Lagrangian Duality in 3D SLAM: Verification Techniques and Optimal Solutions

Luca Carlone, David M. Rosen, Giuseppe Calafiore, John J. Leonard, Frank Dellaert

Abstract—State-of-the-art techniques for simultaneous localization and mapping (SLAM) employ iterative nonlinear optimization methods to compute an estimate for robot poses. While these techniques often work well in practice, they do not provide guarantees on the quality of the estimate. This paper shows that Lagrangian duality is a powerful tool to assess the quality of a given candidate solution. Our contribution is threefold. First, we discuss a revised formulation of the SLAM inference problem. We show that this formulation is probabilistically grounded and has the advantage of leading to an optimization problem with quadratic objective. The second contribution is the derivation of the corresponding Lagrangian dual problem. The SLAM dual problem is a (convex) semidefinite program, which can be solved reliably and globally by off-the-shelf solvers. The third contribution is to discuss the relation between the original SLAM problem and its dual. We show that from the dual problem, one can evaluate the quality (i.e., the suboptimality gap) of a candidate SLAM solution, and ultimately provide a certificate of optimality. Moreover, when the duality gap is zero, one can compute a guaranteed optimal SLAM solution from the dual problem, circumventing non-convex optimization. We present extensive (real and simulated) experiments supporting our claims and discuss practical relevance and open problems.

I. INTRODUCTION

Simultaneous localization and mapping (SLAM) is an enabling technology for many applications, including service and industrial robotics, autonomous driving, search and rescue, planetary exploration, and augmented reality.

The last decade has witnessed several groundbreaking results in SLAM, and state-of-the-art approaches are now transitioning from academic research to industrial applications. Standard techniques compute an estimate (e.g., for robot poses) by minimizing a nonlinear cost function, whose global minimum is the maximum likelihood estimate (or maximum a posteriori estimate in presence of priors). The optimization problem underlying SLAM is commonly solved using iterative nonlinear optimization methods, e.g. the Gauss-Newton method [1], [2], [3], the gradient method [4], [5], trust region methods [6], or ad-hoc approximations [7], [8].

Despite the success of state-of-the-art techniques, some practical and theoretical problems remain open. While iterative approaches are observed to work well in many problem instances, they cannot guarantee the correctness (global optimality) of the estimates that they compute. This is due to the fact that the optimization problem is non-convex, hence iterative techniques may be trapped in local minima(Fig. 1), which correspond to wrong estimates. Recent work [9], [10] shows that iterative techniques fail to converge to a correct estimate even in fairly simple (real and simulated) 3D problems. Recent research efforts have addressed the issue of global convergence from several angles. Olson et al. [4], Grisetti et al. [5], Rosen et al. [6], and Tron et al. [11] study iterative techniques with larger basins of convergence. Carlone et al. [8], [12], and Rosen et al. [10] propose initialization techniques to bootstrap iterative optimization. Huang et al. [13], Wang et al. [14], Carlone [15], and Khosoussi et al. [16] investigate the factors influencing local convergence and the quality of the optimal SLAM solution. While these techniques provide remarkable insights into the problem, and working solutions to improve convergence, none of them can guarantee the recovery of a globally optimal solution to SLAM.

The motivation behind this work is that the transition of SLAM from research topic to industrial technology requires techniques with guaranteed performance. In autonomous vehicle applications, failure to produce a correct SLAM solution may put passengers’ lives at risk. In other applications, SLAM failures can possibly cascade into path planning failures (if the plan is computed using a wrong map), and this may prevent the reliable operation of mobile robots.

Therefore, in this paper we address the following question: Verification Problem: Given a candidate SLAM estimate (e.g., returned by a state-of-the-art iterative solver), is it possible to evaluate the quality of this estimate (e.g., its suboptimality gap), possibly certifying its optimality?

1We use the term “local minimum” to denote a stationary point of the cost which does not attain the optimal objective.
To address this problem, we introduce a powerful tool, \textit{Lagrangian duality}, borrowing the corresponding theory from the optimization community. Duality was first applied to 2D SLAM by Carlone and Dellaert \cite{9}. In this work, we provide a nontrivial extension of \cite{9} to 3D SLAM, including a probabilistic justification of our formulation, the homogenization of the objective function, and extra tools for verification.

This paper contains three main contributions. The first contribution is a revised formulation of the SLAM inference problem. Our formulation is a probabilistically grounded maximum likelihood (ML) estimator, and has the advantage of leading to an optimization problem with quadratic objective, which facilitates the derivation of the dual problem. Our verification techniques, presented in Section IV, can be seamlessly integrated in the dual problem. Our verification techniques, directly from the dual problem. Our verification techniques, when the duality gap is zero, we can compute an optimal SLAM solution. Since current SDP solvers do not scale well to large problems, we also propose a second verification technique that does not require solving the SDP. As a by-product of our derivation, we show that, when the duality gap is zero, we compute an optimal SLAM solution directly from the dual problem. Our verification techniques, presented in Section III, can be seamlessly integrated in standard SLAM pipelines. Experimental evidence (Section IV) confirms that these techniques enable the certification of globally optimal solutions in both real and simulated environments. Extra results and visualizations are given in the supplemental material \cite{17}.

II. 3D Pose Graph Optimization Revisited

We consider the pose graph optimization (PGO) formulation of the SLAM problem. PGO computes the maximum likelihood estimate for $n$ poses $x_1, \ldots, x_n$, given $m$ relative pose measurements $\bar{x}_{ij}$ between pairs of poses $i$ and $j$. In a 3D setup, both the unknown poses and the measurements are quantities in $\text{SE}(3) = \{ (R, t) : R \in \text{SO}(3), t \in \mathbb{R}^3 \}$. We use the notation $x_i = (R_i, t_i)$ and $\bar{x}_{ij} = (R_{ij}, \tilde{t}_{ij})$ to make explicit the rotation and the translation of each pose. PGO can be visualized as a directed graph $G(V, E)$, in which we associate a node $i \in V = \{1, \ldots, n \}$ to each pose $x_i$ and an edge $(i, j) \in E$ to each relative measurement $\bar{x}_{ij}$.

In this section we propose a revised PGO formulation. The key difference w.r.t. related work is the use of the \textit{chordal distance} to quantify the rotation errors (more details in Section II-B). To lay the groundwork for this formulation, we begin with a generative model for our measurements, and then derive the corresponding ML estimator.

A. Generative Noise Model

We assume the following generative model for the relative pose measurements $\bar{x}_{ij} = R_{ij} t_{ij}$.

\[ \bar{x}_{ij} = R_{ij} t_{ij} + \epsilon \quad \epsilon \sim \text{gaussian}(0, \omega_I I) \]

\[ R_{ij} = R_i^T R_j R_{\tilde{t}_{ij}} \quad R_{\tilde{t}_{ij}} \sim \text{vonMises}(I_3, \omega_{R^2}) \] (1)

where “\text{gaussian}(\mu, \Sigma)” denotes a Gaussian distribution with mean $\mu$ and \textit{information} matrix $\Sigma$, while “\text{vonMises}(S, \kappa)” denotes the isotropic von Mises-Fisher distribution, with mean $S \in \text{SO}(3)$ and concentration parameter $\kappa$. The key difference w.r.t. to measurement models in other PGO formulations lies in the use of the von Mises-Fisher distribution as the model for the rotational measurements errors $R_{ij} \in \text{SO}(3)$.

The isotropic von Mises-Fisher (or \textit{Langevin}) distribution on $\text{SO}(n)$ with mean $S \in \text{SO}(n)$ and concentration parameter $\kappa \geq 0$ can be written explicitly as:

\[ P(R_{ij}) = \frac{1}{c_n(\kappa)} \exp (\kappa \text{tr} (S^T R_{ij})) \],

where $\text{tr} (\cdot)$ is the matrix trace and $c_n(\kappa)$ is a normalization term. Closed-form expressions for $c_n(\kappa)$ are given in \cite{18}, these are inconsequential for our derivation. For $\kappa \to 0$, the distribution tends to the uniform distribution over $\text{SO}(n)$. For $\kappa \to \infty$, $R_{ij} = S$ with probability one. Roughly speaking, one may think at $\kappa$ as a measure of \textit{information content}.

We are now ready to introduce the maximum likelihood estimator for the poses, given the measurement model (1).

B. Maximum Likelihood Estimator

The ML estimate corresponds to the set of poses maximizing the likelihood of the measurements, or, equivalently, minimizing the negative log-likelihood:

\[ f_{\text{ML}} = \min_{\{x_i \in \text{SE}(3)\}} \sum_{(i,j) \in E} -\log \mathcal{L}(\bar{x}_{ij} | x) - \log \mathcal{L}(\tilde{t}_{ij} | x). \] (3)

The negative log-likelihood of the Cartesian measurements can be easily computed from the Gaussian distribution:

\[ -\log \mathcal{L}(\tilde{t}_{ij} | x) = \omega_t^2 \| t_j - t_i - R_i \tilde{t}_{ij} \|^2 + \text{const}. \] (4)

Using (2), the negative log-likelihood for $\bar{x}_{ij}$ is:

\[ -\log \mathcal{L}(\bar{x}_{ij} | x) = -\omega_I^2 \text{tr} (R_i^T R_j R_{\tilde{t}_{ij}}) + \text{const}. \]

\[ = \frac{\omega_I^2}{2} \| R_j - R_i \bar{x}_{ij} \|_F^2 + \text{const}. \] (5)

where $\| \cdot \|_F$ is the Frobenius matrix norm (sum of the squares of the entries), and we used $\| S - R \|_F^2 = \text{tr} ((S - R)(S - R)^T)$. The norm $\| S - R \|_F^2$ is usually referred to as the \textit{chordal distance} between two rotations $S$ and $R$ \cite{19}.

Plugging (4) and (5) back into (3) we obtain our ML estimator:

\[ 2^\text{To keep notation simple, we consider measurements with the same distribution. The extension to heterogeneous $\omega_I^2$ and $\omega_t^2$ is trivial.} \]
\[ f_{\text{NL}} = \min_{\{R_i \in \text{SO}(3)\}} \sum_{(i,j) \in \mathcal{E}} \omega_i^2 \| t_j - t_i - R_i t_{ij} \|^2 \quad \text{(PGO)} \]

\[ + \frac{\omega_e^2}{2} \| R_j - R_i R_{ij} \|^2 \]  

(6)

The main difference between (6) and formulations in related work is that we use the chordal distance to quantify rotation errors, as proposed in [9, 19] (related work instead uses the geodesic distance \( \| \log \left( R_i^T R_j \right) \|_2 \)). References [9], [19] show that for small residual errors \( R \) defined as

\[ R = R_i R_j^{-1} - I \]

subject to \( R_i \in \text{SO}(3) \), we can compute close proxies of \( f \) for the PGO problem (6) as:

\[ f^* = \min_{\{\hat{x}, t_i\}} \sum_{(i,j) \in \mathcal{E}} \omega_i^2 \| t_j - t_i - T_{\hat{x}, t_{ij}} \|^2 + \frac{\omega_e^2}{2} \| R_j - Q_{ij} r_{ij} \|_2 \]

(7)

(8)

subject to

\[ r_j^T E_{uv} r_i = 1, \quad u = v \]  

\[ r_j^T E_{uv} r_i = 0, \quad u \neq v \]

(9)

where \( E_{uv} \) is a 9 \times 9 selection matrix composed of 3 \times 3 blocks that are zero everywhere except the 3 \times 3 block in position \((u,v)\), which is the identity matrix. The matrices \( E_{uv} \) are such that \( r_j^T E_{uv} r_i = (R_i^{(uv)})^T R_j^{(uv)} \), hence the constraints in (9) correspond to the orthonormality constraints in (8).

In order to write (9) in a more compact matrix notation, we define the vector \( \hat{x} = [t_1^T, \ldots, t_n^T, r_{11}^T, \ldots, r_{mn}^T] \in \mathbb{R}^{12n} \). Using this notation, (9) becomes:

\[ \hat{f}^* = \min_{\hat{x}} \| \hat{A} \hat{x} - b \|^2 \]

(10)

subject to

\[ \hat{x}^T \hat{E}_{uv} \hat{x} = 1, \quad u = v \]  

\[ \hat{x}^T \hat{E}_{uv} \hat{x} = 0, \quad u \neq v \]

(11)

where the matrices \( \hat{A} \) and \( \hat{E}_{uv} \) are obtained by stacking the coefficient matrices in (9), with suitable zero blocks for padding; \( \omega_e^2 \) and \( \omega_e^2 \) are included in the definition of \( \hat{A} \).

Finally, since absolute poses are not observable from relative measurements, we fix a pose to be our reference frame. Without loss of generality we fix the pose of the first node to the identity pose \((t_1 = 0)\) and \(R_1 = I_3\), or, equivalently \( r_1 = \text{rows}(I_3) \). This process is usually called anchoring. Fixing the first pose modifies (10) as follows:

\[ f^* = \min_{\hat{x}} \| A \hat{x} - b \|^2 \]

(11)

subject to

\[ \hat{x}^T \hat{E}_{uv} \hat{x} = 1, \quad u = v \]  

\[ \hat{x}^T \hat{E}_{uv} \hat{x} = 0, \quad u \neq v \]

(12)

meaning that the two problems have the same optimal objective, and the corresponding solutions can be mapped to each other. Intuitively, if the solution of (12) is \([x^*_i]^T \]

3With slight abuse of notation, \([x^*_i]^T \) indicates a column vector, a somehow more formal, but more cumbersome, notation would be \([ \hat{x}^*_i ]^T \).

Finally, since absolute poses are not observable from relative measurements, we fix a pose to be our reference frame. Without loss of generality we fix the pose of the first node to the identity pose \((t_1 = 0)\) and \(R_1 = I_3\), or, equivalently \( r_1 = \text{rows}(I_3) \). This process is usually called anchoring. Fixing the first pose modifies (10) as follows:

\[ f^* = \min_{\hat{x}} \| A \hat{x} - b \|^2 \]

subject to

\[ \hat{x}^T \hat{E}_{uv} \hat{x} = 1, \quad u = v \]  

\[ \hat{x}^T \hat{E}_{uv} \hat{x} = 0, \quad u \neq v \]

meaning that the two problems have the same optimal objective, and the corresponding solutions can be mapped to each other. Intuitively, if the solution of (12) is \([x^*_i]^T \]

3With slight abuse of notation, \([x^*_i]^T \) indicates a column vector, a somehow more formal, but more cumbersome, notation would be \([ \hat{x}^*_i ]^T \).
then \( x^* = x_H^* \) is also optimal for (11), while if the solution is \([x_H^* - 1, 1] \), then \( x^* = -x_H^* \) will be optimal for (11). The inclusion of the slack variable \( y \) is often referred to as homogenization. We refer to (12) as the primal problem.

### B. The dual problem

In this section we apply Lagrangian duality to the primal problem (12), borrowing the corresponding theory from the optimization community [20], [21]. We begin by recalling basic properties and notions about duality theory and then we tailor these concepts to our SLAM problem.

The key insight of duality is that for every constrained optimization problem of the form:

\[
\begin{align*}
    f^* &= \min_{x} f(x) \\
    \text{subject to} \quad c_i(x) &= 0 \quad \forall i \in C
\end{align*}
\]

(13)

(where \( C \) is a set indexing the constraints \( c_i(x) \)), there is an associated unconstrained optimization problem:

\[
d^* = \max_{\lambda} \left( \inf_{x} f(x) + \sum_{i \in C} \lambda_i c_i(x) \right)
\]

(14)

called the dual problem. The scalar variables \( \lambda_i \) appearing in (14) are called Lagrange multipliers or dual variables, and the function \( d(\lambda) \) is called the dual function; \( \lambda \) is a vector stacking all dual variables. With reference to the dual problem (14), problem (13) is referred to as the primal problem. Intuitively, the minimization ("\( \inf \)") in \( d(\lambda) \) can be understood as a relaxation of the original problem (13) in which the constraints are transformed into penalty terms in the objective, whose "importance" is controlled by \( \lambda \); hence, the maximization (w.r.t. \( \lambda \)) tries to make this relaxation as tight as possible.

The dual problem (14) has two important properties. First, since the dual function \( d(\lambda) \) is the pointwise infimum of a family of affine functions of \( \lambda \), it is always concave, and therefore the dual maximization problem (14) is a convex program [20] Sec. 5.2]. Its convexity guarantees that the dual problem can always be solved globally optimally using local search techniques. Second, given any feasible \( x \) for (13) (for which \( c_i(x) = 0 \)), the definition of \( d(\lambda) \) in (14) shows that \( d(\lambda) \leq f(x) \) for any choice of \( \lambda \). In particular, this must also hold at the optima \( x^* \) and \( \lambda^* \) in (13) and (14), so that:

\[
d^* \leq f^*.
\]

(15)

The inequality (15) is referred to as weak (Lagrangian) duality, and it enables us to lower-bound the optimal value \( f^* \) of the (possibly very difficult, nonconvex) primal problem (13) using the optimal value \( d^* \) of the (convex) dual problem (14). For some problems, the inequality (15) is tight (i.e. \( d^* = f^* \)), for which we say that strong (Lagrangian) duality holds. The quantity \( f^* - d^* \geq 0 \) is called the duality gap.

Using weak duality, it is easy to show that, given a primal feasible point \( \hat{x} \) and a dual point \( \hat{\lambda} \), the following chain of inequality holds:

\[
d(\hat{\lambda}) \leq d(\lambda^*) \leq d^* \leq f^* \leq f(\hat{x})\]

(16)

where the first inequality stems from the fact that \( \lambda^* \) attains the maximum over all \( \lambda \), and the last follows from the optimality of \( f^* \) (which is the global minimum among all feasible \( x \)).

Therefore, in this work we exploit a simple idea: given a candidate solution \( \hat{x} \), if we are able to find a \( \lambda \), for which \( d(\lambda) = f(\hat{x}) \), then the chain of inequalities (16) becomes tight \( d(\lambda) = d^* = f^* = f(\hat{x}) \), which implies that \( \hat{x} \) is an optimal solution. Equation (16) thus provides a means of certifying the global optimality of a candidate solution \( \hat{x} \) for (13) and enables our derivation of algorithmic approaches for certifying the correctness of SLAM solutions.

We are now ready to apply duality to our primal problem (12). From (12) and (13), the dual function is:

\[
d(\lambda) = \inf_{x,y} [Ax - by]^2 + \sum_{i=1}^{n-1} \sum_{u=1,3} \lambda_{iuu}(1 - x^T E_{iuu} x) + \sum_{u,v=1,2,3} \lambda_{iuv}(-x^T E_{iuv} x) + \lambda_y (1 - y^2)^2.
\]

(17)

where \( \lambda \) is the vector of Lagrange multipliers \( \lambda_{iuv} \) and \( \lambda_y \), associated with the orthonormality and homogeneity constraints in (12), respectively. We observe that the quadratic terms in (17) can be written more compactly as:

\[
[Ax - by]^2 - x^T \left[ \sum_{i=1}^{n-1} \sum_{u,v=1}^3 \lambda_{iuu} E_{iuu} \right] x - \lambda_y y^2 = \left[ \begin{array}{c} x \\ y \end{array} \right]^T \left[ \begin{array}{cc} H(\lambda) & -A^T b \\ -b^T A & -b^T b - \lambda_y \end{array} \right] \left[ \begin{array}{c} x \\ y \end{array} \right],
\]

(18)

where

\[
H(\lambda) = A^T A - \sum_{i=1}^{n-1} \sum_{u,v=1}^3 \lambda_{iuv} E_{iuv}.
\]

(19)

Calling \( M(\lambda) \) the matrix in (18), the dual function (17) can thus be written as:

\[
d(\lambda) = \inf_{x,y} \left[ \begin{array}{c} x \\ y \end{array} \right]^T M(\lambda) \left[ \begin{array}{c} x \\ y \end{array} \right] + \sum_{i=1}^{n-1} \sum_{u=1,2,3} \lambda_{iuu} + \lambda_y.
\]

(20)

Now in the dual problem (14) we try to maximize \( d(\lambda) \); however, from (20) we see that \( d(\lambda) = -\infty \) if \( M(\lambda) \) has a negative eigenvalue (by letting \( [x y] \) lie in the corresponding eigenspace). Consequently, we can safely restrict our search to the vectors \( \lambda \) that preserve positive semi-definiteness of \( M(\lambda) \) [20] Sec. 5.1.5. Moreover:

\[
M(\lambda) \succeq 0 \implies \inf_{x,y} \left[ \begin{array}{c} x \\ y \end{array} \right]^T M(\lambda) \left[ \begin{array}{c} x \\ y \end{array} \right] = 0,
\]

(21)

as the minimization over \( [x y] \) in the homogenized problem (21) is unconstrained. The dual problem (14) thus becomes:

\[
d^* = \max_{\lambda} \sum_{i=1}^{n} \lambda_{iuu} + \lambda_y \quad \text{ (dual problem)}
\]

(22)

subject to \( M(\lambda) \succeq 0 \).

The dual SLAM problem turns out to be a semidefinite program (SDP), for which specialized solvers exist [22].
the following section we discuss the relations between the primal and the dual problem, and elucidate on its practical use.

IV. RELATION BETWEEN THE PRIMAL AND THE DUAL PROBLEM AND PRACTICAL USE

In this section we present two powerful applications of the dual problem (22). Section IV-A deals with the case in which one is given a candidate PGO solution, and wants to evaluate its quality, possibly certifying its optimality. Section IV-B shows that in particular cases (when the duality gap is zero) one can obtain an optimal solution of the primal problem from the solution $\lambda^*$ of the dual.

In both sections, we use the following property.

Lemma 1 (Primal optimal solution and zero duality gap): If the duality gap is zero ($d^* = f^*$), then any primal optimal solution $[x^* 1]$ of (12) is in the null space of the matrix $M(\lambda^*)$, where $\lambda^*$ is the solution of the dual problem (22).

Proof: When the duality gap is zero, any minimizer of the primal problem is also a minimizer for the infimum in the dual function $[14] [20]$ Sec. 5.5.5. Consider a primal optimal solution $[x^* 1]$. We already observed that any such minimizer annihilates the quadratic term in (20), and therefore it holds that $[x^* 1]^T M(\lambda^*) [x^* 1] = 0$, which implies that $[x^* 1]$ is in the null space of $M(\lambda^*)$, proving the claim.

We give an alternative proof, which does not require prior knowledge on duality, in the supplemental material [17].

A. Verification

In this section we consider the case in which we are given a candidate solution $\hat{x}$ for the primal problem, and we want to evaluate the quality of this solution. For brevity, we denote with $f(\hat{x})$ the objective evaluated at $\hat{x}$, for both (11) and its homogeneous form (12) (for the latter we imply $y = 1$).

We begin with the following proposition, whose proof easily follows from (16) and the discussion in Section III-B.

Proposition 2 (Verification of Primal Optimal Objective): Given a candidate solution $\hat{x}$ for the primal problem (12), if $f(\hat{x}) = d^*$, then the duality gap is zero and $\hat{x}$ is an optimal solution of (12). Moreover, even if the duality gap is nonzero, $f(\hat{x}) - d^* \geq f(\hat{x}) - f^*$, meaning that $f(\hat{x}) - d^*$ is an upper-bound for the sub-optimality gap of $\hat{x}$.

Proposition 2 ensures that the candidate $\hat{x}$ is optimal when $f(\hat{x}) = d^*$. Moreover, even in the case in which we get $f(\hat{x}) > d^*$, the quantity $f(\hat{x}) - d^*$ can be used as an indicator of how far $\hat{x}$ is from the global optimum.

While Proposition 2 already provides means of verifying a candidate solution, it requires solving the dual problem, to compute $d^*$. The following proposition provides a technique to verify the optimality of $\hat{x}$ without solving the SDP.

Proposition 3 (Verification of Primal Optimal Solution): Given a candidate solution $\hat{x}$ for the primal problem (12), if the linear system

$$
\begin{bmatrix}
M(\lambda) & \hat{x} \\
1 & 1
\end{bmatrix} = 0
$$

(23)

admits a solution, and this solution, called $\hat{\lambda}$, is such that $M(\lambda) \geq 0$ and $d(\hat{\lambda}) = f(\hat{x})$, then the duality gap is zero and $\hat{x}$ is a primal optimal solution.

Proof: From Lemma 1 we know that when the duality gap is zero, it must hold $M(\lambda^*) [x^* 1] = 0$. Therefore, in Proposition 3 we solve the linear system (23), trying to obtain $\lambda^*$. When $M(\lambda) \geq 0$ and $d(\lambda) = f(\hat{x})$, then the chain of inequalities (i) becomes tight, $d(\lambda) = d^* = f^* = f(\hat{x})$, implying that $\hat{x}$ attains the optimal objective $f^*$.

In practice, iterative SLAM solvers optimize (6), rather than the primal problem (12). A natural question is then how to use the results in Propositions 2-3 (which relate $f^*$ and $d^*$) to verify the solution of (6), whose optimal value is $f_{\text{ML}} \geq f^*$. This extension is given by the following proposition, which essentially states that Propositions 2-3 can be applied directly to check the solution of (6).

Proposition 4 (Verification techniques for PGO): The following statements hold true:

(V1) Given a candidate solution $\hat{x}$ for the PGO problem (6), if $f_{\text{ML}}(\hat{x}) = d^*$, then $\hat{x}$ is an optimal solution of (6). Moreover, $f_{\text{ML}}(\hat{x}) - d^* \geq f_{\text{ML}}(\hat{x}) - f_{\text{ML}}$, i.e., $f_{\text{ML}}(\hat{x}) - d^*$ is an upper-bound for the sub-optimality gap of $\hat{x}$.

(V2) Given a candidate solution $\hat{x}$ for the PGO problem (6), if the solution $\lambda$ of the linear system (23) is such that $M(\lambda) \geq 0$ and $d(\lambda) = f_{\text{ML}}(\hat{x})$, then the duality gap is zero and $\hat{x}$ is an optimal solution of (6).
Note that Lemma 1 ensures that the linear system (24) admits a solution. Proposition 5 allows finding an optimal solution to the primal problem (12). If this satisfies the determinant constraints in (6), it follows that this solution is also optimal for the original PGO problem (6). While we cannot prove that the determinant constraints are always satisfied, this was always the case in our experiments.

V. EXPERIMENTS

In this section we show that the first verification technique in Proposition 4 (referred to as V1) enables accurate quantification of the sub-optimality gap of a candidate SLAM solution, but that current SDP solvers do not scale well with increasing problem size. The second verification technique in Proposition 4 (referred to as V2) provides a convenient alternative for large-scale problems: it is reliable and less computationally demanding. Finally, we provide empirical evidence that when the duality gap is zero we can compute an optimal solution from the solution of the dual problem, as per Proposition 5. The SDP (22) is solved using SDPA (22).

Effectiveness of V1. Proposition 4 (V1) ensures that for any candidate solution \( \hat{x} \), \( f_{\text{ML}}(\hat{x}) - d^* \) represents an upper-bound on the sub-optimality gap \( f_{\text{ML}}(\hat{x}) - f_{\text{ML}}^* \) of \( \hat{x} \). However, it is possible that this bound is very loose, in which case it would be of little practical utility. In this section, we show that \( d^* \) is close to \( f_{\text{ML}}^* \) in practice, hence \( f_{\text{ML}}(\hat{x}) - d^* \) is a very good measure of the sub-optimality gap of the candidate \( \hat{x} \).

In our experiments we compute the “optimal” solution \( f_{\text{ML}}^* \) of (6) by refining the chordal initialization of (23), (24) with 10 Gauss-Newton (GN) iterations. While one cannot guarantee a priori that this approach always produces the optimal estimate, using the results of this paper we will be able to check optimality a posteriori.

We evaluated how close \( d^* \) is to \( f_{\text{ML}}^* \) on the cube dataset of Fig. 2. In this dataset, the odometric trajectory is simulated as the robot travels on a 3D grid world, and random loop closures are added between nearby nodes, with probability 0.3. Relative pose measurements are obtained by contaminating the true relative poses with zero-mean Gaussian noise, with standard deviation \( \sigma_T \) and \( \sigma_R \) for the translational and rotational noise, respectively. Statistics are computed over 10 runs: for each run we create a cube with random connectivity and random measurement noise. We consider an example with \( n = 5^3 \) poses and varying noise levels \( \sigma_T \) and \( \sigma_R \).

Fig. 2(a1) shows \( d^* \) and \( f_{\text{ML}}^* \) for different translational noise levels, fixing \( \sigma_R = 0.05\text{rad} \). The figure shows that \( d^* = f_{\text{ML}}^* \) (zero duality gap) independently on the translational noise level, hence \( d^* \) is a very good proxy of \( f_{\text{ML}}^* \).

Fig. 2(a2) shows \( d^* \) and \( f_{\text{ML}}^* \) for different rotational noise, fixing \( \sigma_T = 0.1\text{m} \). In this case the duality gap \( f_{\text{ML}}^* - d^* \) is more sensitive to the noise level, and for large rotational noise \( d^* \) becomes smaller than \( f_{\text{ML}}^* \). However, the gap \( f_{\text{ML}}^* - d^* \) remains small, and is within 20\% of \( f_{\text{ML}}^* \) in all cases.

Fig. 2(a3) shows \( d^* \) and \( f_{\text{ML}}^* \) for different sizes of the cube dataset, fixing \( \sigma_R = 0.05\text{rad} \) and \( \sigma_T = 0.1\text{m} \). Again in this case \( d^* = f_{\text{ML}}^* \) (zero duality gap) independently of the size of the dataset. However, we observe that the cost of solving the SDP (22) increases rather rapidly as a function of the problem size, as shown in Fig. 2(a4).

Primal optimal solution via the dual. Here we demonstrate experimentally that one can recover a primal optimal
solution $x^*$ for the SLAM problem from the dual optimal solution $\lambda^*$ of (22) whenever strong duality holds.

For each of the experimental trials described previously, we also computed an estimate $\hat{x}$ for the SLAM solution directly from the optimal solution $\lambda^*$ for the dual program (22) using equation (24). Figs. 2(b1)-(b3) compare the value $f(\hat{x})$ of this estimate against the value $f_{\text{ML}}$ of the solution obtained by solving the SLAM problem directly using the chordal initialization, and against the cost of the odometric guess.

We can see that for those experimental conditions in which strong duality holds (those experiments in Figs. 2(a1)-(a3) for which the green and red bars are the same height), the estimate $\hat{x}$ in fact achieves the (certified) globally optimal cost $f_{\text{ML}}$ and is therefore a globally optimal solution for the SLAM problem, as guaranteed by Proposition 5. More interestingly, we also find that even in those cases when strong duality does not hold (in which case Proposition 5 no longer guarantees that $\hat{x}$ is a primal optimal solution), the quality of the candidate $\hat{x}$ degrades gracefully (i.e. the gap $f(\hat{x}) - f_{\text{ML}}$ increases gradually) with increasing noise levels. In particular, we find that $\hat{x}$ outperforms the odometric guess in all tested cases.

These experiments confirm that we can extract a primal optimal solution $x^*$ for the SLAM problem directly from the solution $\lambda^*$ of the convex dual problem whenever strong duality holds. Unfortunately, this approach is not currently practical for real-time SLAM applications, due to the high computational cost of solving large-scale SDPs.

### Effectiveness of V2
In this section we show that V2 is a computationally tractable verification approach, and it preserves the desirable properties of V1, i.e., it is able to discern optimal estimates from sub-optimal ones. In contrast to V1, V2 does not quantify the sub-optimality gap, but can only give a binary answer: either it certifies that the estimate $\hat{x}$ is optimal (by producing a dual certificate $\lambda$), or it is inconclusive. Before moving to the real tests, we consider the cube scenario discussed in the previous section.

We perform the following test: for each realization of the cube scenario, we compute the optimal solution $x^*$ (attaining $f_{\text{ML}}$) as in the previous tests, and a sub-optimal solution $x^\dagger$ (attaining $f^\dagger > f^\star$) by bootstrapping the Gauss-Newton method with a random initial guess. Then, we apply the verification technique to both $x^\star$ and $x^\dagger$ and see if they pass the optimality test. Using the results we can compute the precision and recall of our classification:

$$\text{precision} = \frac{|\sqrt{x^\star}|}{|\sqrt{x^\star}| + |\sqrt{x^\dagger}|}, \quad \text{recall} = \frac{|\sqrt{x^\star}|}{|\sqrt{x^\star}| + |X \cdot x^\star|}$$

where $|\sqrt{x^\star}|$ denotes the number of tests in which an optimal solution $x^*$ was accepted as optimal by V2, $|\sqrt{x^\dagger}|$ is the number of tests in which a suboptimal solution was accepted as optimal, and $|X \cdot x^\star|$ is the number of tests in which V2 was not able to certify the optimality of an optimal solution.

We point out that Proposition 5(V2) guarantees that our certification approach always has precision equal to 1; however, recall may be less than 1, and will be for cases in which strong duality does not hold. We plot precision/recall for different levels of translational and rotational noise, and for different sizes of the dataset, as shown in Figs. 2(c1)-(c3). We observe that only for larger rotational noise Fig. 2(c2) the recall decreases; these are exactly the cases in which the duality gap is nonzero. (For $\sigma_R = 0.2\text{rad}$, V2 was not able to certify optimality in any case, which means that the precision becomes undefined ($\sqrt{0}$) and for this reason we do not show the corresponding data point in Fig. 2(c2).) Finally, Fig. 2(c4) shows the CPU time required by V2. This is the time required to solve the linear system (24), and check if $M(\lambda) \succeq 0$. V2 is computationally cheap, as it does not require solving the SDP.

We conclude the experimental part of this paper by testing the performance of the verification technique V2 on large-scale SLAM datasets. We consider the same datasets as [23]: the sphere, sphere-a, torus and cube are simulated datasets, while the garage, cubicle and rim are real datasets. For the results in this paper we substituted the covariances in the SLAM datasets. We consider the same datasets as [23]: $\mu$ is the number of tests in which $f_{\text{ML}}(\hat{x}) = f(\hat{x})$ and $\mu \geq 0$ (if the smallest eigenvalue is non-negative, then $M(\lambda) \succeq 0$). We specify a tolerance in these tests, since the GN estimate $\hat{x}$ will not attain the optimal solution $x^\star$ exactly. Consequently, the solution $\lambda$ of (23) is not exact, so we consider that $f_{\text{GN}}(\hat{x}) = d(\hat{x})$ if $|f_{\text{GN}}(\hat{x}) - d(\hat{x})|/f_{\text{ML}}(\hat{x}) < 20\%$. Similarly, we add some tolerance to the condition $\mu \geq 0$, and accept $\mu \geq -1$. Recall that $\mu$ is expected to be slightly negative, as the objective in (23) tends to push $M(\lambda)$ towards the boundary of the positive definite cone, and the SDP is solved numerically.

Let us start our analysis from the sphere dataset. When using the initialization (“Init.” row), the GN method attains an objective $5.7595 \cdot 10^2$. In this case, the two conditions for V2 are satisfied and we can certify the optimality of the resulting estimate (green check-mark in the rightmost column). When

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$f_{\text{GN}}(\hat{x})$</th>
<th>$d(\lambda)$</th>
<th>$\mu$</th>
<th>Time (s)</th>
<th>V2</th>
</tr>
</thead>
<tbody>
<tr>
<td>sphere</td>
<td>5.7595 \cdot 10^2</td>
<td>5.75 \cdot 10^2</td>
<td>-1.8 \cdot 10^{-4}</td>
<td>609</td>
<td>✓</td>
</tr>
<tr>
<td>sphere-a</td>
<td>5.8019 \cdot 10^2</td>
<td>4.38 \cdot 10^2</td>
<td>-1.45 \cdot 10^{-4}</td>
<td>597</td>
<td>X</td>
</tr>
<tr>
<td>torus</td>
<td>1.2485 \cdot 10^0</td>
<td>1.25 \cdot 10^{-3}</td>
<td>-9.64 \cdot 10^{-4}</td>
<td>332</td>
<td>✓</td>
</tr>
<tr>
<td>garage</td>
<td>3.0413 \cdot 10^0</td>
<td>3.04 \cdot 10^{-3}</td>
<td>-1.01 \cdot 10^{-4}</td>
<td>25</td>
<td>✓</td>
</tr>
<tr>
<td>garage-a</td>
<td>2.7666 \cdot 10^4</td>
<td>2.76 \cdot 10^{-4}</td>
<td>-1.04 \cdot 10^{-4}</td>
<td>39</td>
<td>✓</td>
</tr>
<tr>
<td>cube</td>
<td>4.216 \cdot 10^0</td>
<td>4.22 \cdot 10^{-1}</td>
<td>-1.24 \cdot 10^{-4}</td>
<td>1640</td>
<td>✓</td>
</tr>
<tr>
<td>garage</td>
<td>2.7465 \cdot 10^5</td>
<td>2.74 \cdot 10^{-5}</td>
<td>-9.98 \cdot 10^{-4}</td>
<td>68.2</td>
<td>X</td>
</tr>
<tr>
<td>rim</td>
<td>6.994 \cdot 10^0</td>
<td>6.99 \cdot 10^{-2}</td>
<td>-6.78 \cdot 10^{-7}</td>
<td>12.3</td>
<td>✓</td>
</tr>
<tr>
<td>rim-a</td>
<td>6.389 \cdot 10^0</td>
<td>6.39 \cdot 10^{-2}</td>
<td>-6.75 \cdot 10^{-7}</td>
<td>11.2</td>
<td>X</td>
</tr>
<tr>
<td>rim-b</td>
<td>6.2484 \cdot 10^{-2}</td>
<td>6.16 \cdot 10^{-2}</td>
<td>-4.75 \cdot 10^{-1}</td>
<td>15.5</td>
<td>X</td>
</tr>
<tr>
<td>rim-c</td>
<td>1.235 \cdot 10^0</td>
<td>1.23 \cdot 10^{-1}</td>
<td>-9.77 \cdot 10^{-4}</td>
<td>7.7</td>
<td>X</td>
</tr>
<tr>
<td>torus</td>
<td>1.6985 \cdot 10^{-4}</td>
<td>-2.8 \cdot 10^{-4}</td>
<td>-9.38 \cdot 10^{-3}</td>
<td>7.1</td>
<td>X</td>
</tr>
</tbody>
</table>

Table I: Verification technique V2 on large-scale SLAM datasets.

---

4A conservative estimate of the complexity required to compute an $\epsilon$-solution for the SDP (22) using interior point methods is $O(n^{4.5} \log(1/\epsilon))$, assuming that each outer iteration takes $O(n^2)$ [25].
using the odometric initialization the resulting cost is larger (red entry in the column $f_{ML}(\hat{x})$), hence the estimate is suboptimal. $V2$ is able to identify the suboptimality, since $d(\hat{\lambda})$ becomes much smaller than $f_{ML}(\hat{x})$. Hence $V2$ correctly decides not to certify the optimality of the odometric estimate (red “X” in the last column). Similar considerations hold for the second scenario, the challenging sphere-a: in this case the initialized estimate (shown in Fig. 1 top left) is accepted as optimal. The odometric estimate, instead, is trapped in a local minimum (Fig. 1 bottom left), and the optimality test $V2$ correctly rejects the estimate since it leads to a very negative $\mu$. Similar considerations hold for the torus, cube, and the cubicle datasets: our technique is able to discern optimal solutions from suboptimal estimates in all cases.

For the garage dataset, the initialized estimate is classified as optimal, while the odometric estimate, which has a similar cost ($6.2997 \cdot 10^{-1}$ vs $6.2994 \cdot 10^{-1}$), is rejected as suboptimal. We observed the corresponding trajectory estimates (see [17]) and, while they are both visually correct and have very similar costs, they do not overlap. This may indicate the presence of regions in the cost functions that are nearly flat, i.e., for which different estimates can have similar cost. Allowing for extra GN iterations, the odometric estimate converges to the same cost as the initialized one, and our technique is able to certify its optimality. Empirically, we observed that estimates that are suboptimal because they need extra iterations to converge tend to fail the check on $d(\hat{\lambda})$ (compare with sphere and garage), while estimates that converge to a wrong minimum tend to fail the check on $\mu$ (compare with sphere-a, torus, and cube).

For the rim dataset, both estimates (“init” and “odom”) are rejected by $V2$. While the initialized estimate is visually correct, currently, we cannot conclude anything as it might be that there exists a better estimate (i.e., GN failed), or our technique failed because the duality gap was nonzero.

Finally, in the column “Time”, we report the CPU time (in seconds) required to perform the second verification technique. Most of the time here is spent computing the smallest eigenvalue $\mu$ of $M(\hat{\lambda})$, which was obtained using Matlab’s eigs, specifying $-100$ as guess for the eigenvalue. The CPU time depends on the size of the problem, but also depends on the distance between our guess ($-100$) and the closest eigenvalue. We leave a more thorough investigation of these computational aspects for future work.

VI. CONCLUSION

We show that Lagrangian duality is an effective tool to assess the quality of a given SLAM estimate. We propose two techniques to judge if an estimate is globally optimal (i.e., it is the ML estimate), and we show that when the duality gap is zero one can compute an optimal SLAM solution from the dual problem. The performance of our verification techniques is extensively tested in real and simulated datasets, including large-scale benchmarks. Many theoretical questions remain open, e.g., under what conditions is the duality gap zero? Is it possible to derive bounds on the duality gap? We also leave as future work more practical problems, such as the one of exploring more efficient solvers for large SDPs, possibly exploiting problem structure.

REFERENCES