Fast Global Optimality Verification in 3D SLAM

Jesus Briales Javier Gonzalez-Jimenez

***Abstract*— Graph-based SLAM has proved to be one of the most effective solutions to the Simultaneous Localization and Mapping problem. This approach relies on nonlinear iterative optimization methods that in practice perform both accurately and efficiently. However, due to the non-convexity of the problem, the obtained solutions come with no guarantee of global optimality and may get stuck in local minima. The application of SLAM to many real-world applications cannot be conceived without additional control tools that detect possible suboptimalities *as soon as possible* in order to take corrective action and avoid catastrophic failure of the entire system.**

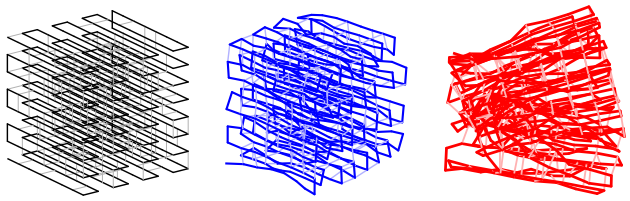
This paper builds upon the state-of-the-art framework [1] in verification for this problem and introduces a novel superior formulation that leads to a much higher efficiency. While retaining the same high effectiveness, the verification times of our proposal reduce up to >50x, paving the way for faster verification in critical real applications or in embedded low-power systems. We support our claims with extensive experiments with real and simulated data.

I. Introduction

The *Simultaneous Localization and Mapping* (SLAM) problem consists of the estimation of both the unknown environment and the trajectory followed by an exteroceptive sensory system from the data it provides. SLAM is at the core of many emerging applications that require to interact with an unknown environment. These include autonomous navigation in any of its modalities (self-driving cars, unmanned aerial vehicles, autonomous underwater devices or even planetary exploration), service, domestic and industrial robotics or augmented reality.

The problem is commonly tackled by two layers: *front-end* in charge of constructing a graph where nodes represent the poses and the arcs encode the relative approximate poses between nodes, and a *back-end* which works towards the coherence of the whole graph through the optimization of a high-dimensional, non-convex, non-linear problem. State-of-the-art approaches for solving this problem are based on iterative optimization techniques that exploits the sparsity of the problem [2]–[4].

However, even though empirically these techniques perform remarkably well, they all share a common limitation: Due to its iterative nature and the non-convexity of the optimization problem, there is no guarantee about the correctness (global optimality) of the obtained solution. Getting stucked in local minima (in general in any stationary point which is not optimal) may lead to estimated solutions that are arbitrarily far from the real optimal point (see Fig. 1). This has inspired different research lines that address the issue of improving convergence from different perspectives: initialization techniques for bootstrapping the iterative optimization [5]–[7], increasing the basin of convergence [4], [8], [9], analysis of the local convergence, problem structure, and influencing factors [10]–[15] or different relaxations of the problem [16] aimed at getting near-optimal global solutions. All these works provide valuable contributions for the aforementioned problem, yet none of them produce any guarantee about the global optimality.



(a) Groundtruth (b) MLE optimum (c) Wrong minimum

Fig. 1. Example of a Pose Graph Optimization (PGO) problem. The synthetic scenario (a) is used to produce noisy odometry and loop closures. The corresponding Maximum Likelihood Estimator (MLE) produces appealing results (b), but poor initialization may lead to wrong suboptimal solutions (c). This work boosts verification techniques for checking the global optimality of a given result.

We see how the increasing success of SLAM techniques is pushing the interest of many technological companies in a range of fields, including autonomous drones and cars, augmented reality, video games, etc. Many of these applications, however, demand a guaranteed performance: As the level of responsibility involved by the automated task grows (e.g. the safety of passengers in the case of self-driving vehicles) techniques that provide awareness on possible suboptimalities of the map or the trajectory estimated by the SLAM pipeline become essential. This way corrective actions could be taken before these errors propagate into the rest of the system (e.g. path planning) and drive this to catastrophic failure.

Very recently, some important progress has been made in this direction by Carlone and collaborators [1], [17], [18], by developing techniques for the SLAM problem of Pose Graph Optimization (PGO) that are able to certify the global optimality of a candidate solution obtained by other means. One of their fundamental contribution is a formulation of the Maximum Likelihood Estimator (MLE) for PGO that turns it into a quadratically constrained optimization problem with quadratic objective. Their second great idea is to apply *duality theory* to yield non-trivial results about the global optimum of the problem.

Mapir Group. Universidad de Málaga. E.T.S. Ingeniería de Informática. Campus de Teatinos, 29071, Málaga, Spain. E-mail: jesusbriales@uma.es

This work has been supported by the Spanish grant program FPU14/06098 and the project PROMOVE (DPI2014-55826-R), funded by the Spanish Government and the "European Regional Development Fund".

Our *main contribution* upon the work [1] for global optimality in 3D PGO is the introduction of a simple yet very effective reformulation of the optimization problem that leads to a smaller *matrix formulation* of the problem and more insightful results. Section II reviews the original formulation in [1] and then elaborates the new *trace-based* formulation. Similar benefits of a good formulation for PGO have been also shown for the planar 2D case [18] although their trick there (namely to exploit the natural bijection between the 2D space and the complex domain) does not seem applicable for the 3D case.

The development of the *Lagrangian dual problem* is a core tool for the subsequent use of duality theory. In Section III we provide a detailed derivation of the *dual problem* corresponding to our new formulation. As for [1], it is a (convex) *semidefinite program* (SDP). Unlike previous works, we avoid anchoring[1] the problem. This way we obtain a simpler dual problem and, most importantly, the results remain fully general, no matter which particular reference frame is chosen for the PGO.

Duality theory can be used to certify the global optimality of a candidate solution for PGO. In Section IV *verification techniques* are obtained as a straightforward application of the verification pipelines proposed in [1], [18] to our new formulation. Again, the steps in the pipeline get simpler under our reformulation.

To prove the benefits of our contribution beyond the formal aspects we conducted experiments on real and simulated data in Section V. Overall, our framework provides a significant boost in computational performance with respect to the state-of-the-art reference [1]. With time improvements of several orders of magnitude, we show that the application of these techniques may not be out of reach for large-scale real problems.

Finally, further insight into this work can be gained from the supplementary video or from the experimental implementation, both available at `https://mapir.isa.uma.es`.

II. Quadratic Pose Graph Optimization problem

The Pose Graph Optimization (PGO) formulation of the 3D SLAM problem consists of the estimation of a set of n unknown poses $(\boldsymbol{R}_i, \boldsymbol{t}_i)$ from m relative measurements $(\bar{\boldsymbol{R}}_{ij}, \bar{\boldsymbol{t}}_{ij})$ between pairs of poses i and j. These observations may originate from different sources, typically, odometry and common observations.

This problem can be visualized as a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, so that each node $i \in \mathcal{V} = \{1, \dots, n\}$ stands for an unknown pose and each of the m relative measurements is represented by an edge $(i, j) \in \mathcal{E}$ in the graph. The unknown poses are estimated as those that maximize the consistency of the model with the observations, yielding an optimization problem of the form

$$f^\star_{\text{ML}} = \min_{\substack{\{\boldsymbol{R}_i \in \text{SO}(3)\} \\ \{\boldsymbol{t}_i \in \mathbb{R}^3\}}} \sum_{(i,j)\in\mathcal{E}} \omega_{\boldsymbol{t}}^2 \|\boldsymbol{t}_j - \boldsymbol{t}_i - \boldsymbol{R}_i \bar{\boldsymbol{t}}_{ij}\|_2^2 + \tfrac{\omega_{\boldsymbol{R}}^2}{2} \|\boldsymbol{R}_j - \boldsymbol{R}_i \bar{\boldsymbol{R}}_{ij}\|_F^2. \quad (1)$$

Note the chordal distance [19] is chosen for the rotation error. According to [1, Sec. II], the problem (1) is the Maximum Likelihood Estimator (MLE) for Pose Graph Optimization under the assumption of Gaussian noise on translation measurements and von Mises noise [20] on rotations.

As a positively weighted sum of distances, the optimization cost f in (1) is a convex function. Furthermore, f is quadratic on the unknown variables, so it can be conveniently expressed in a matrix form to take full advantage of algebra tools. For example, the reference work [1], [17] fiddles with the terms in (1) to get the equivalent quadratic formulation $f(\boldsymbol{x}) = \|\boldsymbol{A}\boldsymbol{x}\|_2^2$. Here the rotation matrices $\boldsymbol{R}_i$ are vectorized as 9×1 vectors $\boldsymbol{r}_i$ and stacked into the unknown vector $\boldsymbol{x} = \begin{bmatrix}\boldsymbol{r}_1, \dots, \boldsymbol{r}_n, \boldsymbol{t}_1, \dots, \boldsymbol{t}_n\end{bmatrix} \in \mathbb{R}^{12n}$. This formulation makes heavy use of vectorization and Kronecker products. As a consequence, the representation dimensions grow unnecessarily (see Fig. 2), as shown by our novel equivalent trace-based formulation.

A. Trace-based formulation

It is a well known fact that inner products naturally induce associated norms. That is the case of the dot product and Euclidean 2-norm for vectors and the trace (tr) product and Frobenius norm for $n \times m$ matrices (Tab. I). Inspired by the fact that the tr product is a generalization of the dot product to matrices, we decide to work backwards and reformulate the Euclidean norm in terms of the tr product. We will see that this allows us to reach a much more compact formulation of f with fewer terms.

First, the translation term $\|(\boldsymbol{t}_j - \boldsymbol{t}_i) - \boldsymbol{R}_i \bar{\boldsymbol{t}}_{ij}\|_2^2$ is rewritten as

$$\begin{aligned}&(\boldsymbol{t}_j - \boldsymbol{t}_i)^\top (\boldsymbol{t}_j - \boldsymbol{t}_i) - 2(\boldsymbol{t}_j - \boldsymbol{t}_i)^\top (\boldsymbol{R}_i \bar{\boldsymbol{t}}_{ij}) + \bar{\boldsymbol{t}}_{ij}^\top \boldsymbol{R}_i^\top \boldsymbol{R}_i \bar{\boldsymbol{t}}_{ij} \\ &= \text{tr}((\boldsymbol{t}_j - \boldsymbol{t}_i)(\boldsymbol{t}_j - \boldsymbol{t}_i)^\top) - 2\,\text{tr}(\boldsymbol{R}_i \bar{\boldsymbol{t}}_{ij} (\boldsymbol{t}_j - \boldsymbol{t}_i)^\top) + \underbrace{\bar{\boldsymbol{t}}_{ij}^\top \bar{\boldsymbol{t}}_{ij}}_{\text{const.}},\end{aligned}$$

where we used the property $\boldsymbol{a}^\top \boldsymbol{b} = \text{tr}(\boldsymbol{a}\boldsymbol{b}^\top)$ together with the implicit constraint $\boldsymbol{R}_i^\top \boldsymbol{R}_i = \boldsymbol{I}_3$.

For the rotation term, we apply that $\|\boldsymbol{A}\|_F^2 = \text{tr}(\boldsymbol{A}^\top \boldsymbol{A})$ to expand the chordal distance $\|\boldsymbol{R}_j - \boldsymbol{R}_i \bar{\boldsymbol{R}}_{ij}\|_F^2$ as

$$-2\,\text{tr}(\bar{\boldsymbol{R}}_{ij} \boldsymbol{R}_j^\top \boldsymbol{R}_i) + \underbrace{\text{tr}(\overbrace{\boldsymbol{R}_j^\top \boldsymbol{R}_j}^{\boldsymbol{I}_3}) + \text{tr}(\overbrace{\bar{\boldsymbol{R}}_{ij}^\top \boldsymbol{R}_i^\top \boldsymbol{R}_i \bar{\boldsymbol{R}}_{ij}}^{\boldsymbol{I}_3})}_{\text{const.}}.$$

Space	Inner product	Induced norm
$\mathbb{R}^n$	[dot product] $\boldsymbol{a}^\top \boldsymbol{b}$	$\|\boldsymbol{a}\|_2$
$\mathbb{R}^{n\times m}$	[tr product] $\text{tr}(\boldsymbol{A}^\top \boldsymbol{B})$	$\|\boldsymbol{A}\|_F$

TABLE I

The trace product extends the dot product to matrix spaces

[1] Absolute poses are not observable in PGO. Anchoring fixes the pose of some node to remove this observability issue. See [1, Sec. III].

We see how in both transformations constant terms appear that may be dropped from the optimization. In doing so we are intrinsically increasing the sparsity of the final quadratic matrix that will define the objective cost, as exemplified in Figure 2.

Let us define now the block-vectors of stacked unknowns

$$\underbrace{\boldsymbol{Q}}_{3n\times 3} = \left[\,\boldsymbol{R}_1\middle|\ldots\middle|\boldsymbol{R}_n\,\right]^\top,\ \underbrace{\boldsymbol{T}}_{n\times 3} = \left[\,\boldsymbol{t}_1\middle|\ldots\middle|\boldsymbol{t}_n\,\right]^\top,\ \underbrace{\boldsymbol{X}}_{4n\times 3} = \begin{bmatrix}\boldsymbol{Q}\\ \boldsymbol{T}\end{bmatrix}.$$

Let $\boldsymbol{e}_i$ stand for the i-th canonical vector and be $\otimes$ the usual Kronecker product of two matrices. Using the relations

$$\boldsymbol{t}_j - \boldsymbol{t}_i = \boldsymbol{T}^\top(\boldsymbol{e}_j - \boldsymbol{e}_i), \quad \boldsymbol{R}_i = \boldsymbol{Q}^\top \underbrace{(\boldsymbol{e}_i \otimes \boldsymbol{I}_3)}_{\boldsymbol{E}_i}$$

the overall translation cost term in f is written as

$$\begin{aligned}&\operatorname{tr}(\boldsymbol{T}^\top \boldsymbol{M}_{\boldsymbol{TT}}\boldsymbol{T}) + 2\operatorname{tr}(\boldsymbol{Q}^\top \boldsymbol{M}_{\boldsymbol{QT}}\boldsymbol{T}),\\ \underbrace{\boldsymbol{M}_{\boldsymbol{TT}}}_{n\times n} &= +\sum_{(i,j)\in\mathcal{E}} \omega_{\boldsymbol{t}_{ij}}(\boldsymbol{e}_j - \boldsymbol{e}_i)(\boldsymbol{e}_j - \boldsymbol{e}_i)^\top,\\ \underbrace{\boldsymbol{M}_{\boldsymbol{QT}}}_{3n\times n} &= -\sum_{(i,j)\in\mathcal{E}} \omega_{\boldsymbol{t}_{ij}} \boldsymbol{E}_i \bar{\boldsymbol{t}}_{ij}(\boldsymbol{e}_j - \boldsymbol{e}_i)^\top,\end{aligned}$$

whereas the rotation part of the cost function becomes

$$\operatorname{tr}(\boldsymbol{Q}^\top \boldsymbol{M}_{\boldsymbol{QQ}}\boldsymbol{Q}), \quad \underbrace{\boldsymbol{M}_{\boldsymbol{QQ}}}_{3n\times 3n} = -\operatorname{sym}\Big(\sum_{(i,j)\in\mathcal{E}} \omega_{\boldsymbol{R}_{ij}} \boldsymbol{E}_i \bar{\boldsymbol{R}}_{ij} \boldsymbol{E}_j^\top\Big).$$

Here $\operatorname{sym}(\boldsymbol{M}) \equiv \frac{\boldsymbol{M}+\boldsymbol{M}^\top}{2}$ stands for the symmetric part of a matrix.

We see from these expressions that the coefficient matrices $\boldsymbol{M}_{\boldsymbol{QQ}}$, $\boldsymbol{M}_{\boldsymbol{QT}}$ and $\boldsymbol{M}_{\boldsymbol{TT}}$ are simple block-matrices obtained by adding the data (weighted relative measurements) into the block pointed by the edge's start and end indices.

Adding both terms, the simplified cost function is

$$f(\boldsymbol{X}) = \text{const.} + \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{M}\boldsymbol{X}), \tag{2}$$

where all the data in the original problem have been condensed into an appropiate constant and the cost matrix

$$\underbrace{\boldsymbol{M}}_{4n\times 4n} = \left[\begin{array}{c|c} \boldsymbol{M}_{\boldsymbol{QQ}} & \boldsymbol{M}_{\boldsymbol{QT}} \\ \hline \boldsymbol{M}_{\boldsymbol{QT}}^\top & \boldsymbol{M}_{\boldsymbol{TT}} \end{array}\right]. \tag{3}$$

III. Relaxations and Lagrangian dual problem

The pursued task in this paper, checking the optimality of a point $\boldsymbol{X}$ for the PGO problem (1), would be straightforward if we knew $f^\star_{\text{ML}}$, but this is unknown. However, it was shown in [1] that *duality theory*, which by definition produces lower bounds on the optimum of an optimization problem, can be effectively exploited for this task. Thus, it follows the derivation of the Lagrangian dual problem, which becomes a fundamental result for the verification tools that will be presented in Section IV.

First, let us consider the implicit constraints $\boldsymbol{R}_i \in \text{SO}(3)$ in (1) or, equivalently, $\boldsymbol{Q}_i \in \text{SO}(3)$ where $\boldsymbol{Q}_i = \boldsymbol{R}_i^\top$ are the blocks in our stacked unknown $\boldsymbol{Q}$. These make the optimization problem non-convex and difficult to solve

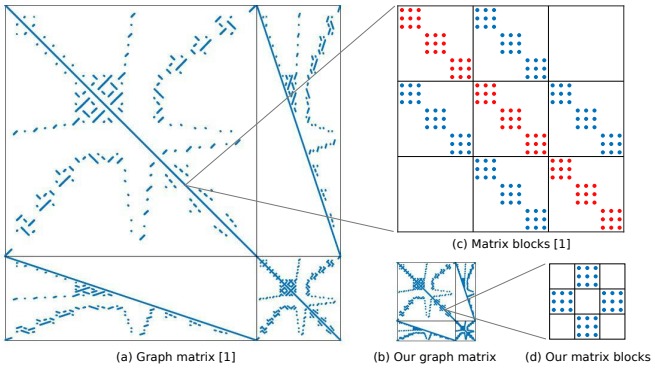


Fig. 2. Quadratic matrix of the PGO problem corresponding to the *garage* dataset for (a) the formulation in [1] and (b) our formulation. Every rotation observation produces a 9×9 block (c), but our formulation reduces it to 3×3 (d). The diagonal blocks (in red) are dropped in our formulation as constant terms. Similar reductions are applied to the $\boldsymbol{M}_{\boldsymbol{TT}}$ and $\boldsymbol{M}_{\boldsymbol{QT}}$ blocks. As a result we get a matrix (b) that is smaller and sparser than (a): (a) $20\text{K}\times 20\text{K}$, $nz = 570\text{K}$, (b) $6.6\text{K}\times 6.6\text{K}$, $nz = 175\text{K}$. (K - thousands, nz - nonzero elements)

globally. Each of these can be written for the rotation matrix representation as

$$\boldsymbol{Q}_i \in \text{SO}(3) \iff \begin{cases}\boldsymbol{Q}_i^\top \boldsymbol{Q}_i = \boldsymbol{I}_3,\\ \det(\boldsymbol{Q}_i) = +1.\end{cases} \tag{4}$$

The determinant constraint is usually dropped, which amounts to performing estimation in $\text{O}(3)$ rather than $\text{SO}(3)$ and checking the determinant value *a posteriori*. For common PGO instances empirical results have shown that this constraint tends to be redundant [21], however, the remaining non-convex equality quadratic constraints produce local minima in which iterative methods may get stuck, making it non-trivial to find the global optimum.

Let us write the *relaxed* version of the MLE (1) that drops the determinant constraint using our trace-based formulation:

$$\begin{aligned} f^\star = \min_{\boldsymbol{X}} \quad & \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{M}\boldsymbol{X}) \\ \text{s.t.} \quad & \underbrace{\boldsymbol{I}_3 - \boldsymbol{Q}_i\boldsymbol{Q}_i^\top}_{\boldsymbol{C}_i(\boldsymbol{Q})} = \boldsymbol{0}, \quad i = 1,\ldots,n. \end{aligned} \tag{5}$$

As a *relaxation* of the original problem, the condition

$$f^\star \leq f^\star_{\text{ML}} \tag{6}$$

stands between the optimal costs of both problems. If the equality is fulfilled, the determinant constraints are *inactive*.

Since the main obstacle in this problem arises because of the constraints, these can be introduced in the objective cost as penalty terms:

$$\begin{aligned}\text{L}(\boldsymbol{X}, \{\boldsymbol{\Lambda}_i\}) &= f(\boldsymbol{X}) + \sum_{i=1}^n \operatorname{tr}(\boldsymbol{\Lambda}_i \boldsymbol{C}_i(\boldsymbol{Q}))\\ &= \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{M}\boldsymbol{X}) + \sum_{i=1}^n \operatorname{tr}(\boldsymbol{\Lambda}_i(\boldsymbol{I}_3 - \boldsymbol{Q}_i\boldsymbol{Q}_i^\top))\\ &= \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{M}(\boldsymbol{\Lambda})\boldsymbol{X}) + \operatorname{tr}(\boldsymbol{\Lambda})\end{aligned}$$

where we define $\boldsymbol{\Lambda} = \text{blkdiag}(\{\boldsymbol{\Lambda}_i\})$ as a block-diagonal matrix with blocks $\boldsymbol{\Lambda}_i$ and the *penalized cost matrix* as

$$\boldsymbol{M}(\boldsymbol{\Lambda}) = \boldsymbol{M}(\boldsymbol{\Lambda})^\top = \begin{bmatrix} \boldsymbol{M}_{\boldsymbol{QQ}} - \boldsymbol{\Lambda} & \boldsymbol{M}_{\boldsymbol{QT}} \\ \boldsymbol{M}_{\boldsymbol{TQ}} & \boldsymbol{M}_{\boldsymbol{TT}} \end{bmatrix}. \tag{7}$$

The cost function $\text{L}(\cdot)$ defined above is the well-known Lagrangian of the constrained optimization problem and the variables $\boldsymbol{\Lambda}_i$ are the matrix version of the Lagrange multipliers. Note that these penalty terms are consistent with the usual Lagrangian definition since they fulfill that

$$\frac{\partial \text{L}(\boldsymbol{X}, \{\boldsymbol{\Lambda}_i\})}{\partial \boldsymbol{\Lambda}_i} = C_i(\boldsymbol{Q}) = \boldsymbol{0}.$$

Furthermore, since the constraint matrices $C_i(\boldsymbol{Q})$ are all symmetric, so must be the Lagrange multipliers:

$$C_i(\boldsymbol{Q}) = C_i(\boldsymbol{Q})^\top \Rightarrow \boldsymbol{\Lambda}_i \in \mathbb{S}^3.$$

The Lagrangian provides an unconstrained *relaxation* of the original problem

$$d(\boldsymbol{\Lambda}) = \min_{\boldsymbol{X}} \text{L}(\boldsymbol{X}, \boldsymbol{\Lambda}) \tag{8}$$

whose optimal value $d(\boldsymbol{\Lambda}) \leq f^\star$ is called the *dual function*. Since the term $\text{tr}(\boldsymbol{X}^\top \boldsymbol{M}(\boldsymbol{\Lambda})\boldsymbol{X})$ is a homogeneous quadratic form (wrt $\boldsymbol{X}$), its minimum value 0 is attained for

$$\begin{aligned} \boldsymbol{X}^\star(\boldsymbol{\Lambda}) &= \{\boldsymbol{X} : \text{tr}(\boldsymbol{X}^\top \boldsymbol{M}(\boldsymbol{\Lambda}^\star)\boldsymbol{X}) = 0\} \\ &= \{\boldsymbol{X} : \boldsymbol{M}(\boldsymbol{\Lambda})\boldsymbol{X} = \boldsymbol{0}_{4n\times 3}\}, \end{aligned} \tag{9}$$

if the penalized matrix is semidefinite positive ($\boldsymbol{M}(\boldsymbol{\Lambda}) \succcurlyeq \boldsymbol{0}$). Otherwise this term is unbounded below (its minimum value is $-\infty$). Thus, the optimum value of the Lagrangian (8) is

$$d(\boldsymbol{\Lambda}) = \begin{cases} \text{tr}(\boldsymbol{\Lambda}) & \text{if } \boldsymbol{M}(\boldsymbol{\Lambda}) \succcurlyeq \boldsymbol{0}, \\ -\infty & \text{otherwise.} \end{cases} \tag{10}$$

The *dual problem* seeks the tightest relaxation by maximizing the lower bound $d(\boldsymbol{\Lambda})$ wrt $\boldsymbol{\Lambda}$ as $d^\star = \max_{\boldsymbol{\Lambda}} d(\boldsymbol{\Lambda})$. In view of the expression for the dual objective (10), the search for the maximum can be safely restricted to values of $\boldsymbol{\Lambda}$ for which $\boldsymbol{M}(\boldsymbol{\Lambda}) \succcurlyeq \boldsymbol{0}$, so the dual problem is a *Semidefinite Program* (SDP):

$$d^\star = \max_{\boldsymbol{\Lambda}} \text{tr}(\boldsymbol{\Lambda}), \quad \text{s.t. } \boldsymbol{M}(\boldsymbol{\Lambda}) \succcurlyeq \boldsymbol{0}. \tag{11}$$

This *convex* problem appears in many practical optimization applications and specialized solvers exist for it.

IV. Optimality conditions and verification

This section provides a tuning under our novel formulation of the verification techniques proposed in [1], [18], exploiting similar duality theory concepts. For a more theoretical background on the topic of duality theory and the properties exploited here we recommend consulting [22, Sec. 5].

From now on, we will refer to the original MLE (1) as our *primal problem*, whereas the SDP (11) is the *dual problem*. A solution is primal (resp. dual) feasible if it fulfills the constraints defined by the primal (resp. dual) problem.

Input: Cost matrix $\boldsymbol{M}$, feasible solution $\hat{\boldsymbol{X}}_{\text{ML}}$
Output: Optimality certificate `isOpt`
solve the SDP problem (11) to get $d^\star$;
if $f(\hat{\boldsymbol{X}}_{ML}) == d^\star$ **then**
  set `isOpt = true`;
else
  set `isOpt = unknown`;
end
return `isOpt`

Algorithm V1: Optimality verification via SDP

A first important relation for the subsequent analysis is the chain of inequalities

$$d(\hat{\boldsymbol{\Lambda}}) \leq d^\star \leq f^\star \leq f^\star_{\text{ML}} \leq f(\hat{\boldsymbol{X}}_{\text{ML}}), \tag{12}$$

where $\hat{\boldsymbol{X}}_{\text{ML}}$ and $\hat{\boldsymbol{\Lambda}}$ are primal and dual feasible points, respectively. The inequalities $d(\hat{\boldsymbol{\Lambda}}) \leq d^\star$ and $f^\star_{\text{ML}} \leq f(\hat{\boldsymbol{X}}_{\text{ML}})$ stem from the definition of the (dual) *maximization* and (primal) *minimization* problems (1) and (11), respectively. The *relaxation* relationship (6) justifies $f^\star \leq f^\star_{\text{ML}}$. Finally, $d^\star \leq f^\star$ states the well-known *weak duality* principle, and the difference $f^\star - d^\star \geq 0$ is called the *duality gap*. If $f^\star = d^\star$ we say there is *strong duality*.

SDP-based optimality verification: As noted in [1], [18], if we know the optimum objective $d^\star$ for the dual problem, the inequality chain (12) provides a straightforward approach for checking the optimality of a primal-feasible candidate $\hat{\boldsymbol{X}}_{\text{ML}}$. Namely, if $d^\star = f(\hat{\boldsymbol{X}}_{\text{ML}})$, the inequality chain (12) forces $f^\star = f(\hat{\boldsymbol{X}}_{\text{ML}})$. This implies that $\hat{\boldsymbol{X}}_{\text{ML}}$ attains the optimal objective $f^\star$ and it globally optimum $\boldsymbol{X}^\star_{\text{ML}} = \hat{\boldsymbol{X}}_{\text{ML}}$.

This simple argument inspires the verification approach V1 in Algorithm 1. Note that the test condition $d^\star = f(\hat{\boldsymbol{X}}_{\text{ML}})$ requires that there is strong duality for $d^\star = f^\star$ and that the determinant constraints (4) are inactive for $f^\star = f^\star_{\text{ML}}$, two conditions that appear to hold in practice.

Also this verification approach requires solving the SDP problem (11) to get $d^\star$. Even though this is a convex problem, in practice the bad scalability of current SDP solvers with increasing problem sizes limit the usability of Algorithm 1. Therefore, as in [1], [18], we present next an alternative verification technique V2 that avoids solving the SDP problem (11).

Faster optimality verification: A second important result is due to duality theory [22, Sec. 5] and states that, if there is strong duality, a primal optimal point $\boldsymbol{X}^\star$ is also a minimizer of the Lagrangian evaluated at the dual optimal point $\boldsymbol{\Lambda}^\star$:

$$d^\star = f^\star_{\text{ML}} \Rightarrow \boldsymbol{X}^\star = \arg\min_{\boldsymbol{X}} \text{L}(\boldsymbol{X}, \boldsymbol{\Lambda}^\star) \tag{13}$$

$$\overset{(9)}{=} \{\boldsymbol{X} : \boldsymbol{M}(\boldsymbol{\Lambda}^\star)\boldsymbol{X} = \boldsymbol{0}_{4n\times 3}\}. \tag{14}$$

Therefore, in our problem, *if there is* strong duality, the columns of any primal optimal solution $\boldsymbol{X}^\star$ must lie in the null space of the matrix $\boldsymbol{M}(\boldsymbol{\Lambda}^\star)$:

$$\boldsymbol{M}(\boldsymbol{\Lambda}^\star)\boldsymbol{X}^\star = \boldsymbol{0}_{4n\times 3}. \tag{15}$$

Input: Cost matrix $\boldsymbol{M}$, feasible solution $\hat{\boldsymbol{X}}_{\text{ML}}$
Output: Optimality certificate `isOpt`
compute $\hat{\boldsymbol{\Lambda}}$ (17);
if $\text{all}(\|\hat{\boldsymbol{\Lambda}}\|_A/n \leq \tau_A, \mu_{\min} \geq \tau_\mu, (\hat{f}-\hat{d})/\hat{f} \leq \epsilon_{rel})$ **then**
  set `isOpt = true`;
else
  set `isOpt = unknown`;
end
return `isOpt`

Algorithm V2: Optimality verification without SDP

This relation provides the following verification approach, analogous to the one proposed in [1]. Given a primal-feasible candidate $\hat{\boldsymbol{X}}_{\text{ML}}$, we make two assumptions:

1) Our candidate is globally optimal: $\hat{\boldsymbol{X}}_{\text{ML}} = \boldsymbol{X}^\star_{\text{ML}}$
2) There is strong duality: $d^\star = f^\star$

Under these assumptions, the linear relation (15) characterizes a dual candidate $\hat{\boldsymbol{\Lambda}}$ that is globally optimal ($\hat{\boldsymbol{\Lambda}} = \boldsymbol{\Lambda}^\star$). Thus, for our original two assumptions to hold we have two necessary conditions:

1) The dual candidate $\hat{\boldsymbol{\Lambda}}$ is dual-feasible, so $\hat{\boldsymbol{\Lambda}} = (\hat{\boldsymbol{\Lambda}})^\top$ and $\boldsymbol{M}(\hat{\boldsymbol{\Lambda}}) \succcurlyeq \boldsymbol{0}$.
2) The duality gap is zero, $f(\hat{\boldsymbol{X}}_{\text{ML}}) - d(\hat{\boldsymbol{\Lambda}}) = 0$.

But if these necessary conditions hold, we already have primal and dual feasible points $\hat{\boldsymbol{\Lambda}}$ and $\hat{\boldsymbol{X}}_{\text{ML}}$ that make the inequality chain (12) tight, forcing $d(\hat{\boldsymbol{\Lambda}}) = d^\star$ and $f^\star = f(\hat{\boldsymbol{X}}_{\text{ML}})$ so that these points must be globally optimal.

If the necessary conditions do not hold, either the primal candidate is not optimal ($\hat{\boldsymbol{X}}_{\text{ML}} \neq \boldsymbol{X}^\star_{\text{ML}}$), or there is no strong duality, or the MLE relaxation is not tight ($f^\star \leq f^\star_{\text{ML}}$). In that case we cannot state anything about the optimality of the candidate.

The reasoning above provides the verification pipeline summarized in Algorithm 2. In the algorithm the strict optimality conditions presented above are relaxed to the verification of certain thresholded parameters, in order to accommodate the presence of numerical errors. Namely, a candidate $\hat{\boldsymbol{\Lambda}}$ is considered dual feasible if

$$\begin{aligned} \|\hat{\boldsymbol{\Lambda}}\|_{\text{A}}/n \leq \tau_A, \quad \tau_A > 0 \\ \mu_{\min} \geq \tau_\mu, \quad \tau_\mu < 0 \end{aligned}$$

where $\|\hat{\boldsymbol{\Lambda}}\|_{\text{A}} = \|\hat{\boldsymbol{\Lambda}} - \hat{\boldsymbol{\Lambda}}^\top\|_F$ measures the antisymmetric component of the matrix and $\mu_{\min} \equiv \mu_{\min}(\boldsymbol{M}(\text{sym}(\hat{\boldsymbol{\Lambda}})))$ is the smallest eigenvalue of $\boldsymbol{M}(\text{sym}(\hat{\boldsymbol{\Lambda}}))$. The zero duality gap condition for optimality of the candidates has been rewritten in terms of the relative accuracy [22, Sec. 5.5.1.]:

$$\frac{\hat{f}-\hat{d}}{\hat{f}} \leq \epsilon_{\text{rel}}, \quad \epsilon_{\text{rel}} > 0.$$

Closed-form expression for dual candidate: Using the definition of $\boldsymbol{M}(\boldsymbol{\Lambda})$ the linear system (15) can be written

$$\begin{bmatrix} \hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{Q}} \\ \boldsymbol{0} \end{bmatrix} = \boldsymbol{M}\hat{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{M}_{\boldsymbol{Q},:} \\ \boldsymbol{M}_{\boldsymbol{T},:} \end{bmatrix} \hat{\boldsymbol{X}}. \tag{16}$$

where $\boldsymbol{M}_{\boldsymbol{Q},:} = \begin{bmatrix} \boldsymbol{M}_{\boldsymbol{QQ}} \big| \boldsymbol{M}_{\boldsymbol{QT}} \end{bmatrix}$ and $\boldsymbol{M}_{\boldsymbol{T},:} = \begin{bmatrix} \boldsymbol{M}_{\boldsymbol{TQ}} \big| \boldsymbol{M}_{\boldsymbol{TT}} \end{bmatrix}$. Thanks to the block-diagonal structure of $\hat{\boldsymbol{\Lambda}}$ each single Lagrange multiplier takes the simple expression

$$\hat{\boldsymbol{\Lambda}}_i = (\boldsymbol{e}_i \otimes \boldsymbol{I}_3)^\top \boldsymbol{M}\hat{\boldsymbol{X}}\boldsymbol{Q}_i^\top, \quad i = 1, \ldots, n, \tag{17}$$

in which matrix inversion is not required at any time, so the problem (15) in terms of $\boldsymbol{\Lambda}$ can be solved very efficiently.

V. Experiments

This section discusses the effectiveness and the performance of the proposed verification algorithms 1 and 2 to check the optimality of a given candidate solution.

We employ the same experimental framework as in the state-of-the-art reference [1] to assess the improvement of our formulation. We strongly encourage the reader to read [1, Sec. V] for a proper understanding of the evaluation framework.

Practical issues: The computation of the most negative eigenvalue for very large-scale problems is far from trivial. The usual numerical methods for this task are of iterative nature, and care must be taken in convergence issues. We used the function *eigs* in Matlab with seed -100, although multiple seeds could be tried for increased robustness. We also limited the numerical tolerance to 10^{-5} to reduce the computational burden.

A. Monte Carlo Analysis

We use the same simulation setup as in [1, Sec. V], namely a variable-size synthetic 3D grid as that in Fig. 1, which is traversed producing an odometric trajectory while loop closures are added with certain probability between nearby nodes. For a comprehensive description of this setup and its parameters, please see [1, Sec. V].

Using the same evaluation metrics as in [1], the obtained statistics where identical to those in [1, Fig. 2] (we skip its representation here for the sake of space). We confirmed again that the new reformulation does not affect the effectiveness of the verification techniques. and the same conclusions as in [1] stand for our framework: The *strong duality* assumption necessary for the proposed tools to work holds independently of the translational noise level or the problem size, although above certain rotation noise threshold this breaks and the duality gap gradually increases with the noise level.

We rather focus then on the comparison of our formulation and the reference one [1] in terms of performance. Recall the computational cost of the algorithm V1 is that of solving the SDP problem (11), whereas for V2 the heaviest step is that computing the most negative eigenvalue of $\boldsymbol{M}(\hat{\boldsymbol{\Lambda}})$. This justifies the results in Fig. 3 where, regardless of the formulation (ours or [1]), V1 is outperformed by V2 in terms of computational efficiency and scalability (with a difference of more than 4 orders of magnitude in time). Besides, the memory usage for SDP solvers scales bad with size as well.

On the other side, Fig. 3 also proves that our formulation excels in both cases (V1 and V2) wrt the reference alternatives [1]. This is clearly supported by the highly

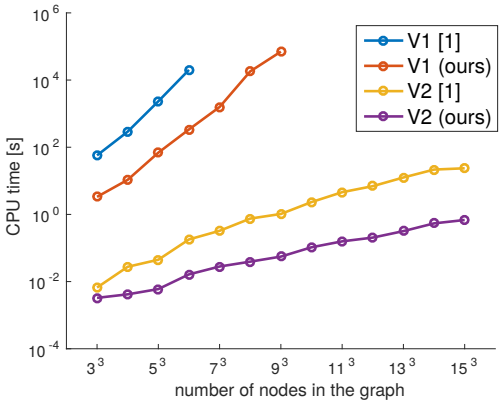


Fig. 3. Mean verification times for problems of increasing size using different approaches. The approach V1 involves solving a Semidefinite Program, whereas V2 reduces to the computation of an eigenvalue. In both cases the smaller size and higher sparsity of our trace-based formulation leads to a significant boost on performance over the reference work [1]. Please note the logarithmic scale.

beneficial effect that a smaller problem size and higher sparsity have in both the resolution of the SDP (11) (for V1) and the computation of $\mu_{\min}(\boldsymbol{M}(\hat{\boldsymbol{\Lambda}}))$ (for V2), reaching a performance improvement of up to 2 orders of magnitude for large-scale problems.

B. Large scale datasets

For the rest of this section we focus on the analysis of the performance in more challenging scenarios. Concretely, we will use the same datasets as [1], [7]. These comprise the *sphere*, *sphere-a*, *torus* and *cube* simulated datasets, as well as the *garage*, *cubicle* and *rim* real datasets. All these datasets consist of large scale scenarios, and we use the more scalable Algorithm 2 only. Since our PGO formulation is for isotropic noise, the covariances in some of the datasets were substituted by isotropic ones, following the indications provided by the authors in [1].

Table II presents the obtained results. Each row of the table corresponds to a different candidate solution we would like to verify. The "Sol" column points how we obtained the candidate. Both *Init.* and *Odom.* perform 10 Gauss-Newton (GN) iterations from the chordal [7] and odometric initialization, respectively. We also computed a groundtruth (*Gt.*) estimate for which we used chordal initialization and then applied a large number of GN iterations (1000) to force numerical convergence. Although this solution is not necessarily globally optimal, in practice it is [7], as proved here by the application of our verification approach.

Let us move to the analyzed parameters. The column "$\hat{f}$" shows the optimization objective value evaluated at the candidate, whereas the column "$\hat{d}$" corresponds to the candidate dual objective returned by Algorithm 2. These parameters are numerically identical to their counterparts in [1, Tab. I], proving again the theoretical equivalence of both formulations.

Next we have the verification parameters used for Algorithm 2. We set $\epsilon_{\text{rel}} = 10^{-2}$, $\tau_A = 10^{-2}$ and $\tau_\mu = -10^{-4}$ as the corresponding verification thresholds. For each row, we show in red the parameters that exceed the thresholds.

The column V2 shows the optimality certificate returned by Algorithm 2 according to the previous parameters. We see that most of the *Init.* estimates successfully converge to the global optimum after 10 GN iterations. We show in yellow those negative certificates where the minimum eigenvalue $\mu_{\min}$ was valid but either the relative duality gap or the symmetry condition failed. Our empirical observation is that those cases often correspond to suboptimal states due to insufficient GN iterations, since iterating further they converged to the global optimal. We also observe that estimates that converge to wrong minima usually produce negative certificates with very negative $\mu_{\min}$, *e.g.* the *sphere-a* or *torus* datasets initialized from odometry.

To get a more intuitive sense of the meaning of the verifications parameters, we added a column with the Absolute Trajectory Error (ATE) [23] of the candidate solution with respect to the corresponding *gt* estimate. We find that the ATE for all those estimates accepted as optimal by our verification approach are of the order of milimeters, which is consistent with what we would expect for a global optimum. On the other side, the ATE is clearly high for wrong minima (as expected) and moderate for those suboptimal estimates (in yellow) that will converge given enough extra GN iterations. The ATE value of the *sphere-a* chordal estimate proves that even though the duality gap is near zero and $\boldsymbol{M}(\boldsymbol{\Lambda})$ is positive semidefinite, the symmetry condition is still necessary for the candidate to be optimal.

Comparison with [1]: We applied the reference fast verification method in [1] to each estimate as well. The columns "t [ours]" and "t [1]" show the verification times (in seconds) for each approach. The conclusion is consistent with that obtained in the Monte Carlo simulation: Our formulation of the verification method yields a remarkable boost in performance, by more than one order of magnitude in most cases[2].

We also observed that, for the cases with zero duality gap, the minimum eigenvalue computed by our technique shows better numerical behavior. Namely, the computed eigenvalues for our approach converged to zero (within the specified tolerance 10^{-5}). However, in [1] these converged only to the range $(10^{-4}, 10^{-1})$.

As for the optimality certificate "V2", the use of more restrictive tests (*e.g.* $\epsilon_{\text{rel}} = 1\%$ vs. $\epsilon_{\text{rel}} = 20\%$ in [1], or the addition of the symmetry threshold τ_A) provokes that some cases classified as optimal in [1] are detected here as suboptimal. Another interesting case is the solution to the challenging *rim* dataset with chordal initialization, which is classified here as globally optimal in contrast to [1]. Our intuition is that the specially large dimensions of the matrix involved in this dataset make the eigenvalue computation more numerically unstable in [1], driving this to a wrong value that causes the optimality test to fail in their case.

[2]Some times obtained for [1] were unusually high, probably due to numerical issues with the function *eigs* and its use of random initialization.

Dataset	Sol.	$\hat{f}$	$\hat{d}$	Verification parameters			V2	ATE (m)	t (s)	
				$\lvert\hat{f}-\hat{d}\rvert/\hat{f}$	$\lVert\hat{\mathbf{\Lambda}}\rVert_A/n$	$\mu_{\min}(\hat{\mathbf{\Lambda}})$			[ours]	[1]
sphere	Gt.	$5.7594\cdot10^{2}$	$5.7594\cdot10^{2}$	$7.7\cdot10^{-12}$	$7.5\cdot10^{-9}$	$2.7\cdot10^{-6}$	✓	-	1.2	2309.1
$n=2500$	Init.	$5.7595\cdot10^{2}$	$5.7528\cdot10^{2}$	$1.2\cdot10^{-3}$	$2.5\cdot10^{-4}$	$2.7\cdot10^{-6}$	✓	$2.1\cdot10^{-3}$	1.2	2285.4
$m=4949$	Odom.	$5.8019\cdot10^{2}$	$4.3785\cdot10^{2}$	$2.5\cdot10^{-1}$	$4\cdot10^{-3}$	$2.7\cdot10^{-6}$	✗	$6.2\cdot10^{-2}$	1.3	2280.6
sphere-a	Gt.	$1.2485\cdot10^{6}$	$1.2485\cdot10^{6}$	$1.5\cdot10^{-11}$	$3.2\cdot10^{-3}$	$1.2\cdot10^{-7}$	✓	-	1.1	397.9
$n=2200$	Init.	$1.2486\cdot10^{6}$	$1.2486\cdot10^{6}$	$8.8\cdot10^{-7}$	$5.4\cdot10^{-2}$	$1.3\cdot10^{-7}$	✗	$7.4\cdot10^{-1}$	1.1	413.4
$m=8647$	Odom.	$3.0413\cdot10^{6}$	$3.04\cdot10^{6}$	$4.3\cdot10^{-4}$	5	$-9.6\cdot10^{1}$	✗	$7.4\cdot10^{1}$	0.5	4.0
torus	Gt.	$1.2114\cdot10^{4}$	$1.2114\cdot10^{4}$	$2.4\cdot10^{-12}$	$9.7\cdot10^{-8}$	$1.1\cdot10^{-8}$	✓	-	1.0	49.5
$n=5000$	Init.	$1.2114\cdot10^{4}$	$1.2114\cdot10^{4}$	$9.1\cdot10^{-6}$	$2\cdot10^{-3}$	$-4.7\cdot10^{-6}$	✓	$4.9\cdot10^{-3}$	0.9	49.5
$m=9048$	Odom.	$2.7666\cdot10^{4}$	$2.7588\cdot10^{4}$	$2.8\cdot10^{-3}$	$2.4\cdot10^{-2}$	$-1.2\cdot10^{2}$	✗	3.8	0.6	4.6
cube	Gt.	$4.216\cdot10^{4}$	$4.216\cdot10^{4}$	$7.8\cdot10^{-13}$	$2.9\cdot10^{-7}$	$6.8\cdot10^{-8}$	✓	-	7.6	2071.3
$n=8000$	Init.	$4.216\cdot10^{4}$	$4.216\cdot10^{4}$	$3.9\cdot10^{-6}$	$1.6\cdot10^{-3}$	$4.1\cdot10^{-8}$	✓	$1.6\cdot10^{-3}$	7.5	2017.8
$m=22236$	Odom.	$2.7465\cdot10^{5}$	$2.7372\cdot10^{5}$	$3.4\cdot10^{-3}$	$1.3\cdot10^{-1}$	$-1\cdot10^{2}$	✗	1.5	6.2	92.5
garage	Gt.	$6.2994\cdot10^{-1}$	$6.2994\cdot10^{-1}$	$4.1\cdot10^{-10}$	$9.3\cdot10^{-11}$	$2\cdot10^{-5}$	✓	-	0.7	14.0
$n=1661$	Init.	$6.2994\cdot10^{-1}$	$6.0426\cdot10^{-1}$	$4.1\cdot10^{-2}$	$1.2\cdot10^{-6}$	$2\cdot10^{-5}$	✗	$4.4\cdot10^{-2}$	0.9	14.4
$m=6275$	Odom.	$6.2997\cdot10^{-1}$	$3.5291\cdot10^{-1}$	$4.4\cdot10^{-1}$	$3.8\cdot10^{-6}$	$2\cdot10^{-5}$	✗	$1.4\cdot10^{-1}$	1.3	14.6
cubicle	Gt.	$6.2481\cdot10^{2}$	$6.2481\cdot10^{2}$	$8.3\cdot10^{-11}$	$2.3\cdot10^{-9}$	$7.5\cdot10^{-7}$	✓	-	1.5	20.0
$n=5750$	Init.	$6.2481\cdot10^{2}$	$6.2481\cdot10^{2}$	$4.7\cdot10^{-7}$	$2.9\cdot10^{-6}$	$7.5\cdot10^{-7}$	✓	$1.1\cdot10^{-5}$	1.5	28.8
$m=16869$	Odom.	$6.2484\cdot10^{2}$	$6.1625\cdot10^{2}$	$1.4\cdot10^{-2}$	$4.9\cdot10^{-4}$	$7.6\cdot10^{-7}$	✗	$1.9\cdot10^{-3}$	1.7	20.2
rim	Gt.	$1.235\cdot10^{4}$	$1.235\cdot10^{4}$	$1.6\cdot10^{-12}$	$4\cdot10^{-8}$	$1.4\cdot10^{-6}$	✓	-	3.0	10.8
$n=10195$	Init.	$1.235\cdot10^{4}$	$1.235\cdot10^{4}$	$4.2\cdot10^{-5}$	$3.9\cdot10^{-4}$	$1.4\cdot10^{-6}$	✓	$2\cdot10^{-3}$	4.2	11.0
$m=29743$	Odom.	$1.6985\cdot10^{4}$	$-2.7965\cdot10^{4}$	2.6	$1.3\cdot10^{-1}$	$-9.3\cdot10^{1}$	✗	1	1.3	11.4

TABLE II

VERIFICATION WITH ALGORITHM 2 ON LARGE-SCALE SLAM DATASETS.

VI. CONCLUSION

Inspired by the state-of-the-art work [1] on global optimality verification for 3D Pose Graph Optimization problems, we propose a novel formulation that substantially reduces the computational burden of the framework. The resulting techniques prove much faster while remaining equally effective.

These verification techniques are a great complement to standard SLAM iterative solvers and, with this improvement in efficiency, we expect to promote its usability in a broader range of applications and systems.

ACKNOWLEDGMENT

We would like to thank Luca Carlone for kindly sharing the code for [1] and for the support with related questions.

REFERENCES

[1] L. Carlone, D. M. Rosen, G. Calafiore, J. J. Leonard, and F. Dellaert, "Lagrangian duality in 3D SLAM: Verification techniques and optimal solutions," in *Intl. Conf. on Intelligent Robots and Systems (IROS)*, pp. 125–132, Sep 2015.

[2] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: A general framework for graph optimization," *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pp. 3607–3613, May 2011.

[3] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping using the bayes tree," *Intl. J. of Robotics Research*, 2011.

[4] D. M. Rosen, M. Kaess, and J. J. Leonard, "Rise: An incremental trust-region method for robust online sparse least-squares estimation," *IEEE Trans. Robotics*, vol. 30, no. 5, pp. 1091–1108, 2014.

[5] L. Carlone, R. Aragues, J. A. Castellanos, and B. Bona, "A fast and accurate approximation for planar pose graph optimization," *Intl. J. of Robotics Research*, 2014.

[6] L. Carlone and A. Censi, "From angular manifolds to the integer lattice: Guaranteed orientation estimation with application to pose graph optimization," *IEEE Trans. Robotics*, vol. 30, no. 2, 2014.

[7] L. Carlone, R. Tron, K. Daniilidis, and F. Dellaert, "Initialization techniques for 3D SLAM: a survey on rotation estimation and its use in pose graph optimization," in *Intl. Conf. on Robotics and Automation (ICRA)*, pp. 4597–4604, IEEE, 2015.

[8] E. Olson, J. Leonard, and S. Teller, "Fast iterative alignment of pose graphs with poor initial estimates," in *Intl. Conf. on Robotics and Automation (ICRA)*, pp. 2262–2269, IEEE, 2006.

[9] R. Tron, B. Afsari, and R. Vidal, "Intrinsic consensus on SO(3) with almost-global convergence," in *CDC*, pp. 2052–2058, 2012.

[10] S. Huang, Y. Lai, U. Frese, and G. Dissanayake, "How far is slam from a linear least squares problem?," in *Intl. Conf. on Intelligent Robots and Systems (IROS)*, pp. 3011–3016, IEEE, 2010.

[11] H. Wang, G. Hu, S. Huang, and G. Dissanayake, "On the structure of nonlinearities in pose graph slam.," in *Robotics: Science and Systems*, 2012.

[12] H. Wang, S. Huang, U. Frese, and G. Dissanayake, "The nonlinearity structure of point feature slam problems with spherical covariance matrices," *Automatica*, vol. 49, no. 10, pp. 3112–3119, 2013.

[13] L. Carlone, "A convergence analysis for pose graph optimization via gauss-newton methods," in *Intl. Conf. on Robotics and Automation (ICRA)*, pp. 965–972, IEEE, 2013.

[14] K. Khosoussi, S. Huang, and G. Dissanayake, "Novel insights into the impact of graph structure on slam," in *Intl. Conf. on Intelligent Robots and Systems (IROS)*, pp. 2707–2714, IEEE, 2014.

[15] R. Sim and N. Roy, "Global a-optimal robot exploration in slam," in *Intl. Conf. on Robotics and Automation (ICRA)*, IEEE, 2005.

[16] D. M. Rosen, C. DuHadway, and J. J. Leonard, "A convex relaxation for approximate global optimization in simultaneous localization and mapping," in *Intl. Conf. on Robotics and Automation (ICRA)*, pp. 5822–5829, IEEE, 2015.

[17] L. Carlone and F. Dellaert, "Duality-based verification techniques for 2D SLAM," in *Intl. Conf. on Robotics and Automation (ICRA)*, 2015.

[18] L. Carlone, G. C. Calafiore, C. Tommolillo, and F. Dellaert, "Planar Pose Graph Optimization: Duality, Optimal Solutions, and Verification," *IEEE Trans. Robotics*, vol. 32, no. 3, pp. 545–565, 2016.

[19] R. Hartley, J. Trumpf, Y. Dai, and H. Li, "Rotation averaging," *Intl. J. of Computer Vision*, vol. 103, no. 3, pp. 267–305, 2013.
[20] K. V. Mardia and P. E. Jupp, *Directional statistics*, vol. 494. John Wiley & Sons, 2009.
[21] R. Tron, D. M. Rosen, and L. Carlone, "On the Inclusion of Determinant Constraints in Lagrangian Duality for 3D SLAM,"
[22] S. Boyd and L. Vandenberghe, *Convex optimization*. 2004.
[23] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Intl. Conf. on Intelligent Robots and Systems (IROS)*, Oct. 2012.