Approximating Marginalization with Sparse Global Priors for Sliding Window SLAM-Graphs

Daniel Wilbers[1] Lars Rumberg[2] Cyrill Stachniss[3]

***Abstract*— Most autonomous vehicles rely on some kind of map for localization or navigation. Outdated maps however are a risk to the performance of any map-based localization system applied in autonomous vehicles. It is necessary to update the used maps to ensure stable and long-term operation. We address the problem of computing landmark updates live in the vehicle, which requires efficient use of the computational resources. In particular, we employ a graph-based sliding window approach for simultaneous localization and incremental map refinement. We propose a novel method that approximates sliding window marginalization without inducing fill-in. Our method maintains the exact same sparsity pattern as without performing marginalization, but simultaneously improves the landmark estimates. The main novelty of this work is the derivation of sparse global priors that approximate dense marginalization. In comparison to state-of-the-art work, our approach utilizes global instead of local linearization points, but still minimizes linearization errors. We first approximate marginalization via Kullback-Leibler divergence and then re-calculate the mean to compensate linearization errors. We evaluate our approach on simulated and real data from a prototype vehicle and compare our approach to state-of-the-art sliding window marginalization.**

I. Introduction

Maps are a central ingredient for most navigation and trajectory planning systems. Updating and refining these maps is a fundamental task for robust long-term operation of autonomous vehicles. Any localization system that relies on a pre-built landmark map suffers from maps getting outdated over time. In the case of automated driving, these changes are for example caused by roads being modified or built completely new. In order to allow stable and steady vehicle operation, a localization and mapping system must be able to refine its map on-the-fly without relying on back-end updates. Although updates via back-end may provide validated information, they are not available in real-time, which makes it necessary to handle environment changes initially in the vehicle. Furthermore, any localization system that is able to update its map can be bootstrapped with a limited number of landmarks and then incrementally add more features to its map, which increases scalability and modularity for initial systems.

In this paper, we focus on the problem of estimating the positions of previously unmapped landmarks on-the-fly in the vehicle based on a state-of-the-art graph-based localization and mapping approach [1]. In particular, we make use of a sliding window formulation as in [2] to keep the problem size computationally tractable. We present a novel approach, which approximates marginalization without inducing any fill-in, as shown in Fig. 1.

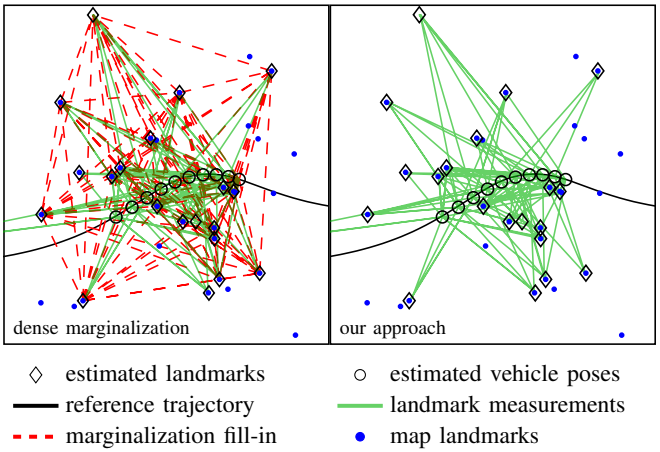


Fig. 1: The influence of marginalization on the graph structure. The illustration shows that our approach does not suffer from fill-in, which reduces computational complexity and thus is faster to compute. Simultaneously, our approach provides a comparable accuracy of the resulting estimate.

In comparison to nonlinear factor recovery [3] and generic linear constraints [4], we focus on the special case of sliding window graphs for which different assumptions hold. On the one side we do not need to cope with the deficient rank problematic, on the other we target for the special approximation topology with individual priors for each state.

The main contribution of this paper is a novel sparsification scheme for marginalization in sliding window graphs. We achieve this by deriving individual global priors for all states involved in marginalization. This allows us to estimate incremental map updates in the context of automated driving. In sum, we make four key claims that our approach

(i) approximates sliding window marginalization with sparse global priors,
(ii) utilizes global linearization points and thus does not require local optimization,
(iii) computes conservative landmark positions for incremental map refinement,
(iv) has exactly the same sparsity pattern compared to using no marginalization, but provides more accurate estimates.

[1]Daniel Wilbers is with Volkswagen Group Research, Wolfsburg, and Institute of Geodesy and Geoinformation, University of Bonn, Germany.
[2]Lars Rumberg is with Volkswagen Group Research, Wolfsburg.
[3]Cyrill Stachniss is with Institute of Geodesy and Geoinformation, University of Bonn, Germany.

II. Related Work

In the last decades a wide range of research has been done in the field of localization and mapping [5]. Cadena et al. [6] provide a general overview of the open challenges in the field, whereas Bresson et al. [7] focus on the current state of the art with respect to automated driving. The predominant methods for solving the SLAM problem are particle methods, extended Kalman filter (EKF), and graph-based approaches. We refer to Stachniss et al. [8] for a comparison of these methods. In this paper, we apply graph-based optimization for vehicle localization and incremental map refinement.

Focused on the special needs in automated driving, mapping companies are working on creating maps, which can be used for vehicle localization. Further work on using pre-built maps is presented by Roh et al. [9], in which the integration of shapefile data is shown and Kümmerle et al. [10], who use aerial images for the derivation of pose-graph constraints. Vysotska and Stachniss [11], [12] show how to include data from Open Street Map for improving localization. A fundamental challenge to all approaches relying on pre-built maps is that the maps are getting outdated over time. For example in automated driving temporary construction sides and newly built streets have a direct impact on the correctness of the map and thus localization. This makes it necessary to refine the pre-built map if changes occur. In this paper, we investigate how to refine the map by adding missing features to it. We especially focus on how to compute the positions of these features online while simultaneously localizing the vehicle. Similar to our prior work [13] we employ a sliding-window filter over poses and landmarks. We make use of delayed marginalization similar to Sibley et al. [2], but here show how to apply a further sparsification step Compared to full-graph solutions, marginalization allows to preserve information, while limiting the state dimension. The latter is necessary to ensure computational tractability in online scenarios. The common drawbacks of marginalization are the missing option of relinearization and induced fill-in. The latter one negatively influences the sparsity pattern and thus increases computation costs. For the case of pose graphs Merfels et al. [14] presented in their work how to compute priors that approximate pose graph marginalization. Several research focuses on reducing the number of constraints induced by marginalization for general graphs. This step is commonly called sparsification. The Sparse Extendend Information Filter (SEIF) by Thrun et al. selectively deactivates constraints between robot pose and landmarks, but keeps intra-landmark constraints [15]. Vial et al. present a technique for conservative graph sparsification via Kullback-Leibler Divergence (KL) minimization [16]. Kretzschmar et al. use a Chow-Liu tree to approximate marginalization and maintain a sparse graph [17]. Carlevaris-Bianco and Eustice extend this work by introducing a conservative tree approximation [18], [19]. Using more complex topologies than trees is suggested by Vallvé et al., who present an optimization technique for sparsification [20], [21]. Without explicit marginalization Choudhary et al. reduce poses and landmarks based on expected information gain [22]. Ta et al. present a near optimal method for pose marginalization by re-parameterizing to local spaces [23]. Recently, Hsiung et al. [24] investigated how to approximate marginalization with sparse priors. While their work focuses on localization errors in the visual-inertial odometry domain, we focus on the estimation of landmark positions for automated driving. Additionally, they do not cover the effects of different linearization points and rely on suboptimal global linearization points. Eckenhoff et al. provide a derivation that suggests to use local linearizations points for marginalization, but requires local optimization of the marginalization blanket [25]. Mazuran et al. show how to recover artificial measurements from marginalized information, which allows for relinearization and further argue that local linearization points provide superior results in incremental mapping scenarios [26], [3]. Compared to such prior work, we show how to avoid using local linearization points and derive how to utilize global linearization points by respecting gradient effects inside our sparse priors. We study the effects for the case of sliding window incremental map refinement.

III. Graph-based Sliding Window Localization

In the following, we explain our sliding window graph-based framework, which uses the graph formulation and notation of Grisetti et al. [1]. We estimate a set of vehicle poses $\boldsymbol{x}^{\mathrm{p}} = \{\boldsymbol{x}^{\mathrm{p}}_{t_S}, \dots, \boldsymbol{x}^{\mathrm{p}}_{t_N}\}$ and landmarks $\boldsymbol{x}^{\mathrm{l}}$, where t_S is the time of the first pose and t_N the time of the last pose in the sliding window. The landmarks $\boldsymbol{x}^{\mathrm{l}}$ are measured in between t_S and t_N. For clarity reasons, we omit the time subscripts in the following. We represent the optimization problem of finding the most likely states $\boldsymbol{x} = \begin{bmatrix}\boldsymbol{x}^{\mathrm{p}} & \boldsymbol{x}^{\mathrm{l}}\end{bmatrix}$ over a set of measurements $\boldsymbol{z}$ as the maximum a posteriori problem

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmax}}\ p(\boldsymbol{z} \mid \boldsymbol{x}) p(\boldsymbol{x}). \tag{1}$$

Assuming Gaussian noise and independent measurements, the optimization problem is equivalent to minimizing a sum over individual cost functions

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmin}} \sum_k \mathbf{e}_k(\boldsymbol{x}, \boldsymbol{z}_k)^\top \boldsymbol{\Omega}_k \mathbf{e}_k(\boldsymbol{x}, \boldsymbol{z}_k), \tag{2}$$

with $\mathbf{e}_k(\boldsymbol{x}, \boldsymbol{z}_k)$ being the error function between measurement $\boldsymbol{z}_k$ and state vector $\boldsymbol{x}$, and the corresponding information matrix $\boldsymbol{\Omega}_k$. In detail, we estimate

$$\begin{aligned}\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmin}} & \sum_i \mathbf{e}^{\text{odo}}(\boldsymbol{x}^{\mathrm{p}}, \boldsymbol{z}_i^{\text{odo}})^\top \boldsymbol{\Omega}_i^{\text{odo}} \mathbf{e}^{\text{odo}}(\boldsymbol{x}^{\mathrm{p}}, \boldsymbol{z}_i^{\text{odo}}) \\ & + \sum_i \mathbf{e}^{\text{abs}}(\boldsymbol{x}^{\mathrm{p}}, \boldsymbol{z}_i^{\text{abs}})^\top \boldsymbol{\Omega}_i^{\text{abs}} \mathbf{e}^{\text{abs}}(\boldsymbol{x}^{\mathrm{p}}, \boldsymbol{z}_i^{\text{abs}}) \\ & + \sum_i \mathbf{e}^{\text{lm}}(\boldsymbol{x}^{\mathrm{p}}, \boldsymbol{x}^{\mathrm{l}}, \boldsymbol{z}_i^{\text{lm}})^\top \boldsymbol{\Omega}_i^{\text{lm}} \mathbf{e}^{\text{lm}}(\boldsymbol{x}^{\mathrm{p}}, \boldsymbol{x}^{\mathrm{l}}, \boldsymbol{z}_i^{\text{lm}}) \\ & + \sum_i \mathbf{e}^{\text{map}}(\boldsymbol{x}^{\mathrm{l}}, \boldsymbol{z}_i^{\text{map}})^\top \boldsymbol{\Omega}_i^{\text{map}} \mathbf{e}^{\text{map}}(\boldsymbol{x}^{\mathrm{l}}, \boldsymbol{z}_i^{\text{map}}) \\ & + F^{\text{marg}}(\boldsymbol{x}),\end{aligned} \tag{3}$$

with the superscripts $^{\text{odo}}$ for odometry measurements between subsequent poses, $^{\text{abs}}$ for absolute pose measurements from

for example GNSS, $^{\text{lm}}$ for relative landmark measurements, $^{\text{map}}$ for map matches, and the term $F^{\text{marg}}(\boldsymbol{x})$ which captures the cost of marginalized measurements outside the sliding window. A description of the error functions is not part of this paper as they are similar to state-of-the-art graph-based optimization techniques [11], [1], [27]. We use Levenberg-Marquardt to solve the nonlinear optimization problem by linearizing (3) and computing iterative updates

$$\boldsymbol{H}\Delta\boldsymbol{x}^* = -\boldsymbol{b}. \tag{4}$$

We refer to Grisetti et al. [1] for a detailed derivation. Important for this work is that the sparsity of the matrix $\boldsymbol{H}$ is strongly related to how fast the linear system can be solved. The sparser the matrix, the lower the computational cost. The sparsity of $\boldsymbol{H}$ depends on the Jacobians of the error functions, which are determined by how we design the structure of the graph. The problem with sliding window graphs is that marginalization induces fill-in in $\boldsymbol{H}$ and thus destroys the sparsity pattern. In the following we tackle exactly this problem.

IV. Sliding Window Marginalization

In this section we describe how to preserve the information of measurements outside of the sliding window through marginalization. A difference to many graph-based approaches is that our temporal sliding window guarantees by design that there are always global factors in the marginalization blanket. We ensure this by including global information from previous time steps. Because of that, there is no need to use relative formulations as in e.g., [3], [25] and we avoid any deficient rank problematic in cases where only relative measurements are marginalized.

A. Calculating the Marginalization Prior

Our sliding window approach is based on marginalizing out the oldest pose and adding a new pose to the front of the graph in every timestep. As a consequence the number of poses in the graph is constant, whereas the number of landmarks is naturally bounded by the environment. Additionally, we marginalize out landmarks, as shown in Fig. 2, if the only connected pose is the last one in the sliding window and subject to marginalization.

We divide the set of all states $\boldsymbol{x} = \{\boldsymbol{x}_m, \boldsymbol{x}_n, \boldsymbol{x}_r\}$ into marginalization nodes $\boldsymbol{x}_m$, directly connected neighbor nodes $\boldsymbol{x}_n$, and the remaining states $\boldsymbol{x}_r$. In the following we call the set $\{\boldsymbol{x}_m, \boldsymbol{x}_n\}$ the *marginalization blanket*. Considering these definitions we rewrite the general optimization problem (2) as

$$\begin{aligned}\boldsymbol{x}^* &= \underset{\boldsymbol{x}}{\operatorname{argmin}} \sum_j \mathbf{e}_j(\boldsymbol{x}_m, \boldsymbol{x}_n, \boldsymbol{z}_j)^\top \boldsymbol{\Omega}_j \mathbf{e}_j(\boldsymbol{x}_m, \boldsymbol{x}_n, \boldsymbol{z}_j) \\ &\quad + \sum_{i_{\setminus j}} \mathbf{e}_i(\boldsymbol{x}_r, \boldsymbol{z}_i)^\top \boldsymbol{\Omega}_i \mathbf{e}_i(\boldsymbol{x}_r, \boldsymbol{z}_i) \\ &= \underset{\boldsymbol{x}}{\operatorname{argmin}}\, F^{m,n}(\boldsymbol{x}_m, \boldsymbol{x}_n) + F^r(\boldsymbol{x}_n, \boldsymbol{x}_r),\end{aligned} \tag{5}$$

where $i_{\setminus j}$ denotes the absence of any measurement connected to the marginalization nodes, $F^{m,n}$ is the cost of the marginalization blanket, and F^r captures the cost of the rest of the graph. Following the argumentation of Eckenhoff et al. [25], we write (5) as a cost function problem

$$\begin{aligned}C &= \min_{\boldsymbol{x}} \left(F^{m,n}(\boldsymbol{x}_m, \boldsymbol{x}_n) + F^r(\boldsymbol{x}_n, \boldsymbol{x}_r)\right) \\ &= \min_{\boldsymbol{x}_n, \boldsymbol{x}_r} \left(\min_{\boldsymbol{x}_m} F^{m,n}(\boldsymbol{x}_m, \boldsymbol{x}_n) + F^r(\boldsymbol{x}_n, \boldsymbol{x}_r)\right).\end{aligned} \tag{6}$$

We minimize the cost of the marginalization blanket $F^{m,n}(\boldsymbol{x}_m, \boldsymbol{x}_n)$ via Taylor approximation and solve for the optimal marginalization state $\boldsymbol{x}_m^*(\boldsymbol{x}_n, \check{\boldsymbol{x}}_m, \check{\boldsymbol{x}}_n)$ with linearization points $\check{\boldsymbol{x}}_m$ and $\check{\boldsymbol{x}}_n$ in the following. The Taylor approximation is given by

$$\begin{aligned}&F^{m,n}(\boldsymbol{x}_m, \boldsymbol{x}_n) \\ &\approx F^{m,n}(\check{\boldsymbol{x}}_m, \check{\boldsymbol{x}}_n) + \boldsymbol{b}^\top \Delta\boldsymbol{x} + \frac{1}{2}\Delta\boldsymbol{x}^\top \boldsymbol{H} \Delta\boldsymbol{x},\end{aligned} \tag{7}$$

where the Hessian $\boldsymbol{H}$, gradient $\boldsymbol{b}$, and $\Delta\boldsymbol{x}_{m,n}$ are defined as

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_m & \boldsymbol{H}_{mn} \\ \boldsymbol{H}_{mn}^\top & \boldsymbol{H}_n \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_m \\ \boldsymbol{b}_n \end{bmatrix}, \tag{8}$$

$$\Delta\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_m - \check{\boldsymbol{x}}_m \\ \boldsymbol{x}_n - \check{\boldsymbol{x}}_n \end{bmatrix} = \begin{bmatrix} \Delta\boldsymbol{x}_m \\ \Delta\boldsymbol{x}_n \end{bmatrix}. \tag{9}$$

Solving (7) for the optimal marginalization state $\boldsymbol{x}_m^*$ results in

$$\boldsymbol{x}_m^*(\boldsymbol{x}_n, \check{\boldsymbol{x}}_m, \check{\boldsymbol{x}}_n) = \check{\boldsymbol{x}}_m - \boldsymbol{H}_m^{-1}(\boldsymbol{b}_m + \boldsymbol{H}_{mn}^\top \Delta\boldsymbol{x}_n). \tag{10}$$

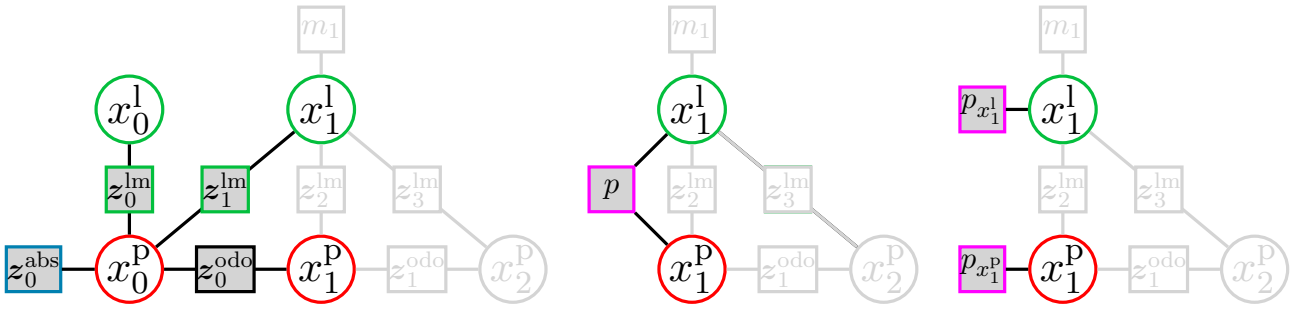
By substituting (10) back into (7), we render the cost of the marginalization blanket independent of the marginalized states $\boldsymbol{x}_m$, such that

$$\begin{aligned}F^{m,n}(\boldsymbol{x}_m, \boldsymbol{x}_n) &\approx F^{m,n}(\boldsymbol{x}_n, \check{\boldsymbol{x}}_n) \\ &= \frac{1}{2}\Delta\boldsymbol{x}_n^\top \boldsymbol{H}_t \Delta\boldsymbol{x}_n + \boldsymbol{b}_t^\top \Delta\boldsymbol{x}_n + c, \\ \text{with} \quad \boldsymbol{H}_t &= \boldsymbol{H}_n - \boldsymbol{H}_{mn}\boldsymbol{H}_m^{-1}\boldsymbol{H}_{mn}^\top, \\ \boldsymbol{b}_t &= \boldsymbol{b}_n - \boldsymbol{H}_{mn}\boldsymbol{H}_m^{-1}\boldsymbol{b}_m,\end{aligned} \tag{11}$$

and c being a constant that is omitted when applying the argmax operator. The interpretation behind (11) is that we have performed marginalization over the marginalization nodes $\boldsymbol{x}_m$ resulting in

$$p(\boldsymbol{x}_n) = \int p(\boldsymbol{x}_n, \boldsymbol{x}_m) \mathrm{d}\boldsymbol{x}_m = \mathcal{N}(\check{\boldsymbol{x}}_n, \boldsymbol{H}_t^{-1}). \tag{12}$$

We call the distribution $p(\boldsymbol{x}_n)$ the *marginalization prior*. Compared to using the Schur complement of (8) to compute $\boldsymbol{H}_t$ and $\boldsymbol{b}_t$, (11) emphasizes the effect of the gradient $\boldsymbol{b}_t$ on the optimization problem. Eckenhoff et al. [25] argue that the performed marginalization is only optimal if the linearization point $\check{\boldsymbol{x}}_n$ is the local optimum of the marginalization blanket and thus the gradient $\boldsymbol{b}_t$ vanishes. Extending their argumentation we show in Sec. V-B how to use a linearization point, that is not the local optimum.



(a) marginalization blanket (b) dense marginalization (c) individual global priors

Fig. 2: Illustration of our approach for computing global priors in order from left to right. All $\boldsymbol{z}$ are measurements, whereas $\boldsymbol{x}^{\mathrm{p}}$ are poses and $\boldsymbol{x}^{\mathrm{l}}$ are landmarks in the sliding window graph. (a) The highlighted nodes and factors are part of the marginalization blanket involved in our marginalization and sparsification process. (b) Graph after marginalizing out the oldest pose x_0^{p}. The dense factor p represents the marginalized information. (c) The result of our sparsification step. We compute an individual global prior for each remaining state of the marginalization blanket.

V. Global Prior Sparsification

In the following we show how to derive sparse global priors that approximate dense marginalization in closed form. Our sparsification scheme is designed to have the exact same sparsity pattern compared to using no marginalization. This means that our approach does not suffer from fill-in and simultaneously computes more precise estimates than without using marginalization. As we derive individual global priors for each involved state, we can visually draw them similar to other measurements. This is not only beneficial for data inspection, but also helps to understand the effect of our approximation.

A. Sparsifying the Marginalization Prior

Instead of applying the marginalization prior directly in the next sliding window we first sparsify the distribution $p(\boldsymbol{x}_n)$ to avoid fill-in and ensure sparsity in the graph. By design we compute a global prior for each state $\boldsymbol{x}_n$ inside the marginalization prior. We formulate the sparsification problem as a minimization of the Kullback-Leibler divergence (*KL*) as

$$\min_{\boldsymbol{\mu}_a, \boldsymbol{\Omega}_a} \mathcal{D}_{KL}\left(\mathcal{N}(\check{\boldsymbol{x}}_n, \boldsymbol{H}_t^{-1}) \parallel \mathcal{N}(\boldsymbol{\mu}_a, \boldsymbol{\Omega}_a^{-1})\right), \quad \text{with} \quad \mathcal{N}(\boldsymbol{\mu}_a, \boldsymbol{\Omega}_a^{-1}) = \prod_i \mathcal{N}(\boldsymbol{\mu}_{a_i}, \boldsymbol{\Omega}_{a_i}^{-1}), \tag{13}$$

where $\mathcal{N}(\boldsymbol{\mu}_{a_i}, \boldsymbol{\Omega}_{a_i}^{-1})$ is the approximated prior for each individual neighbor state in the marginalization blanket. In the multivariate normal case the KL-divergence reaches its minimum if both means are equal ($\boldsymbol{\mu}_a = \check{\boldsymbol{x}}_n$). Thus, the approximation problem reduces to

$$\begin{aligned}\min_{\boldsymbol{\mu}_a, \boldsymbol{\Omega}_a} \mathcal{D}_{KL} &= \min_{\Sigma} \left(\mathrm{tr}(\boldsymbol{\Omega}_a \boldsymbol{H}_t^{-1}) - \ln\left(\det(\boldsymbol{\Omega}_a \boldsymbol{H}_t^{-1})\right)\right) \\ &= \sum_i \left(\mathrm{tr}(\boldsymbol{\Omega}_{a_i}) - \ln\left(\det(\boldsymbol{\Omega}_{a_i}\{\boldsymbol{H}_t^{-1}\}_i)\right)\right),\end{aligned} \tag{14}$$

where we have applied the definition of $\mathcal{N}(\boldsymbol{\mu}_{a_i}, \boldsymbol{\Omega}_{a_i}^{-1})$ and use the notation $\{\boldsymbol{H}_t^{-1}\}_i$ to reference the i-th block diagonal entry of $\boldsymbol{H}_t^{-1}$. Deriving (14) with regard to $\boldsymbol{\Omega}_{a_i}$ yields for each individual prior the optimal information matrix

$$\boldsymbol{\Omega}_{a_i}^{*} = \{\boldsymbol{H}_t^{-1}\}_i^{-1}. \tag{15}$$

Given our specific topology in form of absolute priors (15) shows that we can simply ignore the covariances between states to get the optimal approximation. We denote the sparsified distribution as $\tilde{p}(\boldsymbol{x}_n) = \mathcal{N}(\check{\boldsymbol{x}}_n, \boldsymbol{\Omega}_a^{-1})$, where $\boldsymbol{\Omega}_a^{-1}$ is block-tridiagonal. Inserting the approximation into (11) yields

$$F^{m,n}(\boldsymbol{x}_m, \boldsymbol{x}_n) \propto \frac{1}{2} \Delta\boldsymbol{x}_n^{\top} \boldsymbol{\Omega}_a \Delta\boldsymbol{x}_n + \boldsymbol{b}_t^{\top} \Delta\boldsymbol{x}_n, \tag{16}$$

which is used in the following.

B. Computing Sparse Global Priors

By extending the proof of Eckenhoff et al. [25], we now show that we must not necessarily use the local optimum of the marginalization blanket as the linearization point $\check{\boldsymbol{x}}_n$. Instead we utilize the previous global estimate by including the gradient term inside the quadratic term. We first note that any cost term independent of the state variable is constant and neglected during optimization. This allows us to add the constant term $c = \boldsymbol{\Omega}_a^{-1}\boldsymbol{b}_t\boldsymbol{\Omega}_a\boldsymbol{\Omega}_a^{-1}\boldsymbol{b}_t$ to (16). By rearranging and completing the square, we obtain

$$\begin{aligned}&F^{p}(\boldsymbol{x}_n, \check{\boldsymbol{x}}_n) \propto \\ &\frac{1}{2}(\Delta\boldsymbol{x}_n + \boldsymbol{\Omega}_a^{-1}\boldsymbol{b}_t)^{\top}\boldsymbol{\Omega}_a(\Delta\boldsymbol{x}_n + \boldsymbol{\Omega}_a^{-1}\boldsymbol{b}_t) = \\ &\frac{1}{2}(\boldsymbol{x}_n - \underbrace{(\check{\boldsymbol{x}}_n - \boldsymbol{\Omega}_a^{-1}\boldsymbol{b}_t)}_{\hat{=}\boldsymbol{\mu}_p})^{\top}\boldsymbol{\Omega}_a(\boldsymbol{x}_n - \underbrace{(\check{\boldsymbol{x}}_n - \boldsymbol{\Omega}_a^{-1}\boldsymbol{b}_t)}_{\hat{=}\boldsymbol{\mu}_p}).\end{aligned} \tag{17}$$

We refer to the distribution

$$\hat{p}(\boldsymbol{x}_n) = \mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Omega}_a^{-1}) = \prod_i \mathcal{N}(\boldsymbol{\mu}_{p_i}, \boldsymbol{\Omega}_{a_i}^{-1}) \tag{18}$$

as the *sparse global prior distribution*. It consists of individual prior terms for each state in $\boldsymbol{x}_n$. By including the gradient term inside the quadratic term we compensate the effects of the gradient term without using the local optimum of the marginalization blanket as linearization point. Therefore our approach renders local optimization of the marginalization blanket unnecessary. Returning to our original sliding window optimization problem in (3) our *sparse global prior*

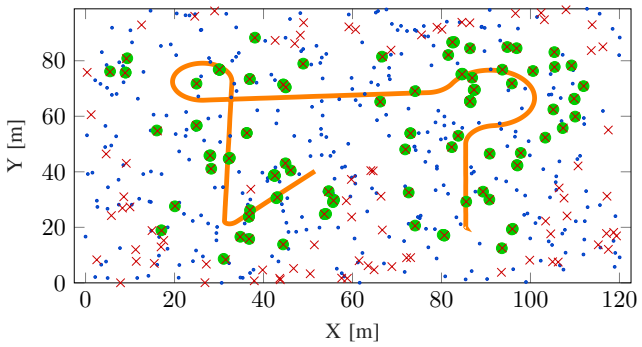


Fig. 3: Map and trajectory of our simulated dataset. All blue landmarks are used for localization, whereas the red crosses represent the ground truth for the estimated landmarks, which are shown in green. The notation is similar to Fig. 4.

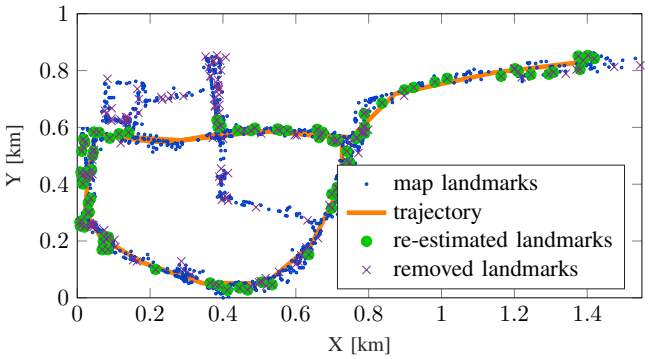


Fig. 4: Map and trajectory of our real world dataset. We randomly remove 20% of the landmarks from the map to evaluate our method. The map was recorded several month before the actual test drive. The trajectory and pole-like landmark measurements were recorded with a prototype vehicle. Shown in green are all the landmarks which are re-estimated for our evaluation.

distribution is included as the marginalization cost

$$F^{\text{marg}}(\boldsymbol{x}) = \sum_i (\boldsymbol{x}_n - \boldsymbol{\mu}_{p_i})^\top \boldsymbol{\Omega}_{a_i} (\boldsymbol{x}_n - \boldsymbol{\mu}_{p_i}), \tag{19}$$

which concludes our derivation.

VI. Experimental Evaluation

The main focus of this paper is a sparsification scheme for marginalization in sliding window graphs in the context of automated driving. Our experiments are designed to show the capabilities of our method and to support our key claims, that our approach

(i) approximates sliding window marginalization with sparse global priors,
(ii) utilizes global linearization points and thus does not require local optimization,
(iii) computes conservative landmark positions for incremental map refinement,
(iv) has exactly the same sparsity pattern compared to using no marginalization, but provides more accurate estimates.

Our evaluation is based on simulated data, which allows us to use error-free ground truth data as reference, and real world data collected with a prototype vehicle. The datasets are illustrated in Fig. 3 and Fig. 4. As landmarks we use pole-like objects similar to Brenner [28]. We randomly delete in both datasets 20% of the map landmarks, which are then re-estimated with our approach. These re-estimations are the basis for our evaluation.

A. Sparse Global Priors

The first experiment is designed to show that we are able to approximate marginalization with our *sparse global prior distribution*. Therefore we calculate the Euclidean distance between the optimized landmark positions, which are added to the map, and the corresponding full graph solution based on simulated data. We compare our sparse approach (18) to dense marginalization as in (12), and no marginalization at all. We use the latter as a baseline and normalize all errors with its mean. Tab. I highlights the benefits of dense marginalization with a local linearization point, as described in [25], with 53.4% of the errors compared

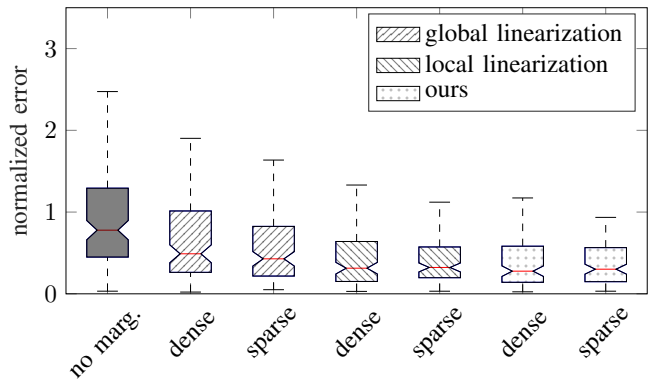


Fig. 5: Normalized Euclidean distances for landmarks compared to the full graph solution based on our simulated dataset.

to no marginalization. Its approximation with our sparse global prior approach performs with 48.2% even slightly better. The effect that the sparse variant performs better than the dense one consistently occurs in all experiments shown in Tab. I regardless of the chosen linearization point. Besides numerical errors, we explain this positive effect by linearization errors, which for our sliding window setting have a stronger impact in the dense case. Our approach is explicitly designed to neglect dependencies between states, whereas for the dense case inaccurate dependencies induced by marginalization accumulate. The comparable results show that we are successfully able to approximate marginalization with sparse global priors.

B. Global vs. Local Linearization vs. Our Approach

Our second experiment supports the claim that our approach utilizes global linearization points and does not

	global linearization	local linearization	ours
dense	73.5 %	53.4 %	46.7 %
sparse	61.1 %	50.9 %	48.2 %

TABLE I: Normalized average errors expressed as percentages. The baseline is the average error without marginalization (100%), such that lower is better in this table. The results are based on our simulated dataset.

require local optimization of the marginalization blanket. We compare the use of global and local linearization points for $\check{\boldsymbol{x}}_m$ and $\check{\boldsymbol{x}}_n$ to our method of utilizing global linearization points and including a gradient term as shown in (17). The comparison for the dense and sparse case is shown in Fig. 5. It can be seen that using naive global linearization points still performs better than no marginalization, but yields the worst performance of all marginalization approaches. Considering the sparse case, the figure shows that our approach performs even slightly better than local linearization. We contribute this effect to our sparsification scheme, in which we first approximate marginalization with a sparse distribution and afterwards include a gradient term in our distribution and not the other way around. By doing so we avoid the effect of inaccurate dependencies induced by marginalization between states on the gradient. Additionally, our experiment suggests that even in the dense case it is beneficial to use the global linearization point and consider the gradient term, as we suggest in Sec. V-B, rather than using the local linearization point. Although this might not generalize to arbitrary graphs, it is beneficial for our sliding window case. Overall, the comparison shows that our approach provides comparable results to local linearization and successfully utilizes global linearization points.

C. Conservative Estimates

The third experiment is to support our claim that our approach estimates conservative landmark positions. Based on our simulated world dataset we calculate the Mahalanobis distance between the estimated landmark with its covariance and the artificially removed landmark position from the map. Fig. 6 shows the result for the different marginalization and sparsification variants. If the given percentile levels from a specific experiment are above the given percentile boundaries the covariances are overconfident (too small) and vice versa if the levels are below the boundaries the variances are underconfident (too large). In our case it is favorable to underestimate the covariances in order to be more robust against outliers. The figure shows that simply using the global linearization point without considering the gradient term, as described in Sec. V-B, is suboptimal. Instead our approach provides underconfident estimates, which is what we desire.

D. Sparsity and Accuracy

Finally, we show that our approach has exactly the same sparsity pattern compared to no marginalization, but provides more accurate estimates. The results of this experiment are based on the Euclidean distance between the estimated landmark positions and the deleted ones from the map. Fig. 7 compares our approach to dense marginalization and shows that our system matrix $\boldsymbol{H}$ achieves a better sparsity pattern. In fact, it is by design exactly similar to using no marginalization. Together with Fig. 8, which shows that our approach achieves better accuracy than using no marginalization and is comparable to dense marginalization, the experiment supports our claim.

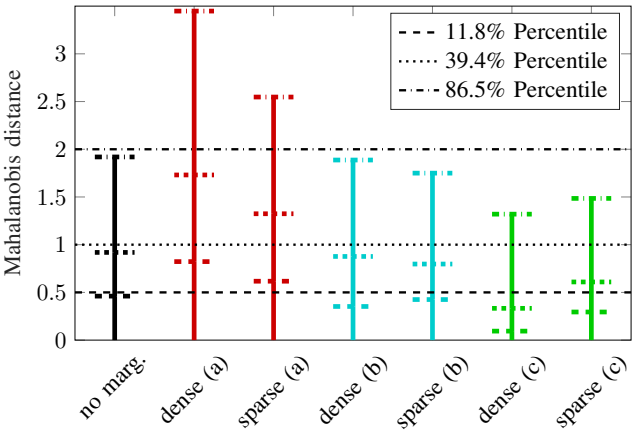


Fig. 6: Based on our simulated dataset the figure shows how the estimated covariances fit to the errors between estimated landmark positions and artificially removed landmarks of our map. The plot compares (a) global linearization, (b) local linearization, (c) our approach. Our approach provides the most conservative estimates.

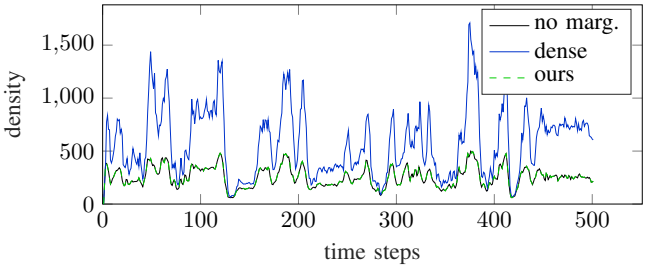


Fig. 7: Based on our real world dataset, the figure shows the sparsity patterns of dense marginalization, without marginalization, and our approach. The density value shown in the plot is the number of non zero block-diagonal entries in $\boldsymbol{H}$ for each time step. Our sparse approximation has the exact same sparsity pattern as without marginalization.

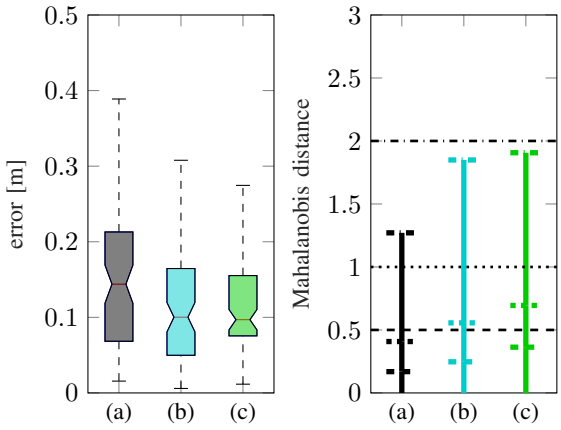


Fig. 8: Results for our real world data set. The plots compare (a) no marginalization, (b) dense marginalization with local linearization points, and (c) our method with sparse globals priors. The plots show that our approach provides similar performance to dense marginalization. Considering that our approach has a favorable sparsity pattern, it is clear that our approach is superior to standard dense marginalization.

In summary, our evaluation suggests that our method provides competitive results to dense sliding window marginalization. At the same time, our method retains the same sparsity pattern as without marginalization and is thus faster to compute. Hence, we supported all our claims with this experimental evaluation.

VII. Conclusion

In this paper, we presented a novel approach for approximating sliding window marginalization. It computes sparse global priors for each state involved in the marginalization and exploits the special structure of sliding window graphs to avoid rank deficiency problems. Our method first uses the Kullback-Leibler divergence to establish sparsity and then adjusts the global linearization points to overcome linearization errors. This allows us to optimally estimate landmark positions while not influencing the problems sparsity pattern and thus keeping computational costs tractable. The computed landmark positions are then utilized to extend a localization map used in autonomous vehicles. We evaluated our approach on simulated and real world data and provided comparisons to state-of-the-art sliding window marginalization. Our experiments showed that our sparsification scheme provides a favorable sparsity pattern and simultaneously achieves comparable accuracy of the resulting landmark estimates.

References

[1] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intell. Transp. Syst. Mag.*, pp. 31–43, 2010.

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

[3] M. Mazuran, W. Burgard, and G. D. Tipaldi, "Nonlinear factor recovery for long-term SLAM," *Int. J. Robot. Research*, vol. 35, no. 1-3, pp. 50–72, 2016.

[4] N. Carlevaris-Bianco, M. Kaess, and R. M. Eustice, "Generic node removal for factor-graph SLAM," *IEEE Trans. Robot.*, vol. 30, no. 6, pp. 1371–1385, 2014.

[5] H. Durrant-Whyte and T. Bailey, "Simultaneous Localization and Mapping: Part I," *IEEE Robot. & Automation Mag.*, vol. 13, no. 2, pp. 99–110, 2006.

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

[7] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Trans. Intell. Vehicles*, 2017.

[8] C. Stachniss, J. J. Leonard, and S. Thrun, *Springer Handbook of Robotics*. Springer, 2016, ch. Simultaneous Localization and Mapping, pp. 1153–1175.

[12] ——, "Exploiting building information from publicly available maps in graph-based SLAM," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Syst. (IROS)*, 2016, pp. 4511–4516.

[9] H. Roh, J. Jeong, Y. Cho, and A. Kim, "Accurate mobile urban mapping via digital map-based SLAM," *Sensors*, vol. 16, no. 8, p. 1315, 2016.

[10] R. Kümmerle, B. Steder, C. Dornhege, A. Kleiner, G. Grisetti, and W. Burgard, "Large scale graph-based SLAM using aerial images as prior information," *Auton. Robots*, vol. 30, no. 1, pp. 25–39, 2011.

[11] O. Vysotska and C. Stachniss, "Improving SLAM by exploiting building information from publicly available maps and localization priors," *J. of Photogrammetry, Remote Sensing and Geoinformation Sci.*, vol. 85, no. 1, pp. 53–65, 2017.

[13] D. Wilbers, C. Merfels, and C. Stachniss, "A comparision of particle filter and graph-based optimization for localization with landmarks in automated vehicles," in *Proc. IEEE Int. Conf. Robotic Computing (IRC)*, 2019.

[14] C. Merfels and C. Stachniss, "Pose fusion with chain pose graphs for automated driving," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Syst. (IROS)*, 2016, pp. 3116–3123.

[15] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, 1st ed. MIT Press, 2005.

[16] J. Vial, H. Durrant-Whyte, and T. Bailey, "Conservative sparsification for efficient and consistent approximate estimation," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Syst. (IROS)*, 2011, pp. 886–893.

[17] H. Kretzschmar, C. Stachniss, and G. Grisetti, "Efficient information-theoretic graph pruning for graph-based SLAM with laser range finders," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Syst. (IROS)*, 2011, pp. 865–871.

[18] N. Carlevaris-Bianco and R. M. Eustice, "Generic factor-based node marginalization and edge sparsification for pose-graph SLAM," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2013, pp. 5748–5755.

[19] ——, "Conservative edge sparsification for graph SLAM node removal," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2014, pp. 854–860.

[20] J. Vallvé, J. Solà, and Andrade-Cette, "Factor descent optimization for sparsification in graph SLAM," in *Proc. European Conf. of Mobile Robots (ECMR)*, 2017.

[21] ——, "Graph SLAM sparsification with populated topologies using factor descent optimization," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, 2018.

[22] S. Choudhary, V. Indelman, H. I. Christensen, and F. Dellaert, "Information-based reduced landmark SLAM," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2015, pp. 4620–4627.

[23] D.-N. Ta, N. Banerjee, S. Eick, S. Lenser, and M. E. Munich, "Fast nonlinear approximation of pose graph node marginalization," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2018.

[24] J. Hsiung, M. Hsiao, E. Westman, R. Valencia, and M. Kaess, "Information sparsification in visual-inertial odometry," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Syst. (IROS)*, 2018.

[25] K. Eckenhoff, L. Paull, and G. Huang, "Decoupled, consistent node removal and edge sparsification for graph-based SLAM," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Syst. (IROS)*. IEEE, 2016, pp. 3275–3282.

[26] M. Mazuran, G. D. Tipaldi, L. Spinello, and W. Burgard, "Nonlinear graph sparsification for SLAM," in *Proc. Robotics: Science and Syst. Conf. (RSS)*, 2014, pp. 1–8.

[27] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2011, pp. 3607–3613.

[28] C. Brenner, "Global localization of vehicles using local pole patterns," in *Lecture Notes in Computer Science*. Springer, Berlin Heidelberg, 2009, vol. 5748.