Characterizing Marginalization and Incremental Operations on the Bayes Tree

Dehann Fourie*, Antonio Terán Espinoza*, Michael Kaess[1], and John Leonard

Massachusetts Institute of Technology, Cambridge MA 20139, USA,
{dehann,teran,jleonard}@mit.edu,
Home page: http://marinerobotics.mit.edu

[1]Carnegie Mellon University, Pittsburgh PA 15213, USA,
kaess@cmu.edu,
Home page: https://rpl.ri.cmu.edu/

Abstract. Perception systems for autonomy are most useful if they can operate within limited/predictable computing resources. Existing algorithms in robot navigation—e.g. simultaneous localization and mapping—employ concepts from filtering, fixed-lag, or incremental smoothing to find feasible inference solutions. Using factor graphs as a probabilistic modeling language, we emphasize the importance of marginalization operations on the equivalent Bayes (junction) tree. The objective is to elucidate the connection between simple tree-based message passing rules with the aforementioned state estimation approaches, and their frequently overlooked relation to direct marginalization on the Bayes tree. We characterize the inherent marginalization operation as part of the fundamental Chapman-Kolmogorov transit integrals which unifies many state-of-the-art approaches. The belief propagation model is then used to define five major tree inference strategies, with regard to computation recycling and resource constrained operation. A series of illustrative examples and results show the versatility of the method.

Keywords: localization, mapping, robotics, Bayes tree, junction tree, Bayes network, Chapman-Kolmogorov, belief propagation, sum-product

1 Introduction

Autonomous systems such as mobile robots or smart factories require a perception system to adequately understand its surroundings and perform meaningful tasks. To this end, such systems employ a combination of sensors that collect data of itself and its environment. Inference algorithms then analyze this data to localize, map, and extract semantic understanding of the world in a timely manner. Real-time use-cases impose additional resource constraints on the system's inference requirements.

*Equal contribution.

Established estimation approaches, such as filtering or fixed-lag smoothing, reduce computational load by marginalizing previous information into a posterior of a few select states (variables), thereby concentrating the history of measurements into only a few variables. State-of-the-art incremental smoothing and mapping (iSAM) algorithms, such as iSAM2 [17], factorize a factor graph [20] probabilistic model into an Bayes tree [16]—similar to a junction tree [15]—which allows for more efficient computations.

This work aims to characterize marginalization and incremental operations on a Bayes tree and elucidate their connection to existing estimation strategies. The contributions of this paper are: i) illustrate the connection between the fundamental Chapman-Kolmogorov transit equation [26] and message passing on the Bayes tree [6]; ii) analysis and characterization of the resulting marginalization operations for recycling computations on the tree-based data structure; iii) highlight the simplicity of reasoning about message passing on the Bayes tree and thereby unify existing estimation methods in a common probabilistic language; iv) to show how the proposed approach can be a unifying framework for the aforementioned estimation approaches; and v) present a description with example results to aid future algorithm development beyond the common parametric Gaussian assumption.

The remainder of the paper is organized as follows: Section 2 provides background and preliminaries to describe the scenario within the context of related work, which is presented in Section 3. The problem is formulated in Section 4 in terms of factor graphs, the Bayes tree, and Chapman-Kolmogorov integrals, which form the basis for defining the proposed generalized recycling operations on the tree. In Section 5, a canonical illustration is shown by means of a toy problem, and further results from a common simultaneous localization and mapping (SLAM) dataset are analyzed. Finally, we conclude with Section 6.

2 Problem Formulation and Background

Combining the various data sources in time to produce reliable consensus on the state of the system and its surroundings is a computationally intensive and complicated task—commonly known as SLAM [5, 21]. This scenario serves as a good proxy for many resource constrained inference tasks beyond robotics.

A major challenge in SLAM is reliable sensor fusion that can extract the maximum amount of information about the world while maintaining control over the computational resource requirements. Surrounding variables of interest and sensor data are encoded into a joint probability model using factor graphs [4, 20]. Factor graphs, illustrated in Figure 1, are bipartite graphs of variables and factors. Each of the variables $\boldsymbol{\Theta}_i$ exist on a manifold domain $\varXi_i$ and represents (possibly hidden) states or parameters of interest. These include landmark features $\mathbf{L} = \{\boldsymbol{l} \in \mathbb{R}^d\}$, robot positions and orientation $\mathbf{X} = \{\boldsymbol{x} \in \mathrm{SE}(3)\}$ [4], sensor calibration $\mathbf{K} = \{\theta_K \in \varXi_K\}$, and many more.

While exploring the world, a robot incrementally collects sets of measurements $\mathbf{Z} = \{\boldsymbol{z}_{1:t}\}$ at each time step t. Each measurement is modeled by a

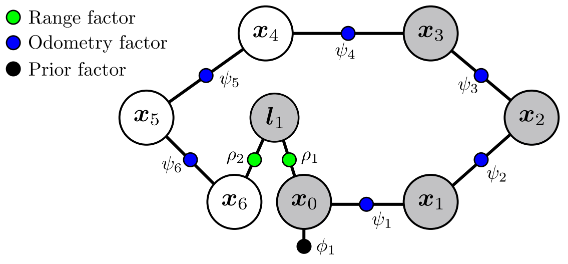

Fig. 1. A sample factor graph denoting a set of poses $\boldsymbol{x}_i$ and an unknown landmark $\boldsymbol{l}_1$ as the variables nodes ($\boldsymbol{\Theta} = \{\boldsymbol{x}_{0:6}, \boldsymbol{l}_1\}$), and three distinct types of factors ($\mathcal{F} = \{\psi_{1:6}, \rho_{1:2}, \phi_1\}$). The shaded nodes correspond to the marginalized variables of the example presented in Section 5.1.

measurement function $\phi_i(\boldsymbol{\Theta}_i, \boldsymbol{z}_i)$ over a subset of variables $\boldsymbol{\Theta}_i$. These functions are used to construct a probabilistic likelihood model, or factor, $p_{\phi_i} = p\,(\,\boldsymbol{Z}_{\phi_i} = \boldsymbol{z}_{\phi_i} \,|\, \boldsymbol{\Theta}_i\,) \in \mathcal{P}$, where $\mathcal{P}$ denotes the space of all allowable (conditional) probability density functions. The union of all likelihoods is called the set of factors $\mathcal{F} = \bigcup_i p_{\phi_i}$ (see Figure 1 caption for an example).

The inference problem can be formulated as a joint probability distribution through a unifying product of all individual likelihoods i and priors j. The variables $\boldsymbol{\Theta}$, factors $\mathcal{F}$, and edges $\mathcal{E}$ encode the factor graph $\mathcal{G} = \{\mathcal{F}, \boldsymbol{\Theta}, \mathcal{E}\}$.

$$p\,(\,\boldsymbol{\Theta} \,|\, \mathbf{Z}\,) = \prod_i p\,(\,\boldsymbol{z}_{\phi_i} \,|\, \boldsymbol{\Theta}_i\,) \prod_j p(\boldsymbol{\Theta}_j) \times \frac{1}{\boldsymbol{Q}} \propto \prod_i p\,(\,\boldsymbol{z}_{\phi_i} \,|\, \boldsymbol{\Theta}_i\,) \prod_j p(\boldsymbol{\Theta}_j), \quad (1)$$

where $\boldsymbol{Q}$ represents the factor graph's partition function, taken as constant due to an independence assumption [20]: we assume that any measurements $\boldsymbol{z}_l, \boldsymbol{z}_{l'}$, where $\forall\, l, l' \in i, j$, are taken from statistically independent processes $\boldsymbol{Z}_l, \boldsymbol{Z}_{l'}$ given the factor graph model $\mathcal{G}$. For example, stereo camera images produce independent monocular measurements ϕ_i given the modeled extrinsics while ignoring common power supply noise; similarly for priors ϕ_j, GPS filter ouputs could be taken with sufficient time separation to have negligible correlation (assuming a loosely coupled architecture) [11]. This allows a further simplification to the unnormalized joint probability function.

By the chain rule, this product of independent measurements $\mathbf{Z}$—through likelihoods $p\,(\,\boldsymbol{z}_{\phi_i} \,|\, \boldsymbol{\Theta}_i\,)$ and variable prior potentials $p(\boldsymbol{\Theta}_j)$—represents the unnormalized, non-Gaussian posterior joint probability density [20]. The inference task therefore requires a system inversion by estimating the belief over state variables given the data $p\,(\,\mathbf{Z} \,|\, \boldsymbol{\Theta}\,) \to p\,(\,\boldsymbol{\Theta} \,|\, \mathbf{Z}\,) \in \mathcal{P}$ that was likely to have produced the received measurements $\mathbf{Z}$ at each time t.

Typically, the *maximum-a-posteriori* estimate $\boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}} p\,(\,\boldsymbol{\Theta} \,|\, \mathbf{Z}\,)$ is sought, however, eq. (1) allows us to pursue something more general; we are interested in directly computing the full posterior marginal beliefs of each variable in the joint distribution, that is, an entire function and not just a parametric point representation.

3 Related Work

While filters [32] offer constant-time updates, they do not scale well in augmented state configurations given the need to track correlations between variables using a dense covariance matrix, as is the case with EKF-SLAM [29] approaches. Extensive previous work shows that the most efficient inference methods exploit structure and sparsity within the problem [4]. Cyclic factor graphs are well suited for modelling heterogeneous sensor data collection, but do not, however, immediately admit simple inference of variable beliefs. For example, loopy belief propagation approaches [25] are expensive for large systems [30].

One avenue is to factorize the joint model factor graph into an acyclic tree representation. Several tree factorizations have been proposed including elimination, cluster, bucket, rake-compress [19], junction trees [15], and more. Tree-structured representations are desirable because trees encode an exact equivalent of the original factor graph in the form of an acyclic graph. One significant example is the Thin Junction Tree Filter (TJTF) algorithm by Paskin [23], for which computational complexity can be controlled.

Around the same time, and despite the similarities in structure to Paskin's TJTF, Frese's Treemap algorithm [9] was independently developed to address similar concerns: bounded uncertainty, linear storage space, and linear update cost. Fundamentally, treemap is different from TJTF in that it does not tradeoff information for computation time but instead uses a hierarchical tree partitioning strategy [10]. The Bayes tree [16]—developed for robotic navigation—is a specialized junction tree used in iSAM2 [17], hybrid-discrete [27], multimodal-iSAM [6, 7], and multi-hypothesis iSAM2 [13] variants. The Bayes tree is based on a globally selected variable ordering (a known NP-hard problem [1]) by well established pivoting algorithms from matrix system solvers, such as approximate minimum degree (AMD) [3].

The transformation of a factor graph into a Bayes tree is described in [16], and pictorially summarized in Figure 2. The factorization mainly consists of two steps: the construction of a chordal Bayes net [24] by means of a *bipartite elimination game* [12]—requiring a specific variable ordering—and the discovery of cliques through *maximum cardinality search* [28]. The elimination process is represented in Figure 2 by the transformation between the first two graphs.

There are several benefits for performing inference on an acyclic Bayes tree factorization of the joint distribution rather than on the original graph itself [16, 18, 23]: the tree structure here implies an exact factorization representing the inherent conditional independence structure between all the variables, and thereby encodes the minimum set of update steps required to find variable posteriors. This fact is exploited by the (non-Gaussian) multimodal-iSAM algorithm [6, 7].

Williams et al. [31] investigate some benefits of concurrent filtering and smoothing using the Bayes tree, but their work does not relate to explicit marginalization of variables beyond the existing iSAM2 incremental updates. Later work in variable selection [14, 22] uses information content to identify which variables are the most valuable and then reduces the SLAM problem size accordingly. These methods, however, require modification of the factor graph before inference. In contrast, our approach presented below illustrates how the Bayes tree can be used to handle similar operations natively within the tree structure as part of a normal single stage factor graph to tree inference process.

4 Method

4.1 General Inference Using the Bayes Tree

This section describes the inference process as tree-based message passing operations which will serve as the foundation for selective message passing strategies in later sections. The development follows a *sum-product* approach outlined by the multimodal-iSAM algorithm [6], which allows general non-Gaussian/multimodal likelihoods to be considered as part of the Bayes tree solution. The discussion applies equally to Gaussian belief propagation, and in the linear case is analogous to Cholesky/QR factorization in matrix systems [16, 17].

The full joint probability distribution $p\,(\boldsymbol{\Theta}\,|\,\mathbf{Z}\,)$ is factorized by the Bayes tree (by extracting cliques from the associated Bayes net) which represents the product of conditional distributions, with one conditional for each clique $\mathbf{C}_k$

$$p\,(\boldsymbol{\Theta}\,|\,\mathbf{Z}\,) \propto \prod_{k=\{\mathbf{C}\}} p\,(\boldsymbol{\Theta}_{F,k}\,|\,\boldsymbol{\Theta}_{S,k}, \boldsymbol{z}_{k^+}\,)\,. \tag{2}$$

Cliques are expressed in the form of separator $\boldsymbol{\Theta}_{S,k}$ and frontal $\boldsymbol{\Theta}_{F,k}$ variables: that is, separators are the intersection with the parent clique $\boldsymbol{\Theta}_{S,k} = \mathbf{C}_k \cap \mathbf{C}_{k'}$, and remaining frontals $\boldsymbol{\Theta}_{F,k} = \mathbf{C}_k \backslash \boldsymbol{\Theta}_{S,k}$ [16]. The set of measurements in each

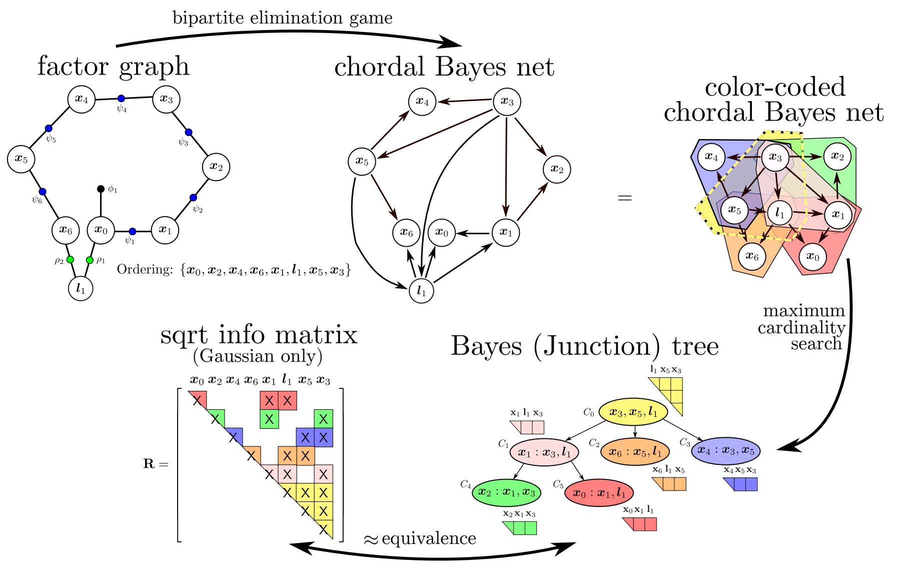

Fig. 2. Pictorial depiction of the steps needed to transform the factor graph from Figure 1 to a Bayes tree representation given a variable ordering.

clique k is given by $\boldsymbol{z}_{k^+}$. Finding these cliques in the original joint probability model eq. (1) is equivalent to grouping statistically dependent variables and factors together (exploiting commutativity of functions from $\mathcal{P}$):

$$p\left(\boldsymbol{\Theta}_{F,k} \,|\, \boldsymbol{\Theta}_{S,k}, \boldsymbol{z}_{k^+}\right) \propto \prod_{i' \in \mathcal{F}_F(\mathbf{C}_k)} p\left(\boldsymbol{z}_{i'} \,|\, \boldsymbol{\Theta}_{i'}\right) \prod_{j' \in \mathcal{F}_F(\mathbf{C}_k)} p\left(\boldsymbol{\Theta}_{j'}\right). \tag{3}$$

The probability in eq. (3) represents a subset of factors $\mathcal{F}_F(C_k)$ from the original factor graph that are directly associated with the frontal variables $\boldsymbol{\Theta}_{F,k}$. This implies that a clique's frontal variables $\boldsymbol{\Theta}_{F,k}$ are rendered conditionally independent from the rest of the graph given the separator $\boldsymbol{\Theta}_{S,k}$.

4.2 Characterizing Marginalization on the Bayes Tree

Consider a **complete batch solve** over the entire tree which would naively involve finding the best estimates (posteriors) for all variables from data $p(\boldsymbol{\Theta} \,|\, \mathbf{Z})$ – as discussed in Section 2. In general, computing the entire joint posterior is hopelessly complicated since the fully generalized solution may have exponential complexity in the dimension of the problem; finding marginal posteriors over each of the cliques is much more feasible [6].

Consider the theoretical operation of marginalizing out all variables except $\boldsymbol{\Theta}_{F,k}$ from the joint probability distribution in eq. (1), with the shorthand $\boldsymbol{z}_{\phi_i} = \boldsymbol{z}_i$. The root clique $k = 1$ in the Bayes tree is somewhat special as having only frontal variables $\boldsymbol{\Theta}_{F,1}$ and an empty separator, $\boldsymbol{\Theta}_{S,1} = \varnothing$. Using the Bayes tree joint factorization from eq. (2), the expression for the root's posterior takes the form

$$p\left(\boldsymbol{\Theta}_{F,1} \,|\, \mathbf{Z}\right) \propto \int_{\Xi} \prod_k p\left(\boldsymbol{\Theta}_{F,k} \,|\, \boldsymbol{\Theta}_{S,k}, \boldsymbol{z}_{k^+}\right) \mathrm{d}\backslash\boldsymbol{\Theta}_{F,1}, \tag{4}$$

where $\backslash\boldsymbol{\Theta}_{F,1}$ expresses all variables except those in the root frontals. From eq. (4), marginalizing all but $\boldsymbol{\Theta}_{F,1}$ produces a unique cascading result in the factorized joint from eq. (1):

$$p\left(\boldsymbol{\Theta}_{F,1} \,|\, \mathbf{Z}\right) \propto \int_{\Xi_2} p\left(\boldsymbol{\Theta}_{F,2} \,|\, \boldsymbol{\Theta}_{S,2}, \boldsymbol{z}_2\right) \int_{\Xi_3} p\left(\boldsymbol{\Theta}_{F,3} \,|\, \boldsymbol{\Theta}_{S,3}, \boldsymbol{z}_3\right) \dots d\boldsymbol{\Theta}_{F,3} \times \\ \underbrace{\int_{\Xi_k} p\left(\boldsymbol{\Theta}_{F,k} \,|\, \boldsymbol{\Theta}_{S,k}, \boldsymbol{z}_k\right) \dots d\boldsymbol{\Theta}_{F,k}}_{m_{k|\mathbf{Y}_k}} d\boldsymbol{\Theta}_{F,2}. \tag{5}$$

where each integral (marginalization) term in the equation corresponds to the summarized information of an upward marginalization message $m_{k|\mathbf{Y}_k}$, eq. (7).

The leaf cliques have no children and can therefore be calculated first using eq. (3). Working upwards from leaves to the root, all the measurements $\mathbf{Z}_{k^+}$ pertaining to cliques $\mathbf{C}_k$ and lower down the tree are collected in the variable $\mathbf{Y}_k = \{\bigcup_k^{L \to k} \boldsymbol{z}_k^+\}$, where index k goes from the leaf L to the current clique

k. For the case of a leaf clique, we have $\mathbf{Y}_k = \boldsymbol{z}_k$, and at the root we have all measurements $\mathbf{Y}_1 = \boldsymbol{z}$. To summarize, given a clique $\mathbf{C}_k$ with a parent clique $\mathbf{C}_{k'}$, the upward message takes the form of a sum-product computation, eq. (7).

The density $m_{k|\mathbf{Y}}\left(\Theta_{S,k}\right)$ obtained from this marginalized clique product is passed up the tree as another belief message to the next parent clique, and is analogous to eq. (10) in Kaess et al. [16]. The parent clique then repeats the process as illustrated in Figure 3. The product and marginalization of potentials continue up the tree until the root node is reached. Since the root has no parent clique, a marginal of the full system posterior density has been obtained. These steps compute all the integrals and products listed in the marginal joint probability density shown in eq. (5), where $\Theta_{F,2}$ must be marginalized last but other variables, such as $\Theta_{F,3}$, can be marginalized sooner.

4.3 Chapman-Kolmogorov and Generalized Belief Propagation

Each of the distributed marginalizations represent the upward message at each clique in the tree. Note that the marginalization (integral) operations of variables are over their respective variable manifold [8] domains $\theta_{F,k} \in \Xi_k$. Each clique is a partial joint posterior over all clique variables, $p\left(\Theta_{C,k} \,|\, \mathbf{Y}_k\right) \doteq M_{k|\mathbf{Y}} \in \mathcal{P}$:

$$M_{k|\mathbf{Y}}\left(\Theta_{C,k}\right) \propto \prod_{i'}^{C_k} p\left(\mathbf{Z}_{i'} \,|\, \Theta_{i'}\right) \prod_{j'}^{C_k} p\left(\Theta_{j'}\right) \prod_{u} m_{u|\mathbf{Y}}\left(\Theta_{S,u}\right). \tag{6}$$

The marginalized message on separator $\Theta_{S,k}$ is sent to the parent clique

$$m_{k|\mathbf{Y}}\left(\Theta_{S,k}\right) \doteq p\left(\Theta_{S,k} \,|\, \mathbf{Y}_k\right) = \int_{\Xi_k} M_{k|\mathbf{Y}}\left(\Theta_{C,k}\right)\, d\Theta_{F,k}. \tag{7}$$

Eqs. (5) through (7) show the sum-product approach to inference, and eq. (7) is a specific instance of the much more general Chapman-Kolmogorov transit integral [26] (which simultaneously describes familiar Bayesian filtering):

$$p\left(\boldsymbol{\Theta}_{S,k} \,|\, \mathbf{Y}_k\right) \propto \int_{\Xi_{F,k}} p\left(\boldsymbol{\Theta}_{F,k} \,|\, \boldsymbol{\Theta}_{S,k}, \mathbf{Z}_{k^+}\right) \prod_{u} p\left(\boldsymbol{\Theta}_{S,u} \,|\, \mathbf{Y}_u\right) \mathrm{d}\boldsymbol{\Theta}_{F,k}, \tag{8}$$

where the clique conditional was defined in eq. (3) and with child cliques u as per Figure 3. Within this context, the role of inference is to compute all messages such that the belief $m_{k|\mathbf{Y}}\left(\Theta_{S,k}\right)$ for any clique $\mathbf{C}_k$ can be obtained.

Practical computation of the Chapman-Kolmogorov transit integral depends on the types of beliefs and messages used in the factor graph. These operations are underpinned by various convolutions and products of infinite functions [6], as shown by eq. (6). Simplifications to the Chapman-Kolmogorov equations are commonly performed using a max-product type approach, for example using linearized Gaussian beliefs and finding the parametric extremum (i.e. mean) of the negative log which results in a non-linear least squares as used in iSAM2 [17]. Other non-parametric methods [7] try to preserve the full belief estimates, and results presented hereafer will all use the multimodal-iSAM algorithm [6] to perform numerical calculations.

4.4 Clique Recycling Strategies

We propose computational controls (similar to out-marginalization in conventional augmented filtering or fixed-lag) that are available through basic symbolic operations on the tree after a complete factor graph has been built. Five closely related strategies with different advantages are presented: 1) a full batch solution; 2) varying downward belief propagation; 3) upward marginalization by criteria; 4) incremental belief propagation; and 5) combination of 2, 3, and 4. We show that the iSAM2 [17] approach is a special case of 2 and 4, and postulate that strategy 5—i.e., naturally incorporating fixed-lag with incremental reuse and downward selection—offers the most versatile resource constrained solution. Each case is illustrated with a canonical example in Section 5.1.

The suggested approach allows the user to simply add all information to the factor graph once and let the solver adapt to fit the available computational resources. An added benefit is that full access to all five message strategies are maintained at all times. The suggested approach does not require preemptive modification of the factor graph as has been required in previous methods.

The calculation of upward messages will depend on the types of variables inside a clique, with three identified possibilities: variables for which a belief exists and an update is not required; variables for which no belief exists or an update is requested; or a combination thereof, producing a standard, fully marginalized, or partially marginalized clique, respectively. Each case is illustrated in Figure 3.

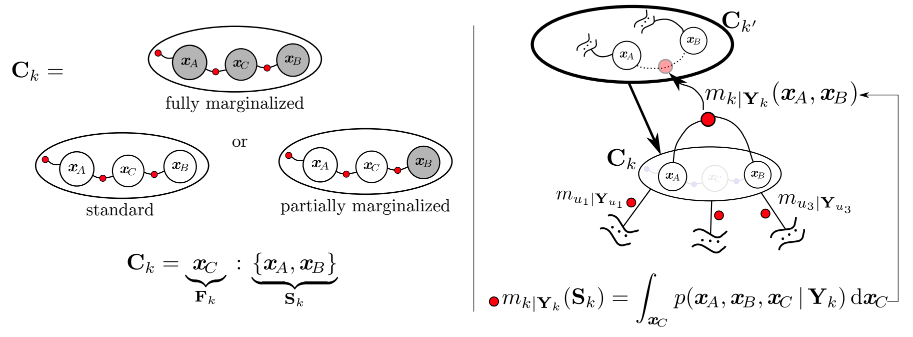
$\mathbf{C}_k =$
$\boldsymbol{x}_A$ $\boldsymbol{x}_C$ $\boldsymbol{x}_B$
fully marginalized
or
standard
partially marginalized
$\mathbf{C}_k = \underbrace{\boldsymbol{x}_C}_{\mathbf{F}_k} : \underbrace{\{\boldsymbol{x}_A, \boldsymbol{x}_B\}}_{\mathbf{S}_k}$
$\mathbf{C}_{k'}$
$m_{k|\mathbf{Y}_k}(\boldsymbol{x}_A, \boldsymbol{x}_B)$
$\mathbf{C}_k$
$m_{u_1|\mathbf{Y}_{u_1}}$
$m_{u_3|\mathbf{Y}_{u_3}}$
● $m_{k|\mathbf{Y}_k}(\mathbf{S}_k) = \int_{\boldsymbol{x}_C} p(\boldsymbol{x}_A, \boldsymbol{x}_B, \boldsymbol{x}_C \mid \mathbf{Y}_k)\, \mathrm{d}\boldsymbol{x}_C$

Fig. 3. *Left*: Clique cases to be handled when carrying out intra-clique operations. *Right*: showcases the role of marginalization messages during the upward solve.

Fully Marginalized Clique is the case where all variables in a clique are marked for marginalization and have all previously been inferred as frontal variables in a downsolve step. Since variable beliefs will not be updated, the only required operation is calculating the upward belief message $m_{k|\mathbf{Y}}$ using eq. (7).

If older factor graph information lands downwards in the tree, a boundary can be found where no changes happen below that point. A previous identical message $m_{k|\mathbf{Y}}\,(\Theta_{S,k})$ can be used and transmitted up to the clique's parent, which can then immediately incorporate it in its Chapman-Kolmogorov computation as shown in eq. (6). By limiting upward propagation of information from that boundary, all information lower down is effectively marginalized—analogous to causal HMMs. Sliding window operation is achieved by considering belief message passing onhy from boundaries of fully marginalized cliques. Stale portions of the tree below the boundary cliques can be deleted, archived, or simply not loaded as part of the tree factorization.

It is also possible to have fully marginalized parent cliques when inference still needs to occur in child cliques below them (shown later in Figure 4). This simply means that upward messages will not be propagated into the marginalized parent, but downward messages from the parent will be used in the unmarginalized child clique calculations as they usually would in a full batch solve case.

Partially Marginalized Cliques have some but not all variables marked for marginalization. Variables marked for marginalization (and which have previously been downsolved as a frontal) should not be updated during the Chapman-Kolmogorov computations, but their existing value should be used to influence surrounding variables which are not marked as frozen.

As an example illustration of solving a specific Chapman-Kolmogorov case, take clique $\mathbf{C}_k = \boldsymbol{x}_C : \boldsymbol{x}_A, \boldsymbol{x}_B$ from Figure 3. Given existing beliefs $\boldsymbol{x}_A, \boldsymbol{x}_B, \boldsymbol{x}_C = \hat{x}_a, \hat{x}_b, \hat{x}_c$, we can build the posterior $M_{k|\mathbf{Y}_k}(\hat{x}_c, \hat{x}_a, \hat{x}_b)$ and use (7) to get the belief message while recalculating only the non-marginalized variables

$$m_{k|\mathbf{Y}}(\boldsymbol{x}_A, \boldsymbol{x}_B) = \int_{\boldsymbol{x}_C} M_{k|\mathbf{Y}_k}(\hat{x}_a, \hat{x}_b, \hat{x}_c)\, \mathrm{d}\boldsymbol{x}_C \tag{9}$$

This marginalized message can be incorporated into the parent's partial posterior computation as one of the right-most terms in eq. (6).

Again, using clique $\mathbf{C}_k$ from the example in Figure 3, we can focus on the case where the separator variable $\boldsymbol{x}_B$ is marked for marginalization, but $\boldsymbol{x}_A$ and $\boldsymbol{x}_C$ are still to be updated. With a previous estimate $\boldsymbol{x}_B = \hat{x}_b$, the goal is to compute the belief message

$$m_{k|\mathbf{Y}}(\boldsymbol{x}_A, \boldsymbol{x}_B = \hat{x}_b) = \int_{\boldsymbol{x}_C} M_{k|\mathbf{Y}_k}(\boldsymbol{x}_A, \boldsymbol{x}_B = \hat{x}_b, \boldsymbol{x}_C)\, \mathrm{d}\boldsymbol{x}_C.$$

Using local likelihoods and priors from the clique, the partial posterior becomes

$$M_{k|\mathbf{Y}_k}(\boldsymbol{x}_A, \boldsymbol{x}_C, \boldsymbol{x}_B) = p\,(\,\boldsymbol{z}_3 \,|\, \boldsymbol{x}_B, \boldsymbol{x}_C\,)\, p\,(\,\boldsymbol{z}_2 \,|\, \boldsymbol{x}_A, \boldsymbol{x}_C\,)\, p_{\boldsymbol{z}_1}(\boldsymbol{x}_A).$$

For example, only one factor $p\,(\,\boldsymbol{z}_3 \,|\, \boldsymbol{x}_B, \boldsymbol{x}_C\,)$ involves the now fixed variable $\boldsymbol{x}_B$. The influence on the marginalized variable, coupled with the measurement $\boldsymbol{z}_3$, needs to be represented in the remaining variables. Thus, a convolution step from fixed $\hat{x}_B$

$$p\,(\,\boldsymbol{z}_3 \,|\, \boldsymbol{x}_B, \boldsymbol{x}_C\,)\, p\,(\,\boldsymbol{x}_B = \hat{x}_b\,) \propto p\,(\,\boldsymbol{x}_C, \boldsymbol{x}_B = \hat{x}_b \,|\, \boldsymbol{z}_3\,)$$

yields the following expression for the partial posterior, where we compute the joint on $\boldsymbol{x}_A, \boldsymbol{x}_C$ but not re-solve marginalized $\boldsymbol{x}_B$

$$M_{k|\mathbf{Y}_k}(\boldsymbol{x}_A, \boldsymbol{x}_C, \boldsymbol{x}_B = \hat{x}_b) \propto p\,(\,\boldsymbol{x}_C, \boldsymbol{x}_B = \hat{x}_b \,|\, \boldsymbol{z}_3\,)\, p\,(\,\boldsymbol{z}_2 \,|\, \boldsymbol{x}_A, \boldsymbol{x}_C\,)\, p_{\boldsymbol{z}_1}(\boldsymbol{x}_A),$$

with which we can proceed to compute the belief message $m_{k|\mathbf{Y}}(\boldsymbol{x}_A, \hat{x}_b)$.

5 Experimental Results

5.1 Simulation: Circular Example

Figure 4 shows a simulated trajectory where a robot travels counter clockwise in one complete circle. Odometry is used to define pose to pose rigid transform constraints. A small turn rate bias is added to stretch the dead reckoning circle into a slightly bigger incomplete ring. A single landmark $\boldsymbol{l}_1$ is observed from the starting pose $\boldsymbol{x}_0$, and later re-observed at the last pose $\boldsymbol{x}_{10}$ (identical to $\boldsymbol{x}_0$). The experiment starts by driving halfway (poses $\boldsymbol{x}_0$ through $\boldsymbol{x}_5$) and calculating a solution, with an associated Bayes tree shown at the top right of Figure 4. Finally, travel for poses $\boldsymbol{x}_6$ through $\boldsymbol{x}_{10}$ is completed and three distinct second solutions are computed using three closely related strategies.

Solving the factor graph after adding poses $\boldsymbol{x}_6$-$\boldsymbol{x}_{10}$ shows the intimate relationship between marginalization and incremental updates [16]. The example is used to showcase the five distinct strategies described in Section 4.4:

1. Batch Solution: The first row in Figure 4 shows a full batch solution (Sections 4.1 through 4.3) of the first five poses with landmark $\boldsymbol{l}_1$, where all messages are passed from the leaves to root and back down again.

2. Limited Down Pass, Tolerance or Priority: The designer may choose at what priority or to what depth to propagate downward messages after an upward pass. Rows 1, 3, and 4 of Figure 4 perform complete downward message passes, while row 2 force-marginalizes a tree branch (blue). When only a few marginals are required, one could manipulate the variable ordering to have those variables in the root and only send upward messages to solve those variables in the root. Further work regarding task-specific variable orderings is forthcoming.

3. Upward Marginalization by Criteria: This strategy is a replacement for the time complexity of conventional fixed-lag operation (i.e., forced variable marginalization) but instead preserves the full factor graph with all data, and only modifies the message passing on the Bayes tree. For example, a naive variable marginalization approach where a fixed window of variables are marked for marginalization based on the sequence they were added to the factor graph, as shown by the shaded nodes in Figure 1. We call this a FIFO marginalization strategy which is directly analogous to augmented or fixed-lag filtering/smoothing. This approach is shown on the second row of Figure 4 with the green trace as the second half and only updated portion of the factor graph.

4. Incremental Belief Propagation: Slight modifications to the factor graph likely mean that major portions of the tree will be repeated. The iSAM2 algorithm [17] specifically uses this fact by minimizing tree changes via variable ordering constraints (using CCOLAMD [3]) by *unhooking* and *rehooking* large parts of the tree after new information is incorporated. The third row in Figure 4 illustrates an incremental update. Given our analysis in Section 4, we can now clearly identify the incremental operation as fundamentally equivalent to clique marginalization. The authors stress that the equivalent fixed-lag smoothing connection on the Bayes tree has to the best of our knowledge not been identified or reported.

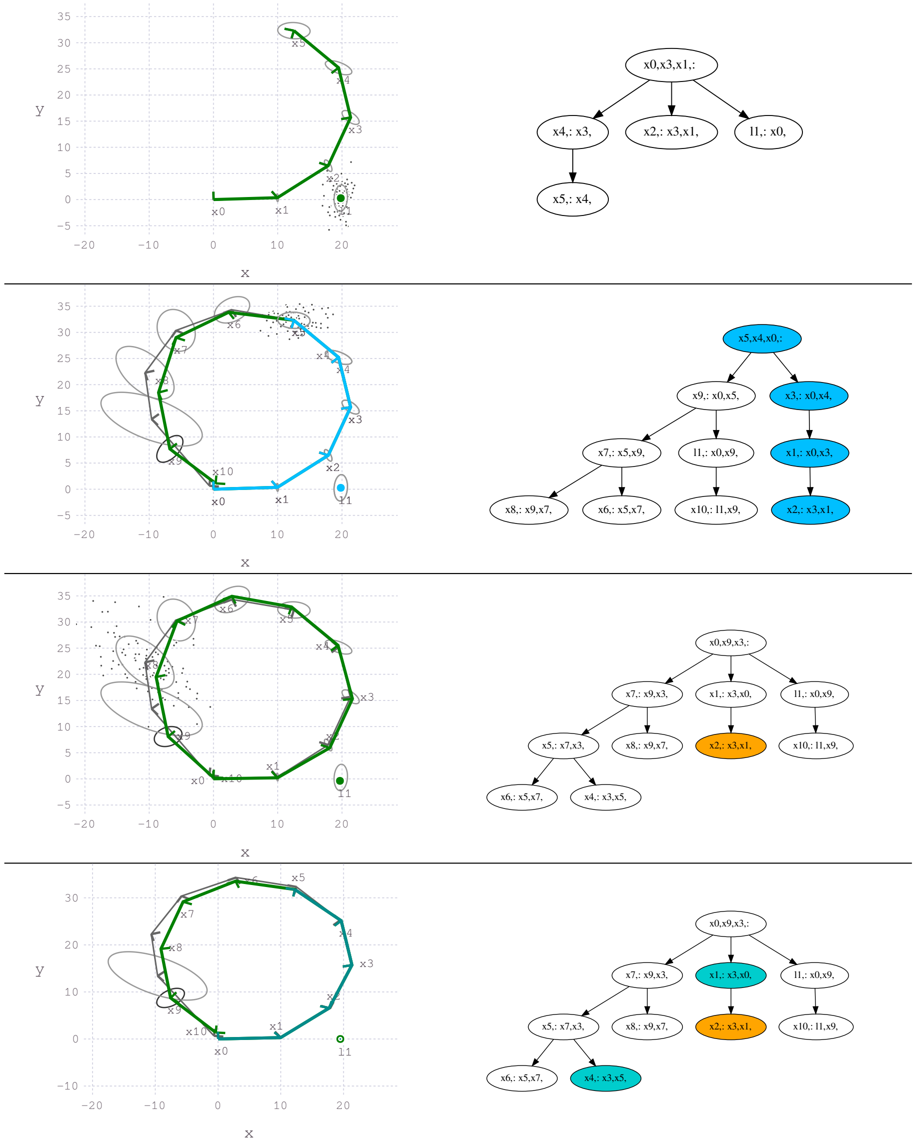

Fig. 4. Canonical example of a robot trajectory with eleven poses driving counter clockwise in a circle with one landmark sighted at the beginning and end – i.e. constructing a factor graph similar to Figure 1. ***Top row***: first five poses after a batch solution with associated Bayes tree. ***Second row***: forced-marginalization for all but last five poses. ***Third row***: incremental tree update similar to iSAM2 [17] with one reused clique in orange. ***Fourth row***: combined marginalized and incremental case. Green traces show latest solution; gray traces show estimates before tree solve. Solutions were computed with a nonparametric solver, multimodal-iSAM [2, 6, 7], with covariance ellipses fitted (for visualization only). The marginalized-only tree was purposefully adjusted from standard AMD variable ordering to show the *marginalized-parent* case; all other trees are from standard AMD variable ordering.

5. Combination of 2, 3, & 4: We define *clique recycling* as the case when, during upward belief propagation, cliques are reused either from a previous incremental update or being marked as marginalization due to some criteria. We emphasize that this approach uses clique marginalization to focus resource utilization, and does not require pre-processing of the factor graph.

5.2 Manhattan World Dataset

A larger scale analysis similar to the toy problem is carried out using the Manhattan world dataset. To quantify the relationship between a full incremental smoothing approach and a tree-based marginalization strategy, the corresponding number of cliques affected are compared at each update step. This metric is additionally paired with the estimation accuracy achieved by each method, thus comparing compute resources and performance.

The graphics in Figure 5 show a comparison between the same type of strategies seen in Figure 4: *incremental* belief propagation, Strategy 4 (iSAM2); *b100* represents *b*oth forced marginalization and clique reuse (Strategy 5, Combination of 2, 3, & 4), updating only the 100 most recent variables; similar for *b20* and *b10*, with corresponding window sizes; and the *batch* solution (Strategy 1).

As expected, the *incremental* strategy very closely approximates the *batch* solution, as it updates variables and incorporates loop closures as needed. As we start to tradeoff accuracy for better time complexity, the estimated trajectory naturally deviates from the reference. This is seen in Figure 5 (left), where the smaller the smoothing window size, the greater the displacement. This hit in accuracy mainly stems from the inability to incorporate loop closures outside the marginalization window.

In turn, the benefit in compute resources from the tradeoff is represented in Figure 5 (right), where the number of cliques to be updated at each time step is shown. The strategies leveraging both recycled computations as well as clique marginalization require fewer updates per time step, and avoid the sharp

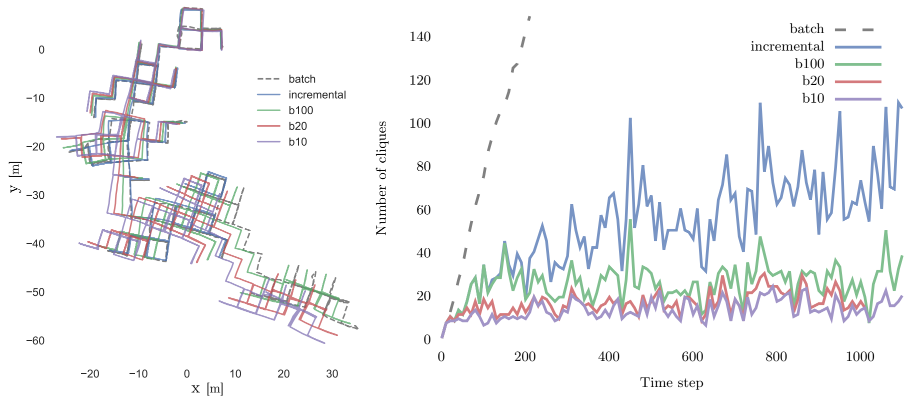

Fig. 5. Trajectory estimates and time history of the number of clique operations for the distinct tree-based inference strategies obtained using a the Manhattan dataset.

Table 1. Statistics for the number of cliques updated at each time step corresponding to the Manhattan scenario presented in Figure 5, with each row representing a different strategy (total of 1,101 time steps). Units represent number of cliques.

	Total updated	Cliques updated per step			
Strategy	[cliques]	**Median**	**Mean**	**Stddev**	**Max**
Incremental	6,026 (100%)	55.00	54.29	23.05	110.00
Both 100	2,919 (48.5%)	28.50	28.89	10.17	56.00
Both 20	1,939 (32.2%)	17.00	17.22	5.21	31.00
Both 10	1,575 (26.1%)	13.00	14.19	4.63	25.00

Table 2. Absolute position error statistics taken with respect to the batch solution for different inference strategies (total of 1,101 time steps). Units represent meters. Table data and plots computed using evo.

	Total updated	Trajectories absolute position error					
Strategy	[cliques]	**Median**	**Mean**	**Stddev**	**Max**	**RMSE**	**SSE**
Incremental	6,026 (100%)	0.23	0.275	0.186	1.203	0.332	82.91
Both 100	2,919 (48.5%)	0.421	0.496	0.312	1.685	0.586	346.216
Both 20	1,939 (32.2%)	0.496	0.529	0.265	1.791	0.592	364.233
Both 10	1,575 (26.1%)	0.628	0.622	0.248	1.385	0.669	504.245

peaks seen by the incremental approach, triggered by the computational demand to close big loops. The statistics for the time history of updated cliques for all approaches are encapsulated in Table 1, whereas the corresponding absolute position error statistics are included in Table 2. It is important to note that even though some loop closures are not being incorporated by the marginalization-driven approaches, they are still able to be detected. This capability comes from keeping the entire, albeit partially "frozen," tree, allowing data association. This choice entails a greater space complexity, but affords on-demand asynchronous updates using the available global structure of the problem and the recovery of the full SLAM solution at any point.

Such an example is shown in Figure 6. After utilizing the "cheap" strategy for a period of time, a window of inactivity opens up (robot standing still, car is idling at a traffic light, etc) and the possibility to "rebase" the current state estimates is presented. The full SLAM solution is computed by "releasing" messages to the hitherto "frozen" cliques, and extracting their new potentials. Afterwards, the system has the choice to operate in an incremental manner, set a marginalization window, or fully filter—all choices easily accomplished by working with simple marginalization operations directly on the Bayes tree.

The ability to explore the presented tradespace opens up an avenue for designing flexible and adaptive inference strategies using the Bayes tree. Characterization of the environment (e.g., how often do loop-closures tend to occur) and knowledge of system requirements (e.g., need to operate above a precision threshold), paired with the ability to seamlessly and trivially transition between inference strategies (a set of simple operations and message passing schemes), give rise to highly versatile inference algorithms. As shown in Figure 6, the method offers flexibility to customize the best marginalization strategy to limit the number of cliques to update.

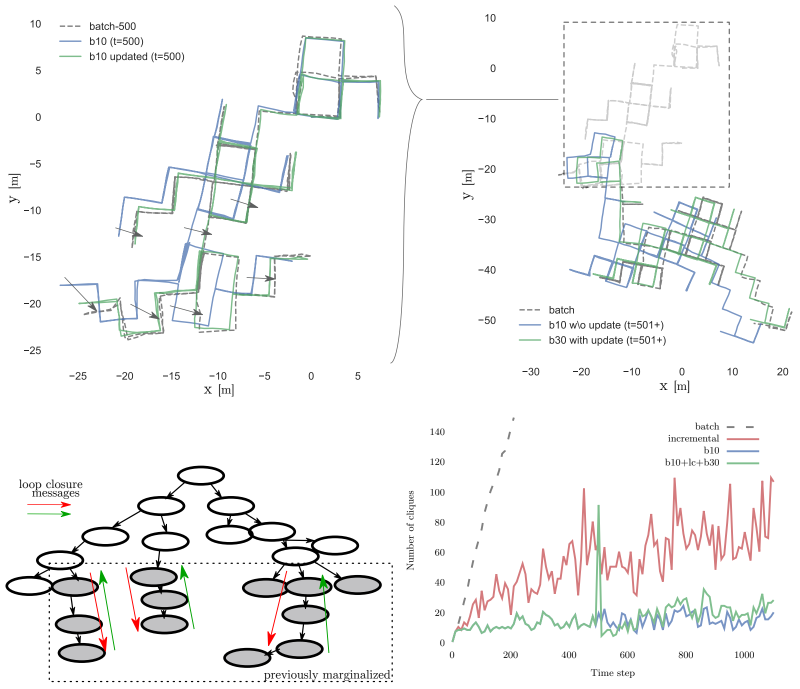

Fig. 6. On-demand loop closure to recover full SLAM solution from a marginalization-based inference strategy. Top row: the effect of the loop closure is shown on the left, and the resulting trajectory estimate obtained with or without the update is shown on the right. Bottom row: a symbolic depiction of how the loop closure is incorporated by releasing and collecting messages to and from the marginalized sections of the tree is shown on the left; of the number of updated cliques is shown on the right, where the green spike at time step 500 corresponds to the loop closure when the hitherto solutions gets "rebased".

6 Conclusions

This work intends to better formalize the inference problem definition for heterogeneous data fusion, SLAM, and state-estimation tasks, where the fundamental operation underlying most, if not all, existing methods is the Chapman-Kolmogorov transit equation. This work provides a framework for developing a variety of future SLAM algorithms beyond the well-known Gaussian minimization approaches. By characterizing the marginalization operations on the Bayes tree, we showed how the incremental update strategy of iSAM2 is a special case, and how this work should allow designers more freedom to develop predictable, resource constrained factor graph-style algorithms.

Acknowledgements This work was partially supported by the Office of Naval Research under grants N00014-18-1-2832 and MURI N00014-19-1-2571, and a National Science Foundation award IIS-1318392 and MIT Portugal Program.

Bibliography

[1] Stefan Arnborg, Derek G Corneil, and Andrzej Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM Journal on Algebraic Discrete Methods*, 8(2):277–284, 1987.

[2] Contributors and Packages. Caesar.jl, 2020. https://github.com/JuliaRobotics/Caesar.jl.

[3] T.A. Davis, J.R. Gilbert, S.I. Larimore, and E.G. Ng. A column approximate minimum degree ordering algorithm. *ACM Trans. Math. Softw.*, 30(3):353–376, 2004.

[4] F. Dellaert and M. Kaess. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *Intl. J. of Robotics Research*, 25(12):1181–1203, December 2006.

[5] Frank Dellaert, Michael Kaess, et al. Factor graphs for robot perception. *Foundations and Trends® in Robotics*, 6(1-2):1–139, 2017.

[6] D. Fourie. *Multi-Modal and Inertial Sensor Solutions to Navigation-type Factor Graphs*. PhD thesis, Massachusetts Institute of Technology and Woods Hole Oceanographic Institution, 2017.

[7] D. Fourie, J.J. Leonard, and M. Kaess. A nonparametric belief solution to the Bayes tree. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, IROS*, Daejeon, Korea, Oct 2016.

[8] Dehann Fourie, Pedro Vaz Teixeira, and John Leonard. Non-parametric Mixed-Manifold Products using Multiscale Kernel Densities. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6656–6662. IEEE, 2019.

[9] U. Frese. Treemap: An $O(\log n)$ algorithm for simultaneous localization and mapping. In *Spatial Cognition IV*, pages 455–476. Springer Verlag, 2005.

[10] Udo Frese and Lutz Schroder. Closing a million-landmarks loop. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5032–5039. IEEE, 2006.

[11] Paul D Groves. *Principles of GNSS, inertial, and multisensor integrated navigation systems*. Artech House, GNSS Technology and Application Series, 2008.

[12] Pinar Heggernes and Pontus Matstoms. *Finding good column orderings for sparse QR factorization*. University of Linköping, Department of Mathematics, 1996.

[13] Ming Hsiao and Michael Kaess. MH-iSAM2: Multi-hypothesis iSAM using Bayes Tree and Hypo-tree. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1274–1280. IEEE, 2019.

[14] Jerry Hsiung, Ming Hsiao, Eric Westman, Rafael Valencia, and Michael Kaess. Information sparsification in visual-inertial odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1146–1153. IEEE, 2018.

[15] Finn V Jensen and Frank Jensen. Optimal Junction trees. In *Uncertainty Proceedings 1994*, pages 360–366. Elsevier, 1994.

[16] M. Kaess, V. Ila, R. Roberts, and F. Dellaert. The Bayes tree: An algorithmic foundation for probabilistic robot mapping. In *Intl. Workshop on the Algorithmic Foundations of Robotics, WAFR*, Singapore, December 2010.

[17] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 31:217–236, February 2012.

[18] Uffe Kjærulff. Inference in Bayesian networks using nested junction trees. In *Learning in Graphical Models*, pages 51–74. Springer, 1998.

[19] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques.* The MIT Press, Cambridge, MA, 2009.

[20] F.R. Kschischang, B.J. Frey, and H-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inform. Theory*, 47(2), February 2001.

[21] J. J. Leonard and H. F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *Proc. IEEE Int. Workshop on Intelligent Robots and Systems*, pages 1442–1447, Osaka, Japan, 1991.

[22] B. Mu, L. Paull, A. Agha-mohammadi, J. Leonard, and J. How. Two-stage focused inference for resource-constrained minimal collision navigation. *IEEE Trans. Robotics*, 33(1):124–140, 2017.

[23] M.A. Paskin. Thin junction tree filters for simultaneous localization and mapping. In *Intl. Joint Conf. on Artificial Intelligence*, pages 1157–1164, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc.

[24] Judea Pearl. Bayesian networks. *Department of Statistics, UCLA*, 2011.

[25] A. Ranganathan, M. Kaess, and F. Dellaert. Loopy SAM. In *Intl. Joint Conf. on Artificial Intelligence*, pages 2191–2196, Hyderabad, India, 2007.

[26] Sheldon M Ross. *Introduction to probability models.* Academic press, 2014.

[27] Aleksandr V Segal and Ian D Reid. Hybrid inference optimization for robust pose graph estimation. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 2675–2682. IEEE, 2014.

[28] Robert E Tarjan and Mihalis Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal on computing*, 13(3):566–579, 1984.

[29] M.R. Walter, R.M. Eustice, and J.J. Leonard. Exactly sparse extended information filters for feature-based SLAM. *Intl. J. of Robotics Research*, 26(4):335–359, 2007.

[30] Yair Weiss and William T Freeman. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. In *Advances in neural information processing systems*, pages 673–679, 2000.

[31] S. Williams, V. Indelman, M. Kaess, J.J. Leonard R. Roberts, and F. Dellaert. Concurrent filtering and smoothing: A parallel architecture for real-time navigation and full smoothing. *The International Journal of Robotics Research*, 33(12):1544–1568, 2014.

[32] Paul Zarchan and Howard Musoff. *Fundamentals of Kalman Filtering: A Practical Approach, 3e.*, volume 232 of *Progress in Astronautics and Aeronautics.* AIAA, 2009.