

No belief propagation required: Belief space planning in high-dimensional state spaces via factor graphs, the matrix determinant lemma, and re-use of calculation

Dmitry Kopitkov¹ and Vadim Indelman²

Abstract

We develop a computationally efficient approach for evaluating the information-theoretic term within belief space planning (BSP), where during belief propagation the state vector can be constant or augmented. We consider both unfocused and focused problem settings, whereas uncertainty reduction of the entire system or only of chosen variables is of interest, respectively. State-of-the-art approaches typically propagate the belief state, for each candidate action, through calculation of the posterior information (or covariance) matrix and subsequently compute its determinant (required for entropy). In contrast, our approach reduces runtime complexity by avoiding these calculations. We formulate the problem in terms of factor graphs and show that belief propagation is not needed, requiring instead a one-time calculation that depends on (the increasing with time) state dimensionality, and per-candidate calculations that are independent of the latter. To that end, we develop an augmented version of the matrix determinant lemma, and show that computations can be re-used when evaluating impact of different candidate actions. These two key ingredients and the factor graph representation of the problem result in a computationally efficient (augmented) BSP approach that accounts for different sources of uncertainty and can be used with various sensing modalities. We examine the unfocused and focused instances of our approach, and compare it with the state of the art, in simulation and using real-world data, considering problems such as autonomous navigation in unknown environments, measurement selection and sensor deployment. We show that our approach significantly reduces running time without any compromise in performance.

Keywords

Belief space planning, active SLAM, informative planning, active inference, autonomous navigation

1. Introduction

Decision making under uncertainty and belief space planning (BSP) are fundamental problems in robotics and artificial intelligence, with applications including autonomous driving, surveillance, sensor deployment, object manipulation, and active simultaneous localization and mapping (SLAM). The goal is to autonomously determine the best actions according to a specified objective function, given the current belief about random variables of interest that could represent, for example, robot poses, a tracked target, or mapped environment, while accounting for different sources of uncertainty.

Since the true state of interest is typically unknown and only partially observable through acquired measurements, it can only be represented through a probability distribution conditioned on available data. BSP and decision-making approaches reason how this distribution (the *belief*) evolves as a result of candidate actions and future expected observations. Such a problem is an instantiation of a partially

observable Markov decision process (POMDP), while calculating an optimal solution of a POMDP was proven to be computationally intractable (Kaelbling et al., 1998) for all but the smallest problems due to curse of history and curse of dimensionality. Recent research has therefore focused on the development of sub-optimal approaches that trade-off optimality and runtime complexity. These approaches can be classified into those that discretize the action, state, and measurement spaces, and those that operate over continuous spaces.

¹Technion Autonomous Systems Program (TASP), Technion - Israel Institute of Technology, Haifa, Israel

²Department of Aerospace Engineering, Technion - Israel Institute of Technology, Haifa, Israel

Corresponding author:

Dmitry Kopitkov, Technion Autonomous Systems Program (TASP), Technion - Israel Institute of Technology, Haifa 32000, Israel.
Email: dimkak@tx.technion.ac.il

Approaches from the former class include point-based value iteration methods (Pineau et al., 2006), simulationbased (Stachniss et al., 2005), and sampling-based approaches (Agha-Mohammadi et al., 2014; Prentice and Roy, 2009). On the other hand, approaches that avoid discretization are often termed direct trajectory optimization methods (e.g. Indelman et al., 2015; Patil et al., 2014; Platt et al., 2010; Van Den Berg et al., 2012; Walls et al., 2015); these approaches typically calculate a locally optimal solution from a given nominal solution.

Decision making under uncertainty, also sometimes referred to as active inference, and BSP can be formulated as selecting an optimal action from a set of candidates, based on some cost function. In information-based decision making, the cost function typically contains terms that evaluate the expected posterior uncertainty upon action execution, with commonly used costs including (conditional) entropy and mutual information (MI). Thus, for Gaussian distributions the corresponding calculations typically involve calculating a determinant of *a posteriori* covariance (information) matrices and, moreover, these calculations are to be performed for *each* candidate action.

Decision making and BSP become even more challenging problems when considering *high-dimensional* state spaces. Such a setup is common in robotics, for example in the context of BSP in uncertain environments, active SLAM, sensor deployment, graph reduction, and graph sparsification. In particular, calculating a determinant of information (covariance) matrix for an n -dimensional state is in general $O(n^3)$, and is smaller for sparse matrices as in SLAM problems (Bai et al., 1996).

Moreover, state-of-the-art approaches typically perform these calculations from scratch for *each* candidate action. For example, in the context of active SLAM, state-of-the-art BSP approaches first calculate the posterior belief within the planning horizon, and then use that belief to evaluate the objective function, which typically includes an information-theoretic term (Huang et al., 2005; Indelman et al., 2015; Kim and Eustice, 2014; Valencia et al., 2013). These approaches then determine the best action by performing the mentioned calculations for each action from a given set of candidate actions, or by local search using dynamic programming or gradient descent (for continuous setting).

Sensor deployment is another example of decision making in high-dimensional state spaces. The basic formulation of the problem is to determine locations to deploy the sensors such that some metric can be measured most accurately through the entire area (e.g. temperature). The problem can also be viewed as selecting the optimal action from the set of candidate actions (available locations) and the objective function usually contains a term of uncertainty, such as the entropy of a posterior system (Krause et al., 2008). Also here, state-of-the-art approaches evaluate a determinant over large posterior covariance (information)

matrices for each candidate action, and do so from scratch (Zimmerman, 2006; Zhu and Stein, 2006).

A similar situation also arises in measurement selection (Carlone et al., 2014; Davison, 2005) and graph pruning (Carlevaris-Bianco et al., 2014; Huang et al., 2012; Mazuran et al., 2014; Vial et al., 2011) in the context of long-term autonomy in SLAM. In the former case, the main idea is to determine the most informative measurements (e.g. image features) given measurements provided by robot sensors, thereby discarding uninformative and redundant information. Such a process typically involves reasoning about MI (see e.g. Chli and Davison, 2009; Davison, 2005), for each candidate selection. Similarly, graph pruning and sparsification can be considered as instances of decision making in high-dimensional state spaces (Carlevaris-Bianco et al., 2014; Huang et al., 2012), with decision corresponding to determining what nodes to marginalize out (Ila et al., 2010; Kretzschmar and Stachniss, 2012), and avoiding the resulting fill-in in an information matrix by resorting to sparse approximations of the latter (Carlevaris-Bianco et al., 2014; Huang et al., 2012; Mazuran et al., 2014; Vial et al., 2011). Also here, existing approaches typically involve calculation of the determinant of large matrices for each candidate action.

In this paper we develop a computationally efficient and exact approach for decision making and BSP in high-dimensional state spaces that addresses the aforementioned challenges. The *key idea* is to use the (augmented) general matrix determinant lemma to calculate action impact with complexity *independent* of state dimensionality n , while *re-using* calculations between evaluating impact for different candidate actions. Our approach supports general observation and motion models, and non-myopic planning, and is thus applicable to a wide range of applications such as those mentioned above, where fast decision making and BSP in high-dimensional state spaces is required.

Although many particular domains can be specified as decision-making and BSP problems, they all can be classified into two main categories, one where state vector is fixed during belief propagation and another where the state vector is augmented with new variables. Sensor deployment is an example of the first case, while active SLAM, where future robot poses are introduced into the state, is an example of the second case. Conceptually the first category is a particular case of the second, but as we will see both will require different solutions. Therefore, in order to differentiate between these two categories, in this paper we consider the first category (fixed-state) as a *BSP* problem, and the second category (augmented-state) as an *augmented BSP* problem.

Moreover, we show the proposed concept is applicable also to active *focused* inference. Unlike the *unfocused* case discussed thus far, active *focused* inference approaches aim to reduce the uncertainty over only a predefined set of the variables. The two problems can have significantly different optimal actions, with an

optimal solution for the *unfocused* case potentially performing badly for the *focused* setup, and vice versa (see e.g. Levine and How, 2013). While the set of *focused* variables can be small, exact state-of-the-art approaches calculate the marginal posterior covariance (information) matrix, for each action, which involves a computationally expensive Schur complement operation. For example, Mu et al. (2015) calculate posterior covariance matrix per each measurement and then use the selection matrix in order to obtain the marginal of the *focused* set. Levine and How (2013) developed an approach that determines MI between *focused* and *unfocused* variables through message-passing algorithms on Gaussian graphs but their approach is limited to only graphs with unique paths between the relevant variables.

In contrast, we provide a novel way to calculate posterior entropy of *focused* variables, which is fast, simple, and general; yet, it does not require calculation of a posterior covariance matrix. In combination with our *re-use* algorithm, it provides a *focused* decision-making solver that is significantly faster compared with standard approaches.

Calculating the posterior information matrix in augmented BSP problems involves augmenting an appropriate prior information matrix with zero rows and columns, i.e. zero padding, and then adding new information due to candidate action (see Figure 1). While the general matrix determinant lemma is an essential part of our approach, unfortunately it is not applicable to the mentioned augmented prior information matrix since the latter is singular (even though the posterior information matrix is full rank). In this paper, we develop a new variant of the matrix determinant lemma, called the *augmented matrix determinant lemma* (AMDL), that addresses the general augmentation of a future state vector. Based on AMDL, we then develop a augmented BSP approach, considering both *unfocused* and *focused* cases.

Finally, there is also a relation to the recently introduced concept of decision making in a conservative sparse information space (Indelman, 2015b, 2016). In particular, considering unary observation models (involving only one variable) and greedy decision making, it was shown that appropriately dropping all correlation terms and remaining only with a diagonal covariance (information) matrix does not sacrifice performance while significantly reducing computational complexity. While the approach presented herein confirms this concept for the case of unary observation models, our approach addresses a general non-myopic decision-making problem, with arbitrary observation and motion models. Moreover, compared with state-of-the-art approaches, our approach significantly reduces runtime while providing identical decisions.

To summarize, our main contributions in this paper are as follows: (a) we formulate (augmented) BSP in terms of factor graphs, which allow us to see the problem in a more intuitive and simple way; (b) we develop an augmented version of matrix determinant lemma (AMDL), where the

subject matrix is first augmented by zero rows/columns and only then new information is introduced; (c) we develop an approach for a non-myopic *focused* and *unfocused* (augmented) BSP in high-dimensional state spaces that uses the (augmented) matrix determinant lemma to avoid calculating determinants of large matrices, with per-candidate complexity independent of state dimension; (d) we show how calculations can be *re-used* when evaluating impacts of different candidate actions; we integrate the calculations *re-use* concept and AMDL into a general and highly efficient BSP solver, that does not involve explicit calculation of posterior belief evolution for different candidate actions, naming this approach *rAMDL*; (e) we introduce an even more efficient *rAMDL* variant specifically addressing a sensor deployment problem.

This paper is an extension of the work presented in Kopitkov and Indelman (2016, 2017). As a further contribution, in this manuscript we show the problem of (augmented) BSP can be formulated through factor graphs (Kschischang et al., 2001), thereby providing insights into the sparsity patterns of Jacobian matrices that correspond to future posterior beliefs. We exploit the sparsity of the augmented BSP problem and provide a solution for this case which outperforms our previous approach from Kopitkov and Indelman (2017). In addition, we present an improved approach, compared with Kopitkov and Indelman (2016), to solve the non-augmented BSP problem. Moreover, we develop a more efficient variant of our approach, specifically addressing the sensor deployment problem (Section 4.1), and show in detail how our augmented BSP method can be applied particularly to autonomous navigation in unknown environments and active SLAM (Section 4.2). This paper also includes an extensive experimental evaluation using synthetic and real-world data (Section 7), with the aim of comparing the time performance of all the approaches proposed herein in a variety of challenging scenarios, and benchmarking against the state of the art. Finally, here we attempt to give the reader some more insights on the problem, discussing theoretical interpretation of information gain (IG) metric, relation to MI (Davison, 2005; Kaess and Dellaert, 2009) (Section 3.5), and practical remarks regarding the proposed approach.

This paper is organized as follows. Section 2 introduces the concepts of BSP, and gives a formal statement of the problem. Section 3 describes our approach *rAMDL* for general formulation. Section 4 tailors the approach for specific domains, providing even more efficient solutions to a number of them. In Section 5 standard approaches are discussed as the main state-of-the-art alternatives to *rAMDL*. Further, in Section 6 we analyze the runtime complexity of approaches presented herein and their alternatives. Section 7 presents experimental results, evaluating the proposed approach and comparing it against the mentioned state of the art. Conclusions are drawn in Section 8. To improve readability, proofs of several lemmas are given in the Appendix.

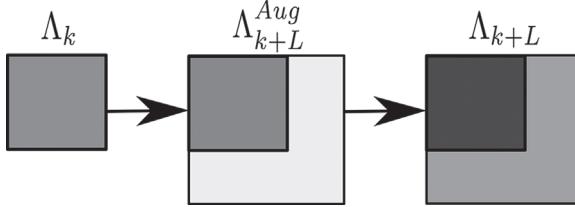


Fig. 1. Illustration of the construction of Λ_{k+L} for a given candidate action in the augmented BSP case. First, Λ_{k+L}^{Aug} is created by adding n' zero rows and columns. Then, the new information of belief is added through $\Lambda_{k+L} = \Lambda_{k+L}^{Aug} + A^T A$.

2. Notation and problem definition

In this paper, we develop computationally efficient approaches for BSP. As evaluating action impact involves inference over an appropriate posterior, we first formulate the corresponding inference problem.

Consider a high-dimensional problem-specific state vector $X_k \in \mathbb{R}^n$ at time t_k . In different applications the state X_k can represent the robot configuration and poses (optionally for whole history), environment-related variables, or any other variables to be estimated. In addition, consider factors $F_i = \{f_i^1(X_i^1), \dots, f_i^{n_i}(X_i^{n_i})\}$ that were added at time $0 \leq t_i \leq t_k$, where each factor $f_i^j(X_i^j)$ represents a specific measurement model, motion model or prior, and as such involves appropriate state variables $X_i^j \subseteq X_i$.

The joint probability distribution function (pdf) can then be written as

$$\mathbb{P}(X_k | \mathbf{H}_k) \propto \prod_{i=0}^k \prod_{j=1}^{n_i} f_i^j(X_i^j), \quad (1)$$

where \mathbf{H}_k is history that contains all the information gathered until time t_k (measurements, controls, etc.).

The inference problem can be naturally represented by a factor graph (Kschischang et al., 2001), which is a bipartite graph $G = (\mathcal{F}, \Theta, \mathcal{E})$ with two node types: variables nodes $\theta_i \in \Theta$ and factor nodes $f_i \in \mathcal{F}$ (see e.g. Figures 2 and 3). Variable nodes represent state variables that need to be estimated, while factor nodes express different constraints between different variables. Each factor node is connected by edges $e_{ij} \in \mathcal{E}$ to variable nodes that are involved in the corresponding constraint. Such a formulation is general and can be used to represent numerous inference problems (e.g. SLAM), while exploiting sparsity. Furthermore, computationally efficient approaches, based on such formulation and exploiting its natural sparsity, have been developed recently (Kaess et al., 2012, 2008).

As is common in many inference problems, we will assume that all factors have a Gaussian form:

$$f_i^j(X_i^j) \propto \exp\left(-\frac{1}{2} \|h_i^j(X_i^j) - r_i^j\|_{\Sigma_i^j}^2\right), \quad (2)$$

with appropriate model

$$r_i^j = h_i^j(X_i^j) + v_i^j, \quad v_i^j \sim \mathcal{N}(0, \Sigma_i^j), \quad (3)$$

where h_i^j is a known nonlinear function, v_i^j is zero-mean Gaussian noise, and r_i^j is the expected value of h_i^j ($r_i^j = \mathbb{E}[h_i^j(X_i^j)]$). Such a factor representation is a general way to express information about the state. In particular, it can represent a measurement model, in which case h_i^j is the observation model and r_i^j and v_i^j are the actual measurement z and measurement noise, respectively. Similarly, it can also represent a motion model (see Section 4.2). A maximum *a posteriori* (MAP) inference can be calculated efficiently (see e.g. Kaess et al., 2012) such that

$$\mathbb{P}(X_k | \mathbf{H}_k) = \mathcal{N}(X_k^*, \Sigma_k), \quad (4)$$

where X_k^* and Σ_k are the mean vector and covariance matrix, respectively.

We shall refer to the posterior $\mathbb{P}(X_k | \mathbf{H}_k)$ as the belief and write

$$b[X_k] \doteq \mathbb{P}(X_k | \mathbf{H}_k). \quad (5)$$

In the context of BSP, we typically reason about the evolution of future beliefs $b[X_{k+l}]$ at different look-ahead steps l as a result of different candidate actions. A particular candidate action can provide unique information (future observations and controls) in a form of newly introduced factors (see Figures 2 and 3) and can be more and less beneficial for specific tasks such as reducing future uncertainty. For example, in a SLAM application, choosing a trajectory that is close to the mapped landmarks (see Figure 3) will reduce uncertainty because of loop closures. Furthermore, conceptually each candidate action can introduce different additional state variables into the future state vector, as in the case of the smoothing SLAM formulation in Figure 3 where the state is augmented by a (various) number of future robot poses.

Therefore, in order to reason about the belief $b[X_{k+l}]$, first it needs to be carefully modeled. More specifically, let us focus on a non-myopic candidate action $a = \{\bar{a}_1, \dots, \bar{a}_L\}$ that is a sequence of myopic actions with planning horizon L . Each action \bar{a}_l can be represented by new factors $F_{k+l} = \{f_{k+l}^1(X_{k+l}^1), \dots, f_{k+l}^{n_{k+l}}(X_{k+l}^{n_{k+l}})\}$ and, possibly, new state variables X_{new}^{k+l} ($1 \leq l \leq L$) that are acquired/added while applying \bar{a}_l . Similar to (1), the future belief $b[X_{k+l}]$ can be explicitly written as

$$b[X_{k+l}] \propto b[X_k] \prod_{l=k+1}^{k+L} \prod_{j=1}^{n_l} f_l^j(X_l^j), \quad (6)$$

where $X_{k+l} \doteq \{X_k \cup X_{new}^{k+1} \cup \dots \cup X_{new}^{k+l}\}$ contains old and new state variables. For example, in Figure 3 at each future time step $k+l$, a new robot pose x_{k+l} is added to the state; new motion model factors $\{f_7, f_8, f_9\}$ and a new observation model factor are also introduced. Similar expressions can be also written for any other lookahead step l . Observe in the above belief (6) that the future factors depend on future observations, whose actual values are unknown.

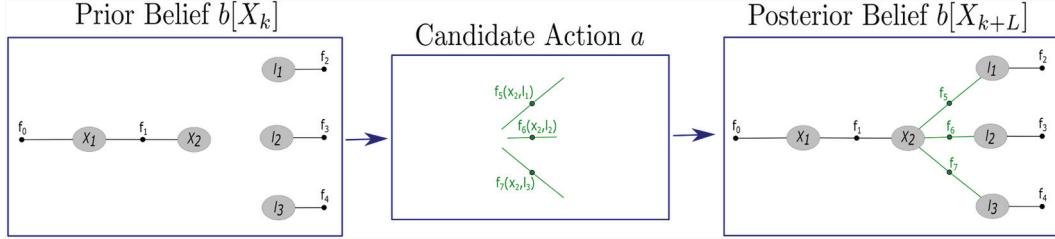


Fig. 2. Illustration of belief propagation in non-augmented BSP SLAM scenario. Prior belief $b[X_k]$ and posterior belief $b[X_{k+L}]$ are represented through factor graphs, with state vector being $X_k = X_{k+L} = \{x_1, x_2, l_1, l_2, l_3\}$ where x_i and l_i denote robot poses and landmarks, respectively. Factor f_0 is prior on the robot's initial pose x_1 ; factors $\{f_2, f_3, f_4\}$ are priors on landmark positions; factors between robot poses represent motion models; factors between pose and landmark represent observation models. Candidate action a introduces new measurements that are represented by new factors $\{f_5, f_6, f_7\}$.

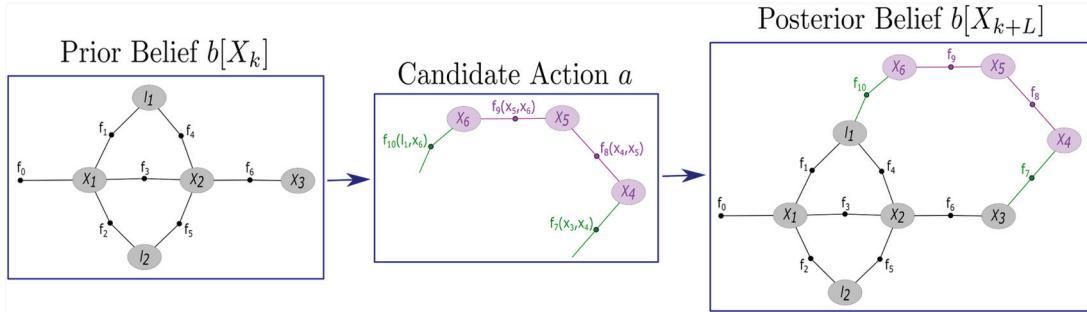


Fig. 3. Illustration of belief propagation in the augmented BSP SLAM scenario. Prior belief $b[X_k]$ and posterior belief $b[X_{k+L}]$ are represented through factor graphs, with appropriate state vectors being $X_k = \{x_1, x_2, x_3, l_1, l_2\}$ and $X_{k+L} = \{x_1, x_2, x_3, l_1, l_2, x_4, x_5, x_6\}$, where x_i and l_i represent robot poses and landmarks, respectively. Factor f_0 is prior on robot's initial pose x_1 ; factors $\{f_2, f_3, f_4\}$ are priors on landmark positions; factors between robot poses represent motion models; factors between pose and landmark represent observation models. Candidate action a makes the robot re-observe landmark l_1 ; in order to do so it introduces new robot poses $\{x_4, x_5, x_6\}$ and new factors $\{f_7, f_8, f_9, f_{10}\}$.

It is important to note that, according to our definition from Section 1, new variables are added only in the augmented setting of the BSP problem, e.g. in the active SLAM context in Figure 3 where new robot poses are introduced by a candidate action. On the other hand, in a non-augmented BSP setting, the states X_{k+L} and X_k are identical, while the beliefs $b[X_{k+L}]$ and $b[X_k]$ are still conditioned on different data. For example, in sensor deployment and measurement selection (see Figure 2) problems, the candidate actions are all possible subsets of sensor locations and of acquired observations, respectively. Here, when applying a candidate action, new information about X_k is brought in, but the state vector itself is unaltered.

In contrast, in the augmented BSP problem new variables are always introduced. In particular, in both smoothing and filtering formulations of SLAM, candidate actions (trajectories) will introduce both new information (future measurements) and also new variables (future robot poses). While in filtering formulation old pose variables are marginalized out, the *smoothing* formulation instead keeps past and current robot poses and newly mapped landmarks in the state vector (see Figure 3), which is beneficial for better estimation accuracy and sparsity. As such, the smoothing formulation is an excellent example for augmented BSP problem,

whereas the filtering formulation can be considered as a *focused* BSP scenario as described below.

As such the non-augmented BSP setting can be seen as a special case of augmented BSP. In order to use similar notation for both problems, however, in this paper we will consider X_{new}^{k+L} to be an empty set for the former case and non-empty for augmented BSP.

It is not difficult to show (see e.g. Indelman et al., 2015) that in the case of non-augmented BSP the posterior information matrix of the *belief* $b[X_{k+L}]$, which is the inverse of *belief*'s covariance matrix, is given by

$$\Lambda_{k+L} = \Lambda_k + \sum_{l=k+1}^{k+L} \sum_{j=1}^{n_l} (H_l^j)^T \cdot (\Sigma_l^j)^{-1} \cdot H_l^j, \quad (7)$$

where Λ_k is the prior information matrix and $H_l^j \doteq \nabla_x h_l^j$ are the Jacobian matrices of h_l^j functions (see (2)) for all the new factor terms in (6).

As was already mentioned, in the case of augmented BSP, the joint state X_{k+L} includes also new variables (with respect to the current state X_k). Considering $X_k \in \mathbb{R}^n$, first, new n' variables are introduced into future state vector $X_{k+L} \in \mathbb{R}^{n'}$

with $N \doteq n + n'$, and then new factors involving appropriate variables from X_{k+L} are added to form a posterior belief $b[X_{k+L}]$, as shown in (6).

Consequently, in the augmented BSP scenario the posterior information matrix of belief $b[X_{k+L}]$, i.e. Λ_{k+L} , can be constructed by first augmenting the current information matrix Λ_k with n' zero rows and columns to obtain $\Lambda_{k+L}^{Aug} \in \mathbb{R}^{N \times N}$, and thereafter adding new information to it, as illustrated in Figure 1 (see e.g. Indelman et al., 2015):

$$\Lambda_{k+L} = \Lambda_{k+L}^{Aug} + \sum_{l=k+1}^{k+L} \sum_{j=1}^{n_l} (H_l^j)^T \cdot (\Sigma_l^j)^{-1} \cdot H_l^j \quad (8)$$

where $H_l \doteq \nabla_x h_l^j$ are augmented Jacobian matrices of all new factors in (6), linearized about the current estimate of X_k and about initial values of newly introduced variables.

After stacking all new Jacobians in (7) and (8) into a single matrix \tilde{A} , and combining all noise matrices into block-diagonal Ψ , we respectively obtain

$$\Lambda_{k+L} = \Lambda_k + \tilde{A}^T \cdot \Psi^{-1} \cdot \tilde{A} = \Lambda_k + A^T \cdot A \quad (9)$$

$$\Lambda_{k+L} = \Lambda_{k+L}^{Aug} + \tilde{A}^T \cdot \Psi^{-1} \cdot \tilde{A} = \Lambda_{k+L}^{Aug} + A^T \cdot A, \quad (10)$$

where

$$A \doteq \Psi^{-\frac{1}{2}} \cdot \tilde{A} \quad (11)$$

is an $m \times N$ matrix that represents both Jacobians and noise covariances of all new factor terms in (6). The above equations can be considered as a single iteration of Gauss–Newton optimization and, similar to prior work (Indelman et al., 2015; Kim and Eustice, 2014; Van Den Berg et al., 2012), we take a maximum-likelihood assumption by assuming they sufficiently capture the impact of candidate action. Under this assumption, the posterior information matrix Λ_{k+L} is independent of (unknown) future observations (Indelman et al., 2015). One can further incorporate reasoning if a future measurement will indeed be acquired (Chaves et al., 2015; Indelman et al., 2015; Walls et al., 2015); however, this is outside the scope of this paper.

Each block row of matrix A represents a single factor from new terms in (6) and has a *sparse* structure. Only a limited number of its sub-blocks is non-zero, i.e. sub-blocks that correspond to the *involved* variables X_l^j in the relevant factor $f_l^j(X_l^j)$.

For notational convenience, we define the set of non-myopic candidate actions by $\mathcal{A} = \{a_1, a_2, \dots\}$ with appropriate Jacobian matrices $\Phi_A = \{A_1, A_2, \dots\}$. Although the planning horizon is not explicitly shown, each $a \in \mathcal{A}$ can represent a future belief $b[X_{k+L}]$ for different number of lookahead steps L .

A general objective function in decision making/BSP can be written as (Indelman et al., 2015)

$$J(a) \doteq \mathbb{E}_{Z_{k+1:k+L}} \left\{ \sum_{l=0}^{L-1} c_l(b[X_{k+l}], u_{k+l}) + c_L(b[X_{k+L}]) \right\}, \quad (12)$$

with L immediate cost functions c_l , for each look-ahead step, and one cost function for terminal future belief c_L . Each such cost function can include a number of different terms related to aspects such as information measure of future belief, distance to goal, and energy spent on control. Arguably, evaluating the information terms involves the heaviest calculations of J .

Thus, in this paper, we focus only on the information-theoretic term of terminal belief $b[X_{k+L}]$, and consider differential entropy \mathcal{H} (further referred to just as entropy) and IG as the cost functions. Both can measure the amount of information of future belief $b[X_{k+L}]$, and will lead to the same optimal action. Yet, calculation of one is sometimes more efficient than the other, as will be shown in Section 3. Therefore, we consider two objective functions:

$$J_{\mathcal{H}}(a) \doteq \mathcal{H}(b[X_{k+L}]), \quad (13)$$

$$J_{IG}(a) \doteq \mathcal{H}(b[X_k]) - \mathcal{H}(b[X_{k+L}]), \quad (14)$$

where the information matrix Λ_{k+L} , that corresponds to the belief $b[X_{k+L}]$, is a function of candidate a 's Jacobian matrix A , see (9) and (10). The optimal candidate a^* , which produces the most certain future belief, is then given by $a^* = \arg \min_{a \in \mathcal{A}} J_{\mathcal{H}}(a)$, or by $a^* = \arg \max_{a \in \mathcal{A}} J_{IG}(a)$ with both being mathematically identical.

In particular, for Gaussian distributions, entropy is a function of the determinant of a posterior information (covariance) matrix, i.e. $\mathcal{H}(b[X_{k+L}]) \equiv \mathcal{H}(\Lambda_{k+L})$, and the objective functions can be expressed as

$$J_{\mathcal{H}}(a) = \frac{n \cdot \gamma}{2} - \frac{1}{2} \ln |\Lambda_{k+L}|, \quad J_{IG}(a) = \frac{1}{2} \ln \frac{|\Lambda_{k+L}|}{|\Lambda_k|} \quad (15)$$

for BSP, and

$$J_{\mathcal{H}}(a) = \frac{N \cdot \gamma}{2} - \frac{1}{2} \ln |\Lambda_{k+L}|, \quad J_{IG}(a) = \frac{n' \cdot \gamma}{2} + \frac{1}{2} \ln \frac{|\Lambda_{k+L}|}{|\Lambda_k|} \quad (16)$$

for augmented BSP, where $\gamma \doteq 1 + \ln(2\pi)$, and Λ_{k+L} can be calculated according to (9) and (10). Thus, evaluating J requires determinant calculation of an $n \times n$ (or $N \times N$) matrix, which is in general $O(n^3)$, per candidate action $a \in \mathcal{A}$. In many robotics applications, state dimensionality can be huge and even increasing with time (e.g. SLAM), and straightforward calculation of the above equations makes real-time planning hardly possible.

So far, the exposition has referred to *unfocused* BSP problems, where the action impact is calculated by considering *all* the random variables in the system, i.e. the entire state vector. However, as will be shown in the following, our approach is applicable also to *focused* BSP problems.

Focused BSP, in both augmented and non-augmented cases, is another important problem, where in contrast to the former case, only a subset of variables is of interest (see e.g. Krause et al., 2008; Levine and How, 2013; Mu et al., 2015). For example, one can look for an action that reduces

the uncertainty of the robot's final pose. The complexity of such a problem is much higher and proposed techniques succeeded to solve it in $O(kn^3)$ (Krause et al., 2008; Levine and How, 2013) with k being the size of candidate actions set, and in $O(\tilde{n}^4)$ (Mu et al., 2015) with \tilde{n} being the size of involved clique within a Markov random field representing the system.

Considering posterior entropy over the *focused* variables $X_{k+L}^F \subseteq X_{k+L}$ we can write

$$J_{\mathcal{H}}^F(a) = \mathcal{H}(X_{k+L}^F) = \frac{n_F \cdot \gamma}{2} + \frac{1}{2} \ln |\Sigma_{k+L}^{M,F}|, \quad (17)$$

where n_F is the dimensionality of the state X_{k+L}^F , and $\Sigma_{k+L}^{M,F}$ is the posterior marginal covariance of X_{k+L}^F (suffix M for marginal), calculated by simply retrieving appropriate parts of posterior covariance matrix $\Sigma_{k+L} = \Lambda_{k+L}^{-1}$.

Solving the above problem in a straightforward manner involves $O(N^3)$ operations for each candidate action, where $N = n + n'$ is the dimension of the posterior system. In the following sections we develop a computationally more efficient approach that addresses both *unfocused* and *focused* (augmented) BSP problems. As will be seen, this approach naturally supports non-myopic actions and arbitrary factor models \mathcal{H}_i^j , and it is, in particular, attractive to BSP in high-dimensional state spaces.

In both the developed approach and the alternative methods described in Section 5, we are making two assumptions. First, we assume that factors have statistical model with Gaussian white noise (see (3)). Second, we take the maximum-likelihood assumption (Platt et al., 2010) and consider a single Gauss–Newton iteration (see (9) and (10)) is sufficient to capture information impact of a candidate action.

3. Approach

Our approach, *rAMD*, utilizes the well-known matrix determinant lemma (Harville, 1998) and re-use of calculations to significantly reduce the computation of the impact of a candidate action, as defined in Section 2, for both augmented and non-augmented cases of the BSP problem. In Section 3.1 we reformulate these problems in terms of factor graphs that will allow us to see another, more simplified, picture of the BSP problem. In Section 3.2.1 we develop a novel way to calculate the information-theoretic term for *unfocused* non-augmented BSP, and then extend it in Section 3.2.2 to the *focused* case. In addition, in order to significantly reduce computational complexity of the augmented BSP problem, as defined in Section 2, in Section 3.3.1 we extend the matrix determinant lemma for the matrix augmentation case. We then discuss in Sections 3.3.2–3.3.3 how this extension can be used within *unfocused* and *focused* augmented BSP. The conclusion from Sections 3.2–3.3 will be that in all investigated types of BSP problem, the information impact of candidate action can be calculated efficiently given specific

prior covariance entries. Further, in Section 3.4 we discuss another key component of *rAMD*: the re-use of covariance calculations, which exploits the fact that many calculations can be shared among different candidate actions. Finally, in Section 3.5 we describe the connection between our technique and the MI approach from Davison (2005) and Kaess and Dellaert (2009), and discuss an interesting conceptual meaning of the IG metric.

3.1. BSP as a factor graph

In the following we show that, similarly to the inference problem, the BSP problem can also be formulated in terms of factor graphs. The *belief* at time t_k , $b[X_k]$ can be represented by a factor graph $G_k = (\mathcal{F}_k, X_k, \mathcal{E}_k)$, where with little abuse of notation, we use X_k to denote the estimated variables, \mathcal{F}_k is the set of all factors acquired until time t_k , and where \mathcal{E}_k encodes connectivity according to the variables X_i^j involved in each factor f_i^j , as defined in (2). The future *belief* $b[X_{k+L}]$ is constructed by introducing new variables and by adding new factors to the belief $b[X_k]$, as was shown in Section 2. Therefore, it can be represented by a factor graph which is an augmentation of the factor graph G_k , as will be shown below.

More specifically, in the case of non-augmented BSP, let $\mathcal{F}(a) = \{f^1, \dots, f^{n_a}\}$ denote all the new factors from (6) introduced by action a , with n_a being the number of such factors. This abstracts the explicit time notations of factors inside (6), which in turn can be seen as unimportant for the solution of the BSP problem. Then the factor graph of $b[X_{k+L}]$ is the prior factor graph G_k with newly introduced factor nodes $\mathcal{F}(a)$ connected to appropriate variable nodes (see Figure 4 for illustration). Thus, it can be denoted by $G_{k+L}(a)$:

$$G_{k+L}(a) = (\mathcal{F}_{k+L}, X_{k+L}, \mathcal{E}_{k+L}), \quad (18)$$

where $\mathcal{F}_{k+L} = \{\mathcal{F}_k, \mathcal{F}(a)\}$, $X_{k+L} \equiv X_k$ are unaltered state variables, and \mathcal{E}_{k+L} represents connectivity between variables and factors according to the definition of each factor (2).

For instance, consider the running example in Figure 2 where action a introduces new measurements in context of active SLAM. The state vector $X_{k+L} \equiv X_k$ contains robot poses and landmarks $\{x_1, x_2, l_1, l_2, l_3\}$. Old factors within \mathcal{F}_k are priors $\{f_0, f_2, f_3, f_4\}$ and motion model f_1 . Candidate action introduces new observation factors $\mathcal{F}(a) = \{f_5, f_6, f_7\}$. Thus, factors of the posterior factor graph $G_{k+L}(a) = (\mathcal{F}_{k+L}, X_{k+L}, \mathcal{E}_{k+L})$ are $\mathcal{F}_{k+L} = \{f_0, f_1, f_2, f_3, f_4, f_5, f_6, f_7\}$, while \mathcal{E}_{k+L} contains edges between nodes in X_{k+L} and nodes in \mathcal{F}_{k+L} as depicted in the figure.

For simplicity, we denote the augmentation of a factor graph G_k with a set of new factors through operator \oplus . Thus, for the non-augmented BSP setting, we have $G_{k+L}(a) \doteq G_k \oplus \mathcal{F}(a)$. In addition, with a slight abuse of notation we will use the same augmentation operator \oplus to define a combination of two factor graphs into one, which will be required in the context of augmented BSP.

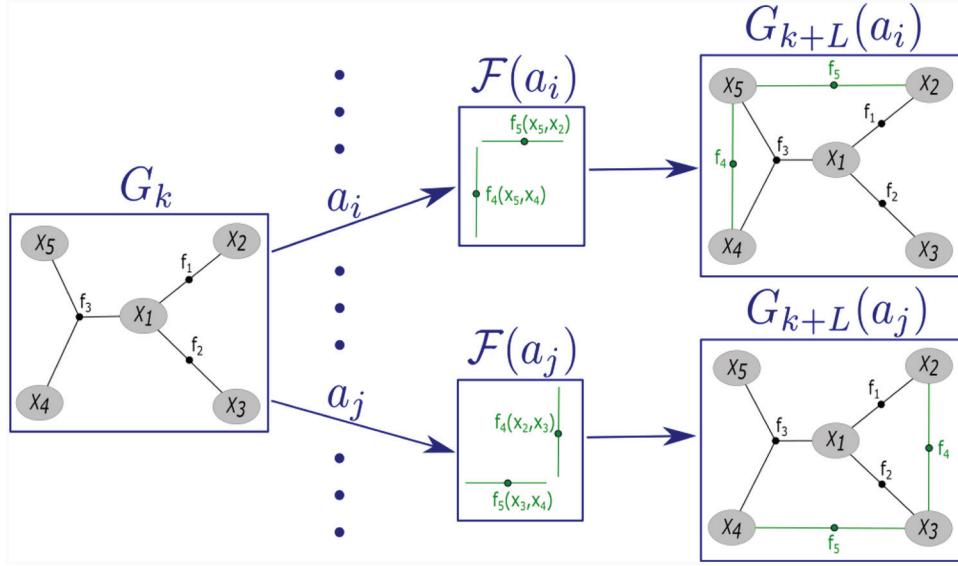


Fig. 4. Illustration of belief propagation in a factor graph representation in the non-augmented case. Two actions a_i and a_j are considered, introducing two new factor sets $\mathcal{F}(a_i)$ and $\mathcal{F}(a_j)$, respectively, into the graph (colored in green).

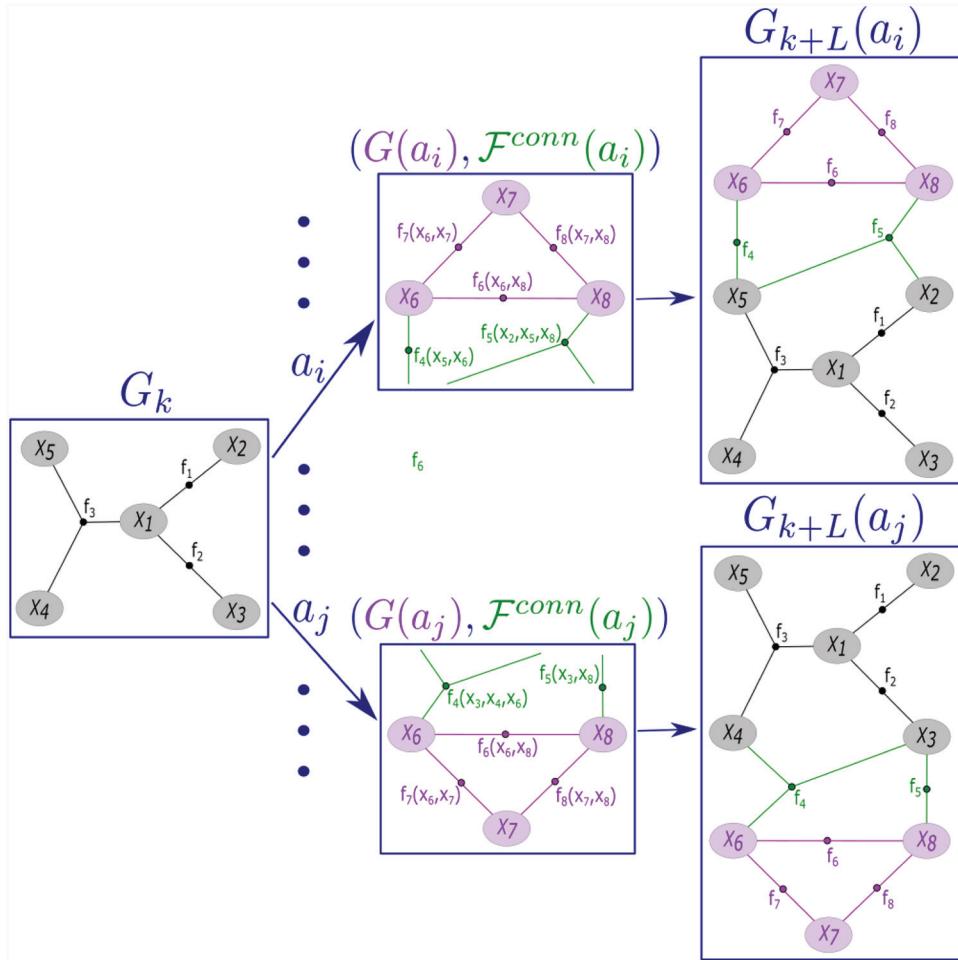


Fig. 5. Illustration of belief propagation in a factor graph representation in the augmented case. Two actions a_i and a_j are considered, introducing their own factor graphs $G(a_i)$ and $G(a_j)$ (colored in pink) that are *connected* to prior G_k through factor sets $\mathcal{F}^{conn}(a_i)$ and $\mathcal{F}^{conn}(a_j)$ (colored in green), respectively.

In the augmented BSP scenario, we denote all new state variables introduced by action a as X_{new} , and also separate all new factors $\mathcal{F}(a)$ from (6) into two groups:

$$\mathcal{F}(a) = \{\mathcal{F}^{new}(a), \mathcal{F}^{conn}(a)\}. \quad (19)$$

Factors connecting only new variables X_{new} are denoted by $\mathcal{F}^{new}(a)$,

$$\mathcal{F}^{new}(a) = \{f^1, \dots, f^{n_{new}}\}, \quad (20)$$

while the rest of the factors are denoted by $\mathcal{F}^{conn}(a)$,

$$\mathcal{F}^{conn}(a) = \{f^1, \dots, f^{n_{conn}}\}, \quad (21)$$

connecting between old and new variables.

Next, let us denote an action's factor graph as $G(a) = (\mathcal{F}^{new}(a), X_{new}, \mathcal{E}_{new})$ with \mathcal{E}_{new} representing connectivity according to *involved* variables in each factor in $\mathcal{F}^{new}(a)$. Then the factor graph that represents the future belief $b[X_{k+L}]$ is a combination of two factor graphs, the prior G_k and action's $G(a)$, connected by factors from $\mathcal{F}^{conn}(a)$ (see Figure 5 for illustration). Thus,

$$G_{k+L}(a) = G_k \oplus G(a) \oplus \mathcal{F}^{conn}(a) = (\mathcal{F}_{k+L}, X_{k+L}, \mathcal{E}_{k+L}), \quad (22)$$

where $\mathcal{F}_{k+L} = \{\mathcal{F}_k, \mathcal{F}^{new}(a), \mathcal{F}^{conn}(a)\}$, $X_{k+L} \equiv \{X_k, X_{new}\}$ is an augmented state vector, and \mathcal{E}_{k+L} represents connectivity between variables and factors according to factors' definition. The separation of factors into two groups allows us to present future *belief*'s factor graph as a simple graph augmentation, and will also be useful during derivation of our approach in Section 3.3. Moreover, the reason for $\mathcal{F}^{conn}(a)$ not to be defined as part of $G(a)$ is due to the fact that factors inside $\mathcal{F}^{conn}(a)$ involve state variables outside of $G(a)$.

For instance, consider the running example in Figure 3 where action a represents a candidate trajectory to perform loop-closure in a SLAM application. Here the prior state vector X_k contains old and current robot poses $\{x_1, x_2, x_3\}$ and mapped until now landmarks $\{l_1, l_2\}$. The newly introduced state variables X_{new} are the future robot poses $\{x_4, x_5, x_6\}$ that represent the candidate trajectory. Old factors within \mathcal{F}_k are prior f_0 on the initial robot pose, motion models $\{f_3, f_6\}$, and observation factors $\{f_1, f_2, f_4, f_5\}$. Here $\mathcal{F}^{new}(a)$ consists of factors $\{f_8, f_9\}$ (colored purple) that are connected only to variables from X_{new} , while $\mathcal{F}^{conn}(a)$ contains factors $\{f_7, f_{10}\}$ (colored green) that connect between variables in X_k and variables in X_{new} . Thus, state variables of the posterior factor graph $G_{k+L}(a)$ are $X_{k+L} = \{x_1, x_2, x_3, x_4, x_5, x_6, l_1, l_2\}$, and factors of $G_{k+L}(a)$ are $\mathcal{F}_{k+L} = \{f_0, f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}\}$.

Note that the new factors in $G_{k+L}(a)$ are not fully defined, as some of them involve future observations that are unknown at planning time. However, the taken maximum-likelihood assumption assumes that the mean vector of $b[X_{k+L}]$ will coincide with current estimate of X_k and with initial values of new variables X_{new} (Indelman et al., 2015; Platt et al., 2010; Van Den Berg et al., 2012). Knowing the mean vector, it is possible to calculate Jacobians of old

and new factors within $G_{k+L}(a)$. Since information matrix $\Lambda = A^T A$ is a product of Jacobian matrices, Λ_{k+L} of future *belief* $b[X_{k+L}]$ can also be calculated without knowing the future observations. Thus, we can reason about the information (and covariance) matrix of $G_{k+L}(a)$, as was shown in Section 2.

Now we can reformulate the information-theoretic objective of the BSP problem. In order to evaluate information impact of action a in a non-augmented BSP setting (15), we need to measure the amount of information added to a factor graph after augmenting it with new factors $G_{k+L}(a) = G_k \oplus \mathcal{F}(a)$. In the case of augmented BSP (16), in order to evaluate the information impact of action a we need to measure the amount of information added to a factor graph after connecting it to another factor graph $G(a)$ through factors in $\mathcal{F}^{conn}(a)$, $G_{k+L}(a) = G_k \oplus G(a) \oplus \mathcal{F}^{conn}(a)$.

In (9) and (10) we expressed the posterior information matrix Λ_{k+L} of $G_{k+L}(a)$ through matrix A , which is the weighted Jacobian of new terms from (6). In non-augmented BSP, each block-row of A represents a specific factor from $\mathcal{F}(a)$, while in augmented BSP block-rows in A represent factors from $\mathcal{F}(a) = \{\mathcal{F}^{new}(a), \mathcal{F}^{conn}(a)\}$. Block-columns of A represent all estimated variables within X_{k+L} . As was mentioned, each factor's block-row is sparse, with only non-zero block entries under columns of variables connected to the factor within the factor graph. For example, the Jacobian matrix's block-row that corresponds to a motion model factor $p(x_{k+l}|x_{k+l-1}, u_{k+l-1})$ will involve only two non-zero block entries for the state variables x_{k+l} and x_{k+l-1} . Factors for many measurement models, such as projection and range models, will also have only two non-zero blocks (see Figure 6).

We define two properties for any set of factors \mathcal{F} that will be used in the following to analyze the complexity of the proposed approach. Denote by $\mathcal{M}(\mathcal{F})$ the sum of dimensions of all factors in \mathcal{F} , where the dimension of each factor is the dimension of its expected value r_i^j from (3). In addition, let $\mathcal{D}(\mathcal{F})$ denote the total dimension of all variables involved in at least one factor from \mathcal{F} . It is not difficult to show that the Jacobian matrix $A \in \mathbb{R}^{m \times n}$ of \mathcal{F} has height $m = \mathcal{M}(\mathcal{F})$, and number of its columns that are not entirely equal to zero is $\mathcal{D}(\mathcal{F})$. The letter \mathcal{D} is used here because the *density* of the information matrix is affected directly by the value of $\mathcal{D}(\mathcal{F})$. It is important to note that, for any candidate action a , the total dimension of new factors $\mathcal{M}(\mathcal{F}(a))$ and the dimension of involved variables $\mathcal{D}(\mathcal{F}(a))$ are *independent* of n , which is the dimension of the belief at planning time $b[X_k]$. Instead, both properties are only functions of the planning horizon L .

In the following sections we describe our BSP approach, using the above notions of factor graphs.

3.2. BSP via the matrix determinant lemma

3.2.1. Unfocused case Information-theoretic BSP involves evaluating the costs from (15), operations that

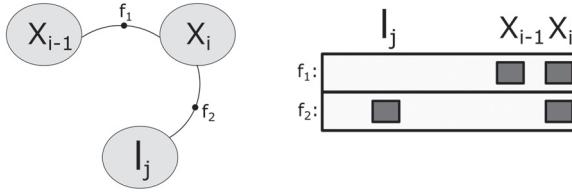


Fig. 6. Concept illustration of A 's structure. Each column represents some variable from the state vector. Each row represents some factor from (6). Here, A represents a set of factors $\mathcal{F} = \{f_1(x_{i-1}, x_i), f_2(x_i, l_j)\}$, where factor f_1 of motion model that involves two poses x_i and x_{i-1} will have non-zero values only at columns of x_i and x_{i-1} . Factor f_2 of observation model that involves together variables x_i and l_j will have non-zero values only at columns of x_i and l_j .

require calculating the determinant of a large $n \times n$ matrix (posterior information matrix), with n being the dimensionality of the state X_{k+L} . State-of-the-art approaches typically perform these calculations from scratch for each candidate action.

In contrast, our approach contains a one-time calculation that depends on the state dimension and will be re-used afterwards to calculate the impact of each candidate action (see Section 3.4). As we show below, the latter depends only on $\mathcal{M}(\mathcal{F}(a))$ and $\mathcal{D}(\mathcal{F}(a))$, while being independent of the state dimension.

Recalling notation from the previous section, we would like to measure the amount of information gained after graph augmentation $G_{k+L}(a) = G_k \oplus \mathcal{F}(a)$. We can measure it through the IG as the utility function. It is not difficult to show that IG from (15) can be written as $J_{IG}(a) = \frac{1}{2} \ln \frac{|\Lambda_k + A^T A|}{|\Lambda_k|}$, where $A \in \mathbb{R}^{m \times n}$ is the Jacobian of factors in $\mathcal{F}(a)$ weighted by their noise, with $m = \mathcal{M}(\mathcal{F}(a))$. Using the generalized matrix determinant lemma (Harville, 1998), this equation can be written as

$$J_{IG}(a) = \frac{1}{2} \ln |I_m + A \cdot \Sigma_k \cdot A^T|, \quad \Sigma_k \equiv \Lambda_k^{-1} \quad (23)$$

as previously suggested in Ila et al. (2010) and Mu et al. (2015) in the context of compact pose-SLAM and *focused* active inference.

Equation (23) provides an exact and general solution for information-based decision making, where each action candidate can produce any number of new factors (non-myopic planning) and where factors themselves can be of any motion or measurement model (unary, pairwise, etc.).

In many problem domains, such as SLAM, inference is typically performed in the information space and, as such, the joint covariance matrix Σ_k is not readily available and needs to be calculated upon demand, which is expensive in general. While, at first sight, it might seem the entire joint covariance matrix needs to be recovered, in practice this is not the case due to *sparsity* of the Jacobian matrix A , as was mentioned above.

Table 1. Different partitions of state variables in BSP

Notation	Description
$X_k = X_{k+L}$	state vector at times k and $k+L$
${}^I X$	subset of X_{k+L} with variables <i>involved</i> in new terms in (6)
${}^{-I} X$	subset of X_{k+L} with variables <i>not involved</i> in new terms in (6)
$X_k^F = X_{k+L}^F = X^F$	subset of X_{k+L} with <i>focused</i> variables
$X_k^U = X_{k+L}^U = X^U$	subset of X_{k+L} with <i>unfocused</i> variables
${}^I X^U$	subset of X^U with variables <i>involved</i> in new terms in (6)
${}^{-I} X^U$	subset of X^U with variables <i>not involved</i> in new terms in (6)

Consequently, only specific entries from the covariance matrix Σ_k are really required, and sparse matrix techniques exist to calculate them efficiently (Golub and Plemmons, 1980; Kaess and Dellaert, 2009). More formally, denote by ${}^I X$ the set of all variables that are connected to factors in $\mathcal{F}(a)$ (see Table 1), i.e. these are the variables that are involved in at least one factor among the new factors generated due to the currently considered candidate action a , see (6). Clearly, the columns of A that correspond to the rest of the variables, ${}^{-I} X$, are entirely filled with zeros (see Figure 6). Thus, equation (23) can be re-written as

$$J_{IG}(a) = \frac{1}{2} \ln |I_m + {}^I A \cdot \Sigma_k^{{M,IX}} \cdot ({}^I A)^T|, \quad (24)$$

where ${}^I A$ is constructed from A by removing all zero columns, and $\Sigma_k^{{M,IX}}$ is a prior joint marginal covariance of variables in ${}^I X$, which should be calculated from the (square root) information matrix Λ_k . Note that dimension of ${}^I X$ is $\mathcal{D}(\mathcal{F}(a))$.

Intuitively, the posterior uncertainty reduction that corresponds to action a is a function of only the prior marginal covariance over variables involved in $\mathcal{F}(a)$ (i.e. $\Sigma_k^{{M,IX}}$) and the new information introduced by the $\mathcal{F}(a)$'s Jacobian A , with the latter also involving the same variables ${}^I X$. Moreover, from the above equation it can be seen that uncertainty reduction in the posterior will be significant for large entries in A and high prior uncertainty over the variables ${}^I X$.

In particular, in the case of myopic decision making with unary observation models (that involve only a single state variable), calculation of $IG(a)$ for different candidate actions only requires recovering the diagonal entries of Σ_k , regardless of the actual correlations between the states, as was recently shown in Indelman (2015b, 2016). However, while in the mentioned papers the per-action calculation takes $O(n)$, the $IG(a)$ calculation is not dependent on n at all, as will be shown in Section 3.4.

Given a prior marginal covariance $\Sigma_k^{{M,IX}}$, whose dimension is $\mathcal{D}(\mathcal{F}(a)) \times \mathcal{D}(\mathcal{F}(a))$, the calculation in (24) is bounded by calculating the determinant of an $\mathcal{M}(\mathcal{F}(a)) \times \mathcal{M}(\mathcal{F}(a))$ matrix, which is, in general,

$O(\mathcal{M}(\mathcal{F}(a))^3)$, where $\mathcal{M}(\mathcal{F}(a))$ is the number of constraints due to new factors (for a given candidate action a). This calculation should be performed for each candidate action in the set \mathcal{A} . Furthermore, in many problems it is logical to assume that $\mathcal{M}(\mathcal{F}(a)) \ll n$, as $\mathcal{M}(\mathcal{F}(a))$ depends mostly on the planning horizon L , which is typically defined and constant, while n (state dimensionality) can be huge and grow with time in real systems (e.g. SLAM). Consequently, given the prior covariance our complexity for selecting best action is $O(|\mathcal{A}|)$, i.e. *independent* of state dimensionality n .

To conclude this section, we showed that calculation of the impact of an action for a single candidate action does not depend on n . While this result is interesting by itself in the context of active inference, in Section 3.4 we go a step further and present an approach to calculate covariance entries, required by all candidates, with one-time calculation that can be re-used afterwards.

3.2.2. Focused case In this section, we present a novel approach to calculate the change in entropy of a *focused* set of variables after factor graph augmentation $G_{k+L}(a) = G_k \oplus \mathcal{F}(a)$, combining it with the ideas from the previous sections (the generalized matrix determinant lemma and IG cost function) and showing that the impact of one candidate action can be calculated independently of state dimension n .

First we recall definitions from Section 2 and introduce additional notation (see also Table 1): $X_k^F \equiv X_{k+L}^F \in \mathbb{R}^{nF}$ denotes the set of *focused* variables (equal to X_{k+L}^F to remind us that prior and posterior states are identical in the non-augmented case), $X_k^U \doteq X_k / X_k^F \in \mathbb{R}^{nU}$ is a set of the remaining *unfocused* variables with $n = n_F + n_U$. The $n_F \times n_F$ prior marginal covariance and information matrices of X_k^F are denoted, respectively, by $\Sigma_k^{M,F}$ (suffix M for marginal) and $\Lambda_k^{M,F} \equiv (\Sigma_k^{M,F})^{-1}$. Furthermore, we partition the joint information matrix Λ_k as

$$\Sigma_k = \begin{bmatrix} \Sigma_k^{M,U} & \Sigma_k^{M,UF} \\ (\Sigma_k^{M,UF})^T & \Sigma_k^{M,F} \end{bmatrix}, \quad \Lambda_k = \begin{bmatrix} \Lambda_k^U & \Lambda_k^{U,F} \\ (\Lambda_k^{U,F})^T & \Lambda_k^F \end{bmatrix}, \quad (25)$$

where $\Lambda_k^F \in \mathbb{R}^{nF \times nF}$ is constructed by retrieving from Λ_k only the rows and the columns related to X_k^F (it is actually conditional information matrix of X_k^F , conditioned on the rest of the variables X_k^U), $\Lambda_k^U \in \mathbb{R}^{nU \times nU}$ is defined similarly for X_k^U , and $\Lambda_k^{U,F} \in \mathbb{R}^{nU \times nF}$ contains the remaining blocks of Λ_k as shown in (25).

The marginal information matrix of X_k^F , i.e. $\Lambda_k^{M,F}$, can be calculated via the Schur complement $\Lambda_k^{M,F} = \Lambda_k^F - (\Lambda_k^{UF})^T \cdot (\Lambda_k^U)^{-1} \cdot \Lambda_k^{UF}$. However, one of the Schur complement's properties (Ouellette, 1981) is $|\Lambda_k| = |\Lambda_k^{M,F}| \cdot |\Lambda_k^U|$, from which we can conclude that

$$|\Lambda_k^{M,F}| = \frac{1}{|\Sigma_k^{M,F}|} = \frac{|\Lambda_k|}{|\Lambda_k^U|}. \quad (26)$$

Therefore, the posterior entropy of X_{k+L}^F (see (17)) is a function of the posterior Λ_{k+L} and its partition Λ_{k+L}^U :

$$J_{\mathcal{H}}^F(a) = \mathcal{H}(X_{k+L}^F) = \frac{n_F \cdot \gamma}{2} - \frac{1}{2} \ln \frac{|\Lambda_{k+L}|}{|\Lambda_{k+L}^U|}. \quad (27)$$

From (9) one can observe that $\Lambda_{k+L}^U = \Lambda_k^U + (A^U)^T \cdot A^U$, where $A^U \in \mathbb{R}^{m \times n_U}$ is constructed from Jacobian A by taking only the columns that are related to variables in X_k^U .

The next step is to use IG instead of entropy, with the same motivation and benefits as in the *unfocused* case (Section 3.2.1). The optimal action $a^* = \arg \max_{a \in \mathcal{A}} J_{IG}^F(a)$ will maximize $J_{IG}^F(a) = \mathcal{H}(X_k^F) - \mathcal{H}(X_{k+L}^F)$, and by combining (27) with the generalized matrix determinant lemma we can write

$$J_{IG}^F(a) = \frac{1}{2} \ln |I_m + A \cdot \Sigma_k \cdot A^T| - \frac{1}{2} \ln |I_m + A^U \cdot \Sigma_k^{U|F} \cdot (A^U)^T|, \quad (28)$$

where $\Sigma_k^{U|F} \in \mathbb{R}^{nU \times nU}$ is a prior covariance matrix of X_k^U conditioned on X_k^F , and it is actually the inverse of Λ_k^U .

Further, A^U can be partitioned into ${}^I A^U$ and ${}^{\neg I} A^U$, representing *unfocused* variables that are, respectively, *involved* (${}^I X^U$) or *not involved* (${}^{\neg I} X^U$) (see also Table 1). Note that ${}^{\neg I} A^U$ contains only zeros, and it can be concluded that

$$|I_m + A^U \cdot \Sigma_k^{U|F} \cdot (A^U)^T| = |I_m + {}^I A^U \cdot \Sigma_k^{I X^U |F} \cdot ({}^I A^U)^T|, \quad (29)$$

where $\Sigma_k^{I X^U |F}$ is the prior covariance of ${}^I X^U$ conditioned on X_k^F .

Taking into account (24) and (29), $J_{IG}^F(a)$ can be calculated through

$$J_{IG}^F(a) = \frac{1}{2} \ln |I_m + {}^I A \cdot \Sigma_k^{M,I X} \cdot ({}^I A)^T| - \frac{1}{2} \ln |I_m + {}^I A^U \cdot \Sigma_k^{I X^U |F} \cdot ({}^I A^U)^T|. \quad (30)$$

We can see that the *focused* and *unfocused* IGs have a simple relation between them

$$J_{IG}^F(a) = J_{IG}(a) - \frac{1}{2} \ln |I_m + {}^I A^U \cdot \Sigma_k^{I X^U |F} \cdot ({}^I A^U)^T|. \quad (31)$$

The second term in (31) is negative and plays the role of a penalty, reducing the action's impact on the posterior entropy of X_{k+L}^F . In Section 3.5 we discuss the intuition behind this penalty term. Note that when all *involved* variables are *focused*, ${}^I X \subseteq X_{k+L}^F$, the variable set ${}^I X^U$ is empty and second term's matrix will be an identity matrix I_m . In such a case, the second term becomes zero and we have $J_{IG}^F(a) = J_{IG}(a)$.

Also here, given prior covariances $\Sigma_k^{M,I X}$ and $\Sigma_k^{I X^U |F}$, calculation of *focused* IG (30) is *independent* of state dimensionality n , with complexity bounded by $O(\mathcal{M}(\mathcal{F}(a))^3)$. In Section 3.4 we show how the required covariances can be efficiently retrieved.

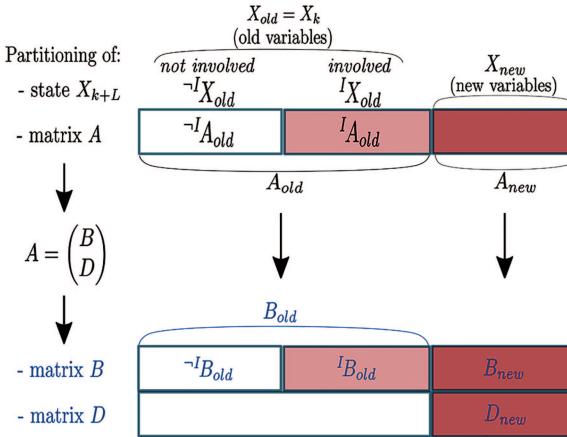


Fig. 7. Partitions of Jacobians and the state vector X_{k+L} in the augmented BSP case, in the *unfocused* scenario. Note that the shown variable ordering is only for illustration, while the developed approach supports *any* arbitrary variable ordering. Also note that all white blocks consist of only zeros. Top: Jacobian A of factor set $\mathcal{F}(a) = \{\mathcal{F}^{conn}(a), \mathcal{F}^{new}(a)\}$. Bottom: Jacobians B and D of factor sets $\mathcal{F}^{conn}(a)$ and $\mathcal{F}^{new}(a)$, respectively.

3.3. Augmented BSP via AMDL

3.3.1. AMDL In order to simplify calculation of IG within augmented BSP (16) one could resort, similar to previous sections, to the matrix determinant lemma. However, due to zero-padding, the information matrix Λ_{k+L}^{Aug} is singular and, thus, the matrix determinant lemma cannot be applied. In this section we develop a variant of the matrix determinant lemma for the considered augmented case (further referred to as AMDL).

Specifically, we want to solve the following problem: recalling $\Lambda^+ = \Lambda^{Aug} + A^T \cdot A$ (see also (10)), and dropping the time indices to avoid clutter, our objective is to express the determinant of Λ^+ in terms of Λ and $\Sigma = \Lambda^{-1}$.

Lemma 1. *The ratio of determinants of Λ^+ and Λ can be calculated through*

$$\frac{|\Lambda^+|}{|\Lambda|} = |\Delta| \cdot |A_{new}^T \cdot \Delta^{-1} \cdot A_{new}|, \quad (32)$$

with $\Delta \doteq I_m + A_{old} \cdot \Sigma \cdot A_{old}^T$, where the matrices $A_{old} \in \mathbb{R}^{m \times n}$ and $A_{new} \in \mathbb{R}^{m \times n'}$ are constructed from A by retrieving columns of only old n variables (denoted as X_{old}) and only new n' variables (denoted as X_{new}), respectively (see Figure 7 and Table 2).

The proof of Lemma 1 is given in Appendix A.1.

Remark 1. It is not difficult to show that AMDL for the matrix update of the form $\Lambda^+ = \Lambda^{Aug} + \tilde{A}^T \cdot \Psi^{-1} \cdot \tilde{A}$ (see (10)) assumes the form

$$\frac{|\Lambda^+|}{|\Lambda|} = |\Psi^{-1}| \cdot |\tilde{\Delta}| \cdot |\tilde{A}_{new}^T \cdot \tilde{\Delta}^{-1} \cdot \tilde{A}_{new}| \quad (33)$$

with $\tilde{\Delta} \doteq \Psi + \tilde{A}_{old} \cdot \Sigma \cdot \tilde{A}_{old}^T$.

Table 2. Different partitions of state variables in augmented BSP

Notation	Description
X_k	state vector at time k
X_{k+L}	state vector at time $k+L$
X_{k+L}^F	subset of X_{k+L} with <i>focused</i> variables
X_{old}	subset of X_{k+L} with old variables, i.e. X_k
X_{new}	subset of X_{k+L} with new variables
$I_{X_{old}}$	subset of X_{old} with variables <i>involved</i> in new terms in (6)
$\neg I_{X_{old}}$	subset of X_{old} with variables <i>not involved</i> in new terms in (6)
Focused augmented BSP ($X_{k+L}^F \subseteq X_{new}$), Section 3.3.3.1	
X_{new}^F	subset of X_{new} with <i>focused</i> variables
X_{new}^U	subset of X_{new} with <i>unfocused</i> variables
Focused augmented BSP ($X_{k+L}^F \subseteq X_{old}$), Section 3.3.3.2	
$I_{X_{old}}^F$	subset of $I_{X_{old}}$ with <i>focused</i> variables
$I_{X_{old}}^U$	subset of $I_{X_{old}}$ with <i>unfocused</i> variables
$\neg I_{X_{old}}^F$	subset of $\neg I_{X_{old}}$ with <i>focused</i> variables
$\neg I_{X_{old}}^U$	subset of $\neg I_{X_{old}}$ with <i>unfocused</i> variables

In addition, we can extend the AMDL lemma for a specific structure of matrix A . As was explained in Section 3.1, in the case of augmented BSP, the new factors can be separated into two sets $\mathcal{F}^{new}(a)$ and $\mathcal{F}^{conn}(a)$. It is not difficult to see that A 's structure in such a case will be

$$A = (A_{old} \ A_{new}) = \begin{pmatrix} B_{old} & B_{new} \\ D_{old} & D_{new} \end{pmatrix} = \begin{pmatrix} B_{old} & B_{new} \\ 0 & D_{new} \end{pmatrix} \quad (34)$$

where B 's rows represent factors from $\mathcal{F}^{conn}(a)$, and D 's rows represent factors from $\mathcal{F}^{new}(a)$ (see also Figure 7). Note that $D_{old} \equiv 0$.

Lemma 2. *The ratio of determinants of Λ^+ and Λ where A has structure from (34) can be calculated through*

$$\frac{|\Lambda^+|}{|\Lambda|} = |\Delta_1| \cdot |B_{new}^T \cdot \Delta_1^{-1} \cdot B_{new} + D_{new}^T \cdot D_{new}| \quad (35)$$

with $\Delta_1 \doteq I_{m_{conn}} + B_{old} \cdot \Sigma \cdot B_{old}^T$ and $m_{conn} = \mathcal{M}(\mathcal{F}^{conn}(a))$, where partitions of B and D are defined above in (34) and can also be seen in Figure 7.

The proof of Lemma 2 is given in Appendix A.2.

We note the above equations are general standalone solutions for any augmented positive-definite symmetric matrix.

To summarize, we developed two augmented determinant lemmas (32) and (35), with the latter exploiting additional knowledge about A 's structure. The dimension of matrix Δ from (32) is $\mathcal{M}(\mathcal{F}(a)) \times \mathcal{M}(\mathcal{F}(a))$, whereas the dimension of Δ_1

from (35) is $\mathcal{M}(\mathcal{F}^{conn}(a)) \times \mathcal{M}(\mathcal{F}^{conn}(a))$. Thus, the complexity of calculation in (35) is lower than in (32) since $\mathcal{M}(\mathcal{F}^{conn}(a)) \leq \mathcal{M}(\mathcal{F}(a))$. In the following sections we use both of the lemmas in order to develop an efficient solution to the augmented BSP problem.

3.3.2. Unfocused augmented BSP through IG Here we show how the AMDL from Section 3.3.1 can be used to efficiently calculate the *unfocused* IG as defined in (16), e.g. change in system's entropy after factor graph augmentation $G_{k+L}(a) = G_k \oplus G(a) \oplus \mathcal{F}^{conn}(a)$ (see Figure 5).

First we introduce different partitions of the joint state X_{k+L} , and the corresponding sub-matrices in the Jacobian matrix A from (10) (see Table 2 and Figure 7). Recall definitions of X_{new} and X_{old} (see Section 3.3.1) and let $'X_{old}$ and $^{-}X_{old}$ denote, respectively, the old involved and the old uninvolvled state variables in the new terms in (6). We represent by $'A_{old}$ and $^{-}A_{old}$ the columns of matrix A that correspond to the state variables $'X_{old}$ and $^{-}X_{old}$, respectively (see Figure 7). Note, $^{-}A_{old} \equiv 0$.

Next, using the AMDL (Lemma 1), the determinant ratio between posterior and prior information matrices is

$$\frac{|\Lambda_{k+L}|}{|\Lambda_k|} = |C| \cdot |A_{new}^T \cdot C^{-1} \cdot A_{new}|, \quad (36)$$

where $C \doteq I_m + A_{old} \cdot \Sigma_k \cdot A_{old}^T$.

Consequently, the IG objective from (16) can be re-written as

$$J_{IG}(a) = \frac{n' \cdot \gamma}{2} + \frac{1}{2} \ln |C| + \frac{1}{2} \ln |A_{new}^T \cdot C^{-1} \cdot A_{new}|. \quad (37)$$

Moreover, considering the above partitioning of A_{old} , we conclude that $A_{old} \cdot \Sigma_k \cdot A_{old}^T = 'A_{old} \cdot \Sigma_k^{M,IX_{old}} \cdot ('A_{old})^T$, where $\Sigma_k^{M,IX_{old}}$ is the marginal prior covariance of $'X_{old}$. Thus, matrix C can be rewritten as

$$C = I_m + 'A_{old} \cdot \Sigma_k^{M,IX_{old}} \cdot ('A_{old})^T. \quad (38)$$

Observe that, given $\Sigma_k^{M,IX_{old}}$, all terms in (38) have relatively small dimensions and $\mathcal{M}(\mathcal{F}(a)) \times \mathcal{M}(\mathcal{F}(a))$ matrix C can be computed efficiently for each candidate action, with time complexity no longer depending on state dimension n , similarly to the non-augmented BSP approach in Section 3.2.1. Calculation of the inverse C^{-1} , which is required in (37), is $O(\mathcal{M}(\mathcal{F}(a))^3)$ and will also not depend on n . The runtime of overall calculation in (37) will have complexity $O(\mathcal{M}(\mathcal{F}(a))^3 + n^3)$ and will depend only on the number of new factors $\mathcal{M}(\mathcal{F}(a))$ and number of new variables n' . Both are functions of the planning horizon L and can be considered as being considerably smaller than the state dimension n . Moreover, higher ratios $n/\mathcal{M}(\mathcal{F}(a))$ lead to a bigger advantage of our approach versus the alternatives (see Section 7).

It is worthwhile to mention a specific case, where $\mathcal{M}(\mathcal{F}(a)) = n'$, which happens for example in SLAM

application when candidate action a introduces only motion (or odometry) factors between the new variables. In such a case, it is not difficult to show that (37) will be reduced to $J_{IG}(a) = \frac{n' \cdot \gamma}{2} + \ln |A_{new}|$. In other words, the IG in such a case depends only on the partition A_{new} of A (see Figure 7), Jacobian entries related to new variables, while the prior Λ_k is not involved in the calculations at all.

Remark 2. It is possible that posterior state dimension $N = n + n'$ will be different for different candidate actions (see e.g. Section 7). In such a case, the entropy (or IG), being a function of the posterior eigenvalues' product, will be of different scale for each candidate and cannot be compared directly. Thus, dimension normalization of (37) may be required. Even though the term $\frac{n' \cdot \gamma}{2}$ may already play the role of such a normalization, the detailed investigation of this aspect is outside the scope of this paper.

We can further enhance the presented above approach by considering the structure of A from (34) (see also Figure 7). This will allow us to slightly improve the complexity of $J_{IG}(a)$'s calculation. By applying the AMDL (Lemma 2), we can show that information gained from connecting G_k (with covariance matrix Σ_k) and $G(a)$ (with information matrix $\Lambda_a = D_{new}^T \cdot D_{new}$) through factors $\mathcal{F}^{conn}(a)$ will be

$$J_{IG}(a) = \frac{n' \cdot \gamma}{2} + \frac{1}{2} \ln |C_1| + \frac{1}{2} \ln |B_{new}^T \cdot C_1^{-1} \cdot B_{new} + \Lambda_a|, \quad (39)$$

$$C_1 = I_{m_{conn}} + B_{old} \cdot \Sigma_k \cdot B_{old}^T, \quad (40)$$

where matrix $B = (B_{old} \ B_{new})$ is the Jacobian of factors in $\mathcal{F}^{conn}(a)$.

Since B_{old} is sparse (see Figure 7), the same as partition A_{old} in (36), C_1 also can be calculated efficiently:

$$C_1 = I_{m_{conn}} + 'B_{old} \cdot \Sigma_k^{M,IX_{old}} \cdot ('B_{old})^T. \quad (41)$$

It is interesting to note that the terms of the above-presented solution for the *unfocused* augmented BSP problem (39) and (41) can be recognized as belonging to different operands in augmentation $G_k \oplus G(a) \oplus \mathcal{F}^{conn}(a)$: prior covariance matrix $\Sigma_k^{M,IX_{old}}$ represents information coming from prior factor graph G_k , information matrix Λ_a provides information of an action's factor graph $G(a)$, and various partitions of matrix B introduce information coming from *connecting* factors $\mathcal{F}^{conn}(a)$.

Although the above solution (39) and (41) look somewhat more complicated, its matrix terms have slightly lower dimensions compared with matrix terms in the general solution presented in (37) and (38), with complexity $O(\mathcal{M}(\mathcal{F}^{conn}(a))^3 + n^3)$, and therefore can be calculated more quickly, as will be shown in our simulations below. Moreover, equations (39) and (41) have more independent terms that can be calculated in parallel, further improving time performance.

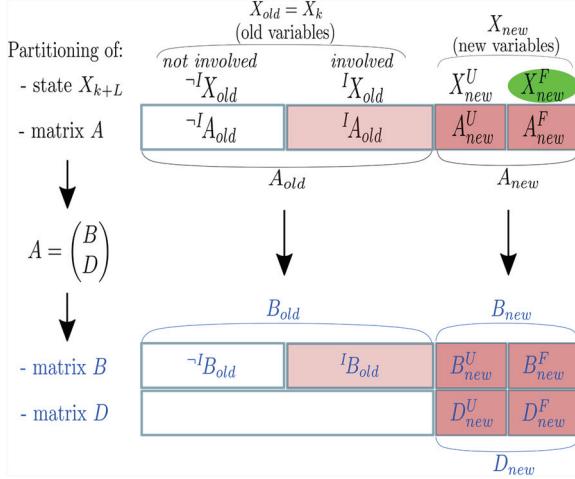


Fig. 8. Partitions of Jacobians and state vector X_{k+L} in the augmented BSP case, *focused* ($X_{k+L}^F \subseteq X_{new}$) scenario. Note that the shown variable ordering is only for illustration, while the developed approach supports *any* arbitrary variable ordering. Also note that all white blocks consist of only zeros. Top: Jacobian A of factor set $\mathcal{F}(a) = \{\mathcal{F}^{conn}(a), \mathcal{F}^{new}(a)\}$. Bottom: Jacobians B and D of factor sets $\mathcal{F}^{conn}(a)$ and $\mathcal{F}^{new}(a)$, respectively.

3.3.3. Focused augmented BSP The *focused* scenario in the augmented BSP setting, with the factor graph augmentation $G_{k+L}(a) = G_k \oplus G(a) \oplus \mathcal{F}^{conn}(a)$, can be separated into different cases. One such case is when the set of focused variables X_{k+L}^F contains only new variables added during BSP augmentation, as illustrated in Figure 8, i.e. $X_{k+L}^F \subseteq X_{new}$ are the variables coming from factor graph $G(a)$. Such a case happens, for example, when we are interested in reducing the entropy of a robot's last pose within the planning horizon. Another case is when the focused variables X_{k+L}^F contain only old variables, as shown in Figure 9, i.e. $X_{k+L}^F \subseteq X_{old} \equiv X_k$ are the variables coming from factor graph G_k . This, for example, could correspond to a scenario where reducing entropy of already-mapped landmarks is of interest (e.g. improve 3D reconstruction quality). The third option is for both new and old variables to be inside X_{k+L}^F . In the following we develop a solution for the first two cases; the third case can be handled in a similar manner.

Remark 3. In most cases, actual variable ordering will be more sporadic than that depicted in Figures 7, 8, and 9. For example, iSAM (Kaess et al., 2012) determines variable ordering using COLAMD (Davis et al., 2004) to enhance the sparsity of the square root information matrix. We note that our approach applies to any arbitrary variable ordering, with the equations derived herein remaining unchanged.

3.3.3.1 Focused augmented BSP ($X_{k+L}^F \subseteq X_{new}$): focused variables belong to $G(a)$ First we define additional partitions of Jacobian A (see Figure 8). The sub-matrices A_{old} , A_{new} , $\neg I_{old}$, and I_{old} were already introduced in the sections above. We now further partition A_{new} into A_{new}^U and

A_{new}^F , that correspond, respectively, to columns of new variables that are focused and unfocused. Denote the former set of variables as X_{new}^F and the latter as X_{new}^U (see also Table 2). Note, $X_{new}^F \equiv X_{k+L}^F$.

Lemma 3. *The posterior entropy of X_{new}^F (equation (17)) is given by*

$$\begin{aligned} J_{\mathcal{H}}^F(a) &= \frac{n_F \cdot \gamma}{2} + \frac{1}{2} \ln |(A_{new}^U)^T \cdot C^{-1} \cdot A_{new}^U| \\ &\quad - \frac{1}{2} \ln |A_{new}^T \cdot C^{-1} \cdot A_{new}|, \end{aligned} \quad (42)$$

where C is defined in (38).

The proof of Lemma 3 is given in Appendix A.3.

We obtained an exact solution for $J_{\mathcal{H}}^F(a)$ that, given $\Sigma_k^{M, I_{old}}$, can be calculated efficiently with complexity $O(\mathcal{M}(\mathcal{F}(a))^3 + n^3)$, similarly to *unfocused* augmented BSP in Section 3.3.2. In Section 3.4, we explain how the prior marginal covariance term ($\Sigma_k^{M, I_{old}}$) can be efficiently retrieved, providing a fast solution for *focused* augmented BSP.

In addition, it is interesting to note that there is an efficient way to calculate the term $\frac{1}{2} \ln |(A_{new}^U)^T \cdot C^{-1} \cdot A_{new}^U| - \frac{1}{2} \ln |A_{new}^T \cdot C^{-1} \cdot A_{new}|$ from (42). First, we calculate the matrix $V \doteq A_{new}^T \cdot C^{-1} \cdot A_{new}$. Note that each row/column of V represents one of the new variables in X_{new} . Next, we reorder rows and columns of V to obtain matrix V^{UF} where first go rows and columns of X_{new}^U , followed by rows and columns of X_{new}^F . Now, we can perform Cholesky decomposition of $V^{UF} = L^T \cdot L$ and retrieve L 's diagonal entries that belong to variables X_{new}^F , denoted by r_{ii}^F . It is not difficult to show that

$$\begin{aligned} &\frac{1}{2} \ln |(A_{new}^U)^T \cdot C^{-1} \cdot A_{new}^U| - \frac{1}{2} \ln |A_{new}^T \cdot C^{-1} \cdot A_{new}| \\ &= - \sum_i \log r_{ii}^F. \end{aligned} \quad (43)$$

Further, as in Section 3.3.2, we additionally exploit the special structure of A from (34) (see also Figure 8). Similarly to *unfocused* augmented BSP, this will allow us to improve the complexity of $J_{\mathcal{H}}^F(a)$'s calculation.

Lemma 4. *The posterior entropy of X_{new}^F (equation (17)), where A has the structure from (34), is given by*

$$\begin{aligned} J_{\mathcal{H}}^F(a) &= \frac{n_F \cdot \gamma}{2} + \frac{1}{2} \ln |(B_{new}^U)^T \cdot C_1^{-1} \cdot B_{new}^U + \Lambda_a^{U|F}| \\ &\quad - \frac{1}{2} \ln |B_{new}^T \cdot C_1^{-1} \cdot B_{new} + \Lambda_a|, \end{aligned} \quad (44)$$

where C_1 is defined in (41), $\Lambda_a = D_{new}^T \cdot D_{new}$ is the information matrix of the action's factor graph $G(a)$, and $\Lambda_a^{U|F} = (D_{new}^U)^T \cdot D_{new}^U$ is the information matrix of variables X_{new}^U conditioned on X_{new}^F and calculated from distribution represented by $G(a)$.

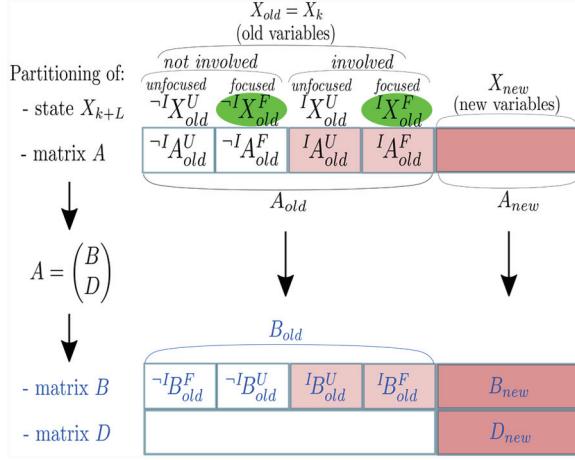


Fig. 9. Partitions of Jacobians and state vector X_{k+L} in the augmented BSP case, *focused* ($X_{k+L}^F \subseteq X_{old}$) scenario. Note that the shown variable ordering is only for illustration, while the developed approach supports *any* arbitrary variable ordering. Also note that all white blocks consist of only zeros. Top: Jacobian A of factor set $\mathcal{F}(a) = \{\mathcal{F}^{conn}(a), \mathcal{F}^{new}(a)\}$. Bottom: Jacobians B and D of factor sets $\mathcal{F}^{conn}(a)$ and $\mathcal{F}^{new}(a)$, respectively.

The proof of Lemma 4 is given in Appendix A.4.

Also here, the matrix terms from the above solution of the *focused* augmented BSP problem (44) have lower dimensions compared with the matrix terms from the general solution presented in (42). Given the prior marginal covariance $\Sigma_k^{M, I^X_{old}}$ its complexity is $O(\mathcal{M}(\mathcal{F}^{conn}(a))^3 + n'^3)$. We demonstrate a runtime superiority of this solution in our simulations below.

It is important to mention that the information-based planning problem for a system that is propagated through the (extended) Kalman filter (Van Den Berg et al., 2012; Walls et al., 2015), where the objective is to reduce the uncertainty of only marginal future state of the system, is an instance of the *focused* augmented BSP ($X_{k+L}^F \subseteq X_{new}$) problem. Thus, the solution provided in this section is applicable also for Kalman filter planning.

3.3.3.2 Focused augmented BSP ($X_{k+L}^F \subseteq X_{old}$): focused variables belong to G_k Similarly to the previous section, we first introduce additional partitions of Jacobian A for the considered case (see Figure 9). From the top part of the figure we can see that $\neg^I A_{old}$ can be further partitioned into $\neg^I A_{old}^U$ and $\neg^I A_{old}^F$. In particular, $\neg^I A_{old}^U$ represents columns of old variables that are both not involved and unfocused, and $\neg^I A_{old}^F$ represents columns of old variables that are both not involved and focused. We denote the former group of variables by $I^X_{old}^U$ and the latter by $I^X_{old}^F$ (see Table 2). Likewise, I^A_{old} can be partitioned into $I^A_{old}^U$ and $I^A_{old}^F$, representing old involved variables that are, respectively, unfocused (X_{old}^U) or focused (X_{old}^F). Note that in this case, the set of focused variables is $X_{k+L}^F = X_k^F = \{\neg^I X_{old}^F \cup I^X_{old}^F\}$ and is contained in factor graph G_k .

Lemma 5. The focused IG of X_k^F is given by

$$J_{IG}^F(a) = \frac{1}{2} (\ln |C| + \ln |A_{new}^T \cdot C^{-1} \cdot A_{new}| - \ln |S| - \ln |A_{new}^T \cdot S^{-1} \cdot A_{new}|), \quad (45)$$

where C is defined in (38), and

$$S \doteq I_m + I^A_{old}^U \cdot \Sigma_k^{I^X_{old}^U|F} \cdot (I^A_{old}^U)^T, \quad (46)$$

and where $\Sigma_k^{I^X_{old}^U|F}$ is the prior covariance of $I^X_{old}^U$ conditioned on X_k^F .

The proof of Lemma 5 is given in Appendix A.5.

Similarly to the cases discussed above (Sections 3.3.2 and 3.3.3.1), given $\Sigma_k^{M, I^X_{old}}$ and $\Sigma_k^{I^X_{old}^U|F}$, calculation of $J_{IG}^F(a)$ per each action a can be performed efficiently with complexity $O(\mathcal{M}(\mathcal{F}(a))^3 + n'^3)$, independently of state dimension n .

It is interesting to note the specific case where $\mathcal{M}(\mathcal{F}(a)) = n'$. In other words, the number of new measurements is equal to the number of new state variables, which can happen for example when only new robot poses and new motion model factors are added. In such a case, it is not difficult to show that (45) will always return zero. We can conclude that for this specific case ($\mathcal{M}(\mathcal{F}(a)) = n'$) there is no new information about the old focused variables X_k^F .

In addition, similar to previous sections, we use the special structure of A from (34) (see also Figure 9) in order to improve the complexity of $J_{IG}^F(a)$'s calculation.

Lemma 6. The focused IG of X_k^F , where A has the structure from (34), is given by

$$J_{IG}^F(a) = \frac{1}{2} (\ln |C_1| + \ln |B_{new}^T \cdot C_1^{-1} \cdot B_{new} + \Lambda_a| - \ln |S_1| - \ln |B_{new}^T \cdot S_1^{-1} \cdot B_{new} + \Lambda_a|), \quad (47)$$

where C_1 is defined in (41), $\Lambda_a = D_{new}^T \cdot D_{new}$ is the information matrix of an action's factor graph $G(a)$, and

$$S_1 = I_{m_{conn}} + I^B_{old}^U \cdot \Sigma_k^{I^X_{old}^U|F} \cdot (I^B_{old}^U)^T, \quad (48)$$

and where $\Sigma_k^{I^X_{old}^U|F}$ is the prior covariance of $I^X_{old}^U$ conditioned on X_k^F .

The proof of Lemma 6 is given in Appendix A.6.

The matrix terms from the above solution (47) and (48) have lower dimensions compared with the matrix terms from the general solution presented in (45) and (46), with complexity $O(\mathcal{M}(\mathcal{F}^{conn}(a))^3 + n'^3)$ given the prior marginal covariance matrices $\Sigma_k^{M, I^X_{old}}$ and $\Sigma_k^{I^X_{old}^U|F}$. The next section presents our approach to calculate the appropriate entries in the prior covariance only once and re-use the result whenever required.

3.4. Re-use calculations technique

As we have seen above, *unfocused* and *focused* (augmented) BSP problems require different prior covariance entries, in order to use the developed expressions. The required entries for each problem are summarized in Table 3. Note that $\Sigma_k^{M,X}$ and $\Sigma_k^{M,I_{X_{old}}}$ both represent exactly the same thing, prior marginal covariance of old variables *involved* in new terms in (6), and have slightly different notation due to the specifics of augmented and non-augmented settings of BSP. The same goes for $\Sigma_k^{I_X^U|F}$ and $\Sigma_k^{I_{X_{old}}^U|F}$, with both representing prior covariance of *unfocused* and *involved* old variables $I_{X_{old}}^U$ conditioned on *focused* variables X^F . In this section we use the notation of augmented BSP ($\Sigma_k^{M,I_{X_{old}}}$ and $\Sigma_k^{I_{X_{old}}^U|F}$), considering the non-augmented BSP setting as its special case.

From Table 3 it is seen that all approaches require prior marginal covariance of the involved old variables, i.e. $I_{X_{old}}$. In terms of factor graphs, in non-augmented BSP the $I_{X_{old}}$ represents variables connected to factors from set $\mathcal{F}(a)$ and has dimension $\mathcal{D}(\mathcal{F}(a))$, whereas in the augmented BSP scenario the $I_{X_{old}}$ represents variables from prior factor graph G_k connected to factors in the set $\mathcal{F}^{conn}(a)$ and has dimension $\mathcal{D}(\mathcal{F}^{conn}(a))$. Although each candidate action may induce a different set of involved variables, in practice these sets will often have many variables in common as they are all related to the belief at the current time (e.g. about the robot pose), in one way or another. With this in mind, we perform a one-time calculation of prior marginal covariance for *all* involved variables (due to at least one candidate action) and re-use it for efficiently calculating IG and entropy of different candidate actions.

More specifically, denote by $X_{All} \subseteq X_k$ the subset of variables that were involved in new terms in (6) for at least one candidate action. We can now perform a one-time calculation of the prior marginal covariance for this set, i.e. $\Sigma_k^{M,X_{All}}$. The complexity of such calculation may be different for different applications. For example, when using an information filter, the system is represented by information matrix Λ_k , and in general the inverse of the Schur compliment of X_{All} variables should be calculated. However, there are techniques that exploit the sparsity of the underlying matrices in SLAM problems, in order to efficiently recover marginal covariances (Kaess and Dellaert, 2009), and more recently, to keep and update them incrementally (Ila et al., 2015). In Section 7 we show that the calculation time of $\Sigma_k^{M,X_{All}}$ while exploiting sparsity (Golub and Plemmons, 1980; Kaess and Dellaert, 2009) is relatively small compared with the total decision-making time of alternative approaches. Still, the more detailed discussion about complexity of covariance retrieval can be found in Kaess and Dellaert (2009) and Ila et al. (2015). The pseudo-code for BSP problems that require only marginal prior covariances $\Sigma_k^{M,I_{X_{old}}}$ (see Table 3) can be found in Algorithm 1.

For *focused* BSP (Section 3.2.2) and for *focused* augmented BSP ($X_{k+L}^F \subseteq X_{old}$) (Section 3.3.3.2) cases, we

also need the term $\Sigma_k^{I_{X_{old}}^U|F}$ (see (30) and (46)). This term can be computed using two different methods as described below.

First method: calculate it through additional marginal covariance entries. First we calculate the prior marginal covariance $\Sigma_k^{M,(I_{X_{old}}^U,F)}$ for the set of variables $\{I_{X_{old}}^U, X_k^F\}$, and then compute the Schur complement over the relevant partitions in $\Sigma_k^{M,(I_{X_{old}}^U,F)}$ (where suffix *M* denotes marginal):

$$\Sigma_k^{I_{X_{old}}^U|F} = \Sigma_k^{M,I_{X_{old}}^U} - \Sigma_k^{M,I_{X_{old}}^U F} \cdot (\Sigma_k^{M,F})^{-1} \cdot \Sigma_k^{M,F I_{X_{old}}^U}. \quad (49)$$

Consequently, we can use a one-time calculation also for the *focused* BSP and for *focused* augmented BSP ($X_{k+L}^F \subseteq X_{old}$) cases as follows. Let us extend the set X_{All} to contain also all focused variables. Once $\Sigma_k^{M,X_{All}}$ is calculated, $\Sigma_k^{M,(I_{X_{old}}^U,F)}$ will be just its partition and can be easily retrieved from it. As a result, the calculation of $\Sigma_k^{I_{X_{old}}^U|F}$ per candidate action becomes computationally cheap (through (49)). Furthermore, the term $(\Sigma_k^{M,F})^{-1}$ can be calculated only once for all candidates. The pseudo-code of this approach can be found in Algorithm 2.

Second method: compute $\Sigma_k^{I_{X_{old}}^U|F}$ through information matrix partitioning. Recall $X_k = \{X_k^F, X_{old}^U\}$ and consider the following partitioning of a prior information matrix:

$$\Lambda_k = \begin{bmatrix} \Lambda_k^F & \Lambda_k^{F,U} \\ (\Lambda_k^{F,U})^T & \Lambda_k^{X_{old}^U} \end{bmatrix}, \quad (50)$$

where first go rows/columns of *focused* variables X_k^F , and then rows/columns of old *unfocused* variables X_{old}^U . Note that a partition of information matrix Λ_k that belongs to X_{old}^U , $\Lambda_k^{X_{old}^U}$, is an information matrix of the conditional distribution

$$\mathbb{P}(X_{old}^U | X_k^F) = \mathcal{N}^{-1}(\times, \Lambda_k^{X_{old}^U}) = \mathcal{N}(\times, \Sigma_k^{X_{old}^U|F}), \quad (51)$$

where \times is the information or mean vector of this distribution. Since $\Sigma_k^{X_{old}^U|F} = (\Lambda_k^{X_{old}^U})^{-1}$ and recalling $I_{X_{old}}^U \subseteq X_{old}^U$, it follows that $\Sigma_k^{I_{X_{old}}^U|F}$ is just a partition of $\Sigma_k^{X_{old}^U|F}$ that belongs to old *unfocused involved* variables $I_{X_{old}}^U$. Therefore, we need to calculate specific entries of the inverse of $\Lambda_k^{X_{old}^U}$. To do so, our one-time calculation will be as follows. We denote by $X_{All}^U \subseteq X_k$ the subset of *unfocused* variables that were involved in new terms in (6) for at least one candidate action. Next, we calculate $\Sigma_k^{X_{All}^U|F}$, the entries of the inverse of $\Lambda_k^{X_{old}^U}$ that belong to X_{All}^U (e.g. via the method from Kaess and Dellaert (2009)). Now, the required $\Sigma_k^{I_{X_{old}}^U|F}$ is just a partition of $\Sigma_k^{X_{All}^U|F}$ and can be retrieved easily for each candidate action. The pseudo-code of this approach is summarized in Algorithm 3.

Table 3. Different problems and required entries of prior covariance: BSP denotes non-augmented belief space planning; Augmented BSP denotes augmented belief space planning

Problem	Required covariance entries
Unfocused BSP, Section 3.2.1	Σ_k^{M, I_X} (prior marginal covariance of variables involved in new terms in (6))
Focused BSP, Section 3.2.2	Σ_k^{M, I_X} and $\Sigma_k^{I_X^U F}$ (prior covariance of <i>unfocused</i> and <i>involved</i> variables I_X^U conditioned on <i>focused</i> variables X^F)
Unfocused Augmented BSP, Section 3.3.2	$\Sigma_k^{M, I_{X_{old}}}$ (prior marginal covariance of old variables involved in new terms in (6))
Focused Augmented BSP ($X_{k+L}^F \subseteq X_{new}$), Section 3.3.3.1	$\Sigma_k^{M, I_{X_{old}}}$
Focused Augmented BSP ($X_{k+L}^F \subseteq X_{old}$), Section 3.3.3.2	$\Sigma_k^{M, I_{X_{old}}}$ and $\Sigma_k^{I_{X_{old}}^U F}$ (prior covariance of <i>unfocused</i> and <i>involved</i> old variables $I_{X_{old}}^U$ conditioned on <i>focused</i> variables X^F)

Algorithm 1: Pseudo-code for BSP problems requiring only prior marginal covariance entries.

```

1 Inputs:
2   |  $\mathcal{A}$ : candidate actions  $\{a_1, a_2, \dots\}$ 
3 Outputs:
4   |  $a^*$ : optimal action
5 begin:
6   |  $X_{all} \leftarrow$  union of old involved variables  $I_{X_{old}}$  from each candidate action  $a_i$ 
    | Calculate prior marginal covariances  $\Sigma_k^{M, X_{all}}$  of variables in  $X_{all}$  (e.g. via the method from Kaess and Dellaert (2009))
    | for  $a_i \in \mathcal{A}$  do
8   |   | Calculate information impact (IG or posterior entropy), using required prior marginal covariances from
     |   |  $\Sigma_k^{M, X_{all}}$ 
8   | end
9   | Select candidate  $a^*$  with maximal IG or minimal posterior entropy
10 end

```

The first method is a good option when the dimension of X_k^F is relatively small. In such a case, equation (49) can be calculated very quickly. When this is not the case, i.e. the number of focused variables is large, the second technique becomes much faster and, thus, is preferable over the first technique.

Remark 4. As we show in Section 4.2, there are cases where $I_{X_{old}}^U$ is identical between all candidate actions. In such cases $\Sigma_k^{I_{X_{old}}^U | F}$ can be calculated only once and further reused by each candidate action.

To summarize this section, the presented technique performs time-consuming calculations in one computational effort; the results are then used for efficiently evaluating the impact of each candidate action. This concept thus preserves expensive CPU resources of any given autonomous system.

3.5. Connection to the MI approach and theoretical meaning of IG

Mutual information $I(a|b)$ is one additional metric from information theory that is used frequently in the field of information-based decision making. Basically it encodes the quantity of information about set of variables a that we would obtain in the case that the value of variables in the other set b would be revealed to us. For example, this metric was used in Davison (2005) and Kaess and Dellaert (2009) to determine the most informative measurements in a measurement selection problem, and more recently in Bai et al. (2016) for information-based active exploration, with both problems being very similar. In addition, it was used in Carlevaris-Bianco et al. (2014) to create a sparse approximation of the true marginalization using a Chow–Liu tree.

In this section we explore the connection between our BSP approach that uses IG (see Section 3.2) and the MI

Algorithm 2: Pseudo-code for BSP problems requiring both prior marginal and conditional covariance entries: *first method.*

```

1 Inputs:
2   |  $\mathcal{A}$ : candidate actions  $\{a_1, a_2, \dots\}$ 
3   |  $X^F$ : focused old variables
4 Outputs:
5   |  $a^*$ : optimal action
6 begin:
7   |  $X_{all} \leftarrow$  union of old involved variables  ${}^I X_{old}$  from each candidate action  $a_i$ 
8   |  $X_{all} = X_{all} \cup X^F$ 
9   | Calculate prior marginal covariances  $\Sigma_k^{M,X_{All}}$  of variables in  $X_{all}$  (e.g. via the method from Kaess and Dellaert (2009))
10  | Calculate  $(\Sigma_k^{M,F})^{-1}$  explicitly, by retrieving  $\Sigma_k^{M,F}$  from  $\Sigma_k^{M,X_{All}}$  and inverting it
11  | for  $a_i \in \mathcal{A}$  do
12    |   | Calculate  $\Sigma_k^{{}^I X_{old}^U|F}$  via (49), by retrieving appropriate  $\Sigma_k^{M,{}^I X_{old}^U}$  and  $\Sigma_k^{M,{}^I X_{old}^U|F}$  from  $\Sigma_k^{M,X_{All}}$ 
13    |   | Calculate IG of  $X^F$  using  $\Sigma_k^{{}^I X_{old}^U|F}$  and required prior marginal covariances from  $\Sigma_k^{M,X_{All}}$ 
14  | end
15  | Select candidate  $a^*$  with maximal IG
16 end

```

Algorithm 3: Pseudo-code for BSP problems requiring both prior marginal and conditional covariance entries: *second method.*

```

1 Inputs:
2   |  $\mathcal{A}$ : candidate actions  $\{a_1, a_2, \dots\}$ 
3   |  $X^F$ : focused old variables
4 Outputs:
5   |  $a^*$ : optimal action
6 begin:
7   |  $X_{all} \leftarrow$  union of old involved variables  ${}^I X_{old}$  from each candidate action  $a_i$ 
8   | Calculate prior marginal covariances  $\Sigma_k^{M,X_{All}}$  of variables in  $X_{all}$  (e.g. via the method from Kaess and Dellaert (2009))
9   |  $X_{all}^U \leftarrow$  union of old involved unfocused variables  ${}^I X_{old}^U$  from each candidate action  $a_i$ 
10  |  $\Lambda_k^{X_{old}^U} \leftarrow$  partition of prior information matrix  $\Lambda_k$  that belongs to old unfocused variables  $X_{old}^U = X_k \setminus X^F$ 
11  |  $\Sigma_k^{X_{All}|F} \leftarrow$  entries of the inverse of  $\Lambda_k^{X_{old}^U}$  that belong to  $X_{All}^U$  (e.g. via the method from Kaess and Dellaert (2009))
12  | for  $a_i \in \mathcal{A}$  do
13    |   | Calculate IG of  $X^F$ , by retrieving  $\Sigma_k^{M,{}^I X_{old}^U}$  from  $\Sigma_k^{M,X_{All}}$  and  $\Sigma_k^{{}^I X_{old}^U|F}$  from  $\Sigma_k^{X_{All}|F}$ 
14  | end
15  | Select candidate  $a^*$  with maximal IG
16 end

```

approach that is applied in Davison (2005) and Kaess and Dellaert (2009); we show that objective functions of both are mathematically identical and calculate exactly the same metric, even though calculations in our approach are made in a much more efficient way. Moreover, we also present the theoretical meaning of IG that provides better intuition for equations (23) and (28).

In a MI approach we would like to select the most informative measurements from the available set $\{z_1, z_2, \dots\}$ and also to account for possible measurement correlation. Each

candidate measurement has a specific measurement model $z_i = h_i(X_k^i) + v_i$ with $v_i \sim \mathcal{N}(0, \Psi_i)$. The candidate measurements are *a priori* unknown and can be viewed as random variables whose statistic properties are fully defined by a random state vector X_k and random noises v_i , due to measurement models. Combining candidate measurements with the state vector, we have

$$W = (X_k, z_1, z_2, \dots)^T, \quad (52)$$

and similarly to the mentioned papers, it can be shown that the covariance matrix of W is

$$\Sigma_W = \begin{pmatrix} \Sigma_k & \Sigma_k \cdot \tilde{A}_1^T & \Sigma_k \cdot \tilde{A}_2^T & \dots \\ \tilde{A}_1 \cdot \Sigma_k & \tilde{A}_1 \cdot \Sigma_k \cdot \tilde{A}_1^T + \Psi_1 & \tilde{A}_1 \cdot \Sigma_k \cdot \tilde{A}_2^T & \dots \\ \tilde{A}_2 \cdot \Sigma_k & \tilde{A}_2 \cdot \Sigma_k \cdot \tilde{A}_1^T & \tilde{A}_2 \cdot \Sigma_k \cdot \tilde{A}_2^T + \Psi_2 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \quad (53)$$

where \tilde{A}_i is the Jacobian of the measurement model function $h_i(X_k^i)$ and where it was not yet combined with model noise Ψ_i , similarly to \tilde{A} defined in (9). The MI approach (Davison, 2005; Kaess and Dellaert, 2009) calculates $I(X_k|z_i)$ for each candidate z_i from Σ_W and selects candidates with the highest MI.

Now we will show that objective $I(X_k|z_i)$ is mathematically identical to our $J_{IG}(z_i)$ from Section 3.2.1 (see also (23)). First, note that $\Sigma_W^{X_k|z_i} = (\Lambda_k + \tilde{A}_i^T \cdot \Psi_i^{-1} \cdot \tilde{A}_i)^{-1}$ (easy to check by using Schur complement from left and Woodbury matrix identity from right). Further, MI for Gaussian distributions can be calculated through covariance matrices as

$$\begin{aligned} I(X_k|z_i) &= \mathcal{H}(X_k) - \mathcal{H}(X_k|z_i) = \frac{1}{2} \ln \left| \frac{\Sigma_W^{M,X_k}}{\Sigma_W^{X_k|z_i}} \right| \\ &= \frac{1}{2} \ln \left| \frac{\Sigma_W^{M,X_k}}{\Sigma_W^{M,X_k} - \Sigma_W^{M,X_k z_i} \cdot (\Sigma_W^{M,z_i})^{-1} \cdot \Sigma_W^{M,z_i X_k}} \right| \\ &= \frac{1}{2} \ln \left| \frac{\Sigma_k}{\Sigma_k - \Sigma_k \cdot \tilde{A}_i^T \cdot (\tilde{A}_i \cdot \Sigma_k \cdot \tilde{A}_i^T + \Psi_i)^{-1} \cdot \tilde{A}_i \cdot \Sigma_k} \right| \end{aligned} \quad (54)$$

and further can be reduced to $I(X_k|z_i) = \frac{1}{2} \ln \left| \frac{\Lambda_k + \tilde{A}_i^T \cdot \Psi_i^{-1} \cdot \tilde{A}_i}{\Lambda_k} \right|$ which is exactly the *unfocused* IG from (23) for the case when the candidate action $a_i \equiv z_i$ introduces a single factor into the factor graph.

While both approaches are obviously calculating the same metric, the computation complexity is not the same. In both Davison (2005) and Kaess and Dellaert (2009), the objective was calculated through (54) and its complexity was dependent on the dimension of X_k . In contrast, our approach *rAMDL* does so independently of state dimension through (24) as has been shown above, making it more efficient compared with the MI technique.

In addition, Kaess and Dellaert (2009) presented the approach to sequentially select informative measurements that accounts for measurements correlation and redundancy, but without the need to update state estimation during each decision. In Section 4.1 we present our algorithm *Sequential rAMDL* where we combine a similar idea together with the *rAMDL* technique in order to eliminate the need for a marginal covariance calculation at each decision.

Most importantly, from the above equations we can see conceptually a very interesting meaning of the metric that is calculated (IG or MI). Without omitting the noise matrix Ψ from our formulation, we can show that the *unfocused* IG of future measurement z is

$$J_{IG}(z) = \frac{1}{2} \ln |I_m + A \cdot \Sigma_k \cdot A^T| = \frac{1}{2} \ln \frac{|\Psi + \tilde{A} \cdot \Sigma_k \cdot \tilde{A}^T|}{|\Psi|}. \quad (55)$$

Further, from (53) we see that $\Sigma^z \doteq \Psi + \tilde{A} \cdot \Sigma_k \cdot \tilde{A}^T$ is the covariance matrix of the random z . Thus, we can see that

$$J_{IG}(z) = \frac{1}{2} \ln |\Sigma^z| - \frac{1}{2} \ln |\Psi| = \mathcal{H}(z) - \mathcal{H}(\nu), \quad (56)$$

where ν is random noise from z 's measurement model, with $\nu \sim \mathcal{N}(0, \Psi)$. From (56) we see that IG is exactly the difference between entropies of future measurement and its noise. It can be explained in the following way: as was mentioned previously, random variable z is fully defined by random variables X_k and ν through measurement model. When z 's value is revealed it obviously provides information about both state and noise. The information about the state (the IG) will then be the whole received information (the entropy of random variable z) minus the information about the noise ν .

From the above we can see that in order for measurement z to be notably informative, three conditions should apply. First, its noise should have small entropy $\mathcal{H}(\nu)$, which also comes from general knowledge about measurement estimation. In addition, z should have large entropy $\mathcal{H}(z)$ from which we can conclude the second and third conditions: the *involved* variables ${}^l X$ from the measurement model should have high prior uncertainty (high prior entropy), as also their \tilde{A} (the Jacobian of measurement model at the linearization point of ${}^l X$) should contain high absolute values (the sign does not matter because of the quadratic term of \tilde{A} in (55)).

In the same way we can review the equation for *focused* IG (28). The first term $\frac{1}{2} \ln |I_m + A \cdot \Sigma_k \cdot A^T|$ measures the amount of information about whole state X_k , while the second term

$$\begin{aligned} &\frac{1}{2} \ln |I_m + A^U \cdot \Sigma_k^{U|F} \cdot (A^U)^T| \\ &= \frac{1}{2} \ln \frac{|\Psi + \widetilde{A}^U \cdot \Sigma_k^{U|F} \cdot (\widetilde{A}^U)^T|}{|\Psi|} = \mathcal{H}(z|X_k^F) - \mathcal{H}(\nu) \end{aligned} \quad (57)$$

measures the information given that X_k^F was provided, meaning information for only *unfocused* variables. The difference between total information and information of only *unfocused* variables will provide the information about the *focused* set X_k^F .

Such interpretation of IG's meaning through the entropy of future measurement and of its noise can be considered not only for the measurement selection problem, but also for the more general formulation from Section 2, thus constituting a possible direction for future research.

4. Application to different problem domains

In Section 3, we provided an efficient solution for a general BSP problem, considering both non-augmented and augmented cases. In this section, we discuss various problem domains of (augmented) BSP and show how our approach can be applied for each case. More concretely, we focus on sensor deployment (Section 4.1), active SLAM (Section 4.2), and graph reduction (Section 4.3), as specific non-augmented and augmented BSP applications. For the former, we develop a more computationally efficient variant of our approach. For each case, we first briefly formulate the problem and then describe our solution.

4.1. Sensor deployment

Sensor deployment is one of the most frequently researched problems of decision making. The basic idea is to measure a specific metric in domain space such as, e.g., temperature within a building space. The goal is to find the best locations for available sensors in order to estimate the metric in the entire domain in the most accurate way.

Typically discretization of the domain space is made due to computation complexity considerations. Thus, we have n available locations in the space, $L \doteq \{l_1, \dots, l_n\}$, where sensors can be deployed. The metric's values in these locations can be modeled as random variables and combined into a state vector: $X = \{x_1, \dots, x_n\}$.

Putting a sensor at location l_i will allow us to take measurement z_i at that location, which will provide information about the metric at place x_i . Assume that the measurement model of a sensor is known and is

$$z_i = h_i(x_i) + v_i, \quad v_i \sim \mathcal{N}(0, \Sigma_{v,i}). \quad (58)$$

In addition, correlation between different locations may be known *a priori*. Such prior can be presented as X 's joint distribution, $\mathbb{P}_0(X)$. Assuming that it is Gaussian, it may be represented as (Indelman, 2016; Krause et al., 2008; Zhu and Stein, 2006; Zimmerman, 2006)

$$X \sim \mathbb{P}_0(X) = \mathcal{N}(\mu, \Sigma_0) = \mathcal{N}^{-1}(\eta, \Lambda_0). \quad (59)$$

Note that, in practice, in typical sensor deployment problems Λ_0 is not actually available and Σ_0 is used instead. Nevertheless, in further formulation we assume that Λ_0 was calculated *a priori* (as Σ_0^{-1}) and therefore is available to us.

Finding the best sensor locations in order to estimate the metric in the most accurate way is another instance of information-based non-augmented BSP and therefore can be viewed through a prism of factor graphs (see Figure 10) as we show below.

Conceptually, the space of candidate actions in a sensor deployment setting contains all subsets of possible sensor locations $S \subseteq L$ with the usual constraint on cardinality of subset S , $|S| \leq c$, to represent that the number of sensors is limited. However, considering all subsets of size c is usually unrealistic as the number of all possible subsets $\binom{n}{c}$

is astronomical due to its combinatorial nature. Therefore, typically the problem is solved in a greedy way.

We propose a sub-optimal approach where a sequence of decisions must be made instead of one decision. During each decision we are looking for subset S' , $|S'| = c'$, with c' locations chosen from locations that were not yet selected. The optimal S' is the one that maximizes X 's estimation accuracy. The algorithm ends when the overall set of locations $S = \{S'_1, S'_2, \dots\}$ grows to cardinality of c . Note that the number of locations in each subset, c' , should be such that the number of S' candidates, $\binom{n}{c'}$, is small enough to be evaluated in a realistic time period. Thus, c' is scenario-dependent and should be selected manually.

More specifically, we assume that until time t_k the disjoint subsets $\{S'_1, \dots, S'_k\}$ of locations were selected, where each location subset $S'_j = \{l_j^1, \dots, l_j^{c'}\}$ provided measurements $Z_j = \{z_j^1, \dots, z_j^{c'}\}$. Given these measurements, the joint pdf at time t_k is

$$\mathbb{P}(X|Z_{1:k}) \propto \mathbb{P}_0(X) \prod_{j=1}^k \prod_{i=1}^{c'} \mathbb{P}(z_j^i|x_j^i), \quad (60)$$

where observation model $\mathbb{P}(z_j^i|x_j^i)$ is defined in (58).

MAP estimation of X according to information in (60) will provide current state *belief* $b_k[X] \doteq \mathbb{P}(X|Z_{1:k}) = \mathcal{N}(X_k^*, \Sigma_k)$, and following (7) the information matrix of $b_k[X]$ is $\Lambda_k = \Sigma_k^{-1} = \Lambda_0 + \sum_{j=1}^k \sum_{i=1}^{c'} (H_j^i)^T \cdot (\Sigma_{v,j,i})^{-1} \cdot H_j^i$ where $H_j^i \doteq \nabla_x h_j^i$ are the Jacobian matrices of observation model (58) for all measurement terms in (60), linearized about the current estimate X_k^* . Note that the *belief* $b_k[X]$ can be naturally represented by a factor graph G_k as was explained in Section 3.1 (see also Figure 10).

The next decision requires us to select next candidate action a : a location subset S'_{k+1} that will minimize posterior uncertainty. Therefore, candidate space contains all subsets of the form $S' \subseteq L \setminus \{S'_1 \cup \dots \cup S'_k\}$ and $|S'| = c'$. Each such candidate subset $a \equiv S' = \{l^1, \dots, l^{c'}\}$ will provide future measurements $Z' = \{z^1, \dots, z^{c'}\}$ and thus future *belief* $b_{k+1}[X]$ and its information matrix will be

$$b_{k+1}[X] = \mathbb{P}(X|Z_{1:k}, Z') \propto b_k[X] \prod_{i=1}^{c'} \mathbb{P}(z^i|x^i), \\ \Lambda_{k+1} = \Lambda_k + \sum_{i=1}^{c'} (H^i)^T \cdot (\Sigma_{v,i})^{-1} \cdot H^i. \quad (61)$$

Thus, the candidate S' introduces to G_k the factor set $\mathcal{F}(a)$, which contains exactly c' factors. Each of the factors is connected to one variable: the x_i that represents location of factor's sensor (see Figure 10).

Similarly to the general formulation in Section 2, stacking all new Jacobians in the above equation together into a single matrix and combining all noise matrices into a block-diagonal one will lead to (9). Hence, the optimal candidate subset S' will be the one that maximizes IG from (15).

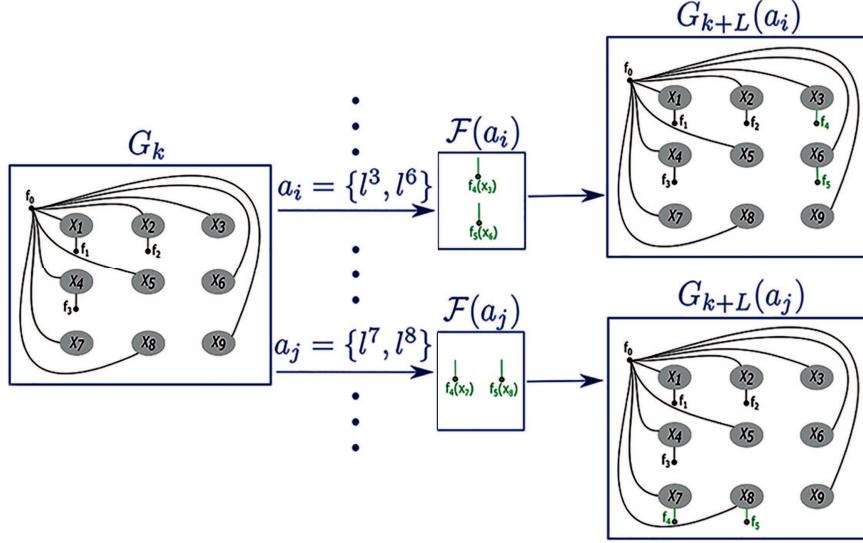


Fig. 10. Illustration of belief propagation in factor graph representation: sensor deployment scenario. The space is discretized through a grid of locations $L \doteq \{l_1, \dots, l_9\}$. Factor f_0 within prior factor graph G_k represents our prior belief about state vector, $\mathbb{P}_0(X)$. Factors f_1-f_3 represent measurements taken from sensors deployed at locations l_1, l_2 , and l_4 . Two actions $a_i = \{l^3, l^6\}$ and $a_j = \{l^7, l^8\}$ are considered, introducing into graph two new factor sets $\mathcal{F}(a_i)$ and $\mathcal{F}(a_j)$, respectively (colored in green). In this example value of c' is 2.

Note that the block-columns of Jacobian matrix $A \in \mathbb{R}^{m \times n}$ from (9) represent all possible sensor locations and block-rows represent new c' measurement factors from (61). As was mentioned before, only *involved* variables will have non-zero values in their block-columns. It is not difficult to show that in the sensor deployment problem, the *involved* variables are x_i that belong to locations in subset S' . Block-columns of all other variables in A will be zeros.

The rest of the problem definition (objective functions, *unfocused* and *focused* settings) for the sensor deployment problem is identical to the general formulation. In particular, in the *unfocused* setting the optimal S'_{k+1} will be found through

$$S'_{k+1} = \arg \max_{S' \subseteq X \setminus \{S'_1, \dots, S'_k\}, |S'|=c'} J_{IG}(S') = \frac{1}{2} \ln |I_m + A_{S'} \cdot \Sigma_k \cdot A_{S'}^T|, \quad (62)$$

where $A_{S'}$ is the Jacobian matrix of candidate S' .

Solution: Sequential rAMDL The above problem can be straightforwardly solved using the *rAMDL* approach, through (24) and (30). However, for each sequential decision the marginal covariance should be calculated for a set of variables involved in any of the candidate actions, and it is not difficult to show that this set will contain all as-yet unoccupied locations. In scenarios with a high number of possible sensor locations, this can negatively affect the overall time performance.

Here we present an enhanced approach, *Sequential rAMDL*, that performs the same sub-optimal sequence of decisions as described above, but uses *only* the prior covariance matrix Σ_0 , *without* recalculating covariance entries

after each decision. Such an approach gives an approximated solution (compared with the sub-optimal sequence of decisions described above), but without paying computation resources for expensive manipulation of high-dimensional matrices.

The first decision will be performed in exactly the same way: we will look for the best subset S'_1 of size c' that maximizes IG (62), for the *unfocused* case. However, upon finding such a subset, the estimation solution of the system will not be updated due to measurements from new sensors. Instead, in each next decision we will look for a subset S'_{k+1} that maximizes the following objective

$$S'_{k+1} = \arg \max_{S' \subseteq X \setminus \{S'_1, \dots, S'_k\}, |S'|=c'} J_{IG}(\tilde{S}) = \frac{1}{2} \ln |I_{\tilde{m}} + A_{\tilde{S}} \cdot \Sigma_0 \cdot A_{\tilde{S}}^T|, \\ A_{\tilde{S}} = \begin{pmatrix} A_{S'_1} \\ \vdots \\ A_{S'_k} \\ A_{S'} \end{pmatrix} \quad (63)$$

where $\tilde{S} \doteq \{S'_1, \dots, S'_k, S'\}$, and $A_{\tilde{S}}$ is a matrix with all appropriate Jacobians combined together.

Note that the sequential decision making through (63) will yield an exact solution, compared with sequential decision making through (62), if Jacobian matrices H^i (equation (61)) do not change after acquiring measurements from newly deployed sensors. This is the case, for instance, when linearization point X_k^* remains the same or when measurement model (58) is linear with respect to x_i (i.e. $z_i = x_i + v_i$). Otherwise, equation (63) will merely be the approximation of the above approach.

After looking into (63) one can see that matrix inside is actually

$$\begin{aligned} I_{\tilde{m}} + A_{\tilde{S}} \cdot \Sigma_0 \cdot A_{\tilde{S}}^T &= \begin{pmatrix} V_{S'_1} & Y_{S'_1, S'_2} & \cdots & Y_{S'_1, S'} \\ Y_{S'_2, S'_1} & V_{S'_2} & \cdots & Y_{S'_2, S'} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{S', S'_1} & Y_{S', S'_2} & \cdots & V_{S'} \end{pmatrix}, \\ V_{S'} &\doteq I_m + A_{S'} \cdot \Sigma_0 \cdot (A_{S'})^T, \quad Y_{S'_i, S'_j} \doteq A_{S'_i} \cdot \Sigma_0 \cdot A_{S'_j}^T, \end{aligned} \quad (64)$$

where $V_{S'}$ and $Y_{S'_i, S'_j}$ can be efficiently calculated (independently of state dimension) due to the sparsity of Jacobians. Moreover, after Σ_0 is calculated (or given) at the beginning of the algorithm, all its required entries are freely accessible through the entire runtime of the algorithm.

It can be seen that all diagonal matrices $V_{S'}$ were already calculated during the first decision and can be kept and reused. In addition, all the correlation matrices $Y_{S'_i, S'_j}$ (except for $Y_{S'_k, S'}$) were calculated in previous decisions. The only required calculation in every decision for each candidate S' is the matrix $Y_{S'_k, S'}$ and determinant of the combined matrix.

Our *unfocused Sequential rAMDL* approach can be seen as providing a little increase in the per-candidate calculation in order to escape the necessity of prior covariance calculation for each decision, similarly to the method of sequential informative measurements selection presented in Kaess and Dellaert (2009). This approach can be a good alternative to the *rAMDL* technique when one-time calculation part of *rAMDL* (Section 3.4) is more time-consuming than the part of candidates evaluation, as will be shown in our simulations. The *focused Sequential rAMDL* approach is also possible, by following similar derivations. Moreover, the same idea is applicable to other sequential domains such as the measurement selection problem.

4.2. Augmented BSP in unknown environments

In this section, we discuss a specific case of the augmented BSP problem from Section 2, considering a SLAM setting. Such a specification provides the reader with an illustrative example of the augmented BSP problem for better intuition.

Let us refine the definition. In the *smoothing* formulation of visual SLAM the state vector X_k represents robot poses per each time step, $\{x_0, \dots, x_k\}$, and landmarks mapped until now, $L_k \doteq \{l_1, \dots, l_{n_k}\}$. Further, we model robot motion dynamics and sensor observations through

$$x_{i+1} = f(x_i, u_i) + \omega_i, \quad \omega_i \sim \mathcal{N}(0, \Sigma_{\omega,i}) \quad (65)$$

$$z_{i,j} = h(x_i, l_j) + v_{i,j}, \quad v_{i,j} \sim \mathcal{N}(0, \Sigma_{v,i,j}), \quad (66)$$

where u_i is control at time t_i , $z_{i,j}$ represents observation of landmark l_j by a robot from position x_i at time t_i , and where ω_i and $v_{i,j}$ are the motion and measurement noises, respectively. Note that the motion model can be easily presented

in the form of a general factor model $r_i^j = h_i^j(X_i^j) + v_i^j$ from (3) by moving the left side to the right:

$$0 = f(x_i, u_i) - x_{i+1} + \omega_i = \bar{f}(x_i, x_{i+1}) + \omega_i, \quad \omega_i \sim \mathcal{N}(0, \Sigma_{\omega,i}). \quad (67)$$

The joint pdf for the SLAM problem at time t_k (or current belief) is then

$$\begin{aligned} b[X_k] &= \mathbb{P}(X_k | Z_{0:k}, u_{0:k-1}) \propto \mathbb{P}(x_0) \prod_{i=1}^k \\ &\quad \left\{ \mathbb{P}(x_i | x_{i-1}, u_{i-1}) \prod_{j=1}^{n_i} \mathbb{P}(z_{i,j} | x_i, l_j) \right\}, \end{aligned} \quad (68)$$

where $\mathbb{P}(x_0)$ is a prior on the robot's first pose, $Z_i = \{z_{i,1}, \dots, z_{i,n_i}\}$ represents all observations at time t_i , with n_i being the number of such observations. The motion and observation models $\mathbb{P}(x_i | x_{i-1}, u_{i-1})$ and $\mathbb{P}(z_{i,j} | x_i, l_j)$ are defined by (65) and (66). A factor graph representation, considering for simplicity only two landmarks l_1 and l_2 , is shown in Figure 11. Performing MAP inference over the belief $b[X_k]$, one can write $b[X_k] = \mathcal{N}(X_k^*, \Sigma_k)$, with appropriate mean vector X_k^* and covariance matrix Σ_k .

The space of candidate actions in SLAM setting contains all control sequences $u_{k+1:k+L-1}$, where L is the planning horizon and can vary between different candidates. Typically a finite set of candidates is pooled from this infinite space according to their relevance to robot's current destination or to loop-closure maneuver, for example through simulation (Stachniss et al., 2005) and sampling (Agha-Mohammadi et al., 2014; Prentice and Roy, 2009). Similar to (6), future belief $b[X_{k+L}] \doteq \mathbb{P}(X_{k+L} | Z_{0:k+L}, u_{0:k+L-1})$ for particular candidate action $a = u_{k+1:k+L-1}$ can be explicitly written as

$$b[X_{k+L}] \propto b[X_k] \prod_{l=k+1}^{k+L} \left\{ \mathbb{P}(x_l | x_{l-1}, u_{l-1}) \prod_{j=1}^{n_l} \mathbb{P}(z_{l,j} | x_l, l_j) \right\}, \quad (69)$$

where X_{k+L} is the state vector at the L th lookahead step. It contains all variables from the current state vector X_k and is augmented by new robot poses $X_{new} = \{x_{k+1}, \dots, x_{k+L}\}$. Also note that in (69) we consider only new observations of landmarks that were already mapped until time t_k . It is also possible to reason about observing not as-yet mapped landmarks (Indelman, 2015a), but it is outside the scope of this paper.

Following the model from Section 3.1, the candidate's factor graph $G(a) = (\mathcal{F}^{new}(a), X_{new}, \mathcal{E}_{new})$ will contain all new robot poses connected by motion model factors $\mathcal{F}^{new}(a) = \{f_{k+1}^M, \dots, f_{k+L-1}^M\}$ with appropriate motion models $\{\bar{f}(x_{k+1}, x_{k+2}), \dots, \bar{f}(x_{k+L-1}, x_{k+L})\}$, whereas factors from $\mathcal{F}^{conn}(a)$, which connect old variables X_k and new variables X_{new} , will contain one motion model factor f_k^M (with motion model $\bar{f}(x_k, x_{k+1})$) and all of observation model factors connecting new poses with observed landmarks (see Figure 11).

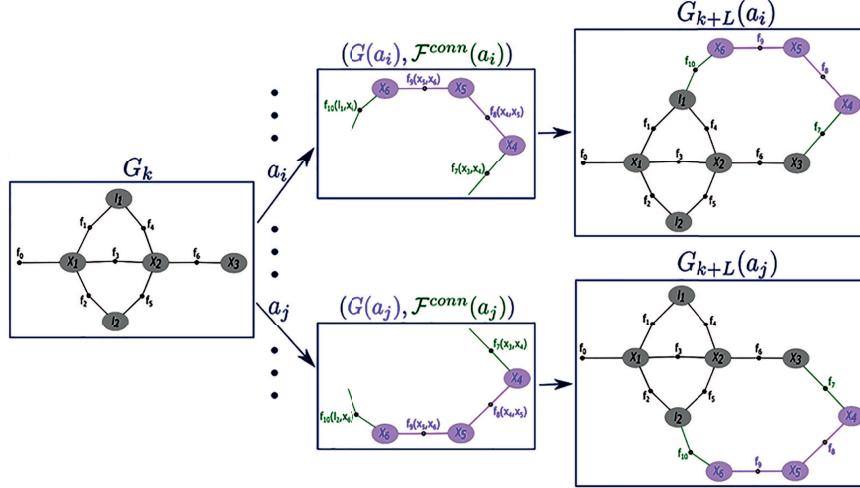


Fig. 11. Illustration of belief propagation in factor graph representation: SLAM scenario. Nodes x_i represent robot poses, while nodes l_i represent landmarks. Factor f_0 is prior on robot's initial position x_1 ; factors between pose and landmark represent the motion model (65); factors between pose and pose represent the observation model (66). Two actions a_i and a_j are considered, performing loop-closure to re-observe landmarks l_1 and l_2 , respectively. Both actions introduce their own factor graphs $G(a_i)$ and $G(a_j)$ (colored in pink) that are connected to prior G_k through factor sets $\mathcal{F}^{conn}(a_i)$ and $\mathcal{F}^{conn}(a_j)$ (colored in green), respectively.

Following the general formulation, the posterior information matrix of belief $b[X_{k+L}]$, i.e. Λ_{k+L} , can be constructed by first augmenting the current information matrix $\Lambda_k \equiv \Sigma_k^{-1}$ with L zero block-rows and block-columns, each block having dimension n_p of robot pose variable, to obtain $\Lambda_{k+L}^{Aug} \in \mathbb{R}^{N \times N}$ with $N = n + L \cdot n_p$, and thereafter adding to it new information, as illustrated in Figure 1 (see e.g. Indelman et al., 2015):

$$\Lambda_{k+L} = \Lambda_{k+L}^{Aug} + \sum_{l=k+1}^{k+L} \left\{ F_l^T \cdot \Sigma_{\omega,l}^{-1} \cdot F_l + \sum_{j=1}^{n_l} H_j^T \cdot \Sigma_{v,l,j}^{-1} \cdot H_j \right\}, \quad (70)$$

where $F_l \doteq \nabla_x f$ and $H_j \doteq \nabla_x h$ are augmented Jacobian matrices of all new factors in (69) (motion and observation terms all together), linearized about the current estimate of X_k and about initial values of newly introduced robot poses.

Again, after stacking together all new Jacobians in the above equation and combining all noise matrices into a block-diagonal matrix, we obtain the same posterior information expression as in (10).

Note that the block-columns of matrix $A \in \mathbb{R}^{m \times N}$ from (10) represent all old robot poses, mapped until now landmarks, and new robot poses from L -horizon future. Here A 's block-rows represent new motion and observation factors from (69). As mentioned before, only *involved* variables will have non-zero values in their block-columns. It is not difficult to see that in SLAM the *involved* ones are: all new robot poses, current robot pose x_k , and all landmarks that will be observed following the current candidate's actions. Block-columns of all other variables in A will be zeros.

The rest of the problem definition (objective functions, *unfocused* and *focused* settings) for the active SLAM problem is identical to the general formulation in Section 2.

Solution: rAMDl applied to SLAM The augmented BSP problem for the SLAM case, described in the previous section, can be naturally solved by our general approach from Section 3.3. However, we go one step further and provide a solution tailored specifically to the SLAM domain, as an example of applying *rAMDl* to a real problem and in order to show the underlying structure of the SLAM solution.

First, let us model informative partitions of Jacobian matrices B and D from (34), B_{old} , B_{new} , and D_{new} (see also Figure 7), for one of the candidate actions, action a . As was mentioned above, the factors from action's factor graph $G(a)$, $\mathcal{F}^{new}(a)$, contain all new motion model factors from (69), except for factor f_k^M . Therefore, D_{new} will have the following form:

$$D_{new} = \begin{pmatrix} \text{block-row for } \bar{f}(x_{k+1}, x_{k+2}) \\ \vdots \\ \text{block-row for } \bar{f}(x_{k+L-1}, x_{k+L}) \end{pmatrix} = \Psi_{new}^{-\frac{1}{2}} \cdot \begin{pmatrix} (x_{k+1}) & (x_{k+2}) & (x_{k+3}) & (x_{k+4}) \\ \mathbb{F}_{k+1} & -I & 0 & 0 \\ 0 & \mathbb{F}_{k+2} & -I & 0 \\ 0 & 0 & \mathbb{F}_{k+3} & -I \\ 0 & 0 & 0 & \mathbb{F}_{k+4} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\begin{aligned} & \cdots \quad (\underline{x}_{k+L-2}) \quad (\underline{x}_{k+L-1}) \quad (\underline{x}_{k+L}) \\ & \left(\begin{array}{cccc} \cdots & 0 & 0 & 0 \\ \ddots & \vdots & \vdots & \vdots \\ \cdots & -I & 0 & 0 \\ \cdots & \mathbb{F}_{k+L-2} & -I & 0 \\ \cdots & 0 & \mathbb{F}_{k+L-1} & -I \end{array} \right) \\ & \doteq \Psi_{new}^{-\frac{1}{2}} \cdot \tilde{D}_{new} \end{aligned} \quad (71)$$

where $\mathbb{F}_{k+l} \doteq \nabla_x f|_{x=x_{k+l}}$ is the Jacobian of motion model function f from (65) with respect to x_{k+l} , $-I$ is Jacobian of \tilde{f} from (67) with respect to second pose and is actually an identity matrix with dimension equal to the dimension of the robot pose. Matrix Ψ_{new} is block-diagonal, combining all noise matrices of $\mathcal{F}^{new}(a)$ factors. In addition, we denote by \tilde{D}_{new} the Jacobian entries of D_{new} not weighted by factors' noise Ψ_{new} .

Assume that following a 's controls the set of landmarks $L_a \subseteq L_k$ will be observed. In addition, define the set of all new observation factors $\mathcal{F}^{obs}(a)$ as

$$\begin{aligned} \mathcal{F}^{obs}(a) = & \{ \text{factor } f_i^O \text{ with observation model } h_i(x, l) : \\ & x \in X_{new}, l \in L_a, 1 \leq i \leq n_o \}, \end{aligned} \quad (72)$$

where n_o is the number of such factors. Thus, the *connecting* factors are $\mathcal{F}^{conn}(a) = \{f_k^M, \mathcal{F}^{obs}(a)\}$; and *involved* old variables will be ${}^lX_{old} = \{L_a, x_k\}$, containing x_k because of first factor's motion model $f(x_k, x_{k+1})$. Therefore, ${}^lB_{old}$ and B_{new} will be

$$\begin{aligned} & \text{(columns of } L_a, x_k, x_{k+1}, \dots, x_{k+L}) \\ & \left({}^lB_{old} \ B_{new} \right) = \left(\begin{array}{c} \text{block-row for } \tilde{f}(x_k, x_{k+1}) \\ \text{block-row for } h_1 \\ \vdots \\ \text{block-row for } h_{n_o} \end{array} \right) \\ & (L_a) \quad (\underline{x}_k) \quad (\underline{x}_{k+1}) \quad \cdots \quad (\underline{x}_{k+L}) \end{aligned}$$

$$= \Psi_{conn}^{-\frac{1}{2}} \cdot \left(\begin{array}{ccccc} 0 & \mathbb{F}_k & -I & \cdots & 0 \\ \mathbb{H}_1^{L_a} & 0 & \mathbb{H}_1^{x_{k+1}} & \cdots & \mathbb{H}_1^{x_{k+L}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbb{H}_{n_o}^{L_a} & 0 & \mathbb{H}_{n_o}^{x_{k+1}} & \cdots & \mathbb{H}_{n_o}^{x_{k+L}} \end{array} \right) \quad (73)$$

$$\begin{aligned} {}^lB_{old} &= \Psi_{conn}^{-\frac{1}{2}} \cdot \begin{pmatrix} 0 & \mathbb{F}_k \\ \mathbb{H}_1^{L_a} & 0 \\ \vdots & \vdots \\ \mathbb{H}_{n_o}^{L_a} & 0 \end{pmatrix} = \Psi_{conn}^{-\frac{1}{2}} \cdot \begin{pmatrix} 0 & \mathbb{F}_k \\ \mathbb{H}^{L_a} & 0 \end{pmatrix}, \\ \mathbb{H}^{L_a} &\doteq \begin{pmatrix} \mathbb{H}_1^{L_a} \\ \vdots \\ \mathbb{H}_{n_o}^{L_a} \end{pmatrix} \end{aligned} \quad (74)$$

$$B_{new} = \Psi_{conn}^{-\frac{1}{2}} \cdot \begin{pmatrix} -I & \cdots & 0 \\ \mathbb{H}_1^{x_{k+1}} & \cdots & \mathbb{H}_1^{x_{k+L}} \\ \vdots & \ddots & \vdots \\ \mathbb{H}_{n_o}^{x_{k+1}} & \cdots & \mathbb{H}_{n_o}^{x_{k+L}} \end{pmatrix} = \Psi_{conn}^{-\frac{1}{2}} \cdot \begin{pmatrix} \mathbb{F} \\ \mathbb{H}^{X_{new}} \end{pmatrix},$$

$$\mathbb{F} \doteq (-I \cdots 0), \quad \mathbb{H}^{X_{new}} \doteq \begin{pmatrix} \mathbb{H}_1^{x_{k+1}} & \cdots & \mathbb{H}_1^{x_{k+L}} \\ \vdots & \ddots & \vdots \\ \mathbb{H}_{n_o}^{x_{k+1}} & \cdots & \mathbb{H}_{n_o}^{x_{k+L}} \end{pmatrix}, \quad (75)$$

where $\mathbb{H}_i^{L_a} \doteq \nabla_{L_a} h_i$ is the Jacobian of the i th observation factor h_i from (66) with respect to variables L_a , and thus only one of its block-columns, corresponding to an observed landmark, is non-zero. Here $\mathbb{H}_i^{x_{k+l}} \doteq \nabla_{x_{k+l}} h_i$ is the Jacobian of h_i with respect to x_{k+l} , and therefore is non-zero only if a factor's observation was taken from pose x_{k+l} . Matrix Ψ_{conn} is block-diagonal, combining all noise matrices of $\mathcal{F}^{conn}(a)$ factors.

As can be seen from the above, the Jacobian matrices ${}^lB_{old}$, B_{new} , and D_{new} are sparse and can be efficiently manipulated. More specifically, the information matrix of factor graph $G(a)$, $\Lambda_a = D_{new}^T \cdot D_{new}$, which is required in our approach, can be calculated quickly as a product of sparse matrices $\Lambda_a = \tilde{D}_{new}^T \cdot \Psi_{new}^{-1} \cdot \tilde{D}_{new}$ due to the formulation in (71); in addition, it can be shown to be singular and block-tridiagonal.

The matrix C_1 from (41) can also be reduced to the following form:

$$\begin{aligned} C_1 &= I_{m_{conn}} + \Psi_{conn}^{-\frac{1}{2}} \cdot \\ & \left(\begin{array}{cc} \mathbb{F}_k \cdot \Sigma_k^{M, x_k} \cdot \mathbb{F}_k^T & \mathbb{F}_k \cdot \Sigma_k^{M, \{x_k / L_a\}} \cdot (\mathbb{H}^{L_a})^T \\ \mathbb{H}^{L_a} \cdot \Sigma_k^{M, \{L_a / x_k\}} \cdot \mathbb{F}_k^T & \mathbb{H}^{L_a} \cdot \Sigma_k^{M, L_a} \cdot (\mathbb{H}^{L_a})^T \end{array} \right) \\ & \cdot \Psi_{conn}^{-\frac{1}{2}} \\ &= \Psi_{conn}^{-\frac{1}{2}} \cdot \\ & \left[\Psi_{conn} + \left(\begin{array}{cc} \mathbb{F}_k \cdot \Sigma_k^{M, x_k} \cdot \mathbb{F}_k^T & \mathbb{F}_k \cdot \Sigma_k^{M, \{x_k / L_a\}} \cdot (\mathbb{H}^{L_a})^T \\ \mathbb{H}^{L_a} \cdot \Sigma_k^{M, \{L_a / x_k\}} \cdot \mathbb{F}_k^T & \mathbb{H}^{L_a} \cdot \Sigma_k^{M, L_a} \cdot (\mathbb{H}^{L_a})^T \end{array} \right) \right] \\ & \cdot \Psi_{conn}^{-\frac{1}{2}} \doteq \Psi_{conn}^{-\frac{1}{2}} \cdot C_2 \cdot \Psi_{conn}^{-\frac{1}{2}}, \end{aligned} \quad (76)$$

$$C_2 \doteq \Psi_{conn} + \left(\begin{array}{cc} \mathbb{F}_k \cdot \Sigma_k^{M, x_k} \cdot \mathbb{F}_k^T & \mathbb{F}_k \cdot \Sigma_k^{M, \{x_k / L_a\}} \cdot (\mathbb{H}^{L_a})^T \\ \mathbb{H}^{L_a} \cdot \Sigma_k^{M, \{L_a / x_k\}} \cdot \mathbb{F}_k^T & \mathbb{H}^{L_a} \cdot \Sigma_k^{M, L_a} \cdot (\mathbb{H}^{L_a})^T \end{array} \right) \quad (77)$$

where $\Sigma_k^{M, \{x_k / L_a\}}$ is the prior cross-covariance between variables x_k and L_a .

In addition, C_1 's determinant and its inverse can be calculated through

$$|C_1| = \frac{|C_2|}{|\Psi_{conn}|}, \quad C_1^{-1} = \Psi_{conn}^{\frac{1}{2}} \cdot C_2^{-1} \cdot \Psi_{conn}^{\frac{1}{2}}. \quad (78)$$

Next, we can calculate term $B_{new}^T \cdot C_1^{-1} \cdot B_{new}$ from (39) as

$$\begin{aligned} B_{new}^T \cdot C_1^{-1} \cdot B_{new} &= (\mathbb{F}^T (\mathbb{H}^{X_{new}})^T) \cdot \\ &\Psi_{conn}^{-\frac{1}{2}} \cdot \Psi_{conn}^{\frac{1}{2}} \cdot C_2^{-1} \cdot \Psi_{conn}^{\frac{1}{2}} \cdot \Psi_{conn}^{-\frac{1}{2}} \cdot \left(\begin{array}{c} \mathbb{F} \\ \mathbb{H}^{X_{new}} \end{array} \right) \\ &= (\mathbb{F}^T (\mathbb{H}^{X_{new}})^T) \cdot C_2^{-1} \cdot \left(\begin{array}{c} \mathbb{F} \\ \mathbb{H}^{X_{new}} \end{array} \right) = \tilde{B}_{new}^T \cdot C_2^{-1} \cdot \tilde{B}_{new}, \end{aligned} \quad (79)$$

where $\tilde{B}_{new} \doteq \left(\begin{array}{c} \mathbb{F} \\ \mathbb{H}^{X_{new}} \end{array} \right)$ contains the Jacobian entries of B_{new} not weighted by factors' noise Ψ_{conn} . Then, the *unfocused* IG objective from (16) in the SLAM setting is given by

$$\begin{aligned} J_{IG}(a) &= \frac{n' \cdot \gamma}{2} - \frac{1}{2} \ln |\Psi_{conn}| + \frac{1}{2} \ln |C_2| + \frac{1}{2} \ln |\tilde{B}_{new}^T \cdot \\ &C_2^{-1} \cdot \tilde{B}_{new} + \tilde{D}_{new}^T \cdot \Psi_{new}^{-1} \cdot \tilde{D}_{new}|. \end{aligned} \quad (80)$$

Above we have shown in detail how our *rAMD* approach can be applied to information-based SLAM planning problem types. The derived equation (80) is very similar to the general solution from (39), having exactly the same runtime complexity. However, within both (80) and (77) we can see a clear separation between noise of factor model and the actual Jacobian entries. Such a separation can provide further theoretical insight about how different terms of the SLAM problem affect the information impact of candidate action $a = u_{k:k+L-1}$. Moreover, it can provide a good starting point for the derivation of $J_{IG}(a)$'s gradient with respect to $u_{k:k+L-1}$, which, in turn, can be used for gradient-descent algorithms that search for locally optimal controls (Indelman et al., 2015; Van Den Berg et al., 2012). Note the variable ordering in the above equation serves only for visualization; the derivation remains valid for an arbitrary variable ordering.

In addition, for the sake of completeness we also provide a SLAM-specific solution for *focused* cases, where we consider either reducing the entropy of the last pose ($X_{k+L}^F \equiv x_{k+L}$) or of all the mapped landmarks ($X_k^F \equiv L_k$). The corresponding derivation can be found in Appendices A.7 and A.8.

4.3. Graph reduction

It is a known fact that in long-term SLAM applications, the state dimension of smoothing techniques can grow unboundedly. In such cases, even the most efficient state-of-the-art estimation algorithms such as iSAM2 (Kaess et al., 2012) can become slow and will not support online operation. Approaches such as graph reduction and graph sparsification try to tackle the problem by reducing the number of

variables (Ila et al., 2010; Kretzschmar and Stachniss, 2012; Paull et al., 2016) and sparsifying entries of the information matrix (Carlevaris-Bianco et al., 2014; Huang et al., 2012; Mazurana et al., 2014; Vial et al., 2011), respectively.

Graph reduction requires us to first select nodes to expel. In such cases, having a state vector X with variables $\{x_1, \dots, x_n\}$, it would be logical to remove the most uncertain node, say x_i , without which the rest of the variables $\bar{X}_i \doteq \{X \setminus x_i\}$ would have the smallest entropy $\mathcal{H}(\bar{X}_i)$. In this section we outline a new approach for such a selection which is closely related to our *rAMD* technique.

Similarly to the *focused* objective function from (17), the best choice for expelled variable x_i^* among state variables will minimize the following objective function:

$$\begin{aligned} x_i^* &= \arg \min_{x_i \in X} J_{GR}(x_i) \\ &= \mathcal{H}(\bar{X}_i) = \frac{(n - n_x) \cdot \gamma}{2} - \frac{1}{2} \ln |\Lambda^{M, \bar{X}_i}|, \end{aligned} \quad (81)$$

where $\mathcal{H}(\bar{X}_i)$ is entropy of the state variables without x_i , and n_x is x_i 's dimension.

Using Equation (26) from our approach, in order to calculate Λ^{M, \bar{X}_i} , we can reduce our objective function to

$$J_{GR}(x_i) = \frac{(n - n_x) \cdot \gamma}{2} - \frac{1}{2} \ln |\Lambda| + \frac{1}{2} \ln |\Lambda^{x_i}|, \quad (82)$$

where Λ is the information matrix of the whole X , and Λ^{x_i} is its partition related to variable x_i .

Given that all x_i variables have the same dimension n_x , eventually we can conclude that optimal x_i^* will also minimize

$$x_i^* = \arg \min_{x_i \in X} J_{GR}(x_i) = \ln |\Lambda^{x_i}| \quad (83)$$

which practically implies calculating the determinant of every partition Λ^{x_i} and choosing the state variable x_i with minimal determinant value. In cases where all x_i are scalars, $|\Lambda^{x_i}|$ is just a value from the diagonal of the information matrix Λ . In cases where x_i 's dimension is n_x , we will have to calculate the determinants of n matrices, each one of dimension $n_x \times n_x$. Taking into account that n_x is usually not big at all (e.g. a 3D pose has six dimensions), the overall calculation is very fast and is just $O(n)$.

5. Alternative approaches

We compare the presented *rAMD* approach with two alternatives, namely the *From-Scratch* and *iSAM* techniques.

In *From-Scratch*, the posterior information matrix Λ_{k+L} is computed by adding new information $A^T \cdot A$, followed by the calculation of its determinant. In the *focused* scenario, the marginal information matrix of X_{k+L}^F is retrieved through the Schur complement performed on Λ_{k+L} , and its determinant is then computed.

The second alternative, uses the *iSAM* algorithm (Kaess et al., 2012) to incrementally update the posterior. Here the

(linearized) system is represented by a square root information matrix R_k , which is encoded, while exploiting sparsity, by the Bayes tree data structure. The posterior matrix R_{k+L} is acquired (e.g. via Givens rotations (Kaess et al., 2012) or another incremental factorization update method), and then the determinant is calculated $|\Lambda_{k+L}| = \prod_{i=1}^N r_{ii}^2$, with r_{ii} being the i th entry on the diagonal of triangular R_{k+L} . For the focused case, the marginal covariance matrix of X_{k+L}^F is computed by recursive covariance per-entry equations (Kaess and Dellaert, 2009) that exploit the sparsity of matrix R_{k+L} .

While the *iSAM* technique outperforms batch *From-Scratch*, it still requires calculating R_{k+L} for each action, which can be expensive, particularly in loop closures, and requires a copy/clone of the original matrix R_k . In contrast, in *rAMDL*, the per-candidate action calculation (e.g. in (37)) has constant complexity in general, given the prior marginal covariance terms that are calculated only once.

6. Computational complexity analysis

In this section, we analyze the computational complexity of the developed-herein family of *rAMDL* algorithms, and of alternative approaches from Section 5. We summarize the runtime complexity of different approaches in Table 4.

The computational complexity of *rAMDL* algorithms consists of two parts. First, there is a one-time computation of prior covariance entries $\Sigma_k^{M, I_{X_{old}}^U | F}$ (and sometimes $\Sigma_k^{I_{X_{old}}^U | F}$, see Section 3.4). Second, there is a per-candidate computation of an approach-specific objective function. Therefore, overall computational complexity is equal to $O(\text{one-time complexity}) + O(\text{per-action complexity} \times \text{number of candidate actions})$.

The prior covariance $\Sigma_k^{M, I_{X_{old}}^U}$ is calculated from the square root information matrix R_k using a recursive method as described in Golub and Plemmons (1980) and Kaess and Dellaert (2009). Its complexity is bounded by $O(n_{nz}^2 \cdot n)$, where n is the state dimension and n_{nz} is the number of non-zero entries in the R_k . However, typically its complexity depends on variable ordering within R_k and is much lower. For more detailed complexity analysis of this covariance recovery method, see Kaess and Dellaert (2009) and Ila et al. (2015). For cases where $\Sigma_k^{I_{X_{old}}^U | F}$ is also required, we consider the method described in Algorithm 2 that calculates the prior conditional covariance entries through the Schur complement. Its complexity is bounded by the inverse calculation $(\Sigma_k^{M, F})^{-1}$, which is $O(|X^F|^3)$ with $|X^F|$ being the dimension of *focused* variables.

The per-action complexity of *rAMDL* algorithms was analyzed in Sections 3.2–3.3, next to the equation definition of each approach, and is summarized in Table 4.

The complexity of both *focused* and *unfocused* *From-Scratch* approaches is governed by the term $O(N^3)$, with N being the posterior state dimension. In the *unfocused* case, the *From-Scratch* method calculates a determinant of dimension N for each candidate action. In the *focused* case, the *From-Scratch* method calculates a Schur complement per each candidate, which is typically more computationally expensive than the computation of a matrix determinant.

The *iSAM* technique propagates posterior belief per each candidate action and then evaluates the appropriate posterior entropy (*unfocused* or *focused*). This belief propagation is done efficiently through a Bayes tree, yet its runtime complexity is difficult to analyze. It is declared in Kaess et al. (2012) that typically belief propagation takes $O(N^{1.5})$. In the *unfocused* case, $|\Lambda_{k+L}|$ is calculated in $O(N)$ through diagonal entries of R_{k+L} , providing a final per-candidate complexity of $O(N^{1.5})$ for the *iSAM Unfocused* approach. In the *focused* case the marginal covariance matrix of *focused* variables X_{k+L}^F is computed via the method from Kaess and Dellaert (2009), which is $O(N_{nz}^2 \cdot N)$ where N is the posterior state dimension and N_{nz} is the number of non-zero entries in the posterior square root information matrix R_{k+L} . Further, the determinant of the marginal covariance matrix is computed, which takes $O(|X^F|^3)$. In total the *iSAM Focused* approach will have per-candidate complexity of $O(N^{1.5} + N_{nz}^2 \cdot N + |X^F|^3)$.

From Table 4 we can see that the per-candidate complexity of *rAMDL* does not depend on the state dimension while the complexity of both *iSAM* and *From-Scratch* does. For example, in the *unfocused* case, *rAMDL* requires $O(\mathcal{M}(\mathcal{F}(a))^3)$, while *iSAM* and *From-Scratch* need $O(N^{1.5})$ and $O(N^3)$, respectively. Since $\mathcal{M}(\mathcal{F}(a))$ represents the total dimension of newly introduced factors by candidate action a , it is typically considerably smaller than the posterior state dimension N , i.e. $\mathcal{M}(\mathcal{F}(a)) \ll N$. This difference makes the *rAMDL* technique significantly faster, as will be shown further in Section 7.

Note that we do not analyze the BSP complexity in terms of the horizon lag of a candidate action. Instead, we use dimensions of different components of prior and posterior factor graphs since these give a better insight on the real complexity of the factor-graph-based algorithms presented herein.

7. Results

In this section, we evaluate the performance of the proposed approach and compare it with alternative approaches considering *unfocused* and *focused* instantiations of several fundamental problems: sensor deployment, measurement selection, and autonomous navigation in unknown environments.

In sensor deployment, each candidate action represents a set of possible locations for deploying a sensor, with a

Table 4. The presented approaches and their alternatives, along with the corresponding runtime complexity. Used symbols are: n is the dimension of prior state vector X_k ; n_{nz} is the number of non-zero entries in a prior square root information matrix R_k ; n' is the dimension of newly introduced variables X_{new} by candidate action a ; $N = n + n'$ is the dimension of posterior state vector X_{k+L} ; N_{nz} is the number of non-zero entries in a posterior square root information matrix R_{k+L} ; $\mathcal{F}(a)$ represents newly introduced factors by candidate action a ; $\mathcal{M}(\mathcal{F}(a))$ is the total dimension of newly introduced factors $\mathcal{F}(a)$; $\mathcal{F}^{conn}(a)$ represents a subset of factors from $\mathcal{F}(a)$ that involves at least one old variable from X_k ; $\mathcal{M}(\mathcal{F}^{conn}(a))$ is the total dimension of factors in $\mathcal{F}^{conn}(a)$. Further details can be found in Section 6.

Approach	Per-action complexity	One-time complexity
Non-augmented BSP rAMDL approaches		
<i>rAMDL Unfocused</i> , equation (24)	$O(\mathcal{M}(\mathcal{F}(a))^3)$	$O(n_{nz}^2 \cdot n)$
<i>rAMDL Focused</i> , equation (30)	$O(\mathcal{M}(\mathcal{F}(a))^3)$	$O(n_{nz}^2 \cdot n + X^F ^3)$
Augmented BSP rAMDL approaches		
<i>rAMDL Unfocused</i> , equations (37), (38)	$O(\mathcal{M}(\mathcal{F}(a))^3)$	$O(n_{nz}^2 \cdot n)$
<i>rAMDL-Extended Unfocused</i> , equations (39), (41)	$O(\mathcal{M}(\mathcal{F}^{conn}(a))^3 + n'^3)$	$O(n_{nz}^2 \cdot n)$
<i>rAMDL Focused New</i> , equations (42), (38)	$O(\mathcal{M}(\mathcal{F}(a))^3 + n'^3)$	$O(n_{nz}^2 \cdot n)$
<i>rAMDL-Extended Focused New</i> , equations (44), (41)	$O(\mathcal{M}(\mathcal{F}^{conn}(a))^3 + n'^3)$	$O(n_{nz}^2 \cdot n)$
<i>rAMDL Focused Old</i> , equations (45), (46)	$O(\mathcal{M}(\mathcal{F}(a))^3 + n'^3)$	$O(n_{nz}^2 \cdot n + X^F ^3)$
<i>rAMDL-Extended Focused Old</i> , equations (47), (48)	$O(\mathcal{M}(\mathcal{F}^{conn}(a))^3 + n'^3)$	$O(n_{nz}^2 \cdot n + X^F ^3)$
Alternative approaches		
<i>From-Scratch Unfocused & Focused</i> , Section 5	$O(N^3)$	-
<i>iSAM Unfocused</i> , Section 5	$O(N^{1.5})$	-
<i>iSAM Focused</i> , Section 5	$O(N^{1.5} + N_{nz}^2 \cdot N + X^F ^3)$	-

single sensor deployment corresponding to a unary factor. We consider a non-myopic setting and let each candidate action represent two sensor locations. In the measurement selection problem, we consider a greedy decision-making paradigm in the context of aerial visual SLAM with pairwise factors.

Further, we present simulation results of applying our approach to autonomous navigation in unknown environments (both *unfocused* and *focused* cases) on synthetic and real-world datasets. The robot has to visit a sequence of goals while minimizing an objective function comprising two terms (to be defined in the sequel): distance to goal, and an uncertainty metric. Candidate actions are non-myopic and involve multiple new and old state variables.

In all cases, the presented simulations reflect the computational performance of different approaches developed within this paper, and alternative methods that are described in Section 5. In Table 5 we summarize the considered approaches in each of the above problems, and refer to appropriate equations for each case. Moreover, we emphasize that all techniques presented herein, *rAMDL* and their alternatives, are mathematically identical in the sense that they determine identical optimal actions given a set of candidate actions.

The code is implemented in Matlab; for measurement selection and autonomous navigation we use the GTSAM library (Dellaert, 2012; Kaess et al., 2012). All scenarios were executed on a Linux machine with an i7 2.40 GHz processor and 32 GB of memory.

7.1. Sensor deployment (*focused* and *unfocused*)

In this section, we apply our approach *rAMDL* to the sensor deployment problem, considering both *focused* and *unfocused* instantiations of this problem (see Section 4.1 for a detailed formulation). The prior of the sensor field is represented by the information matrix Λ and it is dense as usual in the problem of sensor deployment.

We compare our *rAMDL* approach against the batch *From-Scratch* technique that is described in Section 5, and also against the *Sequential rAMDL* described in Section 4.1, which does not require marginal covariance computation at each decision.

While decision making involves evaluating the impact of an action for all candidate actions \mathcal{A} , we first analyze action impact calculation ($J_{IG}(a)$) for a single candidate $a \in \mathcal{A}$, comparing *rAMDL* with the *From-Scratch* approach for the *unfocused* case. Figure 12 shows these timing results as a function of state dimension n (Figure 12a) and as function of Jacobian A 's height m (Figure 12b). As expected, n effects the running time of both the *From-Scratch* technique and calculation of Σ_k (inverse of Λ_k , which is dense in the case of sensor deployment), while m only effects the calculation of the IG objective of *rAMDL* (red line).

One might think, based on Figure 12a and (b), that the proposed approach is slower than the *From-Scratch* alternative because of the time needed for inverse calculation to obtain Σ_k . Yet, it is exactly here that our calculation re-use paradigm comes into play (see Section 3.4): this calculation is performed only *once* for all candidate actions \mathcal{A} ,

Table 5. Considered approaches in different problems from Section 7, along with their appropriate equations

Problem	Approach	Equations/Section
Sensor Deployment, Section 7.1	<i>rAMD</i> L Unfocused <i>rAMD</i> L Focused Sequential <i>rAMD</i> L Partitions <i>From-Scratch</i> , Unfocused & Focused	Equation (24) Equation (30) Equations (63), (64) Givens rotations & Equation (27) Section 5
Measurement selection, Section 7.2	<i>rAMD</i> L Unfocused <i>iSAM</i> Unfocused	Equation (24) Section 5
Autonomous Navigation, Section 7.3	<i>rAMD</i> L Unfocused <i>rAMD</i> -Extended Unfocused <i>rAMD</i> L Focused New <i>rAMD</i> -Extended Focused New <i>rAMD</i> L Focused Old <i>rAMD</i> -Extended Focused Old <i>From-Scratch</i> , Unfocused & Focused <i>iSAM</i> , Unfocused & Focused	Equations (37), (38) Equations (39), (41) Equations (42), (38) Equations (44), (41) Equations (45), (46) Equations (47), (48) Section 5 Section 5

while, given Σ_k , calculating IG for each action is no longer a function of n .

The substantial reduction in running time of our approach, compared with the *From-Scratch* approach, can be clearly seen in Figure 12c, which considers the entire decision-making problem, i.e. evaluation of all candidate actions \mathcal{A} . The figure shows the running time for sequential decision making, where at each time instant we choose the best locations of two sensors, with around $|\mathcal{A}| = 10^5$ candidate actions. The number of all sensor locations is $n = 625$ in this example. Overall, 15 sequential decisions were made. As seen, decision making using our approach requires only about 5 seconds, while the *From-Scratch* approach requires about 400 seconds.

The *Sequential rAMD*L technique is not always faster than *rAMD*L, as can be seen in Figure 12c. As described in Section 4.1 this technique will be superior in cases where the covariance calculation makes up a significant part of the whole decision calculation. We can see that this is the case in Figure 12f, where the number of candidates is limited to 100, and where the covariance calculation time is the biggest part in the decision making of the *rAMD*L approach. There we can see that *Sequential rAMD*L provides better performance than all other alternatives.

We now consider the *focused* version of the sensor deployment problem (17). In other words, the goal is to find sensor locations that maximally reduce uncertainty about chosen *focused* variables X^F . We have 54 such variables, which are shown in Figure 13c, while the rest of the problem setup remains identical to the *unfocused* case.

In Figure 13 we show the corresponding results of *rAMD*L, compared with the *From-Scratch*. The latter first calculates, for each candidate action, the posterior $\Lambda^+ = \Lambda + A^T A$, followed by calculation of the Schur complement $\Lambda^{M,F}$ of the *focused* set X^F , and its determinant $|\Lambda^{M,F}|$ in order to obtain $J_{\mathcal{H}}^F(a)$ (17). We also compare it with an additional approach, termed *Partitions*, which uses

Givens rotations to compute R^+ and instead of performing the Schur complement, calculates the posterior entropy of the *focused* set via (27). This equation is one of our main contributions, being an essential step in the derivation of our approach, and we show here that compared with the *From-Scratch* technique, the *Partitions* approach is considerably faster. Our *focused* approach applies the matrix determinant lemma, transforming (27) into (30), which, together with the *re-use* concept (Section 3.4), makes it possible to drastically reduce the running time as shown in Figure 13a (10 seconds versus about 1000 seconds in *Partitions* and 1300 seconds in *From-Scratch*).

7.2. Measurement selection in SLAM

In this section, we consider a measurement selection problem (see Section 3.5) within a visual aerial SLAM framework, where one has to choose the most informative image feature observations from the numerous image features typically calculated for each incoming new image.

We demonstrate application of our approach in this problem, which, in contrast to the sensor selection problem, involves pairwise factors of the type $p(z_{i,j}|x_i, l_j)$, relating between an image observation $z_{i,j}$, camera pose x_i , and landmark l_j .

A top view of the considered aerial scenario is shown in Figure 14a: an aerial vehicle performs visual SLAM, mapping the environment and at the same time localizing itself. The figure shows the landmarks and the estimated trajectory, along with the uncertainty covariance for each time instant. One can clearly see the impact of loop-closure observations on the latter. In the considered scenario there are about 25,000 landmarks and roughly 500 image features in each view.

The number of image features that correspond to previously seen landmarks is relatively small (around 30–50, see Figure 14b), which corresponds to a much smaller set of

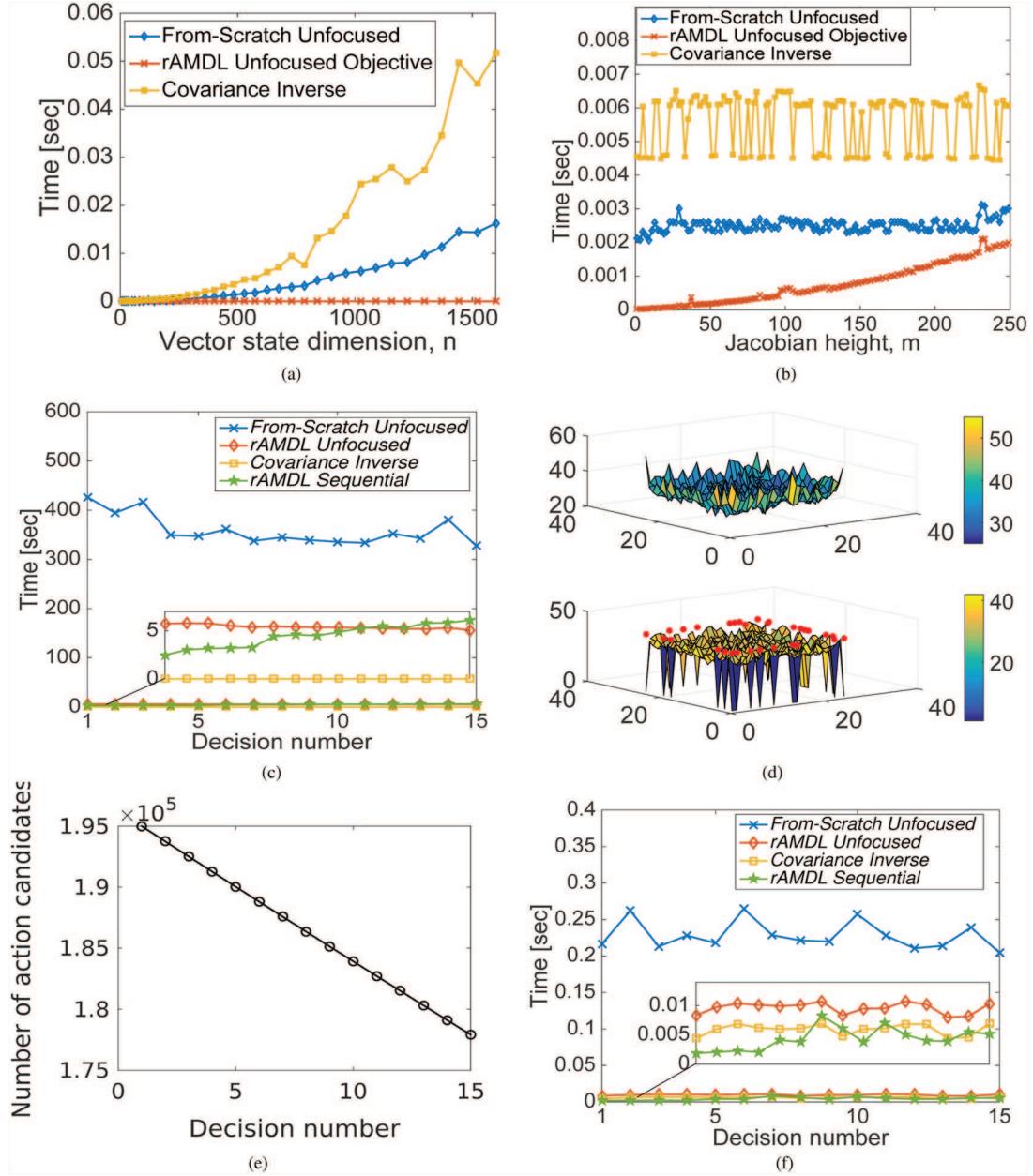


Fig. 12. *Unfocused* sensor deployment scenario. Running time, for simplicity termed as “Time” within graphs, for calculating the impact of a single action as a function of state dimension n (a) and as a function of Jacobian A ’s height m (b). In (a) $m = 2$ and in (b) $n = 625$. *rAMDL Unfocused Objective* represents only the calculation time of candidates’ impacts (IG objective for all actions), without one-time calculation of prior covariance; *Covariance Inverse* represents the time it took to calculate covariance matrix Σ_k from dense information matrix Λ_k , $\Sigma_k = \Lambda_k^{-1}$. (c) Running time for sequential decision making, i.e. evaluating the impact of all candidate actions, each representing candidate locations of two sensors. (d) Prior and final uncertainty of the field, with red dots marking selected locations; note locations with negligible uncertainty are the observed locations where sensors were deployed. (e) Number of action candidates per decision. (f) Running time for sequential decision making, with the number of candidates limited to 100.

actions \mathcal{A} compared with the sensor deployment problem (Section 7.1) where the cardinality of \mathcal{A} was huge (10^5). Such a dataset was chosen on purpose in order to show the behavior of the proposed algorithm in domains with a small number of candidates. In addition, in this scenario the actions are myopic since the measurements are greedily selected.

In addition, as opposed to the sensor deployment problem, in the current problem, state dimensionality n grows with time as more poses and landmarks are added into inference (see Figure 14c) and the information matrix is sparse.

Figure 14d shows the timing results for choosing 10 most informative image observations comparing the proposed

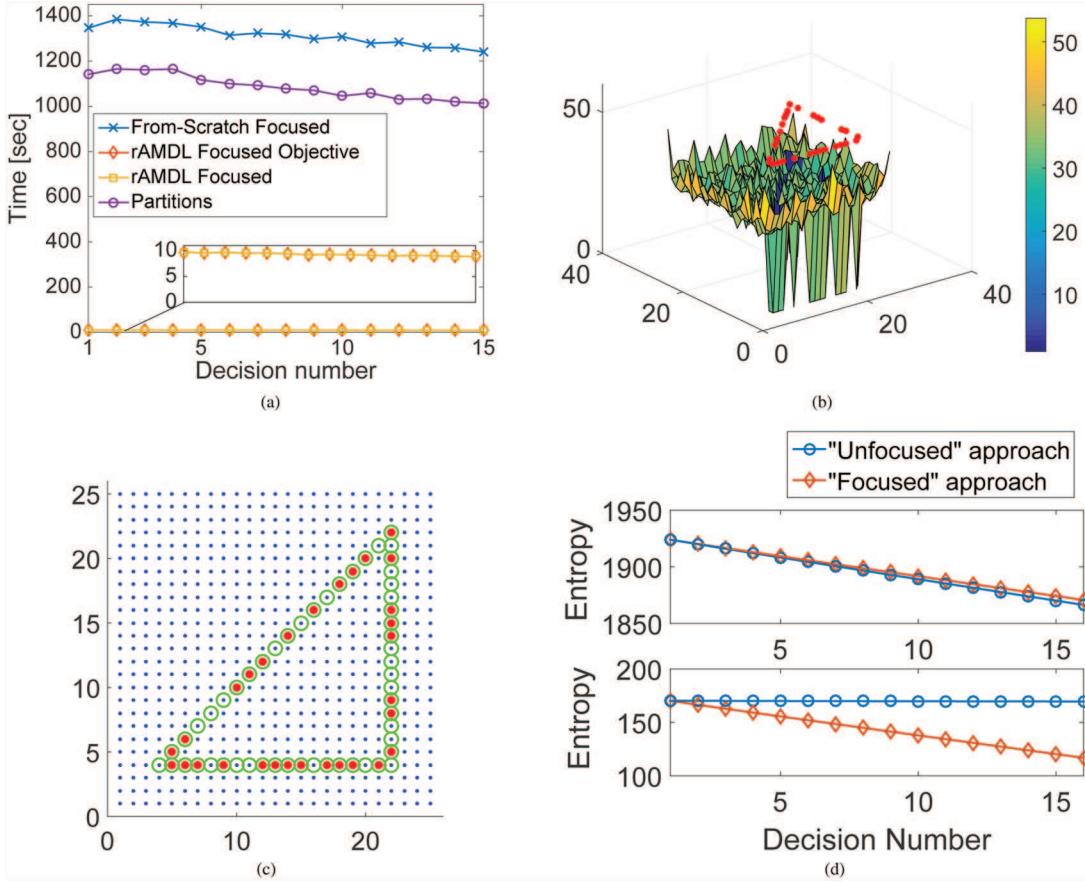


Fig. 13. *Focused* sensor deployment scenario. (a) Overall time it took to make a decision with different approaches; *rAMDL Focused Objective* represents only the calculation of candidates' impacts (IG objective for all actions) while *rAMDL Focused* represents both one-time calculation of prior covariance Σ_k and candidates' evaluation. (b) Final uncertainty of the field, with red dots marking selected locations; note locations with negligible uncertainty are the observed locations where sensors were deployed. (c) *Focused* set of variables (green circles) and locations selected by algorithm (red dots). (d) Overall system entropy (above) and entropy of *focused* set (bottom) after each decision, with the blue line representing the *unfocused* algorithm and the red line representing the *focused* algorithm. Note that all *unfocused* methods make exactly the same decisions, with difference only in their runtime complexity. The same is also true for all *focused* methods.

rAMDL with the *iSAM* approach (computing the posterior square root information matrix using *iSAM*, and then calculating the determinant; see Section 5). This BSP problem is solved sequentially, each time a new image is acquired. As seen, our approach *rAMDL* is substantially faster than the *iSAM*, while providing identical results (the same decisions). In particular, the running time of the *iSAM* approach for the last time index with $n = 10,000$ state dimensionality, is around 7 seconds. In contrast, *rAMDL* takes about 0.05 seconds: calculation time of action impacts via calculation re-use is negligible (red line), while the one-time calculation of marginal covariance $\Sigma_k^{M_{X_{All}}}$ (yellow line) is performed efficiently, in the current implementation, via sparse factorization techniques using GTSAM (Dellaert, 2012; Kaess et al., 2012).

7.3. Autonomous navigation in an unknown environment

In this section we present simulation results of applying our approach to autonomous navigation in unknown environments (both *unfocused* and *focused* cases) on synthetic and real-world datasets.

In the synthetic scenario (Figure 15c), the robot's task is to visit a predefined set of goals $\mathcal{G} = \{G_1, \dots, G_{14}\}$ in an unknown environment while reducing an uncertainty metric. More specifically, the state vector X_k contains all robot poses and landmarks mapped until time t_k (see Section 4.2). At each point of time, the robot autonomously selects an optimal non-myopic action $a = u_{k:k+L-1}$, performs its first control u_k , and subsequently observes landmarks within a radius of 900 meters from its new position. The

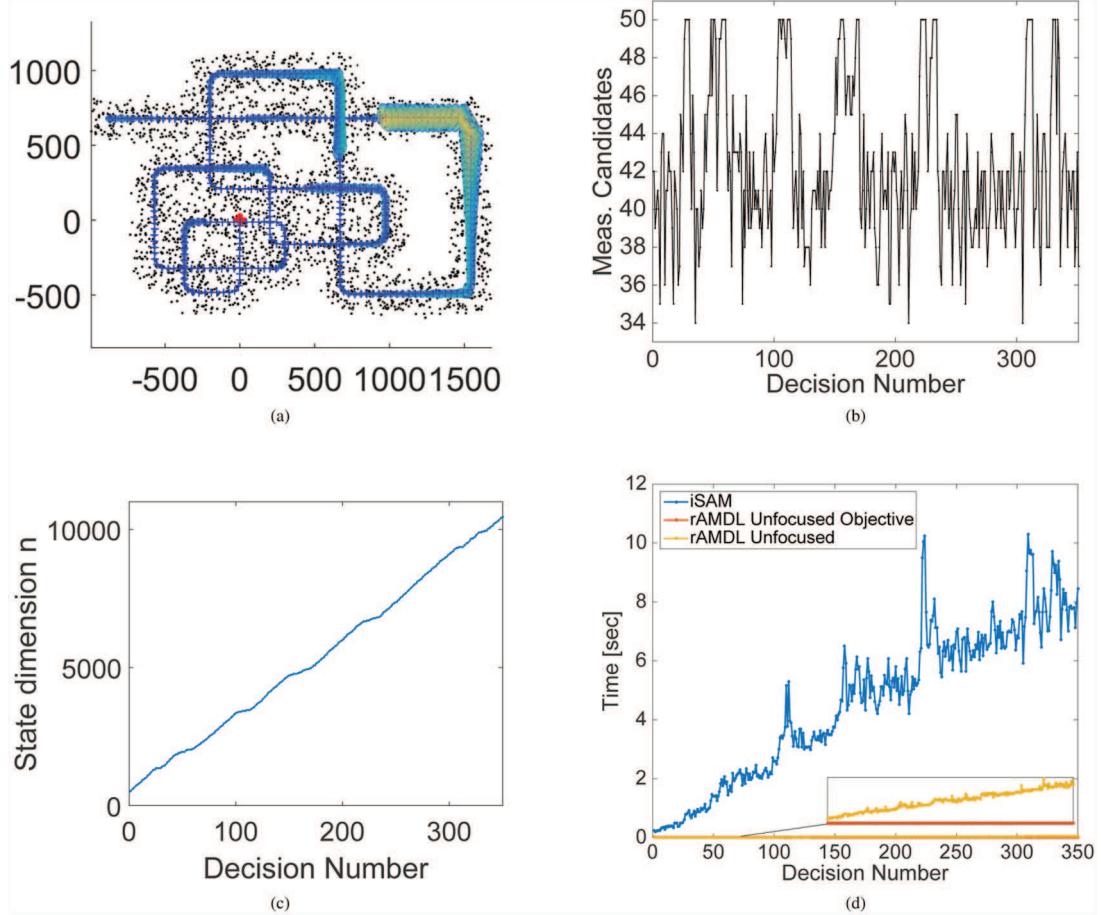


Fig. 14. Measurement selection scenario. (a) Simulated trajectory of a robot; black dots are the landmarks, blue marks and surrounding ellipses are the estimated trajectory along with the uncertainty covariance for each time instant, red mark is the robot's initial pose. (b) Number of measurement candidates per decision. (c) State's dimension n per decision. (d) Overall time it took to evaluate the impacts of all poses' measurements, with different approaches; *rAMDL Unfocused Objective* represents only the calculation of candidates' impacts (IG objective for all actions) while *rAMDL Unfocused* represents both one-time calculation of marginal covariance $\Sigma_k^{M,X_{All}}$ and candidates' evaluation.

landmarks can be either old (seen before) or new (seen for the first time). Next, a SLAM solution is calculated given these new observations and a motion model. To that end, the factor graph from the previous inference time is updated with the new observation and motion model factors, and new variable nodes, representing the current robot pose and new landmarks, are added (see Section 4.2). Afterwards, the next action is chosen and executed, and so on.

The set of candidate actions \mathcal{A} contains one action that navigates the robot from its current pose x_k to the current goal G_i from a predefined set \mathcal{G} (see Figure 15c); it also contains a set of “loop-closure” actions that are generated in the following way. We start by taking all mapped landmarks within a radius of 1000 meters from the robot's current pose. We cluster these landmarks, similarly to Kim and Eustice (2014), and obtain a set of landmark clusters. Each cluster's center g_{cl} represents a “loop-closure” goal and contributes

a “loop-closure” action $a_{cl} = u_{k:k+L-1}$ that navigates the robot from x_k to g_{cl} .

Each action in \mathcal{A} , taking the robot from x_k to location g , is constructed by first discretizing the map into a grid and thereafter searching for an optimal trajectory from the current position to g using an A^* search algorithm, similarly to Kim and Eustice (2014) and Indelman et al. (2015). The optimal candidate action is chosen by evaluating an objective that has the following two terms: distance to the current goal G_i and a term of uncertainty

$$J(a) = d(x_{k+L}, G_i) + J_{\mathcal{H}/IG}^F(a). \quad (84)$$

In the scenarios from Figures 15, 16, 17, and 19 we consider as the term of uncertainty the entropy $J_{\mathcal{H}}^F(a)$ of the last pose x_{k+L} in the planning segment (Section 3.3.3.1), while in the scenario from Figure 18 we instead use the IG of mapped until now landmarks $J_{IG}^F(a)$ (Section 3.3.3.2). Note that the running time presented in the figures refers

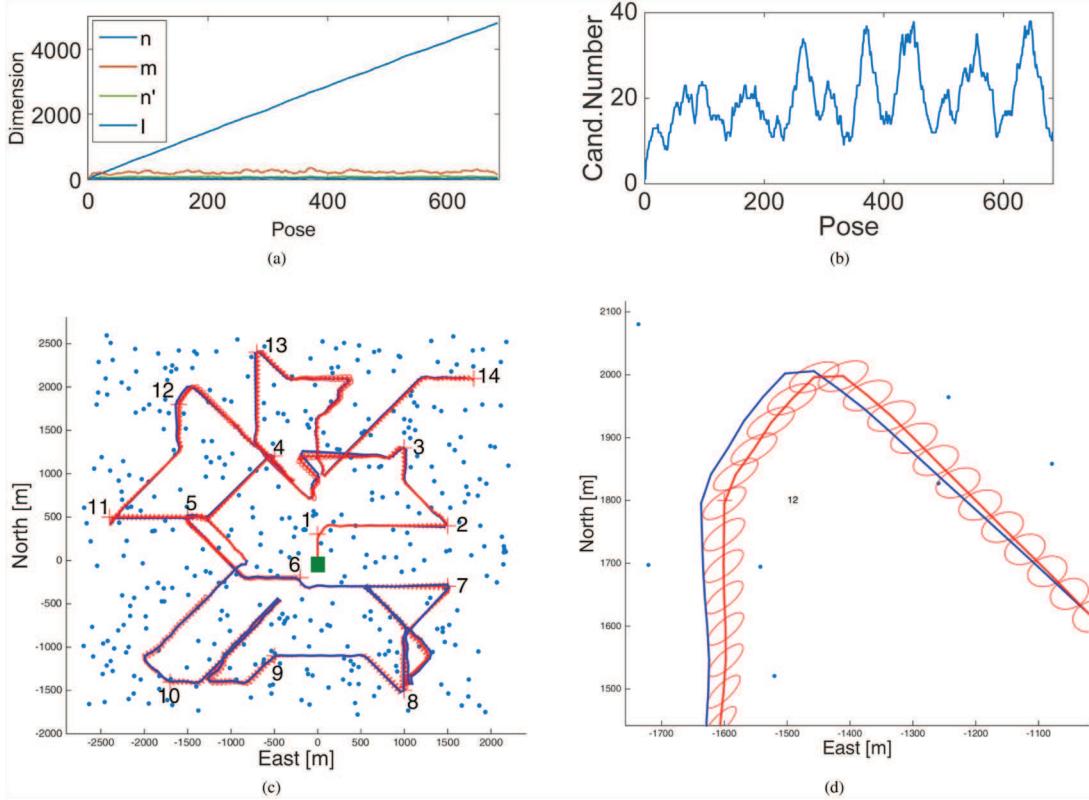


Fig. 15. *Focused* BSP scenario with *focused* robot’s last pose. (a) Dimensions of the BSP problem (state dimension n , average number of new factor terms m , average number of new variables n' , average number of old involved variables l) at each time. (b) Number of action candidates at each time. (c) Final robot trajectory. Blue dots are mapped landmarks, the red line with small ellipses is the estimated trajectory with pose covariances, the blue line is the real trajectory, the red pluses with numbers beside them are robot’s goals. The green mark is the robot’s start position. (d) Enlarged view of the robot’s trajectory near goal 12.

only to the uncertainty term, since it is the focus of this paper and because the calculation complexity of the first term (Euclidean distance $d(x_{k+L}, G_i)$) is relatively insignificant. As can be seen from above, we consider a non-myopic setting and let each candidate action represent trajectories of various length. Limiting the clustering process to a specific radius is done in order to bound the horizon L . In the scenarios from the paper, the radius is set to 100 meters.

In parallel, in scenarios from Figures 16 and 17, an *unfocused* uncertainty objective $J_{IG}(a)$ is calculated (Section 3.3.2), mainly for the purpose of performance comparison between *focused* and *unfocused* cases. The robot’s motion is controlled only by the *focused* objective function.

Four techniques were applied to solve the planning problem: more common techniques *From-Scratch* and *iSAM* (Section 5) and the proposed techniques, our general approach *rAMD* and its extension *rAMD-Extended* that exploits the Jacobian inner structure from (34) (see Table 5, and Sections 3.3.2 and 3.3.3.1). The calculated values of the objective function were numerically compared to validate that all four approaches are calculating exactly the same

metric, thus yielding the same decisions and only differ in running time.

In Figures 16 and 17 it can be clearly seen that while *iSAM* is faster than *From-Scratch*, the running time of both techniques is increasing with state dimensionality, as was mentioned previously. On the other hand, the running time of the *rAMD* approach is shown to be bounded, due to horizon lag of all candidate actions being limited (see Figure 15a). The number of candidate actions in our scenario is around 20 at each planning phase (Figure 15b). Even with such a relatively small candidate set, the *rAMD* approach is faster than its alternatives *iSAM* and *From-Scratch*, while the *rAMD-Extended* approach is the fastest of all. This trend appears to be correct for both *focused* and *unfocused* objective functions, though for the later, *iSAM* comes very close to the *rAMD* technique.

While comparing the running time of both *From-Scratch* and *iSAM* in *focused* and *unfocused* objective functions, it is easy to see that the *unfocused* case is evaluated much faster. The reason for this is that the *focused* calculations contain computation of marginal covariance of

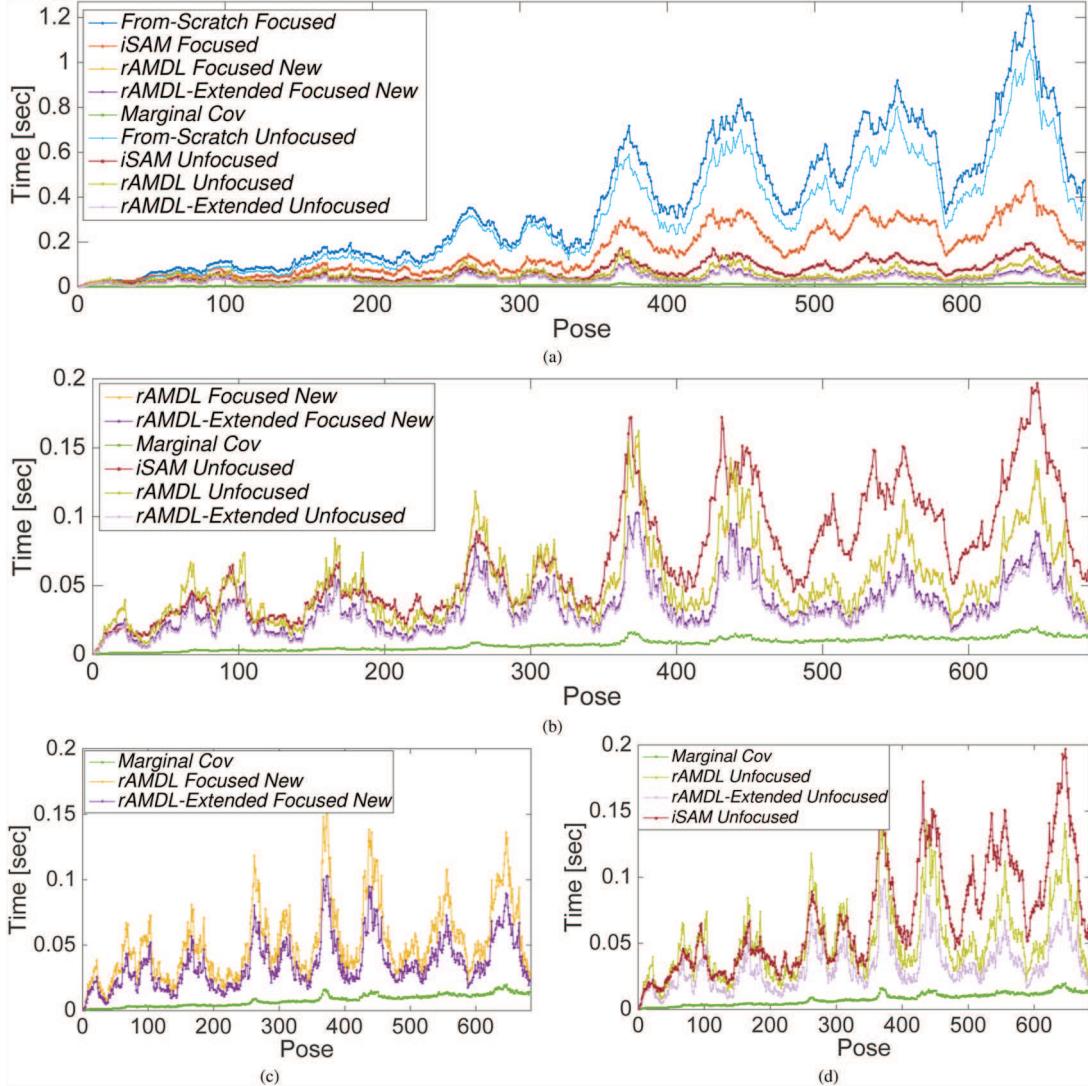


Fig. 16. Focused BSP scenario with *focused* robot’s last pose. (a) Running time of planning, i.e. evaluating the impact of all candidate actions, each representing a possible trajectory. Results are shown both for *focused* and *unfocused* cases. (b) Enlarged view of the fastest approaches from (a). (c) *Focused* approaches from (b). Note that *iSAM Focused* is not depicted because, as seen in (a), it is much slower compared with other *focused* techniques. (d) *Unfocused* approaches from (b). The lowest line, labeled *Marginal Cov*, represents the time it took to calculate prior marginal covariance $\Sigma_k^{M, X_{k+L}}$ in *rAMDL* approach (see Section 3.4). As can be seen, while the *rAMDL* technique (*Unfocused* and *Focused*) is faster than *From-Scratch* and *iSAM*, the *rAMDL-Extended* technique gives even better performance. Further, it is interesting to note that the performance of *Unfocused* and *Focused rAMDL* is almost the same, as so is the performance of *Unfocused* and *Focused rAMDL-Extended*.

the *focused* variable (last pose x_{k+L}) for each candidate action, which requires marginalization over the posterior information matrix Λ_{k+L} . Although this can be performed efficiently by exploiting the sparsity of matrix Λ_{k+L} (Kaess and Dellaert, 2009), the time complexity is significantly affected by variable elimination ordering of the *iSAM* algorithm (Kaess et al., 2012). While in our simulation we did not modify the default ordering of *iSAM* (COLAMD heuristic), different strategies of ordering can be a point for future investigation.

In contrast, for the *rAMDL* approach both *unfocused* and *focused* objective functions (equations (37)

and (42)) have a similar complexity, which is supported by the shown times. The same is correct for the *rAMDL-Extended* approach (equations (39) and (44)).

Next, we repeated our autonomous navigation scenario, but this time X_{k+L}^F contained only landmarks seen by time k (see Figure 18). The IG of such a *focused* set X_{k+L}^F can be used as an objective function for example in the case when we want to improve 3D reconstruction quality. As can be seen in Figure 18, this *focused* set causes both *From-Scratch* and *iSAM* techniques to be much slower compared with their performance in the first scenario, where X_{k+L}^F contained only x_{k+L} . The reason for this is that X_{k+L}^F ’s

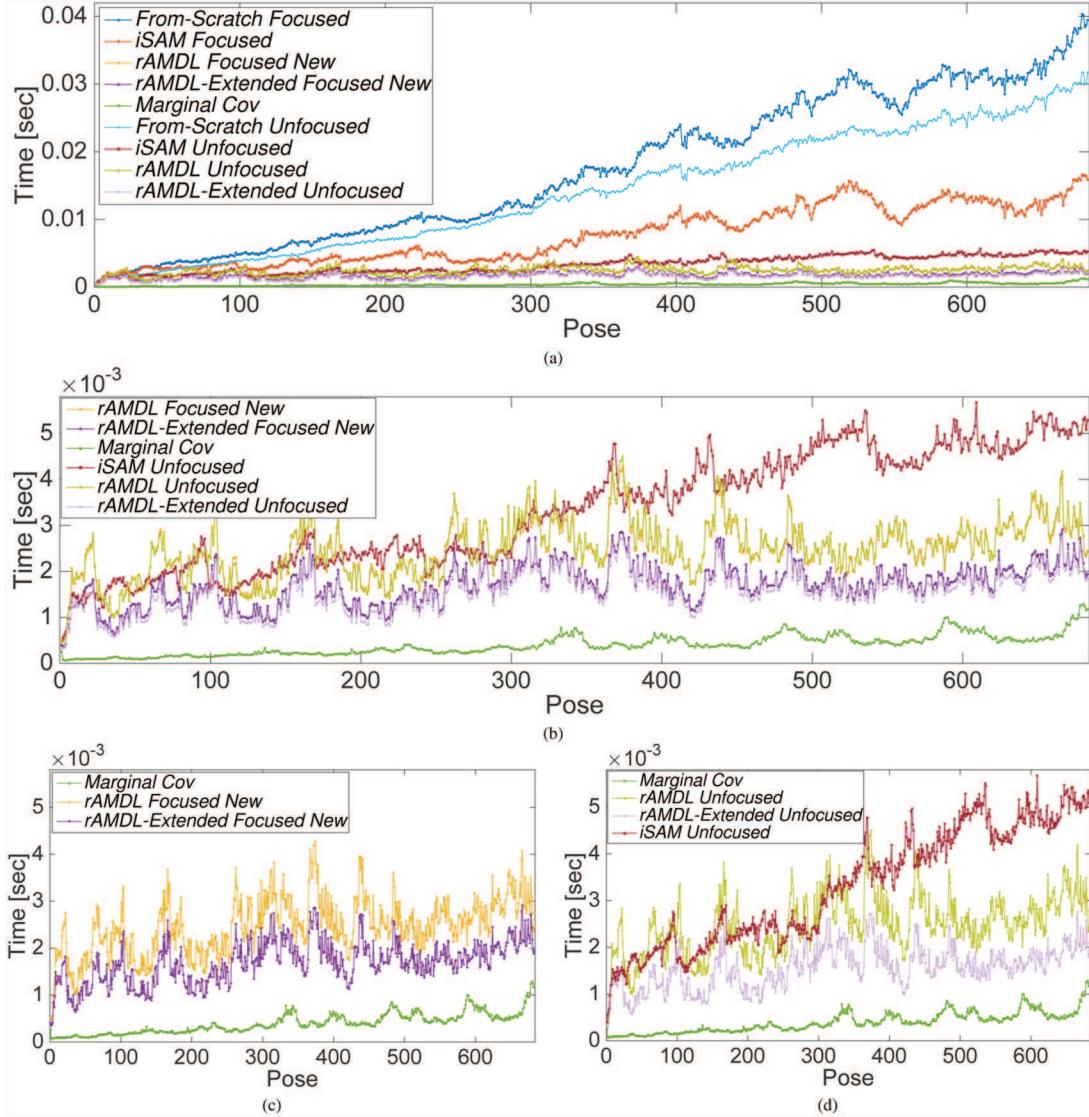


Fig. 17. Focused BSP scenario with *focused* robot's last pose. Running times from Figure 16 normalized by the number of candidates.

dimension is much higher here, representing the dimensions of all landmarks, and computation of its marginal covariance is significantly more expensive. In contrast, the performance of *rAMDl* has barely changed thanks to the *re-use* of calculations (see Section 3.4). Moreover, *rAMDl-Extended* performs even better than *rAMDl*, with candidate action impact evaluation being insignificant compared with the one-time calculation of marginal covariance, as can be seen in Figures 18e and (f).

We also performed a hybrid simulation where part of the real-world Victoria Park dataset (Guivant et al., 2012) was used for offline planning (see Figure 19). At each timestep we collected candidate actions by clustering landmarks seen until that time, just as was done in the first simulation. Further, we considered a focused objective function for each candidate with X_{k+L}^F containing only x_{k+L} . After evaluating all candidates, the robot was moved to the next pose

according to the dataset. Recalling that our main contribution is to reduce time complexity, such an evaluation allowed us to compare the time performance of all of the considered techniques, despite not actually using the calculated actions in the hybrid simulation. As can be seen, here *rAMDl* and *rAMDl-Extended* also outperform both of the alternatives, *From-Scratch* and *iSAM*, keeping the same trends that were observed in previous simulations.

8. Conclusions

We have developed a computationally efficient and exact approach for non-myopic *focused* and *unfocused* BSP in both augmented and non-augmented settings, in high-dimensional state spaces. As a key contribution, we have developed an augmented version of the well-known general matrix determinant lemma and used both of them

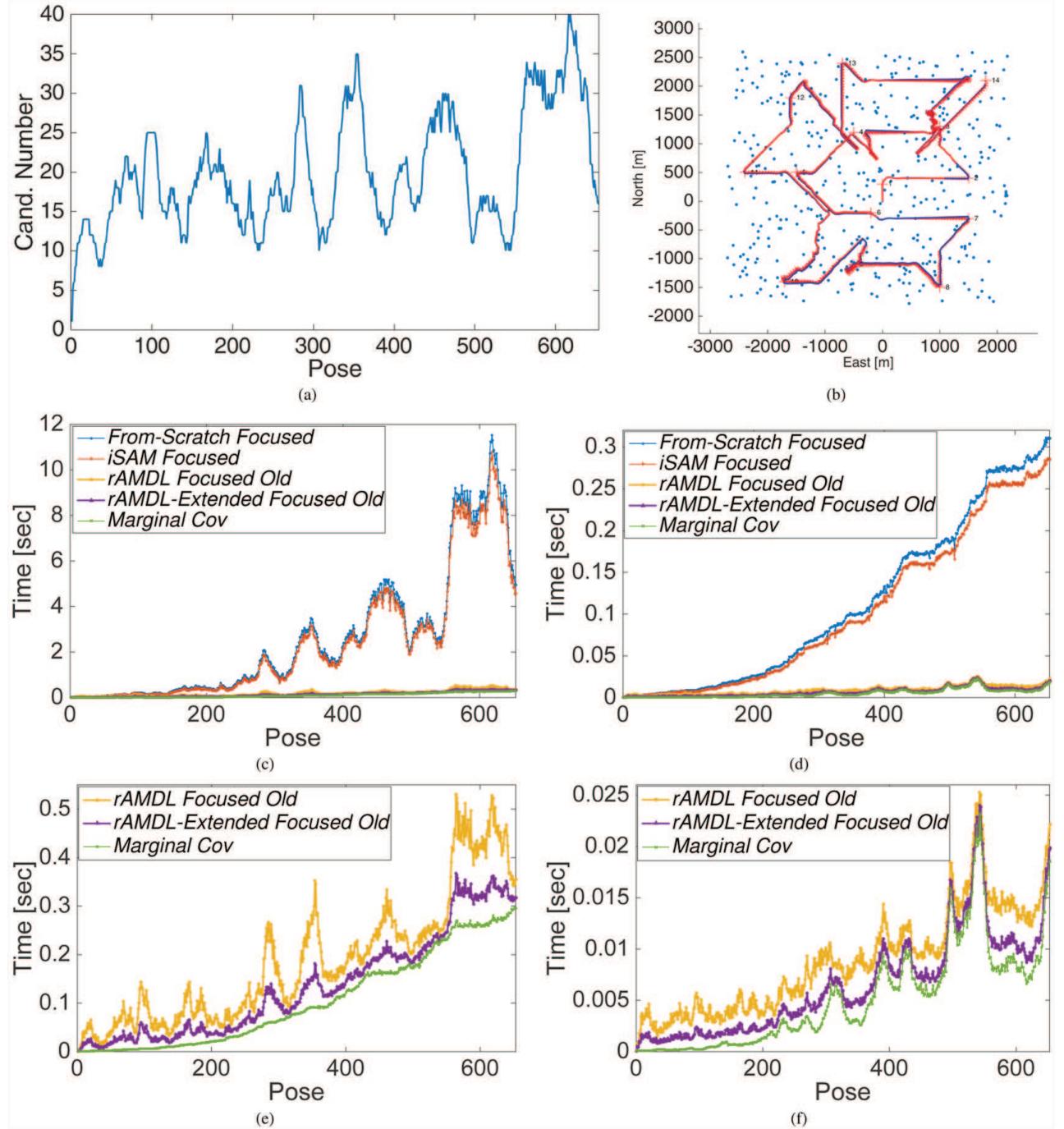


Fig. 18. Focused BSP scenario with *focused* landmarks. (a) Number of action candidates at each time. (b) Final robot trajectory. (c) Running time of planning, i.e. evaluating the impact of all candidate actions, each representing a possible trajectory. (d) Running time from (c) normalized by number of candidates. (e) Enlarged view of the fastest approaches from (c). (f) Enlarged view of the fastest approaches from (d). The lowest line, labeled *Marginal Cov*, represents the time it took to calculate the prior marginal covariance $\Sigma_k^{M,X_{All}}$ in the *rAMDL* approach (see Section 3.4).

to efficiently evaluate the impact of each candidate action on posterior entropy, without explicitly calculating the posterior information (or covariance) matrices. The second ingredient of our approach is the *re-use* of calculations, that exploits the fact that many calculations are shared

among different candidate actions. Our approach drastically reduces running time compared with the state of the art, especially when the set of candidate actions is large, with running time being independent of state dimensionality that increases over time in many of BSP domains. The approach

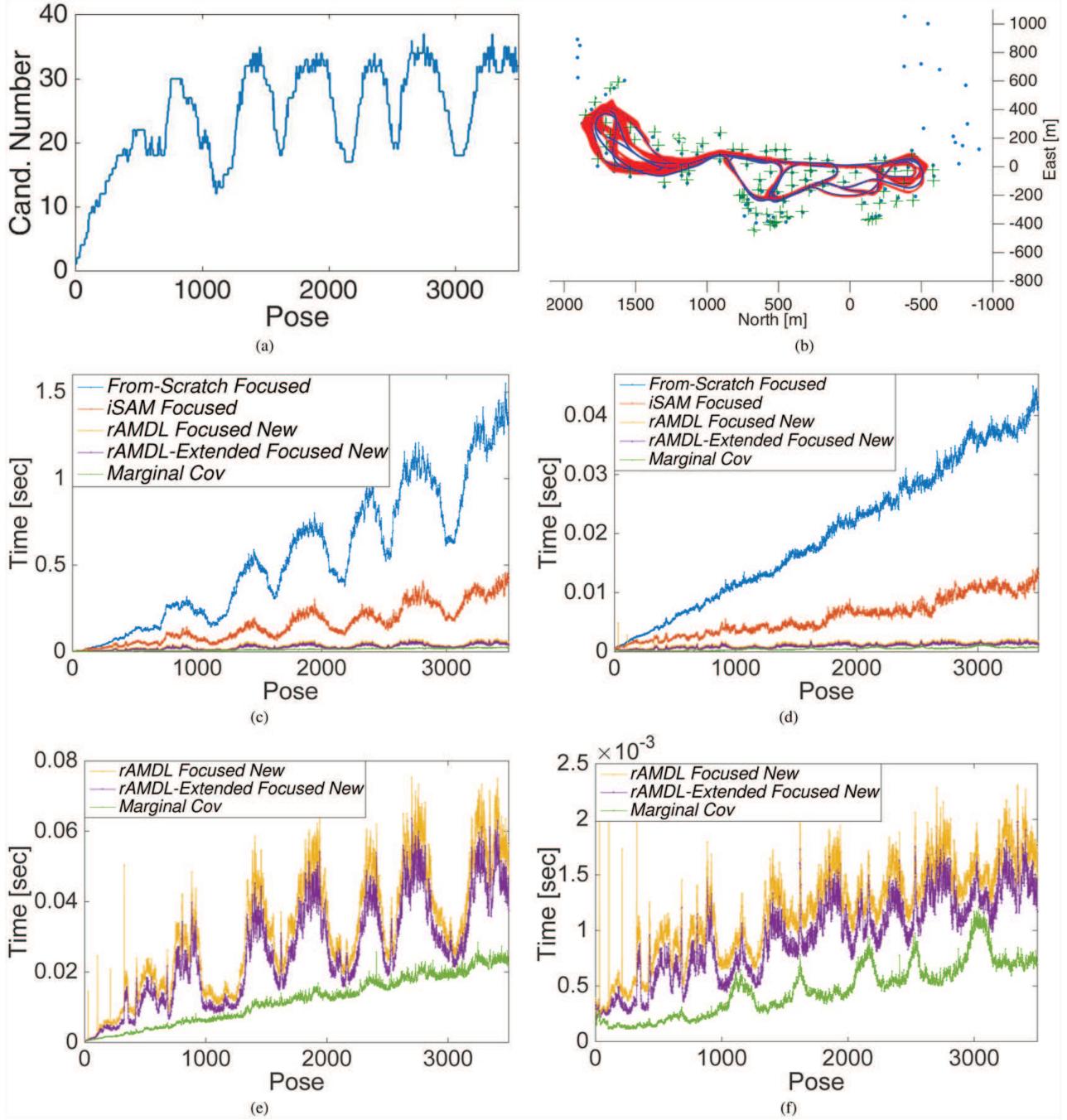


Fig. 19. Focused BSP scenario with *Focused* robot's last pose, using the Victoria Park dataset. (a) Number of action candidates at each time. (b) Final robot trajectory. (c) Running time of planning, i.e. evaluating the impact of all candidate actions, each representing a possible trajectory. (d) Running time from (c) normalized by the number of candidates. (e) Enlarged view of the fastest approaches from (c). (f) Enlarged view of the fastest approaches from (d). The lowest line, labeled *Marginal Cov*, represents time it took to calculate the prior marginal covariance $\Sigma_k^{M,X_{All}}$ in the *rAMDL* approach (see Section 3.4).

has been examined in three problems, sensor deployment, measurement selection in visual SLAM, and autonomous navigation in unknown environments, using both simulated and real-world datasets, and exhibiting in each superior performance compared with the state of the art, and reducing

running time by several orders of magnitude (e.g. 5 versus 400 seconds in sensor deployment).

Funding

This work was supported by the Israel Science Foundation.

References

- Agha-Mohammadi AA, Chakravorty S and Amato NM (2014) FIRM: Sampling-based feedback motion planning under motion uncertainty and imperfect measurements. *The International Journal of Robotics Research* 33(2): 268–304.
- Bai S, Wang J, Chen F and Englot B (2016) Information-theoretic exploration with Bayesian optimization. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 1816–1822.
- Bai Z, Fahey G and Golub G (1996) Some large-scale matrix computation problems. *Journal of Computational and Applied Mathematics* 74(1): 71–89.
- Carlevaris-Bianco N, Kaess M and Eustice RM (2014) Generic node removal for factor-graph SLAM. *IEEE Transactions on Robotics* 30(6): 1371–1385.
- Carlone L, Censi A and Dellaert F (2014) Selecting good measurements via L1 relaxation: A convex approach for robust estimation over graphs. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 2667–2674.
- Chaves SM, Walls JM, Galceran E and Eustice RM (2015) Risk aversion in belief-space planning under measurement acquisition uncertainty. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 2079–2086.
- Chli M and Davison AJ (2009) Active matching for visual tracking. *Robotics and Autonomous Systems* 57(12): 1173–1187.
- Davis T, Gilbert J, Larimore S and Ng E (2004) A column approximate minimum degree ordering algorithm. *ACM Transactions on Mathematical Software* 30(3): 353–376.
- Davison A (2005) Active search for real-time vision. In: *International Conference on Computer Vision (ICCV)*, pp. 66–73.
- Dellaert F (2012) Factor graphs and GTSAM: A hands-on introduction. Technical Report GT-RIM-CP&R-2012-002, Georgia Institute of Technology.
- Golub G and Plemmons R (1980) Large-scale geodetic least-squares adjustment by dissection and orthogonal decomposition. *Linear Algebra and Its Applications* 34: 3–28.
- Guivant J, Nieto J and Nebot E (2012) Victoria park dataset. Available at: http://www-personal.acfr.usyd.edu.au/nebot/victoria_park.htm
- Harville DA (1998) Matrix algebra from a statistician's perspective. *Technometrics* 40(2): 164–164.
- Huang G, Kaess M and Leonard J (2012) Consistent sparsification for graph optimization. In: *Proceedings of the European Conference on Mobile Robots (ECMR)*, pp. 150–157.
- Huang S, Kwok N, Dissanayake G, Ha Q and Fang G (2005) Multi-step look-ahead trajectory planning in SLAM: Possibility and necessity. In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1091–1096.
- Ila V, Polok L, Solony M, Smrz P and Zemcik P (2015) Fast covariance recovery in incremental nonlinear least square solvers. In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 4636–4643.
- Ila V, Porta JM and Andrade-Cetto J (2010) Information-based compact Pose SLAM. *IEEE Transactions on Robotics* 26(1): 78–93.
- Indelman V (2015a) Towards cooperative multi-robot belief space planning in unknown environments. In: *Proceedings of the International Symposium on Robotics Research (ISRR)*.
- Indelman V (2015b) Towards information-theoretic decision making in a conservative information space. In: *American Control Conference*, pp. 2420–2426.
- Indelman V (2016) No correlations involved: Decision making under uncertainty in a conservative sparse information space. *IEEE Robotics and Automation Letters* 1(1): 407–414.
- Indelman V, Carbone L and Dellaert F (2015) Planning in the continuous domain: a generalized belief space approach for autonomous navigation in unknown environments. *The International Journal of Robotics Research* 34(7): 849–882.
- Kaelbling LP, Littman ML and Cassandra AR (1998) Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101(1): 99–134.
- Kaess M and Dellaert F (2009) Covariance recovery from a square root information matrix for data association. *Robotics and Autonomous Systems* 57(12): 1198–1210.
- Kaess M, Johannsson H, Roberts R, Ila V, Leonard J and Dellaert F (2012) iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research* 31: 217–236.
- Kaess M, Ranganathan A and Dellaert F (2008) iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics* 24(6): 1365–1378.
- Kim A and Eustice RM (2014) Active visual SLAM for robotic area coverage: Theory and experiment. *The International Journal of Robotics Research* 33(4–5): 457–475.
- Kopitkov D and Indelman V (2016) Computationally efficient decision making under uncertainty in high-dimensional state spaces. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 1793–1800.
- Kopitkov D and Indelman V (2017) Computationally efficient belief space planning via augmented matrix determinant lemma and re-use of calculations. *IEEE International Conference on Robotics and Automation (ICRA) and IEEE Robotics and Automation Letters (RA-L), Mutual Submission*.
- Krause A, Singh A and Guestrin C (2008) Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research* 9: 235–284.
- Kretzschmar H and Stachniss C (2012) Information-theoretic compression of pose graphs for laser-based SLAM. *The International Journal of Robotics Research* 31(11): 1219–1230.
- Kschischang F, Frey B and Loeliger HA (2001) Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory* 47(2): 498–519.
- Levine D and How JP (2013) Sensor selection in high-dimensional gaussian trees with nuisances. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 2211–2219.
- Mazuran M, Tipaldi GD, Spinello L and Burgard W (2014) Non-linear graph sparsification for SLAM. In: *Robotics: Science and Systems (RSS)*, pp. 1–8.
- Mu B, Agha-mohammadi Aa, Paull L, Graham M, How J and Leonard J (2015) Two-stage focused inference for resource-constrained collision-free navigation. In: *Robotics: Science and Systems (RSS)*.
- Ouellette DV (1981) Schur complements and statistics. *Linear Algebra and its Applications* 36: 187–295.
- Patil S, Kahn G, Laskey M, Schulman J, Goldberg K and Abbeel P (2014) Scaling up Gaussian belief space planning through

- covariance-free trajectory optimization and automatic differentiation. In: *International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, pp. 515–533.
- Paull L, Huang G and Leonard JJ (2016) A unified resource-constrained framework for graph SLAM. In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1–8.
- Pineau J, Gordon GJ and Thrun S (2006) Anytime point-based approximations for large POMDPs. *Journal of Artificial Intelligence Research* 27: 335–380.
- Platt R, Tedrake R, Kaelbling L and Lozano-Pérez T (2010) Belief space planning assuming maximum likelihood observations. In: *Robotics: Science and Systems (RSS)*, Zaragoza, Spain, pp. 587–593.
- Prentice S and Roy N (2009) The belief roadmap: Efficient planning in belief space by factoring the covariance. *The International Journal of Robotics Research* 28(11–12): 1448–1465.
- Stachniss C, Grisetti G and Burgard W (2005) Information gain-based exploration using Rao–Blackwellized particle filters. In: *Robotics: Science and Systems (RSS)*, pp. 65–72.
- Valencia R, Morta M, Andrade-Cetto J and Porta J (2013) Planning reliable paths with pose SLAM. *IEEE Transactions on Robotics* 29(4): 1050–1059.
- Van Den Berg J, Patil S and Alterovitz R (2012) Motion planning under uncertainty using iterative local optimization in belief space. *The International Journal of Robotics Research* 31(11): 1263–1278.
- Vial J, Durrant-Whyte H and Bailey T (2011) Conservative sparsification for efficient and consistent approximate estimation. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 886–893.
- Walls JM, Chaves SM, Galceran E and Eustice RM (2015) Belief space planning for underwater cooperative localization. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 2264–2271.
- Zhu Z and Stein ML (2006) Spatial sampling design for prediction with estimated parameters. *Journal of Agricultural, Biological, and Environmental Statistics* 11(1): 24–44.
- Zimmerman DL (2006) Optimal network design for spatial prediction, covariance parameter estimation, and empirical prediction. *Environmetrics* 17(6): 635–652.

Appendix

A.1. Proof of Lemma 1

Problem definition: given a positive-definite and symmetric matrix $\Lambda \in \mathbb{R}^{n \times n}$ (e.g. a prior information matrix) and its inverse Σ (prior covariance matrix), first Λ is augmented by k zero rows and columns and the result is stored in Λ^{Aug} . Then we have matrix $A \in \mathbb{R}^{m \times (n+k)}$ and calculate $\Lambda^+ = \Lambda^{\text{Aug}} + A^T \cdot A$ (see Figure 1). We would like to express the determinant of Λ^+ in terms of Λ and Σ .

We start by modeling the matrix Λ^{Aug} through Σ . By introducing k new variables, before adding any new constraints involving these variables, we can say that new variables are uncorrelated with old variables, and their uncertainty is infinite (nothing yet is known about them). Then the appropriate covariance matrix after augmentation, Σ^{Aug} , can just be created by adding k zero rows and columns to Σ ,

and setting new diagonal entries with parameter θ , noting that $\theta \rightarrow \infty$:

$$\Sigma^{\text{Aug}} = \begin{bmatrix} \Sigma & 0 \\ 0 & \theta \cdot I \end{bmatrix}. \quad (85)$$

Next, note that the inverse of Σ^{Aug} is given by the following expression:

$$(\Sigma^{\text{Aug}})^{-1} = \begin{bmatrix} \Lambda & 0 \\ 0 & \epsilon \cdot I \end{bmatrix}, \quad (86)$$

where $\epsilon \doteq \frac{1}{\theta}$. Taking the limit $\epsilon \rightarrow 0$ into account, we can see that the above equation converges to Λ^{Aug} as was defined above. Then, in the limit, we have that $(\Lambda^{\text{Aug}})^{-1} = \Sigma^{\text{Aug}}$. In addition, note that $\epsilon \rightarrow 0$, even that it never becomes zero, $\epsilon \neq 0$, thus if needed we can divide by ϵ without worry.

Taking into account the limit of ϵ , expressing Λ^{Aug} through (86) will not change the problem definition. However, such a model allows us to take the inverse of Λ^{Aug} :

$$(\Lambda^{\text{Aug}})^{-1} = \Sigma^{\text{Aug}} = \begin{bmatrix} \Sigma & 0 \\ 0 & \theta \cdot I \end{bmatrix}, \quad (87)$$

and therefore to use the generalized matrix determinant lemma (Harville, 1998):

$$|\Lambda^+| = |\Lambda^{\text{Aug}}| \cdot |I_m + A \cdot \Sigma^{\text{Aug}} \cdot A^T| = |\Lambda| \cdot \epsilon^k \cdot |I_m + A_{\text{old}} \cdot \Sigma \cdot A_{\text{old}}^T + \theta \cdot A_{\text{new}} \cdot A_{\text{new}}^T| \quad (88)$$

where matrices $A_{\text{old}} \in \mathbb{R}^{m \times n}$ and $A_{\text{new}} \in \mathbb{R}^{m \times k}$ are constructed from A by retrieving columns of only old n variables and of only new k variables, respectively (see Figure 7).

Using the matrix determinant lemma once more, we obtain

$$|\Lambda^+| = |\Lambda| \cdot \epsilon^k \cdot |\Delta| \cdot |I_k + \theta \cdot A_{\text{new}}^T \cdot \Delta^{-1} \cdot A_{\text{new}}| \quad (89)$$

where $\Delta \doteq I_m + A_{\text{old}} \cdot \Sigma \cdot A_{\text{old}}^T$.

Moving ϵ inside the last determinant term, we have

$$|\Lambda^+| = |\Lambda| \cdot |\Delta| \cdot |\epsilon \cdot I_k + \epsilon \cdot \theta \cdot A_{\text{new}}^T \cdot \Delta^{-1} \cdot A_{\text{new}}| \quad (90)$$

Recalling that $\epsilon \rightarrow 0$ and $\epsilon \cdot \theta = 1$, we arrive at

$$|\Lambda^+| = |\Lambda| \cdot |\Delta| \cdot |A_{\text{new}}^T \cdot \Delta^{-1} \cdot A_{\text{new}}|. \quad (91)$$

The augmented determinant ratio will be

$$\frac{|\Lambda^+|}{|\Lambda|} = |I_m + A_{\text{old}} \cdot \Sigma \cdot A_{\text{old}}^T| \cdot |A_{\text{new}}^T \cdot (I_m + A_{\text{old}} \cdot \Sigma \cdot A_{\text{old}}^T)^{-1} \cdot A_{\text{new}}| = |\Delta| \cdot |A_{\text{new}}^T \cdot \Delta^{-1} \cdot A_{\text{new}}|. \quad (92)$$

■

A.2. Proof of Lemma 2

For A 's structure given in (34), Δ from (32) will be

$$\Delta = I_m + \begin{pmatrix} B_{old} \\ 0 \end{pmatrix} \cdot \Sigma \cdot \begin{pmatrix} B_{old}^T & 0 \end{pmatrix} = \begin{pmatrix} \Delta_1 & 0 \\ 0 & I_{m_{new}} \end{pmatrix}, \quad (93)$$

where $\Delta_1 = I_{m_{conn}} + B_{old} \cdot \Sigma \cdot B_{old}^T$, $m_{conn} = \mathcal{M}(\mathcal{F}^{conn}(a))$, and $m_{new} = \mathcal{M}(\mathcal{F}^{new}(a))$.

Then we can conclude that

$$|\Delta| = |\Delta_1| \quad (94)$$

and that

$$\Delta^{-1} = \begin{pmatrix} \Delta_1^{-1} & 0 \\ 0 & I_{m_{new}} \end{pmatrix}. \quad (95)$$

Now, by exploiting the structure of A_{new} we obtain

$$\begin{aligned} A_{new}^T \cdot \Delta^{-1} \cdot A_{new} &= (B_{new}^T D_{new}^T) \cdot \begin{pmatrix} \Delta_1^{-1} & 0 \\ 0 & I_{m_{new}} \end{pmatrix} \cdot \begin{pmatrix} B_{new} \\ D_{new} \end{pmatrix} \\ &= B_{new}^T \cdot \Delta_1^{-1} \cdot B_{new} + D_{new}^T \cdot D_{new}. \end{aligned} \quad (96)$$

Then we can conclude that the augmented determinant lemma will be:

$$\frac{|\Lambda^+|}{|\Lambda|} = |\Delta_1| \cdot |B_{new}^T \cdot \Delta_1^{-1} \cdot B_{new} + D_{new}^T \cdot D_{new}|. \quad (97)$$

■

A.3. Proof of Lemma 3

Consider the scenario of *focused* augmented BSP where the focused set X_{k+L}^F contains only newly added variables as defined in Section 3.3.3.1, with an appropriate illustration shown in Figure 8.

First, let us give an overview of the various partitions of Jacobian A that are relevant to our current problem (Figure 8). Here A_{old} , A_{new} , ${}^1A_{old}$, and ${}^{-1}A_{old}$ have been introduced in previous sections. Further, we can partition A_{new} into A_{new}^F columns of new variables that are focused $X_{new}^F \equiv X_{k+L}^F \in \mathbb{R}^{nF}$ and A_{new}^U columns of new unfocused variables X_{new}^U . Considering the figure, the set of all unfocused variables in X_{k+L} will be $X_{k+L}^R \doteq \{X_{old} \cup X_{new}^U\} \in \mathbb{R}^{nR}$, such that $N = n_F + n_R$, providing another A partition $A_R = [A_{old}, A_{new}^U]$.

Next, we partition the posterior information matrix Λ_{k+L} respectively to the sets X_{k+L}^F and X_{k+L}^R defined above as

$$\Lambda_{k+L} = \begin{bmatrix} \Lambda_{k+L}^R & \Lambda_{k+L}^{R,F} \\ (\Lambda_{k+L}^{R,F})^T & \Lambda_{k+L}^F \end{bmatrix}. \quad (98)$$

As was shown in (26), determinant of the marginal covariance of X_{k+L}^F can be calculated through

$$|\Sigma_{k+L}^{M,F}| = \frac{|\Lambda_{k+L}^R|}{|\Lambda_{k+L}|}. \quad (99)$$

Now let us focus on the Λ_{k+L}^R term from the right-hand side.

From (10) we can see that the partition of the posterior information matrix Λ_{k+L}^R can be calculated as

$$\Lambda_{k+L}^R = \Lambda_k^{Aug,R} + A_R^T A_R, \quad (100)$$

where $\Lambda_k^{Aug,R}$ can be constructed by augmenting Λ_k with zero rows and columns in the number of X_{new}^U 's dimension (see Figure 8). The above equation has an augmented determinant form as defined in Section 3.3.1, and so the augmented determinant lemma can be applied on it. Using (32) we have

$$\frac{|\Lambda_{k+L}^R|}{|\Lambda_k|} = |C| \cdot |(A_{new}^U)^T \cdot C^{-1} \cdot A_{new}^U|, \quad (101)$$

where C is defined in (38).

Next, dividing (101) by (36), we obtain

$$\begin{aligned} |\Sigma_{k+L}^{M,F}| &= \frac{|\Lambda_{k+L}^R|}{|\Lambda_k|} \cdot \frac{|\Lambda_k|}{|\Lambda_{k+L}|} = \frac{|\Lambda_{k+L}^R|}{|\Lambda_{k+L}|} \\ &= \frac{|(A_{new}^U)^T \cdot C^{-1} \cdot A_{new}^U|}{|A_{new}^T \cdot C^{-1} \cdot A_{new}|}, \end{aligned} \quad (102)$$

and posterior entropy of X_{k+L}^F is given by

$$\begin{aligned} J_{\mathcal{H}}^F(a) &= \frac{n_F \cdot \gamma}{2} + \frac{1}{2} \ln |(A_{new}^U)^T \cdot C^{-1} \cdot A_{new}^U| \\ &\quad - \frac{1}{2} \ln |A_{new}^T \cdot C^{-1} \cdot A_{new}|. \end{aligned} \quad (103)$$

■
Note that the variables inside information matrices do not have to be ordered in any particular way, and that the proof provided above is correct for any ordering whatsoever.

A.4. Proof of Lemma 4

For A 's structure given in (34), the term $A_{new}^T \cdot C^{-1} \cdot A_{new}$ from (103), similarly to (96), will be

$$A_{new}^T \cdot C^{-1} \cdot A_{new} = B_{new}^T \cdot C_1^{-1} \cdot B_{new} + D_{new}^T \cdot D_{new}, \quad (104)$$

where C_1 is defined in (41).

In the same way, we can conclude (see Figure 8) that

$$(A_{new}^U)^T \cdot C^{-1} \cdot A_{new}^U = (B_{new}^U)^T \cdot C_1^{-1} \cdot B_{new}^U + (D_{new}^U)^T \cdot D_{new}^U. \quad (105)$$

Therefore, the posterior entropy of X_{k+L}^F from (103) is given by

$$\begin{aligned} J_{\mathcal{H}}^F(a) &= \frac{n_F \cdot \gamma}{2} + \frac{1}{2} \ln |(B_{new}^U)^T \cdot C_1^{-1} \cdot B_{new}^U + \Lambda_a^{U|F}| \\ &\quad - \frac{1}{2} \ln |B_{new}^T \cdot C_1^{-1} \cdot B_{new} + \Lambda_a|, \end{aligned} \quad (106)$$

where $\Lambda_a = D_{new}^T \cdot D_{new}$ is the information matrix of an action's factor graph $G(a)$, and where $\Lambda_a^{U|F} = (D_{new}^U)^T \cdot D_{new}^U$ is the information matrix of variables X_{new}^U conditioned

on $X_{new}^F \doteq X_{k+L}^F$ and calculated from the distribution represented by $G(a)$. ■

Note that the variables inside information matrices do not have to be ordered in any particular way, and that the proof provided above is correct for any ordering whatsoever.

A.5. Proof of Lemma 5

Consider the scenario of *focused* augmented BSP where the focused set X_{k+L}^F contains only old variables, with appropriate illustration shown in Figure 9 and with various partitions of Jacobian A defined in Section 3.3.3.2.

First, let us look again over relevant partitions of Jacobian A (Figure 9). The A_{old} , A_{new} , ${}^I A_{old}$, and ${}^{-I} A_{old}$ were already introduced in previous sections. From the figure we can see that ${}^{-I} A_{old}$ can further be separated into ${}^{-I} A_{old}^U$ columns of old variables that are both not involved and unfocused (${}^{-I} X_{old}^U$) and ${}^{-I} A_{old}^F$ columns of old variables that are both not involved and focused (${}^{-I} X_{old}^F$). In addition, ${}^I A_{old}$ can be partitioned into ${}^I A_{old}^U$ columns of old variables that are both involved and unfocused (${}^I X_{old}^U$), and ${}^I A_{old}^F$ columns of old variables that are both involved and focused (${}^I X_{old}^F$) (see Table 2). The set of focused variables is then $X_{k+L}^F = \{{}^{-I} X_{old}^F \cup {}^I X_{old}^F\} \in \mathbb{R}^{n_F}$, containing both involved and not involved variables. We use the notation $X_{k+L}^F \doteq X_k^F$ to remind us that the focused set of variables is part of both X_{k+L} and X_k .

Likewise, the set of all remained, unfocused variables is $X_{k+L}^R \doteq \{{}^{-I} X_{old}^U \cup {}^I X_{old}^U \cup X_{new}\} \in \mathbb{R}^{n_R}$, containing all new variables and some of the old ones (which can be involved or not involved), and providing A 's partition $A_R = [{}^{-I} A_{old}^U, {}^I A_{old}^U, A_{new}]$. Moreover, for the purpose of simplification of coming equations we will denote the set of old variables inside X_{k+L}^R by X_{old}^R , having that $X_{old}^R \doteq \{{}^{-I} X_{old}^U \cup {}^I X_{old}^U\}$, with appropriate Jacobian partition $A_{old}^R \doteq [{}^{-I} A_{old}^U, {}^I A_{old}^U]$.

Next, noting that $X_k = \{X_k^F \cup X_{old}^R\}$ we can partition the prior information matrix Λ_k , respectively,

$$\Lambda_k = \begin{bmatrix} \Lambda_k^F & \Lambda_k^{F,R_{old}} \\ (\Lambda_k^{F,R_{old}})^T & \Lambda_k^{R_{old}} \end{bmatrix}. \quad (107)$$

Similarly, due to $X_{k+L} = \{X_k^F \cup X_{old}^R \cup X_{new}\}$ and $X_{k+L}^R \doteq \{X_{old}^R \cup X_{new}\}$, the posterior information matrix Λ_{k+L} can be respectively partitioned into the next two forms:

$$\begin{aligned} \Lambda_{k+L} &= \begin{bmatrix} \Lambda_{k+L}^F & \Lambda_{k+L}^{F,R_{old}} & \Lambda_{k+L}^{F,X_{new}} \\ (\Lambda_{k+L}^{F,R_{old}})^T & \Lambda_{k+L}^{R_{old}} & \Lambda_{k+L}^{R_{old},X_{new}} \\ (\Lambda_{k+L}^{F,X_{new}})^T & (\Lambda_{k+L}^{R_{old},X_{new}})^T & \Lambda_{k+L}^{X_{new}} \end{bmatrix} \\ &= \begin{bmatrix} \Lambda_{k+L}^F & \Lambda_{k+L}^{F,R} \\ (\Lambda_{k+L}^{F,R})^T & \Lambda_{k+L}^R \end{bmatrix} \end{aligned} \quad (108)$$

with

$$\Lambda_{k+L}^R = \begin{bmatrix} \Lambda_{k+L}^{R_{old}} & \Lambda_{k+L}^{R_{old},X_{new}} \\ (\Lambda_{k+L}^{R_{old},X_{new}})^T & \Lambda_{k+L}^{X_{new}} \end{bmatrix}. \quad (109)$$

We can see from the above partitions (107)–(109) that the posterior information partition Λ_{k+L}^R of X_{k+L}^R is simply the augmentation of prior information partition $\Lambda_k^{R_{old}}$ and can be calculated as

$$\Lambda_{k+L}^R = \Lambda_k^{Aug,R_{old}} + A_R^T A_R, \quad (110)$$

where $\Lambda_k^{Aug,R_{old}}$ can be constructed by first taking the partition of the prior information matrix Λ_k related to X_{old}^R , $\Lambda_k^{R_{old}}$, and augmenting it with n' zero rows and columns (see Figure 9), where n' is just the number of newly introduced variables. The above equation has an augmented determinant form as defined in Section 3.3.1, and so the augmented determinant lemma can be applied also here. Using (32) we have

$$\frac{|\Lambda_{k+L}^R|}{|\Lambda_k^{R_{old}}|} = |S| \cdot |A_{new}^T \cdot S^{-1} \cdot A_{new}|, \quad (111)$$

$$S = I_m + A_{old}^R \cdot (\Lambda_k^{R_{old}})^{-1} \cdot (A_{old}^R)^T. \quad (112)$$

Then by combining (99), (36), and the above equations, we can see that

$$\frac{|\Sigma_{k+L}^{M,F}|}{|\Sigma_k^{M,F}|} = \frac{|\Lambda_{k+L}^R|}{|\Lambda_k^{R_{old}}|} \cdot \frac{|\Lambda_k|}{|\Lambda_k^{R_{old}}|} = \frac{|S| \cdot |A_{new}^T \cdot S^{-1} \cdot A_{new}|}{|C| \cdot |A_{new}^T \cdot C^{-1} \cdot A_{new}|}, \quad (113)$$

where C is defined in (38).

Apparently the IG of X_{k+L}^F can be calculated as

$$\begin{aligned} J_{IG}^F(a) &= \mathcal{H}(X_k^F) - \mathcal{H}(X_{k+L}^F) = \frac{1}{2} \ln |\Sigma_k^{M,F}| - \frac{1}{2} \ln |\Sigma_{k+L}^{M,F}| \\ &= \frac{1}{2} (\ln |C| + \ln |A_{new}^T \cdot C^{-1} \cdot A_{new}| - \ln |S| \\ &\quad - \ln |A_{new}^T \cdot S^{-1} \cdot A_{new}|), \end{aligned} \quad (114)$$

Next, the S term can be further reduced. It is clear that $(\Lambda_k^{R_{old}})^{-1} = \Sigma_k^{R_{old}|F}$, or namely the prior conditional covariance matrix of X_{old}^R conditioned on X_k^F . Moreover, due to the sparsity of A_{old}^R (its sub-block ${}^{-I} A_{old}^U$ contains only zeros) we will actually need only entries of matrix $\Sigma_k^{R_{old}|F}$ that belong to variables involved in new terms of (6) (see Figure 9) and can conclude that

$$S = I_m + A_{old}^R \cdot \Sigma_k^{R_{old}|F} \cdot (A_{old}^R)^T = I_m + {}^I A_{old}^U \cdot \Sigma_k^{{}^I X_{old}^U|F} \cdot ({}^I A_{old}^U)^T. \quad (115)$$

Note that the variables inside information matrices do not have to be ordered in any particular way, and that the proof provided above is correct for any ordering whatsoever.

A.6. Proof of Lemma 6

For A 's structure given in (34), the term S from (115) will be

$$\begin{aligned} S &= I_m + A_{old}^R \cdot \Sigma_k^{R_{old}|F} \cdot (A_{old}^R)^T = I_m + \begin{pmatrix} B_{old}^R \\ 0 \end{pmatrix} \cdot \Sigma_k^{R_{old}|F} \\ &\quad \cdot ((B_{old}^R)^T \ 0) = \begin{pmatrix} S_1 & 0 \\ 0 & I_{m_{new}} \end{pmatrix}, \end{aligned} \quad (116)$$

where $S_1 = I_{m_{conn}} + B_{old}^R \cdot \Sigma_k^{R_{old}|F} \cdot (B_{old}^R)^T$, $m_{conn} = \mathcal{M}(\mathcal{F}^{conn}(a))$ and $m_{new} = \mathcal{M}(\mathcal{F}^{new}(a))$.

Then we can conclude that

$$|S| = |S_1| \quad (117)$$

and that

$$S^{-1} = \begin{pmatrix} S_1^{-1} & 0 \\ 0 & I_{m_{new}} \end{pmatrix}, \quad (118)$$

and similarly to (115) (see also Figure 9) we have that

$$\begin{aligned} S_1 &= I_{m_{conn}} + B_{old}^R \cdot \Sigma_k^{R_{old}|F} \cdot (B_{old}^R)^T \\ &= I_m + {}^I B_{old}^U \cdot \Sigma_k^{I X_{old}^U | F} \cdot ({}^I B_{old}^U)^T. \end{aligned} \quad (119)$$

Next, the term $A_{new}^T \cdot S^{-1} \cdot A_{new}$ from (114), similarly to (96), will be

$$A_{new}^T \cdot S^{-1} \cdot A_{new} = B_{new}^T \cdot S_1^{-1} \cdot B_{new} + D_{new}^T \cdot D_{new}, \quad (120)$$

with S_1 defined in (119).

Then, by applying equations (104), (117), (120), and notion $|C| = |C_1|$, the IG of $X_{k+L}^F \subseteq X_{old}$ from (114) can be calculated as

$$\begin{aligned} J_{IG}^F(a) &= \frac{1}{2} (\ln |C_1| + \ln |B_{new}^T \cdot C_1^{-1} \cdot B_{new} + D_{new}^T \cdot D_{new}| \\ &\quad - \ln |S_1| - \ln |B_{new}^T \cdot S_1^{-1} \cdot B_{new} + D_{new}^T \cdot D_{new}|) \\ &= \frac{1}{2} (\ln |C_1| + \ln |B_{new}^T \cdot C_1^{-1} \cdot B_{new} + \Lambda_a| \\ &\quad - \ln |S_1| - \ln |B_{new}^T \cdot S_1^{-1} \cdot B_{new} + \Lambda_a|), \end{aligned} \quad (121)$$

where C_1 is defined in (41), and where $\Lambda_a = D_{new}^T \cdot D_{new}$ is the information matrix of an action's factor graph $G(a)$. ■

Note that the variables inside information matrices do not have to be ordered in any particular way, and that the proof provided above is correct for any ordering whatsoever.

A.7. SLAM solution: focus on the last pose $X_{k+L}^F \equiv x_{k+L}$

For $X_{k+L}^F \equiv x_{k+L}$ the *focused* entropy objective in the SLAM setting is given by (44). Here, we exploit the inner structure of Jacobian partitions in the SLAM scenario (see (71)–(75)) in order to provide a solution tailored specifically to the SLAM domain. It will provide an illustrated example of applying rAMD to a real problem.

From (75) we can see that B_{new}^U has the following form:

$$\begin{aligned} B_{new}^U &= \Psi_{conn}^{-\frac{1}{2}} \cdot \begin{pmatrix} -I & \cdots & 0 \\ \mathbb{H}_1^{x_{k+1}} & \cdots & \mathbb{H}_1^{x_{k+L-1}} \\ \vdots & \ddots & \vdots \\ \mathbb{H}_{n_o}^{x_{k+1}} & \cdots & \mathbb{H}_{n_o}^{x_{k+L-1}} \end{pmatrix} \\ &= \Psi_{conn}^{-\frac{1}{2}} \cdot \begin{pmatrix} \mathbb{F}^U \\ \mathbb{H}_{X_{new}}^U \end{pmatrix}, \\ \mathbb{F}^U &\doteq (-I \cdots 0), \\ \mathbb{H}_{X_{new}}^U &\doteq \begin{pmatrix} \mathbb{H}_1^{x_{k+1}} \cdots \mathbb{H}_1^{x_{k+L-1}} \\ \vdots \quad \ddots \quad \vdots \\ \mathbb{H}_{n_o}^{x_{k+1}} \cdots \mathbb{H}_{n_o}^{x_{k+L-1}} \end{pmatrix}, \end{aligned} \quad (122)$$

where $\mathbb{H}_i^{x_{k+1}} \doteq \nabla_{x_{k+1}} h_i$ is the Jacobian of h_i with respect to x_{k+1} , and therefore is non-zero only if the factor's observation was taken from pose x_{k+1} . Note that in the SLAM case the X_{new}^U (all new and *unfocused* variables) is $\{x_{k+1}, \dots, x_{k+L-1}\}$.

Similarly to (79), the term $(B_{new}^U)^T \cdot C_1^{-1} \cdot B_{new}^U$ from (44) can be calculated as

$$\begin{aligned} (B_{new}^U)^T \cdot C_1^{-1} \cdot B_{new}^U &= ((\mathbb{F}^U)^T (\mathbb{H}_{X_{new}}^U)^T) \cdot \Psi_{conn}^{-\frac{1}{2}} \\ &\quad \cdot \Psi_{conn}^{\frac{1}{2}} \cdot C_2^{-1} \cdot \Psi_{conn}^{\frac{1}{2}} \cdot \Psi_{conn}^{-\frac{1}{2}} \cdot \begin{pmatrix} \mathbb{F}^U \\ \mathbb{H}_{X_{new}}^U \end{pmatrix} \\ &= ((\mathbb{F}^U)^T (\mathbb{H}_{X_{new}}^U)^T) \cdot C_2^{-1} \cdot \begin{pmatrix} \mathbb{F}^U \\ \mathbb{H}_{X_{new}}^U \end{pmatrix} \\ &= (\tilde{B}_{new}^U)^T \cdot C_2^{-1} \cdot \tilde{B}_{new}^U \end{aligned} \quad (123)$$

where

$$\begin{aligned} \tilde{B}_{new}^U &\doteq \begin{pmatrix} -I & \cdots & 0 \\ \mathbb{H}_1^{x_{k+1}} & \cdots & \mathbb{H}_1^{x_{k+L-1}} \\ \vdots & \ddots & \vdots \\ \mathbb{H}_{n_o}^{x_{k+1}} & \cdots & \mathbb{H}_{n_o}^{x_{k+L-1}} \end{pmatrix} = \begin{pmatrix} \mathbb{F}^U \\ \mathbb{H}_{X_{new}}^U \end{pmatrix} \end{aligned} \quad (124)$$

contains the Jacobian entries of B_{new}^U not weighted by the factors' noise Ψ_{conn} .

In addition, from (71) we can derive the structure of D_{new}^U which is also used in (44):

$$D_{new}^U = \Psi_{new}^{-\frac{1}{2}} \cdot \begin{pmatrix} & & & & & & \\ \underline{(x_{k+1})} & \underline{(x_{k+2})} & \underline{(x_{k+3})} & \underline{(x_{k+4})} & \cdots & \underline{(x_{k+L-2})} & \underline{(x_{k+L-1})} \\ & \mathbb{F}_{k+1} & -I & 0 & 0 & \cdots & 0 & 0 \\ & 0 & \mathbb{F}_{k+2} & -I & 0 & \cdots & 0 & 0 \\ & 0 & 0 & \mathbb{F}_{k+3} & -I & \cdots & 0 & 0 \\ & 0 & 0 & 0 & \mathbb{F}_{k+4} & \cdots & 0 & 0 \\ & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ & 0 & 0 & 0 & 0 & \cdots & -I & 0 \\ & 0 & 0 & 0 & 0 & \cdots & \mathbb{F}_{k+L-2} & -I \\ & 0 & 0 & 0 & 0 & \cdots & 0 & \mathbb{F}_{k+L-1} \end{pmatrix} \doteq \Psi_{new}^{-\frac{1}{2}} \cdot \tilde{D}_{new}^U \quad (125)$$

that due to its sparsity will allow fast calculation of $\Lambda_a^{U|F}$,

$$\Lambda_a^{U|F} = (D_{new}^U)^T \cdot D_{new}^U = (\tilde{D}_{new}^U)^T \cdot \Psi_{new}^{-1} \cdot \tilde{D}_{new}^U. \quad (126)$$

Finally, placing all derived notation into (44), we obtain to the SLAM-specific solution for the entropy of the robot's last pose:

$$J_{\mathcal{H}}^F(a) = \frac{n_F \cdot \gamma}{2} + \frac{1}{2} \ln |(\tilde{B}_{new}^U)^T \cdot C_2^{-1} \cdot \tilde{B}_{new}^U + (\tilde{D}_{new}^U)^T \cdot \Psi_{new}^{-1} \cdot \tilde{D}_{new}^U| - \frac{1}{2} \ln |\tilde{B}_{new}^T \cdot C_2^{-1} \cdot \tilde{B}_{new} + \tilde{D}_{new}^T \cdot \Psi_{new}^{-1} \cdot \tilde{D}_{new}|, \quad (127)$$

where C_2 is defined in (77). ■

Note that the variables inside information matrices do not have to be ordered in any particular way, and that the proof provided above is correct for any ordering whatsoever.

A.8. SLAM solution: focus on mapped landmarks

$$X_{k+L}^F \equiv L_k$$

The *focused* IG of $X_k^F \equiv L_k$ in the SLAM setting is given by (47). Here, we will exploit the inner structure of Jacobian partitions in the SLAM scenario (see (71)–(75)) in order to provide a solution tailored specifically to the SLAM domain. It will provide an illustrated example of applying rAMDL to areal problem.

First, note that all old *involved* and *unfocused* variables ${}^I X_{old}^U$ contain only the current robot's pose x_k . Thus, from (74) we can see that the relevant partition of Jacobian B , the ${}^I B_{old}^U$ used in (48), has the following inner structure:

$${}^I B_{old}^U = \Psi_{conn}^{-\frac{1}{2}} \cdot \begin{pmatrix} \mathbb{F}_k \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \quad (128)$$

Using the above identity, the matrix S_1 from (48) can also be reduced to the following form:

$$\begin{aligned} S_1 &= I_{m_{conn}} + {}^I B_{old}^U \cdot \Sigma_k^{{}^I X_{old}^U|F} \cdot ({}^I B_{old}^U)^T = I_{m_{conn}} + \Psi_{conn}^{-\frac{1}{2}} \\ &\quad \cdot \begin{pmatrix} \mathbb{F}_k \cdot \Sigma_k^{M,x_k} \cdot \mathbb{F}_k^T & 0 \\ 0 & 0 \end{pmatrix} \cdot \Psi_{conn}^{-\frac{1}{2}} \\ &= \Psi_{conn}^{-\frac{1}{2}} \cdot \left[\Psi_{conn} + \begin{pmatrix} \mathbb{F}_k \cdot \Sigma_k^{M,x_k} \cdot \mathbb{F}_k^T & 0 \\ 0 & 0 \end{pmatrix} \right] \\ &\quad \cdot \Psi_{conn}^{-\frac{1}{2}} \doteq \Psi_{conn}^{-\frac{1}{2}} \cdot S_2 \cdot \Psi_{conn}^{-\frac{1}{2}}, \end{aligned} \quad (129)$$

$$\begin{aligned} S_2 &\doteq \Psi_{conn} + \begin{pmatrix} \mathbb{F}_k \cdot \Sigma_k^{M,x_k} \cdot \mathbb{F}_k^T & 0 \\ 0 & 0 \end{pmatrix} \\ &= \left(\Sigma_{\omega,k} + \mathbb{F}_k \cdot \Sigma_k^{M,x_k} \cdot \mathbb{F}_k^T \begin{matrix} 0 \\ \Psi_{obs} \end{matrix} \right), \end{aligned} \quad (130)$$

where $\Sigma_{\omega,k}$ is the noise matrix from the motion model (65), and matrix Ψ_{obs} is block-diagonal, combining all noise matrices of $\mathcal{F}^{obs}(a)$ factors:

$$\Psi_{conn} = \begin{pmatrix} \Sigma_{\omega,k} & 0 \\ 0 & \Psi_{obs} \end{pmatrix}. \quad (131)$$

Further, let us define matrix S_3 :

$$S_3 \doteq \Sigma_{\omega,k} + \mathbb{F}_k \cdot \Sigma_k^{M,x_k} \cdot \mathbb{F}_k^T. \quad (132)$$

Now we can see that S_1 's determinant and inverse can be calculated through

$$|S_1| = \frac{|S_2|}{|\Psi_{conn}|} = \frac{|S_3| \cdot |\Psi_{obs}|}{|\Psi_{conn}|} = \frac{|S_3|}{|\Sigma_{\omega,k}|}, \quad (133)$$

$$S_1^{-1} = \Psi_{conn}^{\frac{1}{2}} \cdot S_2^{-1} \cdot \Psi_{conn}^{\frac{1}{2}} = \Psi_{conn}^{\frac{1}{2}} \cdot \begin{pmatrix} S_3^{-1} & 0 \\ 0 & \Psi_{obs}^{-1} \end{pmatrix} \cdot \Psi_{conn}^{\frac{1}{2}}. \quad (134)$$

Similarly to (79), the term $B_{new}^T \cdot S_1^{-1} \cdot B_{new}$ from (47) can be calculated as

$$\begin{aligned} B_{new}^T \cdot S_1^{-1} \cdot B_{new} &= (\mathbb{F}^T \cdot (\mathbb{H}^{X_{new}})^T) \\ &\quad \cdot \Psi_{conn}^{-\frac{1}{2}} \cdot \Psi_{conn}^{\frac{1}{2}} \cdot S_2^{-1} \\ &\quad \cdot \Psi_{conn}^{\frac{1}{2}} \cdot \Psi_{conn}^{-\frac{1}{2}} \cdot \begin{pmatrix} \mathbb{F} \\ \mathbb{H}^{X_{new}} \end{pmatrix} \\ &= (\mathbb{F}^T \cdot (\mathbb{H}^{X_{new}})^T) \cdot S_2^{-1} \cdot \begin{pmatrix} \mathbb{F} \\ \mathbb{H}^{X_{new}} \end{pmatrix} \\ &= (\mathbb{F}^T \cdot (\mathbb{H}^{X_{new}})^T) \cdot \begin{pmatrix} S_3^{-1} & 0 \\ 0 & \Psi_{obs}^{-1} \end{pmatrix} \cdot \\ &\quad \begin{pmatrix} \mathbb{F} \\ \mathbb{H}^{X_{new}} \end{pmatrix} = \mathbb{F}^T \cdot S_3^{-1} \cdot \mathbb{F} + (\mathbb{H}^{X_{new}})^T \cdot \Psi_{obs}^{-1} \cdot \mathbb{H}^{X_{new}}, \end{aligned} \quad (135)$$

where \mathbb{F} and $\mathbb{H}^{X_{new}}$ are defined in (75) as

$$\mathbb{F} \doteq (-I \quad \cdots \quad 0), \quad \mathbb{H}^{X_{new}} \doteq \begin{pmatrix} \mathbb{H}_1^{x_{k+1}} & \cdots & \mathbb{H}_1^{x_{k+L}} \\ \vdots & \ddots & \vdots \\ \mathbb{H}_{n_o}^{x_{k+1}} & \cdots & \mathbb{H}_{n_o}^{x_{k+L}} \end{pmatrix}. \quad (136)$$

Thus, we can see that $\mathbb{F}^T \cdot S_3^{-1} \cdot \mathbb{F}$ from (135) is an $L \cdot n_p \times L \cdot n_p$ matrix (n_p is the robot pose's dimension and L is the horizon length) that has non-zero entries only at its $n_p \times n_p$ top left corner:

$$\mathbb{F}^T \cdot S_3^{-1} \cdot \mathbb{F} = \begin{pmatrix} S_3^{-1} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}. \quad (137)$$

Finally, placing all derived notation into (47), we arrive at the SLAM-specific solution for IG of already mapped landmarks L_k :

$$\begin{aligned} J_{IG}^F(a) &= \frac{1}{2}(\ln |C_2| - \ln |\Psi_{conn}| + \ln |\tilde{\mathcal{B}}_{new}^T \cdot C_2^{-1} \cdot \tilde{\mathcal{B}}_{new} \\ &\quad + \tilde{\mathcal{D}}_{new}^T \cdot \Psi_{new}^{-1} \cdot \tilde{\mathcal{D}}_{new}| - \ln |S_3| \\ &\quad + \ln |\Sigma_{\omega,k}| - \ln | + \mathbb{F}^T \cdot S_3^{-1} \cdot \mathbb{F} + (\mathbb{H}^{X_{new}})^T \cdot \\ &\quad \Psi_{obs}^{-1} \cdot \mathbb{H}^{X_{new}} + \tilde{\mathcal{D}}_{new}^T \cdot \Psi_{new}^{-1} \cdot \tilde{\mathcal{D}}_{new}|) \\ &= \frac{1}{2} \left(\ln |C_2| + \ln |\tilde{\mathcal{B}}_{new}^T \cdot C_2^{-1} \cdot \tilde{\mathcal{B}}_{new} + \tilde{\mathcal{D}}_{new}^T \cdot \Psi_{new}^{-1} \cdot \right. \\ &\quad \left. \cdot \tilde{\mathcal{D}}_{new}| \right. \\ &\quad \left. - \ln |S_3| - \ln |\mathbb{F}^T \cdot S_3^{-1} \cdot \mathbb{F} + (\mathbb{H}^{X_{new}})^T \cdot \Psi_{obs}^{-1} \cdot \right. \\ &\quad \left. \mathbb{H}^{X_{new}} + \tilde{\mathcal{D}}_{new}^T \cdot \Psi_{new}^{-1} \cdot \tilde{\mathcal{D}}_{new}| - \ln |\Psi_{obs}| \right), \quad (138) \end{aligned}$$

where C_2 is defined in (77). Note that matrix S_3 will be the same for all candidates. Therefore, the terms S_3 , $\ln |S_3|$, and $\mathbb{F}^T \cdot S_3^{-1} \cdot \mathbb{F}$ can be calculated only one time and shared between the candidates thereafter. In addition, the terms C_2 , $\tilde{\mathcal{B}}_{new}^T \cdot C_2^{-1} \cdot \tilde{\mathcal{B}}_{new}$, $\tilde{\mathcal{D}}_{new}^T \cdot \Psi_{new}^{-1} \cdot \tilde{\mathcal{D}}_{new}$, and $(\mathbb{H}^{X_{new}})^T \cdot \Psi_{obs}^{-1} \cdot \mathbb{H}^{X_{new}}$ can be calculated efficiently through sparse matrix operators since we know the exact inner structure of all involved matrix operands. The overall complexity of the above SLAM solution is the same as in (47), $O(\mathcal{M}(\mathcal{F}^{conn}(a))^3 + n'^3)$. ■

Note that the variables inside information matrices do not have to be ordered in any particular way, and that the proof provided above is correct for any ordering whatsoever.