Duality-based Verification Techniques for 2D SLAM

Luca Carlone and Frank Dellaert

***Abstract*— While iterative optimization techniques for Simultaneous Localization and Mapping (SLAM) are now very efficient and widely used, none of them can guarantee global convergence to the maximum likelihood estimate. Local convergence usually implies artifacts in map reconstruction and large localization errors, hence it is very undesirable for applications in which accuracy and safety are of paramount importance. We provide a technique to *verify* if a given 2D SLAM solution is globally optimal. The insight is that, while computing the optimal solution is hard in general, *duality* theory provides tools to compute tight bounds on the optimal cost, via convex programming. These bounds can be used to evaluate the quality of a SLAM solution, hence providing a "sanity check" for state-of-the-art incremental and batch solvers. Experimental results show that our technique successfully identifies wrong estimates (i.e., local minima) in large-scale SLAM scenarios. This work, together with [1], represents a step towards the objective of having SLAM techniques with guaranteed performance, that can be used in safety-critical applications.**

I. Introduction

Simultaneous Localization and Mapping (SLAM) consists in the concurrent estimation of the position of a mobile robot, and the construction of a model of the surrounding environment. SLAM is now a well studied research topic, and the corresponding algorithms are steadily permeating from academic research to industrial applications [2], [3]. Application scenarios include intelligent transportation, search and rescue, and military operation in hostile environment. In those scenarios, preserving high accuracy is critical, as an incorrect map may put human life at risk.

In recent years, optimization-based approaches have become the leading paradigm for SLAM. These approaches compute the SLAM solution by minimizing a nonlinear cost, whose global minimum is the *maximum likelihood estimate* (*maximum-a-posteriori* estimate in presence of priors):

$$f^\star = \min_{\boldsymbol{x}} f(\boldsymbol{x}) \tag{1}$$

where the variable $\boldsymbol{x}$ includes the quantities to be estimated (e.g., robot positions and orientations), and $f(\cdot)$ describes the negative log-likelihood of the measurements. The success of these techniques stems from three main reasons. First, the approach is *general* and one can easily model different sensor measurements [4] and include priors. Second, they are very *fast* in practice, as they exploit problem structure (i.e., *sparsity*) [5], and can operate incrementally when new data is received [6]. Third, they are *robust*, as they can incorporate outlier rejection mechanisms to handle spurious measurements (see [7] and the references therein).

L. Carlone and F. Dellaert are with the School of Interactive Computing, College of Computing, Georgia Institute of Technology, Atlanta, GA, USA, luca.carlone@gatech.edu, frank@cc.gatech.edu.

This work was partially funded by the National Science Foundation Award 11115678 "RI: Small: Ultra-Sparsifiers for Fast and Scalable Mapping and 3D Reconstruction on Mobile Robots".

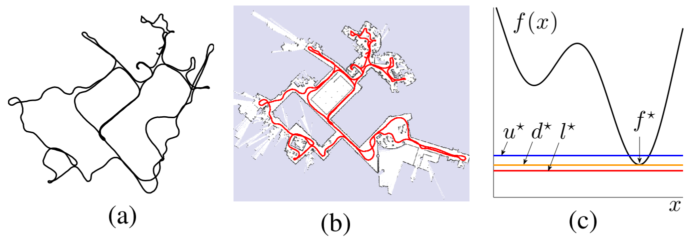


Fig. 1. CSAIL dataset: trajectory estimates from g2o using two different initial guesses. (a) Local minimum. (b) Global minimum, superimposed on the occupancy grid map of the scenario. (c) In this work we use duality to compute bounds $(d^\star, l^\star, u^\star)$ for the global minimum $f^\star$ (illustrative example in the figure): these can be used to verify if a given solution attains the optimal cost or was stuck in a local minimum.

Standard solvers (e.g., gtsam [6] or g2o [8]) minimize the cost $f(\boldsymbol{x})$ iteratively, by refining an initial guess. For instance, Fig. 1a shows the estimated trajectory produced by g2o on the CSAIL dataset [9], using a suitable initial guess.

Typically, after optimization, a human operator evaluates the estimated trajectory from visual inspection, to rule out the possibility that the algorithm converged to a local minimum[1]; local convergence implies artifacts in map reconstruction and large localization errors, hence it is undesirable in practice.

Here we argue that visual inspection cannot be a valid criterion for future robotics applications. First or all, future autonomous robots cannot rely on human supervision for basic tasks, as localization and mapping. Second, in many cases, visual inspection can be deceptive. For instance, the trajectory estimated in Fig. 1a looks reasonable (and the cost attained by the g2o solution is reasonably small, $f(\hat{\boldsymbol{x}}) = 1.9 \cdot 10^1$). However, by comparing it with the actual map of the scenario (Fig. 1b), one realizes that Fig. 1a corresponds to a local minimum (the actual optimal cost is $f^\star = 1.07 \cdot 10^{-1}$). For this reason we look for a grounded approach to autonomously and reliably assess global convergence (Fig. 1c).

Related work tackles local convergence using different strategies. A first set of approaches aims at improving global convergence by adopting techniques with larger basin of convergence or different parameterizations. Examples of this effort are the work from Olson *et al.* [10], Grisetti *et al.* [11], Rosen *et al.* [12], and Tron *et al.* [13]. A second research line proposes to improve convergence by computing an accurate initial guess for iterative techniques. Initialization techniques include [14], [15], [16], [1]. A third line of research involves a theoretical analysis of the optimization problem. Huang *et al.* [17] identify the accumulation of orientation errors as the main cause for divergence of iterative solvers.

[1] We use the term "local minimum" to denote a stationary point of the cost which does not attain the optimal objective.

Wang *et al.* [18] and Huang *et al.* [19] investigate the number of local minima in problems with a small number of poses or when using map-joining techniques. Knuth and Barooah [20] study error accumulation in pose graphs, which is relevant to quantify the quality of the odometric initial guess for optimization. Carlone [21] shows that global convergence is influenced by the information content of the measurements, inter-nodal distances, and structure of the underlying graph. Along this line, Khosoussi *et al.* [22] study the relation between graph structure and quality of the SLAM estimate, discussing the role of node degree, number of spanning trees in the graph, and algebraic connectivity.

The present paper bridges theoretical analysis and practical algorithms by proposing *verification* techniques for SLAM. Rather than analyzing the properties of the optimization problem, we try to answer a fundamental question: given an estimate $\hat{\boldsymbol{x}}$ (say, a solution returned by a state-of-the-art iterative solver), does this estimate correspond to a global optimum of the cost function $f(\boldsymbol{x})$? If the answer is positive, we can trust our estimate; if the answer is negative, we must resort to some *recovery* technique, as the given estimate is not accurate, and it is not safe to use it.

Duality theory in optimization [23] offers well studied tools to obtain a lower bound on the optimal value of an optimization problem. However the standard SLAM formulation is not directly amenable to apply duality.

Our first contribution is to show that using the *chordal* distance [24] in SLAM and choosing a suitable parametrization for rotations allow writing pose graph optimization as a quadratic minimization with quadratic equality constraints; the latter is well suited for duality and can leverage well established results from the optimization community [23].

Therefore, as a second contribution, we exploit duality in SLAM and we obtain a lower bound $d^\star$ on the optimal value $f^\star$ of the cost function $f(\boldsymbol{x})$. This first bound can be computed via semidefinite programming (SDP) and is shown to be tight ($d^\star = f^\star$) in the noise regimes of practical applications. This means that if the candidate solution $\hat{\boldsymbol{x}}$ produces a cost $f(\hat{\boldsymbol{x}})$ that is larger than $d^\star$, it corresponds to a local minimum.

While SDPs are convex, they do not scale to large problems and are currently slow for realistic applications. For this reason, as a third contribution, we develop other two bounds: a lower bound $l^\star$ and an upper bound $u^\star$. These have the advantage of being faster to compute. Therefore, given the candidate solution $\hat{\boldsymbol{x}}$, if the cost $f(\hat{\boldsymbol{x}})$ is outside the interval $[l^\star, u^\star]$, then the estimate is a local minimum. In practice, the interval $[l^\star, u^\star]$ is small, and our technique is able to discern wrong solutions in all tested scenarios.

Our verification techniques can be integrated seamlessly in standard SLAM pipelines, and can be used as a "sanity check" for state-of-the-art incremental and batch solvers. We believe that this contribution represents a step towards the objective of designing SLAM techniques with guaranteed performance, that can be used in safety-critical applications.

Note that the technique [1] already has global convergence guarantees. However, the results in [1] are restricted to the orientation estimates; moreover, [1] performs probabilistic inference, hence it assumes that measurement covariances are reliable. In practice, measurement covariance are only rough estimates, and for this reason, in this paper we are agnostic about the generative model of measurement noise.

The paper is organized as follows. Section II formulates the optimization problem to be solved in SLAM. Section III shows how to rewrite the original problem as a quadratic program with quadratic equality constraints. Section IV shows how to use *duality* to compute the lower bounds $d^\star$ and $l^\star$. Section V shows how to compute the upper bound $u^\star$. Section VI discusses practical use of our findings. Section VII demonstrates our technique in simulated and real datasets. Section VIII provides concluding remarks.

II. Graph Optimization with Chordal Distance

In this section we propose a formulation of optimization-based SLAM that uses the *chordal* distance [24] as a metric for $\mathrm{SO}(2)$. This enables to write the optimization problem in a form that is well suited to apply duality theory. Moreover, Remark 1 in this section shows that minimizing the *chordal* distance is practically equivalent to minimizing the angular distance, which is commonly used in related work.

We consider pose-based SLAM in which we have to estimate n robot poses from m relative pose measurements (*pose graph optimization*). The problem can be visualized as a *graph*, where a pose is attached to each node, and each edge corresponds to a measurement. Each relative pose measurement (between two poses i and j) includes the relative rotation $\boldsymbol{R}_{ij}$ and the relative position $\boldsymbol{\Delta}_{ij}$ between the two poses. Ideally, the measurements should satisfy:

$$\boldsymbol{\Delta}_{ij} = \boldsymbol{R}_i^{\mathsf{T}} \left(\boldsymbol{p}_j - \boldsymbol{p}_i\right), \qquad \boldsymbol{R}_{ij} = \boldsymbol{R}_i^{\mathsf{T}} \boldsymbol{R}_j \;, \tag{2}$$

where $\boldsymbol{R}_i \in \mathrm{SO}(2)$ and $\boldsymbol{p}_i \in \mathbb{R}^2$ are the rotation and the position of node i. However, in presence of noise these relations are not exact and one looks for a set of positions $\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$ and rotations $\{\boldsymbol{R}_1, \ldots, \boldsymbol{R}_n\}$ that minimizes the mismatch with respect to the measurements:

$$f^\star \doteq \min_{\substack{\{\boldsymbol{p}_i\} \in \mathbb{R}^2, \\ \{\boldsymbol{R}_i\} \in \mathrm{SO}(2)}} \sum_{ij} \left(\|\boldsymbol{p}_j - \boldsymbol{p}_i - \boldsymbol{R}_i \boldsymbol{\Delta}_{ij}\|^2 + \frac{1}{2} \|\boldsymbol{R}_i \boldsymbol{R}_{ij} - \boldsymbol{R}_j\|_{\mathrm{F}}^2 \right) \tag{3}$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm of a matrix[2]. The term $\|\boldsymbol{R}_i \boldsymbol{R}_{ij} - \boldsymbol{R}_j\|_{\mathrm{F}}^2$ is the (squared) *chordal distance* between the rotation matrices $\boldsymbol{R}_i \boldsymbol{R}_{ij}$ and $\boldsymbol{R}_j$ [24].

In order to to compute an estimate for robot positions $\{\boldsymbol{p}_i\}$ and rotations $\{\boldsymbol{R}_i\}$, one has to solve problem (3). Before moving on, the following remark ensures that (3) is equivalent to other formulations in related work.

Remark 1 (Chordal distance): We conveniently use the chordal distance in our formulation as it enables to reformulate the problem as in Section III. A more standard cost function would use the squared *angular distance* [1]:

$$\mathbf{dist}_{ij}^{\theta} \doteq \tilde{\theta}^2, \quad \text{with} \quad \tilde{\theta} = \|\mathrm{Log}\left(\boldsymbol{R}_{ij}^{\mathsf{T}} \boldsymbol{R}_i^{\mathsf{T}} \boldsymbol{R}_j\right)\|, \tag{4}$$

[2]The (squared) Frobenius norm of a matrix $\boldsymbol{A} \in \mathbb{R}^{p \times q}$ is defined as $\|\boldsymbol{A}\|_{\mathrm{F}}^2 \doteq \sum_{i=1}^{p} \sum_{j=1}^{q} |a_{ij}|^2$, where a_{ij} are the entries of $\boldsymbol{A}$.

where $\mathrm{Log}(\cdot)$ is the logarithmic map for $\mathrm{SO}(2)$, and $\tilde{\theta}$ is the rotation angle corresponding to the rotation $\boldsymbol{R}_{ij}^\mathsf{T}\boldsymbol{R}_i^\mathsf{T}\boldsymbol{R}_j$.

To clarify the relation between chordal and angular distance, let us develop the chordal distance as follows:

$$\mathbf{dist}_{ij}^{\text{cord}} \doteq \|\boldsymbol{R}_i\boldsymbol{R}_{ij} - \boldsymbol{R}_j\|_\mathsf{F}^2 \quad = \quad \left\|\mathbf{I} - \boldsymbol{R}_{ij}^\mathsf{T}\boldsymbol{R}_i^\mathsf{T}\boldsymbol{R}_j\right\|_\mathsf{F}^2 = \\ 2\left(\sin^2(\tilde{\theta}) + (1-\cos(\tilde{\theta}))^2\right) \quad = \quad 8\sin^2(\tilde{\theta}/2), \tag{5}$$

which also holds for 3D rotations [24]. Eqs. (4)-(5) clarify that both metrics minimize some function of the error angle $\tilde{\theta}$. The residual error for a single measurement is usually small and the following first-order approximation holds:

$$\mathbf{dist}_{ij}^{\text{cord}} = 8\sin^2(\tilde{\theta}/2) \approx 8(\tilde{\theta}/2)^2 = 2\mathbf{dist}_{ij}^{\theta}, \tag{6}$$

meaning that for small residual errors the metrics differ by a constant. In order to compensate this constant we introduced the $\frac{1}{2}$ in front of the chordal distance in (3), such that, for small residual errors, (3) is essentially the same as the standard formulation with angular distance. □

III. Rewriting Problem (3) as a quadratic program with quadratic equality constraints

In this paper we consider the case in which we are given a candidate solution $\hat{\boldsymbol{x}} = \{(\hat{\boldsymbol{p}}_i, \hat{\boldsymbol{R}}_i) : i = 1, \ldots, n\}$ for (3), and we have to check if it is globally optimal. If we knew $f^\star$ this would be easy: if $f(\hat{\boldsymbol{x}}) = f^\star$ then $\hat{\boldsymbol{x}}$ is optimal; unfortunately, $f^\star$ is unknown. Our contribution is to show that we can compute close proxies of $f^\star$ using duality theory, without actually solving (3). To attain this goal, we reformulate (3) as a quadratic program with quadratic equality constraints (eq. (16)): this is a well studied problem in optimization and is directly amenable to apply duality theory.

In order to reformulate Problem (3) as a quadratic program with quadratic equality constraints we have to choose a suitable parametrization for the rotations. So far we attributed to each node a rotation matrix $\boldsymbol{R}_i \in \mathrm{SO}(2)$. In [1] we parametrized each planar rotation with an angle $\theta_i \in (-\pi, +\pi]$. While this is a minimal representation, it requires dealing with the wraparound problem, which is hard to tackle [1]. Here we parametrize the rotation of node i with the cosine and sine of the rotation angle:

$$\boldsymbol{r}_i \doteq [\cos(\theta_i) \;\; \sin(\theta_i)]^\mathsf{T} \doteq [c_i \;\; s_i]^\mathsf{T}. \tag{7}$$

For the vector $\boldsymbol{r}_i$ to represent a rotation, it must hold:

$$\sin^2(\theta_i) + \cos^2(\theta_i) = 1, \quad \text{i.e.,} \quad \|\boldsymbol{r}_i\|^2 = 1. \tag{8}$$

Let us rewrite the cost function (3) using our new parametrization. First, we note that each relative position measurement $\boldsymbol{\Delta}_{ij} \in \mathbb{R}^2$ includes two components, i.e., $\boldsymbol{\Delta}_{ij} = [\Delta_{ij}^x \;\; \Delta_{ij}^y]^\mathsf{T}$. Therefore, we can develop:

$$\boldsymbol{R}_i\boldsymbol{\Delta}_{ij} = \begin{bmatrix} c_i\Delta_{ij}^x - s_i\Delta_{ij}^y \\ c_i\Delta_{ij}^y + s_i\Delta_{ij}^x \end{bmatrix} = \begin{bmatrix} \Delta_{ij}^x & -\Delta_{ij}^y \\ \Delta_{ij}^y & \Delta_{ij}^x \end{bmatrix} \boldsymbol{r}_i \doteq \boldsymbol{D}_{ij}\boldsymbol{r}_i, \tag{9}$$

where the matrix $\boldsymbol{D}_{ij} \in \mathbb{R}^{2\times 2}$ is known as it only includes measurements. Moreover, we can write each relative rotation measurement $\boldsymbol{R}_{ij}$ as $\boldsymbol{R}_{ij} = \begin{bmatrix} c_{ij} & -s_{ij} \\ s_{ij} & c_{ij} \end{bmatrix}$, which implies

$$\boldsymbol{R}_i\boldsymbol{R}_{ij} - \boldsymbol{R}_j = \begin{bmatrix} c_{ij}c_i - s_{ij}s_i - c_j & -s_{ij}c_i - c_{ij}s_i + s_j \\ c_{ij}s_i + s_{ij}c_i - s_j & -s_{ij}s_i + c_{ij}c_i - c_j \end{bmatrix}.$$

Noting that the diagonal entries of $\boldsymbol{R}_i\boldsymbol{R}_{ij} - \boldsymbol{R}_j$ are identical and the off diagonal entries only differ by a minus sign, the Frobenius norm in (3) becomes:

$$\|\boldsymbol{R}_i\boldsymbol{R}_{ij} - \boldsymbol{R}_j\|_\mathsf{F}^2 \quad = \quad 2|c_{ij}c_i - s_{ij}s_i - c_j|^2 + \\ 2|c_{ij}s_i + s_{ij}c_i - s_j|^2 \quad = \quad 2\|\boldsymbol{R}_{ij}\boldsymbol{r}_i - \boldsymbol{r}_j\|^2. \tag{10}$$

Using (9) and (10), and recalling that a valid rotation should satisfy (8), Problem (3) becomes

$$\min_{\{\boldsymbol{p}_i\},\{\boldsymbol{r}_i\}} \sum_{ij} \left(\|\boldsymbol{p}_j - \boldsymbol{p}_i - \boldsymbol{D}_{ij}\boldsymbol{r}_i\|^2 + \|\boldsymbol{R}_{ij}\boldsymbol{r}_i - \boldsymbol{r}_j\|^2\right) \\ \text{subject to} \quad \boldsymbol{r}_i^\mathsf{T}\boldsymbol{r}_i = 1, \quad i = 1, \ldots, n \tag{11}$$

In order to rewrite the previous cost in matrix form we stack the unknown positions and rotation into two vectors:

$$\boldsymbol{p} = [\boldsymbol{p}_1^\mathsf{T} \; \ldots \; \boldsymbol{p}_n^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{2n}, \qquad \boldsymbol{r} = [\boldsymbol{r}_1^\mathsf{T} \; \ldots \; \boldsymbol{r}_n^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{2n}. \tag{12}$$

This allows writing (11) as

$$f^\star = \min_{\boldsymbol{p},\boldsymbol{r}} \left\|\boldsymbol{A}^\mathsf{T}\boldsymbol{p} - \boldsymbol{D}\boldsymbol{r}\right\|^2 + \|\boldsymbol{B}\boldsymbol{r}\|^2 \\ \text{subject to} \quad \boldsymbol{r}^\mathsf{T}\boldsymbol{N}_i\boldsymbol{r} = 1, \quad i = 1, \ldots, n, \tag{13}$$

for suitable (known) matrices $\boldsymbol{A}^\mathsf{T} \in \mathbb{R}^{2m\times 2n}$, $\boldsymbol{D} \in \mathbb{R}^{2m\times 2n}$, $\boldsymbol{B} \in \mathbb{R}^{2m\times 2n}$, $\boldsymbol{N}_i \in \mathbb{R}^{2n\times 2n}$ (we keep the transpose on $\boldsymbol{A}^\mathsf{T}$ since $\boldsymbol{A}$ is the *incidence matrix* of the underlying graph [15]). $\boldsymbol{N}_i$ is a sparse diagonal matrix such that $\boldsymbol{r}^\mathsf{T}\boldsymbol{N}_i\boldsymbol{r} = \boldsymbol{r}_i^\mathsf{T}\boldsymbol{r}_i$.

Since we only applied a re-parametrization, problem (13) is equivalent to the original problem (3), meaning that they have the same optimal value $f^\star$.

Problem (13) has a quadratic objective (involving robot positions $\boldsymbol{p}$ and robot orientations $\boldsymbol{r}$) and quadratic equality constraints. In the following sub-section we reduce the dimension of the optimization problem by eliminating the position vector $\boldsymbol{p}$ via Schur complement; this allows obtaining an optimization problem in the sole rotations. Then, in Sections IV-V we show how to bound the optimal cost $f^\star$.

A. Eliminating robot positions via Schur complement

For every fixed choice of robot orientations $\boldsymbol{r}$, the optimal translations can be computed in closed form:

$$\boldsymbol{p}^\star = (\boldsymbol{A}\boldsymbol{A}^\mathsf{T})^\dagger\boldsymbol{A}\boldsymbol{D}\boldsymbol{r} \tag{14}$$

since $\boldsymbol{p}$ appears linearly in the residual errors and the position vector is unconstrained[3]. Therefore, we can substitute (14) back into (13), so to eliminate the variable $\boldsymbol{p}$ and reduce the dimension of our optimization problem:

$$\min_{\boldsymbol{r}} \left\|(\boldsymbol{A}^\mathsf{T}(\boldsymbol{A}\boldsymbol{A}^\mathsf{T})^\dagger\boldsymbol{A} - \mathbf{I}_{2m})\boldsymbol{D}\boldsymbol{r}\right\|^2 + \|\boldsymbol{B}\boldsymbol{r}\|^2 \\ \text{subject to} \quad \boldsymbol{r}^\mathsf{T}\boldsymbol{N}_i\boldsymbol{r} = 1, \quad i = 1, \ldots, n \tag{15}$$

[3]The pseudo inverse in eq. (14) becomes an inverse as soon as the position of one node in the pose graph is fixed [16].

where $\mathbf{I}_{2m}$ is the identity matrix of size $2m$. Stacking the matrices $(\boldsymbol{A}^{\mathsf{T}}(\boldsymbol{A}\boldsymbol{A}^{\mathsf{T}})^{\dagger}\boldsymbol{A}-\mathbf{I}_{2m})\boldsymbol{D}$ and $\boldsymbol{B}$ into a single matrix $\boldsymbol{M} \in \mathbb{R}^{4m\times 2n}$, problem (15) becomes:

$$f^{\star} = \min_{\boldsymbol{r}} \quad \|\boldsymbol{M}\boldsymbol{r}\|^2 \quad \text{subject to} \quad \boldsymbol{r}^{\mathsf{T}}\boldsymbol{N}_i\boldsymbol{r} = 1, \quad i = 1,\ldots,n \tag{16}$$

Problem (16) is now a quadratic problem (in the sole rotations $\boldsymbol{r}$) with quadratic equality constraints. Problem (16) is equivalent to the original problem (3), meaning that the two problems share the same optimal objective (we only eliminated some of the variables) and the solution of (3) is uniquely determined by the solution of (16) via (14).

Later we refer to (16) as the *primal* problem, whose optimal value is $f^{\star}$. Problem (16) is still hard to solve, as equality quadratic constraints are nonconvex.

IV. Lower bounds for $f^{\star}$

Despite the non-convexity of (16), the key advantage of our reformulation is that (16) resembles well studied problems in optimization. For instance, (16) is a formulation of the *two-way partitioning* problem [23], in which one minimizes a (homogeneous) least-squares objective subject to unit norm constraints. Moreover, variants of the problem have been studied in optimization literature, including quadratic programming with one [25] and two quadratic equalities [26], [27]. The same insight is used in [28] to force the unit norm of quaternions when estimating rotations.

A common denominator of these techniques is the use of *duality*. The idea is simple: rather than solving the original (primal) problem (16), which is hard, one solves the *dual* problem, which is always convex. The solution of the dual problem provides a lower bound on the optimal value of the original problem and in some cases this lower bound is tight [23], i.e., the optimal dual cost coincides with $f^{\star}$.

A. The dual problem

In this section we exploit *duality* to obtain a lower bound $d^{\star}$ on the optimal value $f^{\star}$ of (16). Experimental evidence (Section VII) suggests that this bound is tight ($d^{\star} = f^{\star}$) in common SLAM problems, hence $d^{\star}$ is a very good indicator of the cost we should expect from an optimal estimate.

Let us define the *Lagrangian* of the primal problem (16):

$$\mathcal{L}(\boldsymbol{r},\boldsymbol{\lambda}) = \|\boldsymbol{M}\boldsymbol{r}\|^2 + \sum_{i=1}^{n}\lambda_i(1-\boldsymbol{r}^{\mathsf{T}}\boldsymbol{N}_i\boldsymbol{r}), \tag{17}$$

where $\lambda_i \in \mathbb{R}$, $i = 1,\ldots,n$, are the *Lagrange multipliers* (or *dual variables*) associated to problem (16). Roughly speaking, the Lagrangian (17) transforms the hard constraints in (16) into penalty terms, in which the amount of penalty is controlled by the Lagrange multipliers. Developing the squared norm in (17) and rearranging the terms we get

$$\mathcal{L}(\boldsymbol{r},\boldsymbol{\lambda}) = \boldsymbol{r}^{\mathsf{T}}\left(\boldsymbol{M}^{\mathsf{T}}\boldsymbol{M} - \sum_{i=1}^{n}\lambda_i\boldsymbol{N}_i\right)\boldsymbol{r} + \sum_{i=1}^{n}\lambda_i \tag{18}$$

The *dual function* is then defined as the infimum of the Lagrangian with respect to $\boldsymbol{r}$:

$$d(\boldsymbol{\lambda}) = \inf_{\boldsymbol{r}}\mathcal{L}(\boldsymbol{r},\boldsymbol{\lambda}) = \inf_{\boldsymbol{r}}\left(\boldsymbol{r}^{\mathsf{T}}\left(\boldsymbol{M}^{\mathsf{T}}\boldsymbol{M}-\sum_{i=1}^{n}\lambda_i\boldsymbol{N}_i\right)\boldsymbol{r}+\sum_{i=1}^{n}\lambda_i\right) \tag{19}$$

The dual function has an interesting property: $d(\boldsymbol{\lambda}) \leq f^{\star}$ for all choices of $\boldsymbol{\lambda}$. This motivates the definition of the *dual problem*, which simply looks for the $\boldsymbol{\lambda}$ that makes this lower bound as tight as possible:

$$d^{\star} = \max_{\boldsymbol{\lambda}}\ d(\boldsymbol{\lambda}) = \max_{\boldsymbol{\lambda}}\ \inf_{\boldsymbol{r}}\mathcal{L}(\boldsymbol{r},\boldsymbol{\lambda}) \tag{20}$$

The importance of the dual problem in optimization is twofold [23]. First, it holds:

$$d^{\star} \leq f^{\star} \tag{21}$$

meaning that the optimal objective of the dual problem provides a lower bound on the objective $f^{\star}$ of the primal problem (16); this property is called *weak duality*. For particular problems, the inequality (21) becomes an equality, and in those cases we say that *strong duality* holds [23].

Second, the dual problem (20) is always convex in $\boldsymbol{\lambda}$, regardless the convexity properties of the primal problem.

In the following section we provide an explicit expression for (20) and show how to compute $d^{\star}$.

B. Solving the dual problem

The solution of the dual problem (20) has been studied in the context of the two-way partitioning problem [23]. Here we recall it, adapting it to our SLAM setup.

We first observe that, if $\boldsymbol{M}^{\mathsf{T}}\boldsymbol{M} - \sum_{i=1}^{n}\lambda_i\boldsymbol{N}_i$ in (19) is non-positive definite, then the infimum drifts to minus infinity. Since the dual problem (20) looks for a maximum with respect to $\boldsymbol{\lambda}$, this case is not of interest and, as in [23], we limit the search to values of λ_i that make $\boldsymbol{M}^{\mathsf{T}}\boldsymbol{M} - \sum_{i=1}^{n}\lambda_i\boldsymbol{N}_i \succeq 0$, where $\succeq$ denotes positive definiteness.

If $\boldsymbol{M}^{\mathsf{T}}\boldsymbol{M} - \sum_{i=1}^{n}\lambda_i\boldsymbol{N}_i \succeq 0$, the minimum of the quadratic term in (19) is zero, and the dual function simplifies to $d(\boldsymbol{\lambda}) = \sum_{i=1}^{n}\lambda_i$. This allows writing (20) as:

$$d^{\star} = \max_{\boldsymbol{\lambda}} \quad \sum_{i=1}^{n}\lambda_i \quad \text{subject to} \quad \boldsymbol{M}^{\mathsf{T}}\boldsymbol{M} - \sum_{i=1}^{n}\lambda_i\boldsymbol{N}_i \succeq 0 \tag{22}$$

Problem (22) is an SDP (semidefinite program) and it is convex, hence can be solved globally using standard solvers.

C. A fast(er) lower bound

Convex programming has polynomial-time complexity. The solution of the SDP (22) can be computed quickly for small problems. However, solving the SDP can be still computationally expensive for large-scale problems, and this is the case in SLAM, where it is not infrequent to have problems with tens of thousands poses.

In this section we show how to compute a cheaper lower bound. The basic idea is to substitute the constraint $\boldsymbol{M}^{\mathsf{T}}\boldsymbol{M} - \sum_{i=1}^{n}\lambda_i\boldsymbol{N}_i \succeq 0$ with a simpler inequality that involves $\boldsymbol{\lambda}$.

Let us define $\boldsymbol{H} \doteq \boldsymbol{M}^\mathsf{T}\boldsymbol{M} - \sum_{i=1}^{n} \lambda_i \boldsymbol{N}_i$ and let us call μ_H the smallest eigenvalue of $\boldsymbol{H}$. From linear algebra, we know that the condition $\boldsymbol{H} \succeq 0$ is equivalent to $\mu_H \geq 0$. Since $\boldsymbol{H}$ is obtained as sum of two matrices ($\boldsymbol{M}^\mathsf{T}\boldsymbol{M}$ and $-\sum_{i=1}^{n} \lambda_i \boldsymbol{N}_i$), the Weyl inequality [29] guarantees that the smallest eigenvalue of $\boldsymbol{H}$ is larger than the sum of the smallest eigenvalues of the matrices $\boldsymbol{M}^\mathsf{T}\boldsymbol{M}$ and $-\sum_{i=1}^{n} \lambda_i \boldsymbol{N}_i$. More formally, if we call μ_M and μ_λ the smallest eigenvalues of $\boldsymbol{M}^\mathsf{T}\boldsymbol{M}$, and $-\sum_{i=1}^{n} \lambda_i \boldsymbol{N}_i$, respectively, then

$$\mu_H \geq \mu_M + \mu_\lambda. \tag{23}$$

Now the interesting observation is that the matrix $\boldsymbol{M}^\mathsf{T}\boldsymbol{M}$ is known (hence we can compute μ_M), and the matrix $-\sum_{i=1}^{n} \lambda_i \boldsymbol{N}_i$ is diagonal by construction (with entries $-\lambda_i$ on the diagonal), hence $\mu_\lambda = \min(-\boldsymbol{\lambda}) = -\max(\boldsymbol{\lambda})$, where $\min(\cdot)$ and $\max(\cdot)$ denote the minimum and the maximum entry of a vector, respectively. Therefore (23) becomes:

$$\mu_H \geq \mu_M - \max(\boldsymbol{\lambda}). \tag{24}$$

Substituting the condition $\boldsymbol{H} \succeq 0$ with $\mu_H \geq 0$, and using (24), problem (22) can be relaxed to:

$$l^\star = \max_{\boldsymbol{\lambda}} \sum_{i=1}^{n} \lambda_i, \quad \text{subject to} \quad \max(\boldsymbol{\lambda}) \leq \mu_M \tag{25}$$

Note that the condition $\mu_M - \max(\boldsymbol{\lambda}) \geq 0$ is sufficient but not necessary for $\mu_H \geq 0$ to hold. For this reason (25) is a relaxation of (22).[4]

Problem (25) is a linear program, and one can see that the optimal solution is $\lambda_1^\star = \ldots = \lambda_n^\star = \mu_M$, which implies:

$$l^\star = n\,\mu_M \tag{26}$$

The lower bound $l^\star$ is cheap to obtain since it only requires the computation of the smallest eigenvalue of the matrix $\boldsymbol{M}^\mathsf{T}\boldsymbol{M}$. Moreover, since (26) is a relaxation of (22), it holds:

$$l^\star \leq d^\star \tag{27}$$

We obtained the bound $l^\star$ by relaxing the dual problem (22). However, $l^\star$ also has an intuitive relation with the primal problem (16), as explained in the following proposition (proof is given in appendix). This will be useful to devise the upper bound of Section V.

Proposition 2 (Primal interpretation of $l^\star$): The lower bound $l^\star$ is the optimal cost attained by the following relaxation of problem (16):

$$l^\star = \min_{\boldsymbol{r}} \|\boldsymbol{M}\boldsymbol{r}\|^2, \quad \text{subject to} \sum_{i=1}^{n} \boldsymbol{r}^\mathsf{T}\boldsymbol{N}_i\boldsymbol{r} = n \tag{28}$$

where the constraint that each of the n vectors $\boldsymbol{r}_i$ has unit norm in (16) is replaced by the condition that the sum of the (squared) norms is equal to n. Moreover, the optimal solution $l^\star$ of (28) is attained by the right singular vector $\boldsymbol{s}^\star$ corresponding to the smallest singular value of $\boldsymbol{M}^\mathsf{T}\boldsymbol{M}$. □

To wrap-up the discussion of this section: exploiting duality, one can compute a first lower bound $d^\star$ solving (22). This bound is tight in common problem instances, but may be slow to compute. Therefore, we proposed a second lower bound $l^\star$ that only requires the computation of the smallest eigenvalue of $\boldsymbol{M}^\mathsf{T}\boldsymbol{M}$, and is attained by the corresponding right singular vector $\boldsymbol{s}^\star$.

[4] It turns out that analyzing the spectrum of the sum of Hermitian matrices is not an easy task. This problem has a long history, starting from the *Horn's conjecture* [30], later proven by Knutson and Tao [31].

V. An Upper Bound for $f^\star$

Let us consider the primal problem (16). By definition of optimality, any estimate $\hat{\boldsymbol{x}}$ that is feasible (i.e., that satisfies the constraints) is such that $f^\star \leq f(\hat{\boldsymbol{x}})$, hence every feasible solution provides a valid upper bound for $f^\star$. However, a loose bound would be useless and we look for a *tight* upper bound, which is as close as possible to $f^\star$.

We propose an upper bound that leverages the result of Proposition 2. Currently, we do not have theoretical guarantees on the tightness of this bound, but we show in the experimental section that it works well for reasonable measurement noise.

Proposition 2 states that the right singular vector $\boldsymbol{s}^\star$ corresponding to the smallest singular value of the matrix $\boldsymbol{M}^\mathsf{T}\boldsymbol{M}$ attains the lower bound $l^\star$. Unfortunately, in general, $\boldsymbol{s}^\star$ is not feasible for problem (16): if we partition $\boldsymbol{s}^\star$ in two-vectors $\boldsymbol{s}^\star = \{\boldsymbol{s}_1^\star, \ldots, \boldsymbol{s}_n^\star\}$, each $\boldsymbol{s}_i^\star$ is not guaranteed to have unit norm, as required by the constraints in (16).

To obtain our upper bound, we simply normalize each $\boldsymbol{s}_i^\star$:

$$\bar{\boldsymbol{s}}_i = \boldsymbol{s}_i^\star / \|\boldsymbol{s}_i^\star\|, \quad i = 1, \ldots, n.$$

Since, by construction the vector $\bar{\boldsymbol{s}} = \{\bar{\boldsymbol{s}}_1, \ldots, \bar{\boldsymbol{s}}_n\}$ satisfies the constraints in (16), the cost attained by $\bar{\boldsymbol{s}}$ is an upper bound of the optimal cost:

$$u^\star = f(\bar{\boldsymbol{s}}) \geq f^\star \tag{29}$$

In the following section we clarify how to use the upper and lower bounds we computed so far in SLAM.

VI. Verification of Optimality

In this section we wrap-up our discussion and focus on practical use of our results. From the derivation in this paper we know that the optimal value $f^\star$ satisfies:

$$l^\star \leq d^\star \leq f^\star \leq u^\star \tag{30}$$

where $l^\star$, $d^\star$, and $u^\star$ are given in Eqs. (26), (22), (29). In the experimental section we show that in common SLAM problems $d^\star = f^\star$, hence (30) becomes $l^\star \leq d^\star = f^\star \leq u^\star$.

Now, let us assume we are given a *candidate SLAM solution* $\hat{\boldsymbol{x}}$. The candidate solution can be the result of the application of an iterative optimization technique (e.g., Gauss-Newton or Levenberg-Marquardt methods) to the SLAM problem (3). Our task is to check whether $\hat{\boldsymbol{x}}$ corresponds to a global minimum of problem (3).

Plugging $\hat{\boldsymbol{x}}$ back into (3) we compute $f(\hat{\boldsymbol{x}})$. Then, from our derivation, we can conclude that:

(C.1) if $f(\hat{\boldsymbol{x}}) = d^\star$, then $\hat{\boldsymbol{x}}$ is optimal;
(C.2) if $f(\hat{\boldsymbol{x}}) \notin [l^\star, u^\star]$, then $\hat{\boldsymbol{x}}$ is *not* optimal;

We will show in the next section that the interval $[l^\star, u^\star]$ is small in practice, hence the last check (C.2) also provides a reliable way to disambiguate global from local minima. This is particularly useful when one cannot afford to compute $d^\star$.

	n	m	$l^\star$		$d^\star$		$f^\star$		$u^\star$	$f(\hat{\boldsymbol{x}})$
INTEL	1228	1505	$6.06 \cdot 10^{-1}$	$\leq$	$7.89 \cdot 10^{-1}$	$\leq$	$7.89 \cdot 10^{-1}$	$\leq$	$8.14 \cdot 10^{-1}$	$7.89 \cdot 10^{-1}$
INTEL-a	1228	1505	$3.49 \cdot 10^{+2}$	$\leq$	$3.92 \cdot 10^{+2}$	$\leq$	$3.92 \cdot 10^{+2}$	$\leq$	$4.86 \cdot 10^{+2}$	$5.96 \cdot 10^{+2}$
FR079	989	1217	$5.39 \cdot 10^{-2}$	$\leq$	$7.19 \cdot 10^{-2}$	$\leq$	$7.19 \cdot 10^{-2}$	$\leq$	$7.49 \cdot 10^{-2}$	$7.19 \cdot 10^{-2}$
FR079-a	989	1217	$0.27 \cdot 10^{+2}$	$\leq$	$2.93 \cdot 10^{+2}$	$\leq$	$2.93 \cdot 10^{+2}$	$\leq$	$3.37 \cdot 10^{+2}$	$3.26 \cdot 10^{+3}$
CSAIL	1045	1172	$0.89 \cdot 10^{-1}$	$\leq$	$1.07 \cdot 10^{-1}$	$\leq$	$1.07 \cdot 10^{-1}$	$\leq$	$2.39 \cdot 10^{-1}$	$1.07 \cdot 10^{-1}$
CSAIL-a	1045	1172	$0.44 \cdot 10^{+2}$	$\leq$	$1.51 \cdot 10^{+2}$	$\leq$	$1.51 \cdot 10^{+2}$	$\leq$	$3.23 \cdot 10^{+2}$	$1.51 \cdot 10^{+2}$
M3500	3500	5453	$1.24 \cdot 10^{2}$	$\leq$	n/a	$\leq$	$1.38 \cdot 10^{2}$	$\leq$	$1.66 \cdot 10^{2}$	$1.38 \cdot 10^{2}$
M3500-a	3500	5453	$7.76 \cdot 10^{2}$	$\leq$	n/a	$\leq$	$9.12 \cdot 10^{2}$	$\leq$	$3.70 \cdot 10^{3}$	$1.13 \cdot 10^{6}$
M3500-b	3500	5453	$1.57 \cdot 10^{3}$	$\leq$	n/a	$\leq$	$2.05 \cdot 10^{3}$	$\leq$	$8.65 \cdot 10^{3}$	$1.50 \cdot 10^{6}$
M3500-c	3500	5453	$1.61 \cdot 10^{3}$	$\leq$	n/a	$\leq$	$2.55 \cdot 10^{3}$	$\leq$	$1.39 \cdot 10^{4}$	$1.08 \cdot 10^{9}$

TABLE I

NUMBER OF NODES (n), NUMBER OF MEASUREMENTS (m), PROPOSED BOUNDS ($l^\star, d^\star, u^\star$), OPTIMAL COST ($f^\star$), AND COST OBTAINED BY g2o ($f(\hat{\boldsymbol{x}})$), FOR EACH TESTED DATASET. FOR THE SCENARIOS M3500-a-b-c, THE VALUES $f^\star$ AND $f(\hat{\boldsymbol{x}})$ ARE TAKEN FROM [1].

VII. EXPERIMENTAL VALIDATION

In this section we evaluate the quality of the bounds $l^\star, d^\star, u^\star$ in practical problem instances. As we will see, standard benchmarking scenarios available in SLAM literature are particularly easy, and our bounds are very tight for those. For this reason we introduce new challenging datasets to test the limits of applicability of our technique.

We consider the following datasets:

INTEL: *Intel Research Lab* [9];
FR079: *University of Freiburg*, building 079 [9];
CSAIL: *MIT*, CSAIL building [9];
M3500: simulated dataset, proposed in [10];
INTEL-a, FR079-a, CSAIL-a: variants of the INTEL, FR079, and CSAIL datasets with extra additive noise on rotation measurements (std: 0.1 rad);
M3500-a, M3500-b, M3500-c: variants of the M3500 dataset with extra additive noise on rotation measurements (std: 0.1, 0.2, and 0.3 rad, respectively) [1].

For each dataset we report the number of poses (n) and number of measurements (m) in Table I.

We implemented the computation of the bounds $l^\star, d^\star, u^\star$ in Matlab. All the bounds require the computation of the matrix $\boldsymbol{M}^\mathsf{T}\boldsymbol{M}$; this is a dense matrix in general. The bounds $l^\star, u^\star$ can be computed from the smallest eigenvalue of $\boldsymbol{M}^\mathsf{T}\boldsymbol{M}$ and the corresponding singular vector; this can be done using standard Matlab functions. In order to compute $d^\star$, we implemented the optimization problem (22) in CVX [32]. This worked for small datasets (as the GRID scenarios discussed later in this section), but for large scenarios (as the ones in Table I) we incurred in memory problems. In those cases we solved the SDP using NEOS [33], [34], which is an online service designed to solve large optimization problems. We chose sdpt3 as SDP solver in NEOS.

Table I reports the bounds $l^\star, d^\star, u^\star$ and the optimal cost $f^\star$, following the order of the inequality chain (30). Moreover, in the last column, we report the value of the cost $f(\hat{\boldsymbol{x}})$ attained by g2o, using odometry as initial guess. The optimal cost $f^\star$ is obtained by bootstrapping g2o with the algorithm presented in [1]. In our derivation, $f^\star$ should be obtained by optimizing the cost (3), (chordal distance), while g2o uses the angular distance (Remark 1). We implemented a routine optimizing directly the cost (3), and we noticed that it produced the same optimal cost as g2o in all scenarios[5], which essentially confirms Remark 1. For this reason, in the following, $f^\star$ denotes interchangeably the optimal value for the angular cost (g2o) and the chordal cost (3).

In all tested scenarios the lower bound $d^\star$ was tight, i.e., $d^\star = f^\star$ (blue entries in Table I). In the largest scenarios, also NEOS incurred in memory problems and was not able to solve the problem (n/a entries in the table). The bounds $l^\star, u^\star$ are very close to $f^\star$ in the scenarios INTEL, FR079, CSAIL, M3500, which are the usual benchmarking scenarios in related works. For those and for the scenario CSAIL-a also g2o was able to attain the optimal cost $f^\star$.

	Time $d^\star$	Time $(l^\star, u^\star)$
INTEL	3573.3	2.2
INTEL-a	3321.9	2.3
FR079	3121.5	1.3
FR079-a	1663.6	1.4
CSAIL	2297.8	1.5
CSAIL-a	2389.9	1.6
M3500	n/a	20.9
M3500-a	n/a	23.4
M3500-b	n/a	25.1
M3500-c	n/a	24.2

TABLE II

TIME IN SECONDS TO COMPUTE THE BOUND $d^\star$ AND THE BOUNDS $(l^\star, u^\star)$ FOR EACH TESTED SCENARIO.

In the noisy scenarios INTEL-a, FR079-a, and M3500-a-b-c, the solution produced by g2o was suboptimal. In all cases, our bounds were able to detect the wrong solution: the red entries in Table I denote the case in which $f(\hat{\boldsymbol{x}}) \notin [l^\star, u^\star]$; in these cases our verification technique guarantees that the solution corresponds to a local minimum (Section VI). For a visual comparison, Fig. 2 shows the estimated trajectory corresponding to the global minimum against the g2o estimate, for some of the datasets. It is interesting to notice that our technique was able to detect wrong solutions even when they only imply small imperfections: for instance, the local minimum for the dataset INTEL-a is globally correct and only has a wrong wraparound in the bottom-right loop.

[5] Maximum observed difference was less than 1% of the cost in all tests.

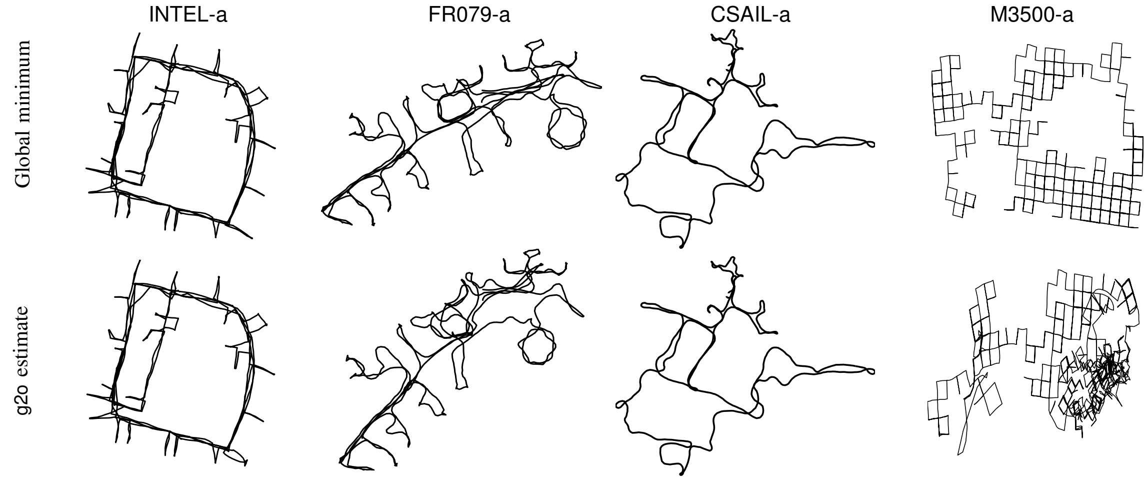


Fig. 2. Trajectory estimates corresponding to the global minimum $f^\star$ against the estimate returned by g2o for different tested scenarios. The scenarios INTEL-a, FR079-a, and CSAIL-a are proposed in this paper and are noisy versions of the datasets INTEL, FR079, and CSAIL [9]. In the scenarios INTEL-a, FR079-a, and M3500-a, g2o is trapped in a local minimum. Further details on the scenarios M3500-a, M3500-b and M3500-c are given in [1].

Table II reports the CPU time required to compute the bounds proposed in this paper, for the scenarios of Table I. We put together the bounds $l^\star, u^\star$ as they are computed by a single Matlab instruction (the time to normalize the vectors as per eq. (29) is negligible). The computation of $d^\star$ is prohibitive in practically all cases. However, $l^\star, u^\star$ are relatively cheap to compute: recall that the verification technique can be executed periodically and it is not subject to strict timing constraints, as in standard SLAM algorithms.

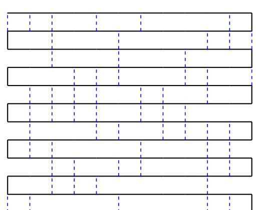


Fig. 3. Example of GRID scenario with 12^2 nodes. Solid black line denotes the odometric path, while loop closures are shown as dashed blue lines.

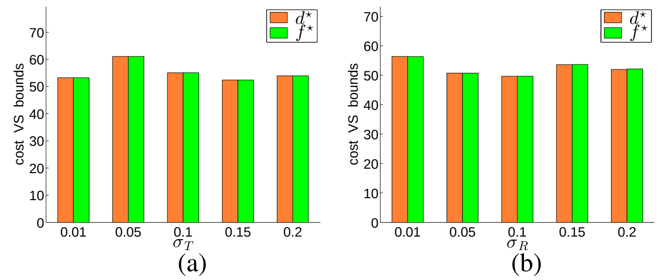


Fig. 4. Optimal cost $f^\star$ VS our lower bound $d^\star$ for different levels of (a) translation noise (std: σ_T), and (b) rotation noise (std: σ_R).

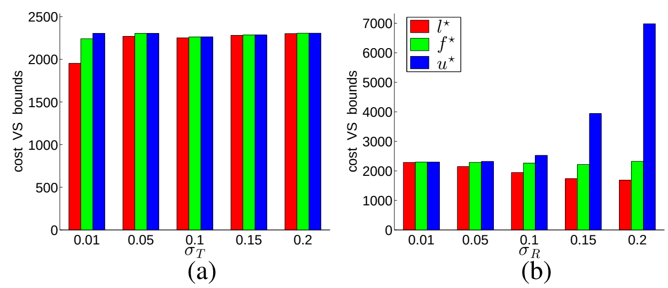


Fig. 5. Optimal cost $f^\star$ VS the lower bound $l^\star$ and the upper bound $u^\star$ for different levels of (a) translation noise, and (b) rotation noise.

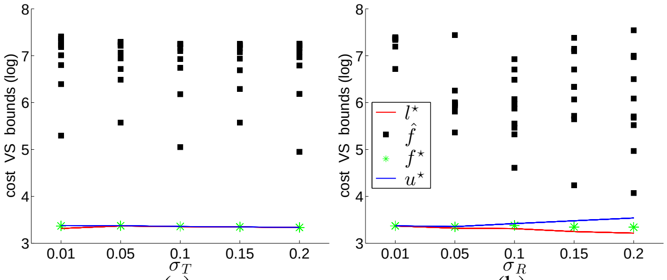
A. Further tests

To better evaluate our bounds, we performed a Monte Carlo analysis on the simulated GRID dataset of Fig. 3. Ground truth robot odometry is shown as a solid line in the figure, while loop closures are added randomly (with probability 0.5) between nearby nodes. The actual measurements are obtained by adding Gaussian noise to the ground truth. We indicate with σ_T and σ_R the standard deviations of translation and rotation noise, respectively. Unless specified otherwise, we consider $\sigma_T = 0.1$ m and $\sigma_R = 0.01$ rad. All results are averaged over 10 Monte Carlo runs.

First, we consider a relatively small GRID dataset with 7^2 nodes on which we can solve the SDP and compute $d^\star$. Fig. 4 shows the bound $d^\star$ versus the optimal cost $f^\star$ for increasing levels of translation noise and rotation noise. As in the previous tests, the bound $d^\star$ is tight.

Now, we consider a larger scenario with 40^2 nodes to test $l^\star$ and $u^\star$. Fig. 5a shows the bounds versus the optimal value for increasing levels of translation noise. The bounds are fairly close to $f^\star$ independently on σ_T. Fig. 5b shows the bounds for increasing levels of the rotation noise. The upper bound $u^\star$ degrades for large noise, while the degradation is more graceful for $l^\star$. Both bounds are close to the optimal value in the standard range of operation ($\sigma_R < 0.15$ rad).

In order to show that the bounds $(l^\star, u^\star)$ are adequate to discern global and local minima, in Fig. 6 we show the bounds versus the global minimum and different local minima obtained by initializing g2o with random initial guesses (10 runs). In all cases the interval $[l^\star, u^\star]$ only contains the global minimum, meaning that the bounds allow to accurately identify wrong solutions. Note that the plot is on log scale, meaning that our bounds are orders of magnitude better than the values produced by local minima.



(a) (b)

Fig. 6. Global minimum ($f^\star$) and local minima ($\hat{f}$) versus the proposed bounds $[l^\star, u^\star]$, for different levels of (a) translation noise (std: σ_T), and rotation noise (std: σ_R).

VIII. Conclusion

We propose techniques to verify whether a given SLAM estimate is globally optimal. These techniques are based on duality theory, and rely on the computation of lower and upper bounds on the optimal cost. Experimental results show that these bounds can successfully discern globally correct estimates from wrong solutions corresponding to local minima. Our verification techniques can be integrated seamlessly in standard SLAM pipelines, and provide a sanity check for the solution returned by standard iterative solvers.

References

[1] L. Carlone and A. Censi, "From angular manifolds to the integer lattice: Guaranteed orientation estimation with application to pose graph optimization," *IEEE Trans. Robotics*, 2014.

[2] *Handbook of Robotics*. B. Siciliano and O. Khatib: Springer, 2008.

[3] G. Rose and S. Thrun, "Google's X-Man A conversation with Sebastian Thrun," *Foreign Affairs*, vol. 92, no. 6, pp. 2–8, 2013.

[4] V. Indelman, S. Wiliams, M. Kaess, and F. Dellaert, "Factor graph based incremental smoothing in inertial navigation systems," in *Intl. Conf. on Information Fusion, FUSION*, 2012.

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

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

[7] L. Carlone, A. Censi, and F. Dellaert, "Selecting good measurements via ℓ_1 relaxation: a convex approach for robust estimation over graphs," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2014.

[8] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, Shanghai, China, May 2011.

[9] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner, "Slam benchmarking webpage," 2009.

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

[11] G. Grisetti, C. Stachniss, and W. Burgard, "Non-linear constraint network optimization for efficient map learning," *Trans. on Intelligent Transportation systems*, vol. 10, no. 3, pp. 428–439, 2009.

[12] D. Rosen, M. Kaess, and J. Leonard, "RISE: An incremental trust-region method for robust online sparse least-squares estimation," *IEEE Trans. Robotics*, 2014.

[13] R. Tron, B. Afsari, and R. Vidal, "Intrinsic consensus on SO(3) with almost global convergence," in *IEEE Conference on Decision and Control*, 2012.

[14] F. Dellaert and A. Stroupe, "Linear 2D localization and mapping for single and multiple robots," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2002.

[15] L. Carlone, R. Aragues, J. Castellanos, and B. Bona, "A linear approximation for graph-based simultaneous localization and mapping," in *Robotics: Science and Systems (RSS)*, 2011.

[16] ——, "A fast and accurate approximation for planar pose graph optimization," *Intl. J. of Robotics Research*, 2014.

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

[18] H. Wang, G. Hu, S. Huang, and G. Dissanayake, "On the structure of nonlinearities in pose graph SLAM," in *Robotics: Science and Systems (RSS)*, 2012.

[19] S. Huang, H. Wang, U. Frese, and G. Dissanayake, "On the number of local minima to the point feature based SLAM problem," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2012, pp. 2074–2079.

[20] J. Knuth and P. Barooah, "Error growth in position estimation from noisy relative pose measurements," *Robotics and Autonomous Systems*, vol. 61, no. 3, pp. 229–224, 2013.

[21] L. Carlone, "Convergence analysis of pose graph optimization via Gauss-Newton methods," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2013, pp. 965–972.

[22] K. Khosoussi, S. Huang, and G. Dissanayake, "Novel insights into the impact of graph structure on SLAM," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2014.

[23] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[24] R. Hartley, J. Trumpf, Y. Dai, and H. Li, "Rotation averaging," *IJCV*, vol. 103, no. 3, pp. 267–305, 2013.

[25] D. Sorensen, "Minimization of a large-scale quadratic fuction subject to a spherical constraint," *SIAM J. Optim.*, vol. 7, pp. 141–161, 1997.

[26] J. R. Bar-On and K. A. Grasse, "Global optimization of a quadratic functional with quadratic equality constraints," *Journal of Optimization Theory and Applications*, vol. 82, no. 2, pp. 379–386, 1994.

[27] ——, "Global optimization of a quadratic functional with quadratic equality constraints, part 2," *Journal of Optimization Theory and Applications*, vol. 93, no. 3, pp. 547–556, 1997.

[28] J. Fredriksson and C. Olsson, "Simultaneous multiple rotation averaging using lagrangian duality," in *Asian Conf. on Computer Vision (ACCV)*, 2012.

[29] C. Meyer, *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000.

[30] A. Horn, "Eigenvalues of sums of Hermitian matrices," *Pacific J. Math.*, vol. 12, pp. 225–241, 1962.

[31] A. Knutson and T. Tao, "Honeycombs and sums of Hermitian matrices," *Notices Amer. Math. Soc.*, vol. 48, no. 2, pp. 175–186, 2001.

[32] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming." [Online]. Available: http://cvxr.com/cvx

[33] W. Gropp and J. J. Moré, *Optimization Environments and the NEOS Server*. Approximation Theory and Optimization, M. D. Buhmann and A. Iserles, eds., Cambridge University Press, 1997.

[34] J. Czyzyk, M. P. Mesnier, and J. J. Moré, "The NEOS server," *IEEE Journal on Computational Science and Engineering*, vol. 5, no. 3, pp. 68–75, 1998.

Appendix

Proof of Proposition 2. Because of the structure of the matrices $\boldsymbol{N}_i$, the following equality holds:

$$\sum_{i=1}^{n} \boldsymbol{r}^\mathsf{T} \boldsymbol{N}_i \boldsymbol{r} = \sum_{i=1}^{n} \boldsymbol{r}_i^\mathsf{T} \boldsymbol{r}_i = \boldsymbol{r}^\mathsf{T} \boldsymbol{r}. \tag{31}$$

Using (31) and applying the change of variables $\boldsymbol{q} = \boldsymbol{r}/\sqrt{n}$, problem (28) becomes:

$$\min_{\boldsymbol{q}} \; n \left(\boldsymbol{q}^\mathsf{T} \boldsymbol{M}^\mathsf{T} \boldsymbol{M} \boldsymbol{q} \right), \quad \text{subject to} \quad \boldsymbol{q}^\mathsf{T} \boldsymbol{q} = 1 \tag{32}$$

Recalling that the definition of the smallest eigenvalue of the matrix $\boldsymbol{M}^\mathsf{T}\boldsymbol{M}$ is:

$$\mu_M \doteq \min_{\boldsymbol{q}, \|\boldsymbol{q}\|=1} \boldsymbol{q}^\mathsf{T} (\boldsymbol{M}^\mathsf{T} \boldsymbol{M}) \boldsymbol{q}, \tag{33}$$

it follows that the optimal value of (32) is $n \; \mu_M$, which coincides with $l^\star$ in (26), proving the first claim. The second claim easily follows, noting that the minimum of (33) is attained, by definition, by the right singular vector of $\boldsymbol{M}^\mathsf{T}\boldsymbol{M}$ corresponding to the smallest singular value. $\square$