Consistent Sparsification for Graph Optimization

Guoquan Huang, Michael Kaess, and John J. Leonard

***Abstract*— In a standard pose-graph formulation of simultaneous localization and mapping (SLAM), due to the continuously increasing numbers of nodes (states) and edges (measurements), the graph may grow prohibitively too large for long-term navigation. This motivates us to systematically reduce the pose graph amenable to available processing and memory resources. In particular, in this paper we introduce a *consistent* graph sparsification scheme: i) sparsifying nodes via marginalization of old nodes, while retaining all the information (consistent relative constraints) – which is conveyed in the discarded measurements – about the remaining nodes after marginalization; and ii) sparsifying edges by formulating and solving a consistent ℓ_1-regularized minimization problem, which automatically promotes the sparsity of the graph. The proposed approach is validated on both synthetic and real data.**

I. Introduction

Recently, most popular solutions to the simultaneous localization and mapping (SLAM) problem, either in batch or incremental fashion, are based on graph optimization (i.e., all robot poses and/or landmark positions comprise the nodes of the graph, while each edge encodes a measurement constraint) [1]–[4]. However, a standard graph formulation may suffer from unbounded complexity of both processing and memory, which can grow continuously over time. This is due to the fact that new robot poses (and new landmarks in the case of feature-based SLAM) are constantly being added into the graph, resulting in the number of nodes increasing constantly in time; and moreover, if frequent loop-closing events occur in SLAM, loop-closure constraints (edges) can significantly increase the density of the graph. This, for example, can be the case where a service robot is operating inside an office building for an extended period of time.

Even though the issue of reducing the complexity of graph optimization, in particular, for SLAM, has recently been addressed [5]–[9], to the best of our knowledge, little work has yet explicitly taken into account estimation consistency (i.e., unbiased estimates, and the estimated covariance larger than or equal to the true covariance [10]) in the design of graph reduction (sparsification) scheme. This is a significant limitation, since if an estimator is inconsistent, then the accuracy of the computed state estimates is unknown, which in turn makes the estimator unreliable.

To address the aforementioned issue, in this paper we study in-depth *consistent* graph sparsification for SLAM, and explicitly enforce consistency during sparsifying *both* nodes and edges of the graph. In particular, the main contributions of this work are the following:

- We sparsify the nodes of the graph by marginalization of old, matured (accurately estimated) nodes when loop closure occurs, and derive *consistent* relative constraints for (a subset of) the remaining nodes, which encapsulate *all* the information – conveyed in the discarded measurements at the time of marginalization – about the remaining nodes. As a result, the problem size (i.e., the number of optimization variables) is bounded *only* by the explored area, and independent of the exploration time. Although marginalization (Schur complement) is commonly used to reduce the size of the state vector in batch estimation [11], when applied to graph-based incremental estimation, to the best of our knowledge, little work has been shown to optimally (and thus consistently) extract all the information about the remaining nodes from the discarded measurements and then propagate it to subsequent optimization. In contrast, we explicitly derive from the discarded measurements the consistent (i.e., without reuse or loss of information) relative constraints for the remaining nodes that are also involved in the discarded measurements. These induced constraints, in general, are *correlated*, instead of independent as assumed in most prior work, and are used as new measurements in subsequent estimation.
- Furthermore, we sparsify the edges of the graph by formulating and solving a *consistent* ℓ_1-regularized minimization problem. In particular, due to marginalization of old nodes as well as frequent loop closures, it may become necessary to systematically remove some edges so as to reduce the processing and memory requirements. Towards this end, we formulate the sparsification of graph edges as a novel consistency-constrained ℓ_1-regularized minimization problem, which implicitly imposes the sparsity of the solution due to ℓ_1 norm. As a result, its solution renders the sparsified graph which consistently approximates the original dense graph (i.e., by definition, the covariance of the resulting sparsified graph is larger than or equal to that of the original one). It is interesting to point out that, while ℓ_1-minimization has been extensively studied in compressive sensing [12], the application to graph sparsification treated in this paper is expected to promote its widespread adoption in robotics.

II. Related Work

The SLAM problem has received considerable attention over the past two decades and many different estimators were

This work was partially supported by ONR grants N00014-12-1-0093, N00014-10-1-0936, N00014-11-1-0688, and N00014-12-10020. The authors are with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, USA. Email: {gqhuang|kaess|jleonard}@mit.edu

employed for solving it. In particular, filtering methods such as the extended Kalman filter (EKF) recursively estimate a state vector consisting of the current robot pose and the observed landmarks [13]–[16]. One appealing property of such a filtering algorithm is its bounded runtime with respect to the size of the explored environment, while the quadratic computational complexity limits its applications to large areas. Moreover, due to the fact that any linearization-based filter marginalizes out the previous robot poses, it cannot relinearize the nonlinear system and measurement models at the past states, which may result in large linearization errors and degrade the performance.

A. Graph optimization

Graph-based batch optimization methods for SLAM recently have prevailed (e.g., [1], [3], [4], [17]–[23]). These methods follow the paradigm of bundle-adjustment (BA) algorithms originally developed in photogrammetry and computer vision [11], and formulate and solve a batch least-squares problem for the entire robot trajectory (and all landmarks), with no marginalization. These BA-based approaches exploit the sparsity of the measurement graph so as to speed up computation. However, for large-scale problems, an optimal batch solution may still be too computationally expensive to obtain in real time [24].

Therefore, different approximate methods for graph optimization have been developed, which either use a subset of the data to optimize over only a subset of variables, or solve the BA problem intermittently. Specifically, sliding-window filters (SWFs) [25], [26] compute a solution for a constant-size, sliding window of states (robot poses and landmark positions) using only the measurements corresponding to that time interval. Similarly, keyframe-based approaches [27]–[29] perform batch optimization over only a (heuristically) selected subset of views or keyframes. On the other hand, incremental approaches to BA, such as the incremental smoothing and mapping (iSAM) algorithm [2], greatly reduce computation by employing factorization-updating methods that allow reusing the information-matrix factorization available from previous time steps. Computationally demanding procedures, such as relinearization and batch factorization, are only performed periodically. Nevertheless, the incremental methods still suffer from increased computational cost. For instance, due to the increased robot trajectory over time as well as the accumulation of fill-in between periodic batch steps, iSAM's efficiency degrades with the increasingly dense graph (e.g., if the number of constraints is more than five times the number of poses as reported in [24]). It becomes necessary to sparsify both the nodes and the edges of the graph in order to fully take advantage of the efficiency offered by the incremental estimation algorithms in large environments.

B. Graph reduction

The keyframe-based approximate approaches for graph optimization [28], [29] are among the first attempt to reduce the pose graph for visual SLAM. As mentioned earlier, these approaches create a sparse skeleton graph of views (poses) and then perform batch optimization only on these keyframes. In particular, in order to keep the density of poses constant in a given region, old poses are marginalized out from the skeleton graph. However, it is unclear that if applying these methods to an incremental setting, how the information (which is conveyed in the discarded measurements after marginalization) about the remaining poses is utilized in future optimization. In this work, we explicitly derive the *consistent* relative constraints for (a subset of) the remaining poses that are involved in the discarded measurements. These induced constraints describe *all* the information for the remaining states available in the discarded measurements, and are used as new constraints in future estimation.

Similarly, in [5] marginalization is employed to reduce graph complexity. However, since it composes the relative-pose constraints for the remaining states directly from the discarded measurements by geometry, during this process, the same measurement information may be used multiple times, which results in *inconsistent* constraints. In addition, it heuristically removes edges based on the degrees of their nodes; when the degree of a node exceeds an ad-hoc, pre-chosen threshold, the associated edge with the least residual error is pruned out. We also notice that the recent work of [8] bounds the size of the pose graph with respect to the explored area, instead of the exploration time. To this end, this approach does not add new nodes at loop closures (without marginalization), while retaining new relative-pose constraints between the existing nodes involved in the loop closures, which are inferred directly from the discarded measurements in a similar way as in [5].

Recently, compact pose SLAM [6] uses an information-theoretic method to determine which poses should be added into the state vector of the information filter [30] and which measurements should be utilized in the estimation. In particular, only nonredundant poses are included in the graph if no other poses are nearby, and only highly informative loop-closure edges (computed based on the mutual information gain) are added into the graph, while other poses and measurements are simply discarded. This clearly results in loss of information. Interestingly, the proposed sparsification scheme presented in Section IV can combine the node reduction idea of [6] to determine which nodes to marginalize out, while retaining all the information about the remaining nodes from the discarded measurements.

More recently, an information-theoretic approach is proposed to compress the pose graph, *only* for laser-based SLAM [7]. Specifically, this method selects only the most informative laser scans (nodes) with respect to the built map of the environment; and moreover, it employs the *approximate* marginalization based on Chow-Liu tree to keep the sparsity of the pose graph. However, it remains unclear that, during this approximate marginalization, how the information conveyed in the discarded laser scans about the remaining nodes, is extracted and utilized in an incremental framework. Moreover, the Chow-Liu tree approximation employed in the marginalization in theory does not guarantee consistency. In contrast, both issues are explicitly addressed in our proposed approach (see Section IV).

We also notice that the most recent publication [9] addresses the same problem of pose-graph sparsification. In [9], a marginalization process similar to Section IV-A is used to sparsify the nodes; and a better Chow-Liu tree-based approximation than that of [7] is proposed to sparsify the edges. However, the Chow-Liu tree-based sparsification yet does not guarantee consistent estimates, which is addressed in this paper by formulating and solving a consistent ℓ_1-regularized sparsification problem.

On the other hand, conservative (consistent) sparsification of information matrix is introduced in [31] for sparse extended information filter (SEIF)-based SLAM [32], [33]. While this approach is closely related to the proposed sparsification of graph edges (see Section IV-B), we formulate such consistent sparsification as an ℓ_1-regularized minimization problem which implicitly promotes the sparsity by ℓ_1 norm and whose solution automatically renders the sparsified information matrix (graph). As a result, the proposed approach does not need to know which edges to eliminate beforehand, which, however, is assumed in [31].

III. Problem Formulation

While many problems in robotics and computer vision can be formulated as graph (network) optimization, in this section we focus on SLAM to illustrate the graph-based formulation. In particular, graph-based SLAM [3] consists of a front-end and a back-end. The former aims to extract all the relative-pose constraints from the raw sensor measurements (e.g., [34]), while the latter is to compute the most likely configuration of the poses, which is the focus of this paper.

Specifically, in the graph built during the front-end, the robot poses are described by the nodes of the graph, and the edges represent the spatial constraints between the connecting nodes, which are constructed either from proprioceptive (e.g., odometry) or exteroceptive measurements (e.g., images). Let $\mathbf{x} = \begin{bmatrix}\mathbf{x}_1^T, \dots, \mathbf{x}_n^T\end{bmatrix}^T$ be the robot poses (nodes), and $\mathbf{z}_{ij} = \mathbf{h}(\mathbf{x}_i, \mathbf{x}_j) + \mathbf{n}_{ij}$ be the measurement constraint (edge) between the nodes i and j, where noise is commonly assumed to be zero-mean white Gaussian, i.e., $\mathbf{n}_{ij} \sim \mathcal{N}(\mathbf{0}, \mathbf{W}_{ij})$. The objective of the back-end is to compute the maximum likelihood estimate (MLE) of all the robot poses (nodes) using all the measurement constraints (edges). By assuming independent measurements, we have the following factorization of the measurement likelihood function:

$$p(\mathcal{Z}|\mathbf{x}) = \prod_{\mathbf{z}_{ij} \in \mathcal{Z}} p(\mathbf{z}_{ij}|\mathbf{x}_i, \mathbf{x}_j) \tag{1}$$

where $\mathcal{Z} = \{\mathbf{z}_{ij}\}$ denotes all the measurements (edges). By using the Gaussian-noise assumption and taking the negative logarithm of (1), we can easily show that maximizing the likelihood (1) is equivalent to minimizing the following least-squares cost function:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \; c(\mathbf{x}) := \arg\min_{\mathbf{x}} \sum_{\mathbf{z}_{ij} \in \mathcal{Z}} ||\mathbf{z}_{ij} - \mathbf{h}(\mathbf{x}_i, \mathbf{x}_j)||^2_{\mathbf{W}_{ij}} \tag{2}$$

where we have used the notation $||\mathbf{r}||^2_{\mathbf{W}} = \mathbf{r}^T \mathbf{W}^{-1} \mathbf{r}$, i.e., the squared Mahalanobis distance of residual $\mathbf{r}$ with covariance $\mathbf{W}$. This is a *nonlinear* least-squares problem because of the nonlinearity of the constraints $\mathbf{h}(\cdot)$.

A standard iterative Gauss-Newton approach is often used for solving (2). In particular, at the k-th Gauss-Newton iteration, a correction, $\delta\mathbf{x}^{(k)}$, to the current estimate, $\hat{\mathbf{x}}^{(k)}$, is computed by minimizing the second-order Taylor-series approximation of (2), which can be written as:

$$\begin{aligned} c(\hat{\mathbf{x}}^{(k)} + \delta\mathbf{x}) &\simeq \sum_{\mathbf{z}_{ij} \in \mathcal{Z}} ||\mathbf{z}_{ij} - \mathbf{h}\left(\hat{\mathbf{x}}_i^{(k)}, \hat{\mathbf{x}}_j^{(k)}\right) - \mathbf{H}_{ij}^{(k)} \delta\mathbf{x}||^2_{\mathbf{W}_{ij}} \\ &=: ||\mathbf{J}^{(k)} \delta\mathbf{x} - \mathbf{r}^{(k)}||^2 \end{aligned} \tag{3}$$

where we linearize the measurement constraints at the current state estimate with Jacobians, $\mathbf{H}_{ij}^{(k)} = \frac{\partial \mathbf{h}(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}}\Big|_{\mathbf{x} = \hat{\mathbf{x}}^{(k)}}$. Note that hereafter we drop the iteration index (k) for simplicity of notation. We now have a *linear* least-squares problem with respect to $\delta\mathbf{x}$ (3), where $\mathbf{J}$ is the full Jacobian matrix obtained by stacking and weighting all the measurement Jacobians, and $\mathbf{r}$ is the corresponding stacked residual vector:

$$\mathbf{J} = \begin{bmatrix} \mathbf{W}_{12}^{-\frac{1}{2}} \mathbf{H}_{12} \\ \vdots \\ \mathbf{W}_{ij}^{-\frac{1}{2}} \mathbf{H}_{ij} \\ \vdots \end{bmatrix} \qquad \mathbf{r} = \begin{bmatrix} \mathbf{W}_{12}^{-\frac{1}{2}} \left(\mathbf{z}_{12} - \mathbf{h}(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2)\right) \\ \vdots \\ \mathbf{W}_{ij}^{-\frac{1}{2}} \left(\mathbf{z}_{ij} - \mathbf{h}(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j)\right) \\ \vdots \end{bmatrix} \tag{4}$$

We employ QR factorization to solve (3), i.e.,

$$\begin{aligned} &\min_{\delta\mathbf{x}} \; ||\mathbf{J}\delta\mathbf{x} - \mathbf{r}||^2 = \left|\left| \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \delta\mathbf{x} - \mathbf{r} \right|\right|^2 = \\ &\qquad \left|\left| \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \delta\mathbf{x} - \mathbf{Q}^T \mathbf{r} \right|\right|^2 =: \left|\left| \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \delta\mathbf{x} - \begin{bmatrix} \mathbf{d} \\ \mathbf{e} \end{bmatrix} \right|\right|^2 \\ &\Leftrightarrow \min_{\delta\mathbf{x}} \; ||\mathbf{R}\delta\mathbf{x} - \mathbf{d}||^2 \end{aligned} \tag{5}$$

where we have used the reduced QR of $\mathbf{J}$ [35], since it in general is tall, i.e.,

$$\mathbf{J} = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} = \mathbf{Q}_1 \mathbf{R} \tag{6}$$

Once $\delta\mathbf{x}$ is found by back substitution (5), the new state estimate is updated as: $\hat{\mathbf{x}}^{(k+1)} = \hat{\mathbf{x}}^{(k)} + \delta\mathbf{x}^{(k)}$.

A. Incremental smoothing and mapping

When a new measurement, $\mathbf{z}_{mn}$, sequentially becomes available, ideally we need to recompute the full Jacobian $\mathbf{J}$ and solve the batch problem from scratch. However, this is an expensive operator, even by exploiting the sparsity of the Hessian matrix [1], [4], [20]. To save computations, we focus on incremental smoothing and mapping (iSAM) [2], which reuses the previously-computed Jacobian and *incrementally* updates the QR factorization directly.

In particular, we augment $\mathbf{J}$ (without recomputing it) with the new measurement Jacobian $\mathbf{H}_{mn}$ [see (6)]:

$$\mathbf{J}_a := \begin{bmatrix} \mathbf{J} \\ \mathbf{W}_{mn}^{-\frac{1}{2}} \mathbf{H}_{mn} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{W}_{mn}^{-\frac{1}{2}} \mathbf{H}_{mn} \end{bmatrix} \tag{7}$$

We now aim to decompose $\mathbf{J}_a$ into triangular form (i.e., square-root information matrix). Since $\mathbf{J}$ was already factorized into the triangular $\mathbf{R}$, we only need to zero out the new block row, i.e., the new measurement Jacobian $\mathbf{H}_{mn}$, in order to obtain the updated square-root information matrix $\mathbf{R}_a$. This can be achieved efficiently, for example, by using Givens QR [35]. Similarly, the corresponding new vector, $\mathbf{d}_a$, can be obtained by applying the same Givens rotations to the augmented residual vector, $\mathbf{d}_a := \begin{bmatrix} \mathbf{d} \\ \mathbf{W}_{mn}^{-\frac{1}{2}}\left(\mathbf{z}_{mn} - \mathbf{h}(\hat{\mathbf{x}}_m, \hat{\mathbf{x}}_n)\right) \end{bmatrix}$.

It is important to note that, although relinearization is not needed at each time step when a new measurement becomes available, in order to reduce the linearization errors, we relinearize the system at the latest, and thus the best, state estimates periodically [2] or as needed when the linearization point significantly deviates from the current state estimate [24]. In addition, we can combine variable reordering [2] with this batch factorization to reduce fill-in of the resulting triangular system [see (5)], which can further speed up the subsequent incremental estimation.

IV. Consistent Graph Sparsification

Even though different algorithms, such as the iSAM presented in the preceding section (also see [2]), are available to improve the efficiency of batch least-squares solutions, their performance degrades with the growing size of the graph as the robot continuously navigates in its environments. In order to make the processing and memory requirements amenable for long-term operation, in this section we introduce consistent sparsification of both the nodes and edges of the graph within the incremental estimation framework.

A. Consistent node sparsification via marginalization

First of all, we sparsify the nodes of the graph by marginalizing out the selected old poses, while consistently retaining all the information that discarded measurements convey about the remaining nodes, so as to bound the size of the graph (state vector) only with respect to the explored area, instead of the exploration time. Note that this temporally-scalable idea has been explored in [8], which, however, follows an approximate composition of discarded measurements. In particular, we partition the nodes (states), $\mathbf{x} = \begin{bmatrix} \mathbf{x_M} \\ \mathbf{x_R} \end{bmatrix}$, where $\mathbf{x_M}$ denote the nodes to marginalize out, and $\mathbf{x_R}$ are those remaining in the graph. In this work we choose $\mathbf{x_M}$ to be the old poses when loop closure occurs. This, however, is not a necessary assumption; instead, one can selectively discard the most matured (accurately estimated) or the least informative nodes as compared to the rest of the graph [6], [7]. We should repeat that in this work we focus more on how to optimally (and thus consistently) retrieve all the information from the discarded measurements about the remaining nodes, instead of on studying which nodes are the best to marginalize.

The marginalization process is carried out with Schur complement, and yields the following reduced-Hessian matrix for the remaining state $\mathbf{x_R}$:

$$\mathbf{A}_r = \mathbf{A_{RR}} - \mathbf{A_{RM}}\mathbf{A}_{\mathbf{MM}}^{-1}\mathbf{A_{MR}} \tag{8}$$

where the matrices appearing in the above equation are defined as partitions of the full-Hessian matrix [see (3) and (4)]:

$$\mathbf{A} = \mathbf{J}^T\mathbf{J} = \sum_{\mathbf{z}_{ij}\in\mathcal{Z}} \mathbf{H}_{ij}^T\mathbf{W}_{ij}^{-1}\mathbf{H}_{ij} =: \begin{bmatrix} \mathbf{A_{MM}} & \mathbf{A_{MR}} \\ \mathbf{A_{RM}} & \mathbf{A_{RR}} \end{bmatrix} \tag{9}$$

Note that the estimates of the remaining nodes, $\hat{\mathbf{x}}_\mathbf{R}$, remain the same before and after this marginalization process [30], due to the fact that the marginal distribution of a joint Gaussian distribution is also Gaussian. After marginalization, all the states in $\mathbf{x_M}$, as well as all the measurements that involve these states, denoted by $\mathcal{Z}_\mathbf{M}$, are discarded, and no longer participate in the future optimization.

It is important to note that, the discarded measurements $\mathcal{Z}_\mathbf{M}$ typically involve, and thus convey information about, a subset of the remaining states $\mathbf{x_R}$. In order for consistent (i.e., without loss or reuse of information) estimation in an incremental fashion, we need to retain all such information and propagate it to subsequent optimization. Therefore, we now focus on deriving such consistent constraints from the discarded measurements, which will be used whenever the batch relinearization occurs in the incremental estimation.

1) Derive consistent relative constraints from $\mathcal{Z}_\mathbf{M}$: We further partition the remaining states into two subsets, $\mathbf{x_{R_1}}$ which together with $\mathbf{x_M}$ are involved in the discarded measurements and $\mathbf{x_{R_2}}$ which are not involved, i.e., $\mathbf{x_R} = \begin{bmatrix} \mathbf{x_{R_1}} \\ \mathbf{x_{R_2}} \end{bmatrix}$. In order to optimally retrieve all the information conveyed in the discarded measurements about the subset of the remaining states, ideally we should solve the MLE (least-squares) problem of minimizing $c_m(\mathbf{x_M}, \mathbf{x_{R_1}})$, for $\mathbf{x_{R_1}}$ [see (2)]. However, in most cases, $\mathcal{Z}_\mathbf{M}$ contains only *relative* pose-to-pose measurement constraints, and does not include any *global* information about the involved states. This implies that the corresponding system is *unobservable* and we cannot find the optimal estimates of $\mathbf{x_M}$ and $\mathbf{x_{R_1}}$ in the global frame of reference without ambiguity [10]. Instead, we can *locally* find MLE for the remaining nodes $\mathbf{x_{R_1}}$, by transforming the involved states into a local frame of reference of one remaining state (e.g., of the first node in $\mathbf{x_{R_1}}$). The computed MLE of the relative-state estimates essentially are the relative-pose constraints that encapsulate all the information contained in the discarded measurements for the remaining states, and will be used for computing the Jacobians and Hessians in the future optimization.

Specifically, we now solve the following least-squares (MLE) problem using *only* the discarded measurements $\mathcal{Z}_\mathbf{M}$, with respect to the *relative* states, ${}^L\mathbf{x_M}$ and ${}^L\bar{\mathbf{x}}_{\mathbf{R_1}}$, [see (2)]. Here the symbol " $^-$ " denotes that the remaining nodes exclude the node used to define the local frame $\{L\}$.

$$\min_{{}^L\mathbf{x_M}, {}^L\bar{\mathbf{x}}_{\mathbf{R_1}}} c_m({}^L\mathbf{x_M}, {}^L\bar{\mathbf{x}}_{\mathbf{R_1}}) := \sum_{\mathbf{z}_{ij}\in\mathcal{Z}_\mathbf{M}} ||\mathbf{z}_{ij} - \mathbf{h}({}^L\mathbf{x}_i, {}^L\mathbf{x}_j)||^2_{\mathbf{W}_{ij}} \tag{10}$$

Similarly, a Gauss-Newton approach is employed to solve the above problem. The information (Hessian) matrix of the estimates of the relative remaining states (i.e., the relative-pose constraints), is again obtained using Schur complement by marginalizing out ${}^L\mathbf{x}_\mathbf{M}$ [see (9) and (8)]:

$$\bar{\mathbf{A}}_m = \sum_{\mathbf{z}_{ij}\in\mathcal{Z}_\mathbf{M}} \bar{\mathbf{H}}_{ij}^T \mathbf{W}_{ij}^{-1} \bar{\mathbf{H}}_{ij} = \begin{bmatrix} \bar{\mathbf{A}}_{\mathbf{MM}} & \bar{\mathbf{A}}_{\mathbf{MR_1}} \\ \bar{\mathbf{A}}_{\mathbf{R_1M}} & \bar{\mathbf{A}}_{\mathbf{R_1R_1}} \end{bmatrix} \Rightarrow \quad (11)$$

$$\bar{\mathbf{A}}_p = \bar{\mathbf{A}}_{\mathbf{R_1R_1}} - \bar{\mathbf{A}}_{\mathbf{R_1M}} \bar{\mathbf{A}}_{\mathbf{MM}}^{-1} \bar{\mathbf{A}}_{\mathbf{MR_1}} \quad (12)$$

where the symbol "$^{-}$" denotes that all the involved Jacobians and Hessians are computed with respect to the *relative* states, ${}^L\mathbf{x}_\mathbf{M}$ and ${}^L\bar{\mathbf{x}}_{\mathbf{R_1}}$, which are different from (8) and (9).

Close inspection reveals that the information matrix $\bar{\mathbf{A}}_m$ encapsulates all the information contained in the discarded measurements $\mathcal{Z}_\mathbf{M}$; and thus $\bar{\mathbf{A}}_p$, the Schur complement of $\bar{\mathbf{A}}_{\mathbf{MM}}$ in $\bar{\mathbf{A}}_m$, describes *all* the information that the discarded measurements provide us about the relative remaining states, ${}^L\bar{\mathbf{x}}_{\mathbf{R_1}}$. Therefore, the relative constraints for the remaining states induced from the discarded measurements, ${}^L\bar{\mathbf{x}}_{\mathbf{R_1}} \sim \mathcal{N}\left({}^L\hat{\bar{\mathbf{x}}}_{\mathbf{R_1}}, \bar{\mathbf{A}}_p^{-1}\right)$, are consistent (i.e., without loss/reuse of information). Note that these essentially are the relative-frame generic linear constraints (GLC) as derived in [9]. Note also that, due to marginalization (12), in general, the information matrix $\bar{\mathbf{A}}_p$, and thus the corresponding covariance matrix $\bar{\mathbf{A}}_p^{-1}$, is full. This indicates that the induced relative constraints for the remaining states are *correlated*, instead of independent.

At this point, we stress that although marginalization of old nodes is often used to reduce the pose graph (e.g., [5], [7]), prior to [9] that was developed in parallel to this work, to the best of our knowledge, little work has yet explicitly derived optimal (consistent) relative constraints for the remaining nodes from the discarded edges, for the use in future estimation. In particular, in [5] the authors compose a new *independent* edge (relative constraint) – induced by marginalization – directly from the connected discarded edges by geometry. However, since same discarded edges may be used multiple times in composing new edges, these new edges should be *correlated*. As a result of this reuse of information of same measurements, inconsistent relative constraints for the remaining nodes are inferred, which will negatively impact subsequent estimation. In contrast, in the proposed approach we infer the optimal relative constraints by solving a batch MLE (least-squares) problem followed by marginalization, and hence capture all the correlations between the induced constraints.

B. Consistent edge sparsification via ℓ_1-minimization

Due to marginalization as well as loop closures, the information (Hessian) matrix, and thus the graph, inevitably becomes dense, which can significantly increase the computational burden for the graph optimizer (or batch least-squares solver). To address this issue, we now aim to sparsify the edges of the graph in a *consistent* manner – the covariance of the sparsified graph is larger than or equal to that of the original one.

Mathematically, suppose the original dense graph representing a Gaussian distribution, $\mathcal{N}(\hat{\mathbf{x}}, \mathbf{\Sigma})$. We aim to find a sparse graph to consistently approximate the original one, with the mean unchanged, i.e., the corresponding Gaussian distribution, $\mathcal{N}(\hat{\mathbf{x}}, \mathbf{X}^{-1})$, where $\mathbf{X}$ is the resulting sparse information matrix. To this end, as in [31] we choose to minimize the Kullback-Leibler Divergence (KLD) between the two Gaussians, while enforcing the consistency constraint on the covariance matrix, i.e.,

$$\min_{\mathbf{X}} \quad \mathrm{D_{KL}}\left(\mathcal{N}(\hat{\mathbf{x}}, \mathbf{X}^{-1}) \middle|\middle| \mathcal{N}(\hat{\mathbf{x}}, \mathbf{\Sigma})\right) = \quad (13)$$

$$-\frac{1}{2}\left[\ln\left(\frac{\det \mathbf{X}^{-1}}{\det \mathbf{\Sigma}}\right) + \mathrm{tr}\left(\mathbf{X}\mathbf{\Sigma}\right) - \dim(\mathbf{x})\right]$$

$$\text{subject to} \quad \mathbf{X}^{-1} \succeq \mathbf{\Sigma} \quad (14)$$

By noting that the original covariance matrix $\mathbf{\Sigma}$ as well as the dimension of the state $\mathbf{x}$ is known, this is a *convex* problem, and hence the global minimum can be guaranteed [36]. In [31] an optimal solution is constructed by assuming that the edges of the graph to remove are known before sparsification, which clearly is not always the case in practice.

In contrast, we notice that ℓ_1-minimization, which is convex and ensures a sparse solution, has been extensively studied in compressive sensing [12]. By exploiting this idea, we reformulate the problem (13) and (14) as the following convex consistency-constrained ℓ_1-regularized minimization problem, whose solution automatically renders a consistent sparse information matrix:

$$\min_{\mathbf{X}} \quad -\ln\left(\det \mathbf{X}\right) + \mathrm{tr}\left(\mathbf{X}\mathbf{\Sigma}\right) + \lambda||\mathbf{X}||_1 \quad (15)$$

$$\text{subject to} \quad \mathbf{X} \preceq \mathbf{\Sigma}^{-1} \quad (16)$$

where $||\cdot||_1$ denotes the ℓ_1 norm, and λ is a design parameter to control the level of sparsity of the solution, which can be selected heuristically as [37]. Note that the equivalence between the constraints (14) and (16) is established based on Corollary 7.7.4 in [38]. While many different convex optimization algorithms [36], such as interior point methods, are available to solve the above problem (15) and (16), we adapt the alternating direction method of multipliers (ADMM) [39] to solve it more efficiently.

In particular, ADMM is often used to solve problems of the following form:

$$\min_{\mathbf{X}\in\mathcal{S}_{++}^n} \quad f(\mathbf{X}) + g(\mathbf{Y}) \qquad \text{subject to} \;\; \mathbf{X} - \mathbf{Y} = \mathbf{0} \quad (17)$$

where the optimization variable $\mathbf{X}$ is n-dimensional symmetric positive definite, i.e., $\mathbf{X} \in \mathcal{S}_{++}^n$. By defining $f(\mathbf{X}) := -\ln\left(\det \mathbf{X}\right) + \mathrm{tr}\left(\mathbf{X}\mathbf{\Sigma}\right)$, and $g(\mathbf{Y}) := \lambda||\mathbf{Y}||_1$, we have the same problem as in (15), but *without* consistency constraint (16). To solve the above ℓ_1-regularized problem (17), the ADMM algorithm iterates as follows [39]:

$$\mathbf{X}^{(k+1)} = \arg\min_{\mathbf{X}} \left(f(\mathbf{X}) + \frac{\rho}{2}||\mathbf{X} - \mathbf{Y}^{(k)} + \mathbf{U}^{(k)}||_F^2\right) \quad (18)$$

$$\mathbf{Y}^{(k+1)} = \arg\min_{\mathbf{Y}} \left(g(\mathbf{Y}) + \frac{\rho}{2}||\mathbf{X}^{(k+1)} - \mathbf{Y} + \mathbf{U}^{(k)}||_F^2\right) \quad (19)$$

$$\mathbf{U}^{(k+1)} = \mathbf{U}^{(k)} + \mathbf{X}^{(k+1)} - \mathbf{Y}^{(k+1)} \quad (20)$$

where $\rho > 0$ is a penalty parameter used in the augmented Lagrangian and $||\cdot||_F$ denotes the Frobenius norm.

In order to find the desired *consistent* sparse information matrix, $\mathbf{X}$, we now include the consistency constraint (16) into the ADMM step (18). In particular, the optimality condition of (18) is given by:

$$-\mathbf{X}^{-1}+\mathbf{\Sigma}+\rho\left(\mathbf{X}-\mathbf{Y}^{(k)}+\mathbf{U}^{(k)}\right)=\mathbf{0}\Rightarrow \tag{21}$$

$$\rho\mathbf{X}-\mathbf{X}^{-1}=-\mathbf{\Sigma}-\rho\left(-\mathbf{Y}^{(k)}+\mathbf{U}^{(k)}\right)=:\mathbf{V}\mathbf{D}\mathbf{V}^{T} \tag{22}$$

where we have employed the eigen-decomposition in the second equality with $\mathbf{D}=\mathrm{Diag}(d_i)$. By inspection, the solution has the following form, $\mathbf{X}^{(k+1)}:=\mathbf{V}\mathbf{\Gamma}\mathbf{V}^{T}$, where $\mathbf{\Gamma}=\mathrm{Diag}(\gamma_i)$, $\gamma_i>0$. To find $\mathbf{\Gamma}$ and thus $\mathbf{X}$, substitution of the above form of $\mathbf{X}^{(k+1)}$ into (22) yields:

$$\mathbf{V}\left(\rho\mathbf{\Gamma}-\mathbf{\Gamma}^{-1}\right)\mathbf{V}^{T}=\mathbf{V}\mathbf{D}\mathbf{V}^{T}\quad\Rightarrow \tag{23}$$

$$\rho\gamma_i-\frac{1}{\gamma_i}=d_i\ \Rightarrow\ \gamma_i=\frac{d_i+\sqrt{d_i^2+4\rho}}{2\rho} \tag{24}$$

In order to satisfy the consistency constraint (16), we need to ensure γ_i (24) is smaller than the minimum eigenvalue of $\mathbf{\Sigma}^{-1}$, denoted by $\lambda_{\min}(\mathbf{\Sigma}^{-1})$. Therefore, we compute γ_i as:

$$\gamma_i=\min\left(\frac{d_i+\sqrt{d_i^2+4\rho}}{2\rho},\lambda_{\min}(\mathbf{\Sigma}^{-1})\right) \tag{25}$$

On the other hand, (19) is a well-known lasso problem and its optimal solution is given by [40]:

$$\mathbf{Y}^{(k+1)}=\mathrm{shrink}\left(\mathbf{X}^{(k+1)}+\mathbf{U}^{(k)},\frac{\lambda}{\rho}\right) \tag{26}$$

where the shrinkage operator, $\mathrm{shrink}(\mathbf{\Xi},\xi)$, updates each element $\mathbf{\Xi}_{ij}$ as: $\mathrm{shrink}(\mathbf{\Xi},\xi)_{ij}=\mathrm{sgn}(\mathbf{\Xi}_{ij})\max\left(|\mathbf{\Xi}_{ij}|-\xi,0\right)$. Since no closed-form solution can be found to ensure its consistency, i.e., $\mathbf{Y}^{(k)}<\mathbf{\Sigma}^{-1}$ [see (16)], we relax this consistency constraint in this step. This can be justified by the fact that, although $\mathbf{Y}^{(k)}$ may not satisfy (16) or even not be positive definite, $\mathbf{X}^{(k+1)}$, which is what we are ultimately seeking, always remains so by construction.

V. Experimental Results

To validate the proposed consistent graph sparsification, we have preliminarily tested the algorithm on both synthetic and real-world data: i) Manhattan3500 (3500 nodes, 5598 edges), ii) City4000 (2815 nodes, 20687 edges), iii) Intel research lab (909 nodes, 4453 edges), and iv) MIT Killian court (1940 nodes, 2190 edges). The first two datasets are synthetic, while the last two are real. For the results presented in this section, we integrate the proposed graph sparsification scheme into the iSAM algorithm [2] performing incremental estimation, termed S-iSAM,[1] which is compared with the ground truth for the simulated data and with the batch least-squares (BLS) – that is the best result can be obtained in practice – for the real data. We marginalize out the old poses when loop closures occur, and sparsify the graph when the number of non-zeros of the information (Hessian) matrix exceeds a pre-chosen threshold (for instance, 10% of the total matrix entries).

In particular, Fig. 1 shows, as an example, the information (Hessian) matrices before and after the proposed sparsification of edges in the case of Manhattan3500. It is clear that the sparsified information (Hessian) matrix significantly reduces the non-zeros, by 77% in this particular case, while preserving consistency (which can be easily verified numerically). Note also that, in practice it is not necessary to perform the proposed edge sparsification, i.e., to solve the ℓ_1-regularized minimization, every time step, thus saving substantial computational cost, particularly when the graph becomes large. This is due to the fact that the graph becomes dense only when marginalization or loop closure occurs, which, in general, is an occasional event (i.e., not as frequent as adding an edge/node).

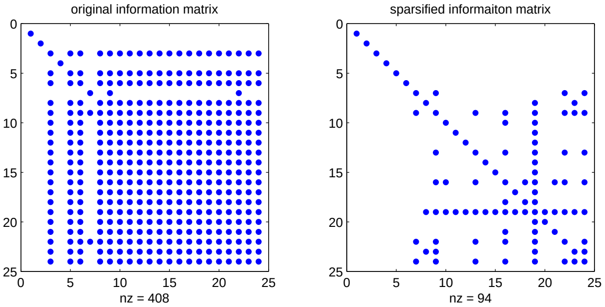


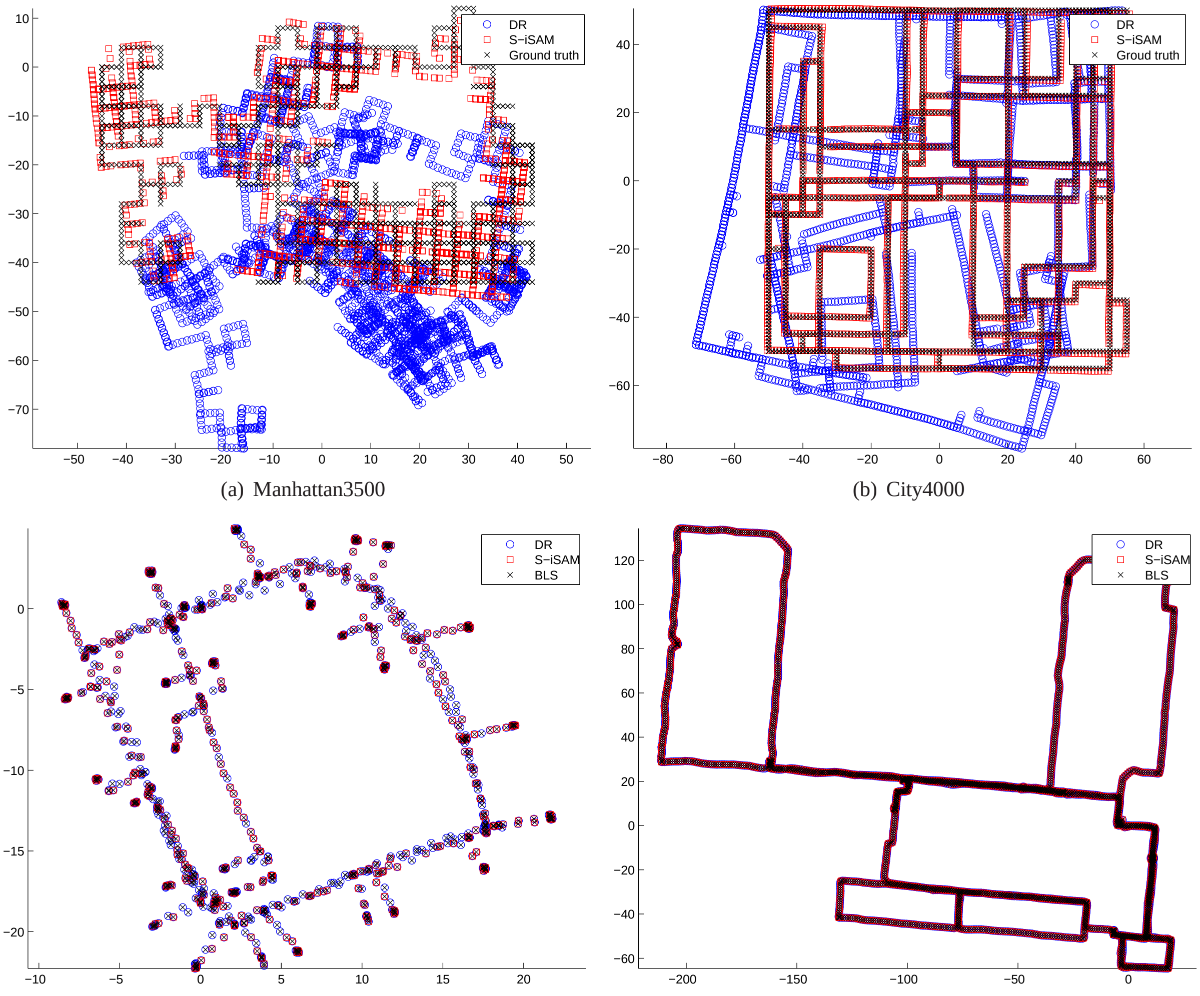
Fig. 1. Illustration of the proposed consistent sparsification of edges for the Manhattan3500 graph occurring at a particular time step.

Moreover, Figs. 2 and 3 show the estimates and their errors, respectively, as compared to the benchmarks (i.e., ground truth for simulated data, and BLS for the real data). As evident, the proposed S-iSAM achieves results close to the benchmarks, while keeping significantly fewer nodes, i.e., 1971 nodes for Manhattan3500 (56%), 2098 nodes for City4000 (75%), 507 nodes for Intel (56%), and 1718 nodes for MIT (89%). We would like to point out that the proposed node sparsification provides a consistent way of retrieving the information about the remaining nodes from the discarded measurements, rather than intends to find the minimum number of nodes to represent the original graph. It should be also stressed that, when a node is marginalized out, all the measurements (edges) connected to it – including the future measurements of this kind – are discarded, and hence cannot be used for the subsequent estimation in order to obtain causal estimates (without using discarded information). This implies that the proposed approach may utilize significantly fewer measurements (depending on the particular graphs) than the BLS that uses all available measurements, while achieving comparable performance. This result is attributed to the fact that *all* the information, conveyed in the discarded measurements, about the remaining states, is retained during the proposed consistent node sparsification.

VI. Conclusions and Future Work

In this paper, we have proposed a consistent sparsification scheme for graph optimization with applications to SLAM. In particular, we first sparsify the nodes of the graph by marginalizing out old poses at the times of loop closures, while retaining all the information for the remaining nodes conveyed in the discarded measurements. To this end, we formulate a batch MLE (least-squares) problem using only the discarded measurements, with respect to the relative, instead

[1] It is important to point out that the proposed sparsification scheme can be integrated into the SEIF as in [31], besides the iSAM as considered here.



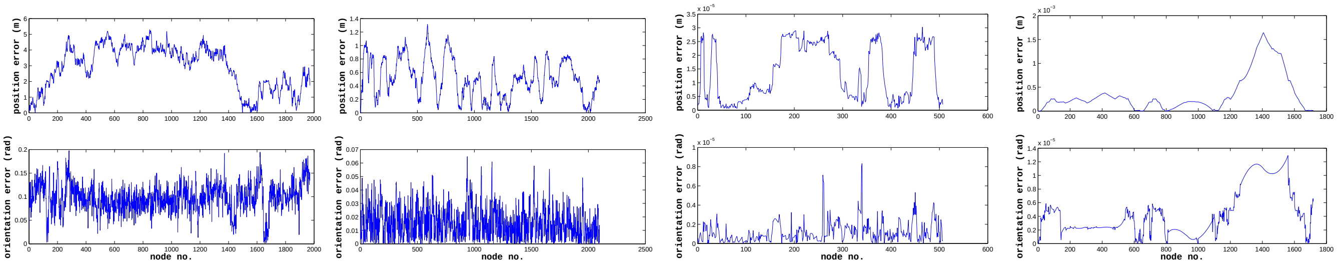
(a) Manhattan3500

(b) City4000

(c) Intel Research Lab

(d) MIT Killian Court

Fig. 2. Experimental results on the synthetic (a and b) and real (d and c) data: In these plots, the dots correspond to the dead-reckoning (DR), the crosses to the benchmark which is either the ground truth or the batch least-squares (BLS), the squares to the iSAM with the proposed sparsification scheme (S-iSAM). It is clear that the proposed method achieves results close to the benchmarks, even though it utilizes less information from the available measurements as some measurements are discarded for the future usage due to marginalization. Note also that in the two real-world datasets (c and d), the compared approaches produce very similar results (see Fig. 3), which makes the corresponding lines difficult to distinguish.



(a) Manhattan3500

(b) City4000

(c) Intel Research Lab

(d) MIT Killian Court

Fig. 3. Estimation errors of the proposed S-iSAM as compared to the benchmarks (ground truth or BLS) in the synthetic and real data. In these plots, the upper subplots are the estimation errors of the positions and the bottom ones are the orientation estimation errors. It becomes evident that the iSAM with the proposed graph sparsification scheme attains comparable performance as that of the benchmarks.

of global, states (i.e., in the local frame of reference of one remaining node involved in the discarded measurements). Furthermore, we sparsify the edges of the graph based on a consistent ℓ_1-regularized convex minimization problem which promotes the sparsity of the solution imposed by ℓ_1 norm and is solved by adapting the ADMM algorithm [39]. The solution to this convex problem automatically renders a sparsified graph that consistently approximates the original one. The proposed graph sparsification scheme has been integrated in the iSAM algorithm [2] and validated on both synthetic and real data.

It should be pointed out that, although the ADMM method is efficient to solve a relatively small problem of consistent edge sparsification [see (15) and (16)] (e.g., when the number of nodes of the graph is in the order of thousands), it may not be satisfactory when the graph becomes even larger. For this reason, in the future we will investigate different ways to speed up the solution to the ℓ_1-regularized minimization problem, for example, by selecting only a dense subgraph to sparsify via the ADMM approach, instead of the whole graph as we currently do. Moreover, since, by no means, it would be optimal to marginalize out the old nodes at loop closures, we will study in depth the problem of which nodes should be marginalized out and when, in order to maximize the information retained in the sparsified graph while achieving a minimal representation of the original graph. In addition, we will conduct more thorough evaluations on various large-scale real-world experiments.

References

[1] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, Dec. 2006.

[2] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, Dec. 2008.

[3] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[4] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, Shanghai, China, May 9–13, 2011, pp. 3607–3613.

[5] E. Eade, P. Fong, and M. Munich, "Monocular graph SLAM with complexity reduction," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, Taipei, Taiwan, Oct. 18–22, 2010, pp. 3017–3024.

[6] V. Ila, J. Porta, and J. Andrade-Cetto, "Information-based compact pose SLAM," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 78–93, Feb. 2010.

[7] H. Kretzschmar and C. Stachniss, "Information-theoretic compression of pose graphs for laser-based SLAM," *International Journal of Robotics Research*, vol. 31, no. 11, pp. 1219–1230, Sep. 2012.

[8] H. Johannsson, M. Kaess, M. Fallon, and J. Leonard, "Temporally scalable visual SLAM using a reduced pose graph," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, Karlsruhe,Germany, May 6–10, 2013, pp. 54–61.

[9] N. Carlevaris-Bianco and R. M. Eustice, "Generic factor-based node marginalization and edge sparsification for pose-graph SLAM," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, Karlsruhe, Germany, May 6–10, 2013, pp. 5728–5735.

[10] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation*. New York: Wiley, 2001.

[11] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment – a modern synthesis," *Lecture Notes in Computer Science*, vol. 1883, pp. 298–375, Jan. 2000.

[12] D. Donoho, "Compressed sensing," *IEEE Transactions on Signal Processing*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[13] S. Thrun, "Robotic mapping: A survey," in *Exploring artificial intelligence in the new millennium*, G. Lakemeyer and B. Nebel, Eds. San Francisco, CA: Morgan Kaufmann Inc., 2003, pp. 1–35.

[14] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. Cambridge, MA: The MIT Press, 2005.

[15] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robotics Automation Magazine*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[16] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robotics Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[17] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, vol. 4, no. 4, pp. 333–349, Oct. 1997.

[18] J. Folkesson and H. Christensen, "Graphical SLAM - a self-correcting map," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, vol. 1, Apr. 26– May 1, 2004, pp. 383–390.

[19] E. Olson, J. Leonard, and S. Teller, "Fast iterative alignment of pose graphs with poor initial estimates," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, Orlando, FL, May 15–19, 2006, pp. 2262–2269.

[20] K. Konolige, G. Grisetti, R. Kummerle, W. Burgard, B. Limketkai, and R. Vincent, "Efficient sparse pose adjustment for 2D mapping," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, Taipei, Taiwan, Oct. 18–22, 2010, pp. 22–29.

[21] G. Grisetti, C. Stachniss, and W. Burgard, "Nonlinear constraint network optimization for efficient map learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 428–439, Sep. 2009.

[22] G. Grisetti, R. Kummerle, C. Stachniss, U. Frese, and C. Hertzberg, "Hierarchical optimization on manifolds for online 2D and 3D mapping," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, Anchorage, AK, May3–8, 2010, pp. 273–278.

[23] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Visual SLAM: Why filter?" *Image and Vision Computing*, vol. 30, no. 2, pp. 65–77, Feb. 2012.

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

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

[26] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "An observability constrained sliding window filter for SLAM," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, San Francisco, CA, Sep. 25–30, 2011, pp. 65–72.

[27] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. of the IEEE and ACM International Symposium on Mixed and Augmented Reality*, Nara, Japan, Nov. 13–16, 2007.

[28] K. Konolige and M. Agrawal, "FrameSLAM: From bundle adjustment to real-time visual mapping," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1066–1077, Oct. 2008.

[29] K. Konolige, J. Bowman, J. D. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua, "View-based maps," *International Journal of Robotics Research*, vol. 29, no. 8, pp. 941–957, Jul. 2010.

[30] R. M. Eustice, H. Singh, and J. J. Leonard, "Exactly sparse delayed-state filters for view-based SLAM," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1100–1114, Dec. 2006.

[31] J. Vial, H. Durrant-Whyte, and T. Bailey, "Conservative sparsification for efficient and consistent approximate estimation," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, San Francisco, CA, Sep. 25–30, 2011, pp. 886 –893.

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

[33] M. R. Walter, R. M. Eustice, and J. J. Leonard, "Exactly sparse extended information filters for feature-based SLAM," *International Journal of Robotics Research*, vol. 26, no. 4, pp. 335–359, 2007.

[34] E. Olson, "Robust and efficient robotic mapping," Ph.D. dissertation, MIT, Cambridge, MA, USA, 2008.

[35] G. H. Golub and C. F. V. Loan, *Matrix Computations*. The Johns Hopkins University Press, 1996.

[36] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[37] O. Banerjee, L. E. Ghaoui, A. d'Aspremont, and G. Natsoulis, "Convex optimization techniques for fitting sparse Gaussian graphical models," in *Proc. of the International Conference on Machine Learning*, New York, NY, Jun. 25–29, 2006, pp. 89–96.

[38] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1990.

[39] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[40] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society (Series B)*, vol. 58, no. 1, pp. 267–288, 1996.