CPL-SLAM: Efficient and Certifiably Correct Planar Graph-Based SLAM Using the Complex Number Representation

Taosha Fan, Hanlin Wang, Michael Rubenstein and Todd Murphey

Abstract—In this paper, we consider the problem of planar graph-based simultaneous localization and mapping (SLAM) that involves both poses of the autonomous agent and positions of observed landmarks. We present CPL-SLAM, an efficient and certifiably correct algorithm to solve planar graph-based SLAM using the complex number representation. We formulate and simplify planar graph-based SLAM as the maximum likelihood estimation (MLE) on the product of unit complex numbers, and relax this nonconvex quadratic complex optimization problem to convex complex semidefinite programming (SDP). Furthermore, we simplify the corresponding complex semidefinite programming to Riemannian staircase optimization (RSO) on the complex oblique manifold that can be solved with the Riemannian trust region (RTR) method. In addition, we prove that the SDP relaxation and RSO simplification are tight as long as the noise magnitude is below a certain threshold. The efficacy of this work is validated through applications of CPL-SLAM and comparisons with existing state-of-the-art methods on planar graph-based SLAM, which indicates that our proposed algorithm is capable of solving planar graph-based SLAM certifiably, and is more efficient in numerical computation and more robust to measurement noise than existing state-of-the-art methods. The C++ code for CPL-SLAM is available at https://github.com/MurpheyLab/CPL-SLAM.

I. INTRODUCTION

Simultaneous localization and mapping (SLAM) estimates poses of an autonomous agent and positions of observed landmarks from noisy measurements [1]–[3]. For an autonomous agent, the ability to construct a map of the environment and concurrently estimate its location within the map is essential to navigation and exploration in unknown scenarios, such as autonomous driving [4], disaster response [5], underwater exploration [6], precision agriculture [7], floor plan building [8], virtual and augmented reality [9], to name a few. An intuitive way to formulate SLAM problems is to use a graph whose vertices are associated with either poses of the autonomous agent or positions of observed landmarks and whose edges are associated with the available measurements [10]. In graph-based SLAM, the estimation problem is usually addressed as a difficult nonconvex optimization problem that involves up to thousands of variables and constraints, and the procedure of solving the optimization problem greatly affects the overall performance of estimation. Even though a number

of optimization methods have been developed [1], [10], [11], it is generally NP-hard to solve a nonconvex optimization problem globally [11], and it is common to get stuck at local minima in solving graph-based SLAM, which results in bad estimates.

In robotics, most graph-based SLAM techniques rely on local search methods for nonlinear optimization to estimate poses of the autonomous agent and positions of observed landmarks. Lu and Milios [12] formulate SLAM as pose graph optimization (PGO) and use iterative nonlinear optimization methods to solve PGO. Duckett and Frese et al. [13], [14] exploit the sparsity of graph-based SLAM and propose relaxations for the resulting nonlinear optimization problem. Olson et al. [15] propose a stochastic gradient descent on an alternative state space representation of graph-based SLAM that has good stability and scalability. Grisetti et al. [16] extend Olson's work by presenting a novel tree parametrization that improves the convergence of stochastic gradient descent. Fan and Murphey [17] propose an accelerated proximal method for pose graph optimization. Dellaert and Kaess et al. [18]–[21] propose incremental smoothing algorithms that enable online updates of large-scale graph-based SLAM with nonlinear optimization. Huang and Wang et al. [22], [23] study the least-square structure of graph-based SLAM and indicate the possibility of reducing the nonlinearity and nonconvexity of SLAM. Kümmerle et al. [24] propose g^2o framework that solves graph-based SLAM using Gauss-Newton method. Carlone et al. [25], [26] propose approximations for planar pose graph optimization that reduce the risks of getting stuck at local minima. Khosoussi et al. [27] exploit the separable structure of SLAM problems using variable projection and propose algorithms to improve the efficiency of Gauss-Newton methods. However, all of the aforementioned nonlinear optimization techniques are local search methods, and as a result, there are no guarantees of the correctness for the resulting solutions.

To address the issues of local minima in nonlinear optimization, several efforts have been made to relax graph-based SLAM as convex optimization problems. Liu et al. [28] propose a suboptimal convex relaxation to solve SLAM problems. Carlone et al. [29]–[31] propose a tight semidefinite relaxation and analyze its optimality using Lagrangian duality. Briales et al. [32] present a fast method for the optimality verification of 3D PGO based on [31]. A further breakthrough of the semidefinite relaxation of PGO is made in [33] that results in a fast and certifiable algorithm for pose graph optimization. Rosen et al. propose SE-Sync to solve the semidefinite relaxation of PGO using Riemannian staircase optimization on the Stiefel manifold that is orders of magnitude faster than interior point methods [33]. Furthermore, it is shown in [33]

Taosha Fan and Todd Murphey are with the Department of Mechanical Engineering, Northwestern University, Evanston, IL 60201, USA. E-mail: taosha.fan@u.northwestern.edu, t-murphey@northwestern.edu

Hanlin Wang is with the Department of Computer Science, Northwestern Univerity, Evanston, IL 60201, USA. E-mail: hanlinwang@u.northwestern.edu

Michael Rubenstein is with the Departments of Computer Science and Mechanical Engineering, Northwestern Univerity, Evanston, IL 60201, USA. E-mail: rubenstein@northwestern.edu

that the semidefinite relaxation of PGO is tight as long as the magnitude of the measurement noise is below a certain threshold. Similar to SE-Sync, Briales et al. [34] propose Cartan-Sync that uses the Cartan motion group and introduce a novel preconditioner to accelerate the algorithm. Mangelson et al. [35] formulate planar pose graph and landmark SLAM using sparse-bounded sum of squares programming that is guaranteed to find the globally optimal solution regardless of the noise level.

In fields other than robotics, problems such as angular and rotation synchronization that share a similar mathematical formulation with graph-based SLAM have been extensively studied. Singer et al. [36], [37] propose semidefinite relaxations to solve angular and rotation synchronization by finding the eigenvectors that correspond to the greatest eigenvalues. Bandeira et al. [38] prove the tightness of the semidefinite relaxation of angular synchronization and show that the Riemannian staircase optimization is significantly more scalable to solve the resulting problem. Boumal [39] proposes the generalized power method that can recover the globally optimal solution to angular synchronization. Eriksson et al. [40] explore the role of strong duality in rotation averaging, which has important applications in computer vision.

In applied mathematics, it is common to use unit complex numbers in synchronization problems over $SO(2)$ [38], [39]. In robotics, it is not new to use the complex number representation in planar robot localization and mapping problems, either. Betke et al. [41] use the complex number to represent positions of landmarks to localize a mobile robot with bearing measurements. Carlone et al. [30] use the complex number representation of $SO(2)$ and $SE(2)$ to verify the optimality of planar PGO and the tightness of semidefinite relaxations, and the analysis is much clearer and simpler than that using the matrix representation, and to our knowledge, this is the first implementation of the complex number representation in planar PGO.

In general, a certifiably correct algorithm for an optimization problem not only finds a solution to the problem, but also is capable of certifying the global optimality of the resulting solution [42]. For many estimation problems, it is usually intractable to attain a globally optimal solution and we have to either solve these problems using local search methods, or relax them to a more reasonable formulation. As a result, the certifiable correctness of the algorithm is important for estimation problems in which globally optimal solutions are preferred. Even though a number of optimization methods are proposed to planar graph-based SLAM, to our knowledge, only [29]–[35] are certifiably correct.

In this paper, we consider the problem of planar graph-based SLAM that involves both poses of the autonomous agent and positions of observed landmarks. We present CPL-SLAM, which means the **ComPLex** number **S**imultaneous **L**ocalization **A**nd **M**apping, an efficient and certifiably correct algorithm to solve planar graph-based SLAM using the complex number representation.

This paper is built upon the works of [30], [33], [38] that use the complex number representation, the semidefinite relaxation and the Riemannian staircase optimization [43] to efficiently

and certifiably correctly solve large-scale estimation problems. In [30], Carlone et al. were first to formulate planar PGO using the complex number representation of $SE(2)$, in which the optimality and tightness of the semidefinite relaxation are studied; in CPL-SLAM, we use the same representation of $SE(2)$ as the one in [30]. In [33], Rosen et al. analyze the optimality and tightness of the semidefinite relaxation of PGO and introduce the Riemannian staircase optimization to solve the semidefinite relaxation, which, though using the matrix representation, motivates this paper. In [38], Bandeira et al. prove the tightness of semidefinite relaxation of phase synchronization on $SO(2)$ using the complex number representation, which is helpful to our theoretical analysis of CPL-SLAM.

In graph-based SLAM, poses are special Euclidean groups $SE(d)$ [10], [21], [24], [33], [34], which are isomorphic to a semidirect product of real space $\mathbb{R}^d$ and special orthogonal groups $SO(d)$ [44]. In general, it is possible to identify $SE(d)$ as a pair (t, R), in which the translation is represented as a real vector t in $\mathbb{R}^d$ and the rotation is represented as a matrix R in $SO(d)$, and such an identification results in the matrix representation of $SE(d)$ that is commonly used in robotics. However, for planar graph-based SLAM, the matrix representation of $SO(2)$ is redundant, which needs four real numbers, whereas a unit complex number that can be represented with two real numbers is sufficient to capture the topological and geometric structures of $SO(2)$ [30], [45]. Furthermore, as is later shown in this paper, the complex number representation of $SO(2)$ and $SE(2)$ brings significant analytical and computational benefits, and renders the resulting CPL-SLAM algorithm a lot more efficient in numerical computation and much more robust to measurement noise. As a result, the CPL-SLAM outperforms existing methods of planar graph-based SLAM in terms of both numerical scalability and theoretical guarantees.

In contrast to the state-of-the-art local search methods in [12]–[16], [18]–[26], CPL-SLAM is a lot faster and capable of certifying the correctness of the solutions. As for [30], [33], [35] that also seek to use convex relaxation to certifiably correctly solve graph-based SLAM, CPL-SLAM has better scalability, and more explicitly, CPL-SLAM is expected to be several orders of magnitude faster than [30], [35] and several times faster than [33]. Moreover, [30], [33] are designed for pose-only problems, whereas CPL-SLAM extends the works of [30], [33] by accepting pose-landmark measurements. Even though [30] uses complex semidefinite relaxation to verify the optimality and tightness of planar pose graph optimization, we present stronger, more complete and more concise theoretical results and more scalable algorithms to solve the complex semidefinite relaxation. Furthermore, the conciseness of the complex number representation makes the semidefinite relaxation in CPL-SLAM much tighter than that in [33] using the matrix representation, and thus, CPL-SLAM has greater robustness to measurement noise.

This paper extends the preliminary results of [46] in which we only use the complex number representation to solve planar pose graph optimization without landmarks. In this paper, our proposed CPL-SLAM uses the graph-based SLAM measure-

ment model in [33], [34], [46] and can handle planar graph-based SLAM with landmarks. Similar to [33], [34], [46] for pose graph optimization, CPL-SLAM is a certifiable algorithm that is guaranteed to attain the globally optimal solution to planar graph-based SLAM with landmarks as long as the magnitude of measurement noise is below a certain threshold. Furthermore, even though it is not new to involve landmarks in graph-based SLAM [10], [16], [47], we propose a novel preconditioner making better use of translation and landmark information in planar graph-based SLAM. As a result, the performance of the truncated conjugate gradient method is improved. In addition, we also provide additional proofs of lemmas and propositions, extensive experimental results on numerous datasets and much more detailed discussions, all of which are not covered in [46].

In summary, the contributions of this paper are the following:

1) We formulate planar graph-based SLAM with poses of the autonomous agent and positions of observable landmarks using the complex number representation and simplify the resulting estimation problem as an optimization problem on the product of unit complex numbers.
2) We relax the nonconvex optimization problem as complex semidefinite programming and prove that the complex semidefinite relaxation is tight as long as the magnitude of measurement noise is below a certain threshold.
3) We recast the complex semidefinite programming as a series of rank-restricted complex semidefinite programming on complex oblique manifolds that can be efficiently solved with the Riemannian staircase optimization [43], and it is almost guaranteed to retrieve the true solution to the complex semidefinite programming if the rank of the Riemannian staircase optimization is appropriately selected.
4) The resulting CPL-SLAM algorithm is certifiably correct, and more importantly, a lot faster in numerical computation and much more robust to measurement noise than existing state-of-the-art methods [12]–[16], [18]–[26], [30], [33], [47].

The rest of this paper is organized as follows. Section II introduces notations that are used throughout this paper. Section III reviews the complex number representation of $SO(2)$ and $SE(2)$. Section IV formulates planar graph-based SLAM with poses of the autonomous agent and positions of observable landmarks using the complex representation and Section V relaxes planar graph-based SLAM to complex semidefinite programming. Section VI presents the CPL-SLAM algorithm to solve planar graph-based SLAM. Section VII presents and discusses comparisons of CPL-SLAM with existing methods [21], [33], [47] on a series of simulated Tree and City datasets and a suite of large 2D simulated and real-world SLAM benchmark datasets. The conclusions are made in Section VIII.

II. NOTATION

$\mathbb{R}$ and $\mathbb{C}$ denote the sets of real and complex numbers, respectively; $\mathbb{R}^{m \times n}$ and $\mathbb{C}^{m \times n}$ denote the sets of $m \times n$ real and complex matrices, respectively; $\mathbb{R}^n$ and $\mathbb{C}^n$ denote the sets of $n \times 1$ real and complex vectors, respectively. $\mathbb{C}_1$ and $\mathbb{C}_1^n$ denote the sets of unit complex numbers and $n \times 1$ vectors over unit complex numbers, respectively. $\mathbb{P}$ denotes the group of $(\mathbb{C}, +) \rtimes (\mathbb{C}_1, \cdot)$ in which "$\rtimes$" denotes the semidirect product of groups [44] under complex number addition "$+$" and multiplication "$\cdot$". $\mathbb{S}^n$ and $\mathbb{H}^n$ denote the sets of $n \times n$ real symmetric matrices and complex Hermitian matrices, respectively. The notation "$\mathbf{i}$" is reserved for the imaginary unit of complex numbers. The notation $|\cdot|$ denotes the absolute value of real and complex numbers, and the notation $\overline{(\cdot)}$ denote the conjugate of complex numbers. The superscripts $(\cdot)^\top$ and $(\cdot)^H$ denote the transpose and conjugate transpose of a matrix, respectively. For a complex matrix W, $[W]_{ij}$ denotes its (i, j)-th entry; the notations $\Re(W)$ and $\Im(W)$ denote real matrices such that $W = \Re(W) + \Im(W)\mathbf{i}$; $W \succcurlyeq 0$ means that W is Hermitian and positive semidefinite; $\mathrm{trace}(W)$ denotes the trace of W; $\mathrm{diag}(W)$ extracts the diagonal of W into a vector and $\mathrm{ddiag}(W)$ sets all off-diagonal entries of W to zero; the notations $\|W\|_F$ and $\|W\|_2$ denote the Frobenius norm and the induced-2 norm, respectively. The notation $\langle \cdot, \cdot \rangle$ denotes the real inner product of matrices. For a vector v, the notation $[v]_i$ denotes its i-th entry; $\|v\|^2 = \|v\|_2^2 = \sqrt{\sum_i |[v]_i|^2} = \sqrt{v^H v}$; the notation $\mathrm{diag}(v)$ denotes the diagonal matrix with $[\mathrm{diag}(v)]_{ii} = v_i$. The notation $\mathbf{1} \in \mathbb{C}^n$ denotes the vector of all-ones. The notation $\mathbf{0} \in \mathbb{C}^n$ denotes the vector of all-zeros. The notation $\mathbf{I} \in \mathbb{C}^{n \times n}$ denotes the identity matrix. The notation $\mathbf{O} \in \mathbb{C}^{n \times n}$ denotes the zero matrix. For a hidden parameter x whose value we wish to infer, the notations $\underline{x}$, $\tilde{x}$ and $\hat{x}$ denote the true value of x, a noisy observation of x and an estimate of x, respectively.

III. THE COMPLEX NUMBER REPRESENTATION OF $SO(2)$ AND $SE(2)$

In this section, we give a brief review of $SO(2)$ and $SE(2)$, and show that $SO(2)$ and $SE(2)$ can be represented using complex numbers. It should be noted that the complex number representation used in this paper, though presented in a different way, is in fact equivalent to that in [30].

It is known that the set of unit complex numbers

$$\mathbb{C}_1 \triangleq \{a_1 + a_2 \mathbf{i} \in \mathbb{C} | a_1^2 + a_2^2 = 1\}$$

forms a group under complex number multiplication "$\cdot$" for which the identity is 1 and the inverse is the conjugate, i.e., for $z, z' \in \mathbb{C}_1$, we obtain [45]

$$z \cdot z' \in \mathbb{C}_1, \quad 1 \cdot z = z \cdot 1 = z, \quad z \cdot \overline{z} = \overline{z} \cdot z = 1.$$

In addition, the group of unit complex numbers $(\mathbb{C}_1, \cdot)$ is diffeomorphic and isomorphic to the matrix Lie group $SO(2)$:

$$SO(2) \triangleq \{\begin{bmatrix} a_1 & -a_2 \\ a_2 & a_1 \end{bmatrix} \in \mathbb{R}^{2 \times 2} | a_1^2 + a_2^2 = 1\}$$
$$\triangleq \{R \in \mathbb{R}^{2 \times 2} | R^\top R = \mathbf{I}, \det(R) = 1\}$$

under matrix multiplication. As a result, $SO(2)$ can be represented using unit complex numbers $\mathbb{C}_1$. More explicitly, if $R \in SO(2)$ is

$$R = \begin{bmatrix} a_1 & -a_2 \\ a_2 & a_1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \tag{1}$$

the corresponding unit complex number representation $z \in \mathbb{C}_1$ is

$$z = a_1 + a_2\mathbf{i} = e^{\mathbf{i}\theta} = \cos\theta + \sin\theta\mathbf{i} \tag{2}$$

in which $e^{\mathbf{i}\theta} = \cos\theta + \sin\theta\mathbf{i}$. Furthermore, if $b' = \begin{bmatrix} b_1' & b_2' \end{bmatrix}^\top \in \mathbb{R}^2$ is rotated by $R \in SO(2)$ in Eq. (1) from $b = \begin{bmatrix} b_1 & b_2 \end{bmatrix}^\top \in \mathbb{R}^2$, i.e.,

$$b_1' = a_1 b_1 - a_2 b_2 = b_1 \cos\theta - b_2 \sin\theta,$$
$$b_2' = a_1 b_2 + a_2 b_1 = b_2 \cos\theta + b_1 \sin\theta,$$

we obtain

$$s' = z \cdot s = \underbrace{a_1 b_1 - a_2 b_2}_{b_1'} + \underbrace{(a_1 b_2 + a_2 b_1)}_{b_2'}\mathbf{i}, \tag{3a}$$

or equivalently,

$$\begin{aligned} s' = z \cdot s &= e^{\mathbf{i}\theta} \cdot s \\ &= \underbrace{b_1 \cos\theta - b_2 \sin\theta}_{b_1'} + \underbrace{(b_2 \cos\theta + b_1 \sin\theta)}_{b_2'}\mathbf{i}, \end{aligned} \tag{3b}$$

in which z is a unit complex number as that given in Eq. (2), and

$$s = b_1 + b_2\mathbf{i} \quad \text{and} \quad s' = b_1' + b_2'\mathbf{i} \tag{4}$$

are the complex number representation of b and b', respectively. As a result, rotating a vector can also be described using the complex number representation.

In general, the special Euclidean group $SE(2)$ is the matrix Lie group

$$SE(2) \triangleq \left\{ \begin{bmatrix} R & t \\ \mathbf{0} & 1 \end{bmatrix} \in \mathbb{R}^{3\times3} | R \in SO(2), t \in \mathbb{R}^2 \right\}, \tag{5}$$

whose group multiplication is matrix multiplication. In terms of group theory, $SE(2)$ is also represented as the semidirect product of $(\mathbb{R}^2, +)$ and $SO(2)$:

$$SE(2) \triangleq (\mathbb{R}^2, +) \rtimes SO(2),$$

in which "$\rtimes$" denotes the semidirect product of groups under vector addition and matrix multiplication [44] and whose group multiplication "$\circ$" using the matrix representation of Eq. (5) is defined to be

$$g \circ g' = (Rt' + t, RR'), \tag{6}$$

in which $g = (t, R)$, $g' = (t', R') \in SE(2)$.[1] Following the complex number representation of $SO(2)$ and $\mathbb{R}^2$, the representation of $SE(2)$ as Eq. (5) is diffeomorphic and isomorphic to the semidirect product of $(\mathbb{C}, +)$ and $(\mathbb{C}_1, \cdot)$:

$$\mathbb{P} \triangleq (\mathbb{C}, +) \rtimes (\mathbb{C}_1, \cdot),$$

[1] In group theory, the definition of the semidirect product relies on the choice of the group multiplication rule.

whose group multiplication "$\odot$" is defined to be

$$\rho \odot \rho' = (z \cdot c' + c, z \cdot z') \in \mathbb{P}, \tag{7}$$

in which $\rho = (c, z)$, $\rho' = (c', z') \in \mathbb{P}$. In Eq. (7), $z, z' \in \mathbb{C}_1$ and $c, c' \in \mathbb{C}$ are the complex number representation of $R, R' \in SO(2)$ and $t, t' \in \mathbb{R}^2$, respectively, which follow the same representation as that in Eqs. (2) and (4). It is obvious from Eqs. (2) and (3) that the group multiplication of $\rho \odot \rho'$ in Eq. (7) is equivalent to that of $g \circ g'$ in Eq. (6). Furthermore, the identity of $\mathbb{P}$ is $(0, 1) \in \mathbb{P}$ and the inverse of $\rho = (c, z) \in \mathbb{P}$ is

$$\rho^{-1} = (-\overline{z} \cdot c, \overline{z}) \in \mathbb{P}. \tag{8}$$

As a result, instead of using the matrix representation, we represent $SE(2)$ with a 2-tuple of complex numbers. In addition, if $b' \in \mathbb{R}^2$ is transformed by $g \in SE(2)$ from $b \in \mathbb{R}^2$, we obtain

$$s' = z \cdot s + c,$$

in which $\rho = (c, z) \in \mathbb{P}$ is the complex number representation of $g \in SE(2)$, and s and s' are the complex number representation of b and b', respectively.

For notational convenience, in the rest of paper, we will omit the complex number multiplication "$\cdot$" if there is no ambiguity.

In terms of the computation of group multiplication and transformation only, the complex number representation of $SO(2)$ and $SE(2)$ has the same complexity as the matrix representation. In spite of this, as shown in the following sections, the complex number representation greatly simplifies the analysis for planar graph-based SLAM, and most importantly, the semidefinite relaxation and Riemannian optimization of planar graph-based SLAM using the complex number representation is simpler for problem formulation, more efficient in numerical computation and more robust to measurement noise than that using the matrix representation in [33].

In the following sections, we will use the complex number representation of $SO(2)$ and $SE(2)$ to formulate and solve planar graph-based SLAM.

IV. PROBLEM FORMULATION AND SIMPLIFICATION

In this section, we formulate planar graph-based SLAM as maximum likelihood estimation, and further simplify it to complex quadratic programming on the product of unit complex numbers.

A. Problem Formulation

Planar graph-based SLAM consists of estimating n unknown poses $g_1, g_2, \cdots, g_n \in SE(2)$, in which $g_{(\cdot)} = (t_{(\cdot)}, R_{(\cdot)})$ with $t_{(\cdot)} \in \mathbb{R}^2$ and $R_{(\cdot)} \in SO(2)$, and n' landmark positions $l_1, l_2, \cdots, l_{n'} \in \mathbb{R}^2$ given m noisy pose-pose measurements $\tilde{g}_{ij} \in SE(2)$ of

$$g_{ij} \triangleq g_i^{-1} g_j \in SE(2)$$

and m' noisy pose-landmark measurements $\tilde{l}_{ij} \in \mathbb{R}^2$ of

$$l_{ij} \triangleq R_i^\top (l_j - t_i) \in \mathbb{R}^2. \tag{9}$$

From Section III, the problem is equivalent to estimating n 2-tuples of complex numbers $\rho_1, \rho_2, \cdots, \rho_n \in \mathbb{P}$, in which

$\rho_{(\cdot)} = (c_{(\cdot)}, z_{(\cdot)}) \in \mathbb{P}$ with $c_{(\cdot)} \in \mathbb{C}$ and $z_{(\cdot)} \in \mathbb{C}_1$, and n' complex numbers $s_1, s_2, \cdots, s_{n'} \in \mathbb{C}$ given m noisy pose-pose measurements $\tilde{\rho}_{ij} \in \mathbb{P}$ of

$$\rho_{ij} \triangleq \rho_i^{-1} \odot \rho_j \in \mathbb{P}$$

and m' noisy pose-landmark measurements $\tilde{s}_{ij} \in \mathbb{C}$ of

$$s_{ij} \triangleq \overline{z}_i(s_j - c_i) \in \mathbb{C}.$$

The unknown n poses and n' landmark positions and the noisy relative measurements can be described with a directed graph $\overrightarrow{G} = (\mathcal{V} \cup \mathcal{V}', \overrightarrow{\mathcal{E}} \cup \overrightarrow{\mathcal{E}'})$ in which $i \in \mathcal{V} \triangleq \{1, \cdots, n\}$ is associated with g_i or ρ_i, and $i \in \mathcal{V}' = \{1, \cdots, n'\}$ is associated with l_i or s_i, and $(i, j) \in \overrightarrow{\mathcal{E}} \subset \mathcal{V} \times \mathcal{V}$ if and only if the pose-pose measurement $\tilde{g}_{ij}$ or $\tilde{\rho}_{ij}$ exists, and $(i, j) \in \overrightarrow{\mathcal{E}'} \subseteq \mathcal{V} \times \mathcal{V}'$ if and only if the pose-landmark measurement $\tilde{l}_{ij}$ or $\tilde{s}_{ij}$ exists. If the orientation of edges in $\overrightarrow{\mathcal{E}}$ and $\overrightarrow{\mathcal{E}'}$ are ignored, we obtain the undirected graph of $\overrightarrow{G}$ that is denoted as $G = (\mathcal{V} \cup \mathcal{V}', \mathcal{E} \cup \mathcal{E}')$. In the rest of this paper, we assume that $\overrightarrow{G}$ is weakly connected and G is (equivalently) connected. In addition, we assume that the noisy relative measurements $\tilde{\rho}_{ij} = (\tilde{c}_{ij}, \tilde{z}_{ij})$ and $\tilde{s}_{ij}$ are random variables that satisfy

$$\tilde{c}_{ij} = \underline{c}_{ij} + c_{ij}^\epsilon \qquad c_{ij}^\epsilon \sim N(0, \tau_{ij}^{-1}), \qquad (10a)$$

$$\tilde{z}_{ij} = \underline{z}_{ij} z_{ij}^\epsilon \qquad z_{ij}^\epsilon \sim \text{vMF}(1, \kappa_{ij}), \qquad (10b)$$

$$\tilde{s}_{ij} = \underline{s}_{ij} + s_{ij}^\epsilon \qquad s_{ij}^\epsilon \sim N(0, \nu_{ij}^{-1}), \qquad (10c)$$

for all $(i, j) \in \overrightarrow{\mathcal{E}} \cup \overrightarrow{\mathcal{E}'}$. In Eq. (10), $\underline{\rho}_{ij} = (\underline{c}_{ij}, \underline{z}_{ij})$ and $\underline{s}_{ij}$ are the true (latent) values of ρ_{ij} and s_{ij}, respectively, $N(\mu, \Sigma)$ denotes the complex normal distribution with mean $\mu \in \mathbb{C}$ and covariance $\Sigma \succcurlyeq 0$, and $\text{vMF}(z_0, \kappa)$ denotes the von Mises-Fisher distribution on $\mathbb{C}_1$ with mode $z_0 \in \mathbb{C}_1$, concentration number $\kappa \geq 0$ and the probability density function of $\text{vMF}(z_0, \kappa)$ is [48]

$$f(z; z_0, \kappa) = \frac{1}{c_d(\kappa)} \exp\left(\kappa(\overline{z}_0 z + z_0 \overline{z})\right),$$

in which $c_d(\kappa)$ is a function of κ.

If $\tilde{c}_{ij}$, $\tilde{z}_{ij}$ and $\tilde{s}_{ij}$ are independent from each other, from Eqs. (3), (7) and (8), a straightforward algebraic manipulation indicates that the maximum likelihood estimation (MLE) is a least square problem as follows

$$\min_{\substack{s_i \in \mathbb{C}, \\ c_i \in \mathbb{C}^n, z_i \in \mathbb{C}_1^n}} \sum_{(i,j) \in \overrightarrow{\mathcal{E}}} \left[\kappa_{ij}|z_i \tilde{z}_{ij} - z_j|^2 + \tau_{ij}|c_j - c_i \right.$$
$$\left. - z_i \tilde{c}_{ij}|^2\right] + \sum_{(i,j) \in \overrightarrow{\mathcal{E}'}} \nu_{ij}|s_j - c_i - z_i \tilde{s}_{ij}|^2, \quad \text{(MLE)}$$

in which κ_{ij}, τ_{ij} and ν_{ij} are as given in Eqs. (10a) to (10c). From Eqs. (1) and (2), it should be noted that

$$|z_i \tilde{z}_{ij} - z_j|^2 = \frac{1}{2}\|R_i \widetilde{R}_{ij} - R_j\|_F^2,$$

and it is also trivial to show that

$$|c_j - c_i - z_i \tilde{c}_{ij}|^2 = \|t_j - t_i - R_i \tilde{t}_{ij}\|_F^2,$$

$$|s_j - c_i - z_i \tilde{s}_{ij}|^2 = \|l_j - t_i - R_i \tilde{l}_{ij}\|_F^2.$$

As a result, (MLE) is equivalent to

$$\min_{\substack{R_i \in SO(2), \\ p_i, l_i \in \mathbb{R}^2}} \sum_{(i,j) \in \overrightarrow{\mathcal{E}}} \left[\frac{\kappa_{ij}}{2}\|R_i \widetilde{R}_{ij} - R_j\|_F^2 + \tau_{ij}\|t_j - t_i - \right.$$
$$\left. R_i \tilde{t}_{ij}\|_F^2\right] + \sum_{(i,j) \in \overrightarrow{\mathcal{E}'}} \nu_{ij}\|l_j - t_i - R_i l_{ij}\|_F^2. \quad \text{(SE-MLE)}$$

Even though there are landmarks present in (MLE) and (SE-MLE), this does not create a significant distinction from [33] in terms of problem formulation. As a matter of fact, if there are no landmarks, i.e., $\mathcal{V}' = \emptyset$ and $\overrightarrow{\mathcal{E}'} = \emptyset$, (SE-MLE) is almost the same as the formulation of pose graph optimization using the matrix representation in SE-Sync [33] except for the weight factors. In addition, (SE-MLE) can also be constructed as a specialized case of SE-Sync's [33] measurement model if we interpret pose-landmark measurements as pose-pose measurements whose rotational weight factors are zero.

In the next subsection, we will simplify (MLE) to quadratic programming on the product of unit complex numbers $\mathbb{C}_1^n$.

B. Problem Simplification

The simplification of (MLE) is similar to that of [33, Appendix B], the difference of which is that ours uses the complex number representation while [33] uses the matrix representation and ours has landmarks involved while [33] does not.

For notational convenience, we define $z_{ji} = \overline{z}_{ij}$, $\kappa_{ji} = \kappa_{ij}$ and $\tau_{ji} = \tau_{ij}$, and (MLE) can be reformulated as

$$\min_{\xi \in \mathbb{C}^{n'} \times \mathbb{C}^n \times \mathbb{C}_1^n} \xi^H \widetilde{\Gamma} \xi \qquad \text{(P)}$$

in which

$$\xi \triangleq \begin{bmatrix} s_1 & \cdots & s_{n'} & c_1 & \cdots & c_n & z_1 & \cdots & z_n \end{bmatrix}^\top.$$

In (P), $\widetilde{\Gamma}$ is a $(2n + n')$-by-$(2n + n')$ Hermitian matrix

$$\widetilde{\Gamma} \triangleq \begin{bmatrix} \Sigma^s & U & \widetilde{N} \\ * & L(W^c) + \Sigma^c & \widetilde{E} \\ * & * & L(\widetilde{G}^z) + \widetilde{\Sigma}^z \end{bmatrix}, \qquad (11)$$

in which $\widetilde{\Sigma}^s \in \mathbb{H}^{n'}$, $\widetilde{U} \in \mathbb{C}^{n' \times n}$, $\widetilde{N} \in \mathbb{C}^{n' \times n}$, $L(W^c) \in \mathbb{H}^n$, $\widetilde{\Sigma}^c \in \mathbb{H}^n$, $\widetilde{E} \in \mathbb{C}^{n \times n}$, $L(\widetilde{G}^z) \in \mathbb{H}^n$ and $\widetilde{\Sigma}^z \in \mathbb{H}^n$ are defined as

$$[\Sigma^s]_{ij} \triangleq \begin{cases} \sum_{(k,i) \in \overrightarrow{\mathcal{E}'}} \nu_{ki}, & i = j, \\ 0 & \text{otherwise}, \end{cases}$$

$$[U]_{ij} \triangleq \begin{cases} -\nu_{ji}, & (j, i) \in \overrightarrow{\mathcal{E}'}, \\ 0 & \text{otherwise}, \end{cases}$$

$$[\widetilde{N}^s]_{ij} \triangleq \begin{cases} -\nu_{ji} \tilde{s}_{ji}, & (j, i) \in \overrightarrow{\mathcal{E}'}, \\ 0 & \text{otherwise}, \end{cases}$$

$$[L(W^c)]_{ij} \triangleq \begin{cases} \sum_{(i,k) \in \mathcal{E}} \tau_{ik}, & i = j, \\ -\tau_{ij}, & (i, j) \in \mathcal{E}, \\ 0 & \text{otherwise}, \end{cases}$$

$$[\Sigma^c]_{ij} \triangleq \begin{cases} \sum_{(i,k)\in\overrightarrow{\mathcal{E}'}} \nu_{ik}, & i=j, \\ 0 & \text{otherwise}, \end{cases}$$

$$[\widetilde{E}]_{ij} \triangleq \begin{cases} \sum_{(i,k)\in\overrightarrow{\mathcal{E}}} \tau_{ik}\tilde{c}_{ik}+ \\ \quad + \sum_{(i,k)\in\overrightarrow{\mathcal{E}'}} \nu_{ik}\tilde{s}_{ik}, & i=j, \\ -\tau_{ij}\tilde{c}_{ji}, & (j,i)\in\overrightarrow{\mathcal{E}}, \\ 0 & \text{otherwise}, \end{cases}$$

$$[L(\widetilde{G}^z)]_{ij} \triangleq \begin{cases} \sum_{(i,k)\in\mathcal{E}} \kappa_{ik}, & i=j, \\ -\kappa_{ij}\tilde{z}_{ji}, & (i,j)\in\mathcal{E}, \\ 0 & \text{otherwise}, \end{cases}$$

$$[\widetilde{\Sigma}^z)]_{ij} \triangleq \begin{cases} \sum_{(i,k)\in\overrightarrow{\mathcal{E}}} \tau_{ik}|\tilde{c}_{ik}|^2+ \\ \quad + \sum_{(i,k)\in\overrightarrow{\mathcal{E}'}} \nu_{ik}|\tilde{s}_{ik}|^2, & i=j \\ 0 & \text{otherwise}, \end{cases}$$

respectively.

It is possible to marginalize the translational states and landmarks and reformulate planar graph-based SLAM as an optimization problem on the rotational states only, which has been used in [27], [33], [49] to improve the computational efficiency. In a similar way, if rotational states $z \triangleq \begin{bmatrix} z_1 & \cdots & z_n \end{bmatrix}^\top \in \mathbb{C}_1^n$ are known, (P) is reduced to unconstrained complex quadratic programming on translational states $c \triangleq \begin{bmatrix} c_1 & \cdots & c_n \end{bmatrix}^\top \in \mathbb{C}^n$ and landmark positions $s \triangleq \begin{bmatrix} s_1 & \cdots & s_{n'} \end{bmatrix}^\top \in \mathbb{C}^w$:

$$\min_{\beta\in\mathbb{C}^{n'+n}} \beta^H\Lambda\beta + 2\langle\beta,\widetilde{\Theta}z\rangle + \underbrace{z^H L(\widetilde{G}^z)z + z^H\widetilde{\Sigma}^z z}_{\text{constant}}, \quad (12)$$

in which $\beta \triangleq \begin{bmatrix} s^\top & c^\top \end{bmatrix}^\top \in \mathbb{C}^{n+n'}$, $\widetilde{\Theta} \triangleq \begin{bmatrix} \widetilde{N} \\ \widetilde{E} \end{bmatrix} \in \mathbb{C}^{(n+n')\times n}$, and $\Lambda \triangleq \begin{bmatrix} \Sigma^s & U \\ * & L(W^c)+\Sigma^c \end{bmatrix} \in \mathbb{H}^{n+n'}$. It can be shown that according to [50, Proposition 4.2][2], one of the optimal solutions to Eq. (12) is

$$\beta = -\Lambda^\dagger\widetilde{\Theta}z. \quad (13)$$

Substituting Eq. (13) into (P) and simplifying the resulting equation, we obtain the complex quadratic programming on the product of unit complex numbers $\mathbb{C}_1^n$ as follows

$$\min_{z\in\mathbb{C}_1^n} z^H\widetilde{M}z, \quad (14)$$

in which $\widetilde{M} = L(\widetilde{G}^z) + \widetilde{\Sigma}^z - \widetilde{\Theta}^H\Lambda^\dagger\widetilde{\Theta} \succeq 0$.

---

[2]It should be noted that [50, Proposition 4.2] was originally derived for real matrices, however, the results can be generalized to complex matrices as well.

Furthermore, let $\Omega \in \mathbb{R}^{(m+m')\times(m+m')}$ be the diagonal matrix indexed by $e \in \overrightarrow{\mathcal{E}} \cup \overrightarrow{\mathcal{E}}'$ and $e' \in \overrightarrow{\mathcal{E}} \cup \overrightarrow{\mathcal{E}}'$ whose (e,e')-element is given by

$$[\Omega]_{ee'} \triangleq \begin{cases} \nu_e, & e=e' \text{ and } e\in\overrightarrow{\mathcal{E}'}, \\ \tau_e, & e=e' \text{ and } e\in\overrightarrow{\mathcal{E}}, \\ 0, & \text{otherwise}, \end{cases} \quad (15)$$

in which ν_e and $\tau_e \in \mathbb{R}$ are the precisions of the landmark positional observations and the translational observations as given in Eqs. (10c) and (10a), respectively; and let $\widetilde{T} \in \mathbb{C}^{(m+m')\times n}$ be the matrix indexed by $e \in \overrightarrow{\mathcal{E}} \cup \overrightarrow{\mathcal{E}}'$ and $k \in \mathcal{V} \cup \mathcal{V}'$ whose (e,k)-element is given by

$$[\widetilde{T}]_{ek} \triangleq \begin{cases} -\tilde{s}_{kj}, & e=(k,j)\in\overrightarrow{\mathcal{E}'}, \\ -\tilde{c}_{kj}, & e=(k,j)\in\overrightarrow{\mathcal{E}}, \\ 0, & \text{otherwise}; \end{cases} \quad (16)$$

and let $A(\overrightarrow{G}) \in \mathbb{R}^{n\times m}$ to the matrix indexed by $k \in \mathcal{V} \cup \mathcal{V}'$ and $e \in \overrightarrow{\mathcal{E}} \cup \overrightarrow{\mathcal{E}}'$ whose (k,e)-element is given by

$$[A(\overrightarrow{G})]_{ke} = \begin{cases} 1, & e=(i,k)\in\overrightarrow{\mathcal{E}}\cup\overrightarrow{\mathcal{E}'}, \\ -1, & e=(k,j)\in\overrightarrow{\mathcal{E}}\cup\overrightarrow{\mathcal{E}'}, \\ 0, & \text{otherwise}. \end{cases} \quad (17)$$

In addition, without loss of any generality, we also introduce the ordering over $\overrightarrow{\mathcal{E}}\cup\overrightarrow{\mathcal{E}'}$ and $\mathcal{V}\cup\mathcal{V}'$ such that $e' \in \overrightarrow{\mathcal{E}'}$ precedes $e \in \overrightarrow{\mathcal{E}}$ and $k' \in \mathcal{V}'$ precedes $k \in \mathcal{V}$. As a result of Eqs. (15) to (17), $\widetilde{M} = L(\widetilde{G}^z) + \widetilde{\Sigma}^z - \widetilde{\Theta}^H\Lambda^\dagger\widetilde{\Theta}$ can be rewritten as

$$\widetilde{M} = L(\widetilde{G}^z) + \widetilde{T}^H\Omega^{\frac{1}{2}}\Pi\Omega^{\frac{1}{2}}\widetilde{T}, \quad (18)$$

in which $\Pi \in \mathbb{R}^{(m+m')\times(m+m')}$ is the matrix of the orthogonal projection operator $\pi : \mathbb{C}^{m+m'} \to \ker(A(\overrightarrow{G})\Omega^{\frac{1}{2}})$ onto the kernel of $A(\overrightarrow{G})\Omega^{\frac{1}{2}}$. Therefore, Eq. (14) is equivalent to

$$\min_{z\in\mathbb{C}_1^n} \operatorname{trace}(\widetilde{M}zz^H), \\ \widetilde{M} = L(\widetilde{G}^z) + \widetilde{T}^H\Omega^{\frac{1}{2}}\Pi\Omega^{\frac{1}{2}}\widetilde{T}. \quad \text{(QP)}$$

Interested readers can refer to Appendix A for a detailed derivation of (QP).

In the next section, we will relax (QP) to complex semidefinite programming and show that the semidefinite relaxation is tight as long as the noise magnitude is below a certain threshold.

V. THE SEMIDEFINITE RELAXATION

In a similar way to [30], [38], [39], it is straightforward to relax (QP) to

$$\min_{X\in\mathbb{H}^n} \langle\widetilde{M},X\rangle \\ \text{s.t.} \quad X \succeq 0, \quad \operatorname{diag}(X) = \mathbf{1}. \quad \text{(SDP)}$$

It should be noted that if $\hat{X} \in \mathbb{H}^n$ has rank one and solves (SDP), then a solution $\hat{z} \in \mathbb{C}_1^n$ to (QP) can be exactly recovered from $\hat{X}$ through singular value decomposition with which we have $\hat{X} = \hat{z}\hat{z}^H$.

In this paper, it is without loss of any generality to assume that all the manifolds are Riemannian submanifolds of Euclidean space [51], whose differential geometric properties, e.g., Riemannian gradients and Riemannian Hessians, are defined accordingly.

In the rest of section, we will analyze and derive the conditions for the optimality of (QP) and (SDP), and conditions for the tight relaxation of (SDP), all the proofs of which can be found in Appendix B.

From [51], the necessary conditions for the local optimality of (QP) can be well characterized in terms of the Riemannian gradients and Hessians.

Lemma 1. If $\hat{z} \in \mathbb{C}_1^n$ is a local optimum of (QP), then there exists a real diagonal matrix $\hat{\Lambda} \triangleq \Re\{\text{ddiag}(\widetilde{M}\hat{z}\hat{z}^H)\} \in \mathbb{R}^{n \times n}$ such that $\hat{S} \triangleq \widetilde{M} - \hat{\Lambda} \in \mathbb{H}^n$ satisfies the following conditions:

(1) $\hat{S}\hat{z} = \mathbf{0}$;
(2) $\langle \dot{z}, \hat{S}\dot{z} \rangle \geq 0$ for all $\dot{z} \in T_{\hat{z}}\mathbb{C}_1^n$.

If $\hat{z}$ satisfies (1), it is a first-order critical point, and if $\hat{z}$ satisfies (1) and (2), it is a second-order critical point.

Proof. See Appendix B.1. □

Since (SDP) is convex and the identity matrix $\mathbf{I} \in \mathbb{C}^{n \times n}$ is strictly feasible, the sufficient and necessary conditions for the global optimality of (SDP) can be derived in terms of the Karush-Kuhn-Tucker (KKT) conditions.

Lemma 2. A Hermitian matrix $\hat{X} \in \mathbb{H}^n$ is a global optimum of (SDP) if and only if there exists $\hat{S} \in \mathbb{H}^n$ such that the following conditions hold:

(1) $\text{diag}(\hat{X}) = \mathbf{1}$;
(2) $\hat{X} \succeq 0$;
(3) $\hat{S}\hat{X} = \mathbf{0}$;
(4) $\widetilde{M} - \hat{S}$ is real diagonal;
(5) $\hat{S} \succeq 0$.

Furthermore, if $\text{rank}(\hat{S}) = n - 1$, then $\hat{X}$ has rank one and is the unique global optimum of (SDP).

Proof. See Appendix B.2. □

As a result of Lemmas 1 and 2, we obtain the sufficient conditions for the exact recovery of (QP) from (SDP).

Lemma 3. If $\hat{z} \in \mathbb{C}_1^n$ is a first-order critical point of (QP) and $\hat{S} = \widetilde{M} - \hat{\Lambda} \succeq 0$ in which $\hat{\Lambda} = \Re\{\text{ddiag}(\widetilde{M}\hat{z}\hat{z}^H)\}$, then $\hat{z}$ is a global optimum of (QP) and $\hat{X} = \hat{z}\hat{z}^H$ is a global optimum of (SDP). Moreover, if $\text{rank}(\hat{S}) = n - 1$, then $\hat{X}$ is the unique optimum of (SDP).

Proof. See Appendix B.3. □

Lemma 3 gives sufficient conditions to check whether (SDP) is a tight relaxation of (QP). As a matter of fact, if the measurement noise is not too large, it is guaranteed that (SDP) is always a tight relaxation of (QP) as the following proposition states.

Proposition 1. Let $\underline{M} \in \mathbb{H}^n$ be the data matrix of the form Eq. (18) that is constructed with the true (latent) pose-pose measurements $\underline{\rho}_{ij} = (\underline{c}_{ij}, \underline{z}_{ij})$ and pose-landmark measurements $\underline{s}_{ij}$, then there exists a constant $\gamma = \gamma(\underline{M}) > 0$ such

that if $\|\widetilde{M} - \underline{M}\|_2 < \gamma$, then (SDP) attains the unique global optimum at $\hat{X} = \hat{z}\hat{z}^H \in \mathbb{H}^n$, in which $\hat{z} \in \mathbb{C}_1^n$ is a global optimum of (QP).

Proof. See Appendix B.4. □

Lemma 3 verifies the tightness of the complex semidefinite relaxation and Proposition 1 guarantees that the tightness of the complex semidefinite relaxation, which makes (SDP) certifiably correct for graph-based SLAM. It should be noted that similar results to Lemma 3 and Proposition 1 have been presented for synchronization problems on general special Euclidean groups using the matrix representation in [33] and for phase synchronization using the complex number representation in [38].

In spite of the tightness of the semidefinite relaxation of planar graph-based SLAM, solving large-scale complex semidefinite programming remains challenging and time-consuming. In the next section, we will further relax (SDP) as a series of rank-restricted complex semidefinite programming such that (SDP) can be efficiently solved with the Riemannian staircase optimization.

VI. THE CPL-SLAM ALGORITHM

In this section, we show that it is possible to recast (SDP) as Riemannian optimization on complex oblique manifolds, which is one of the most important contributions of this paper. A brief introduction to the complex oblique manifold can be found in Appendix D, and it is also helpful to read [52] that is about the real oblique manifold.

In general, interior point methods to solve (SDP) take polynomial time, which, however, may still be slow when the polynomial exponent is large. Instead of solving (SDP) directly, Boumal et al. found that (SDP) can be relaxed to a series of rank-restricted complex semidefinite programing [43]:

$$\min_{Y \in \text{OB}(r,n)} \text{trace}(\widetilde{M}YY^H) \qquad (r\text{-SDP})$$

in which

$$\text{OB}(r, n) \triangleq \{Y \in \mathbb{C}^{n \times r} | \text{ddiag}(YY^H) = \mathbf{I}\}$$

is the complex oblique manifold. It should be noted that (r-SDP) can be a tight relaxation of (SDP) if some conditions are met as stated in Propositions 2 and 3, whose proofs are immediate from [43, Theorem 2].

Proposition 2. If $\hat{Y} \in \text{OB}(r, n)$ is rank-deficient and second-order critical for (r-SDP), then it is globally optimal for (r-SDP) and $\hat{X} = \hat{Y}\hat{Y}^H \in \mathbb{H}^n$ is globally optimal for (SDP).

Proposition 3. If $r \geq \lceil \sqrt{n} \rceil$, then for almost all $\widetilde{M} \in \mathbb{C}^{n \times n}$, every first-order critical $\hat{Y} \in \text{OB}(r, n)$ for (r-SDP) is rank-deficient.

Propositions 2 and 3 are referred as the Burer-Monteiro guarantees for smooth semidefinite programming [43] that apply to a number of classic estimation problems. From Propositions 2 and 3, it can be concluded that (SDP) is equivalent to successively solving (r-SDP) with the Riemannian trust region (RTR) method [53] for $2 \leq r_1 < r_2 < \cdots < r_k \leq n + 1$ until

Algorithm 1 The Riemannian staircase optimization (RSO)

1: **Input**: Integers $2 \leq r_0 < r_1 < \cdots < r_k \leq n + 1$; an initial iterate $z_0 \in \mathbb{C}_1^n$
2: $Y_0 = \begin{bmatrix} \hat{z}_0 & \mathbf{0} \end{bmatrix} \in \mathrm{OB}(r_0, n)$
3: **for** $i = 1 \to k$ **do**
4:     Implement the Riemannian optimization to solve
$$Y_i^* = \arg \min_{Y \in \mathrm{OB}(r_i, n)} \mathrm{trace}(\widetilde{M} Y Y^H)$$
    locally with Y_i as an initial guess
5:     **if** $\mathrm{rank}(Y_i^*) < r_i$ **then**
6:         **return** $Y_i^* \in \mathrm{OB}(r_i, n)$
7:     **else**
8:         $Y_{i+1} = \begin{bmatrix} \hat{Y}_i & \mathbf{0} \end{bmatrix} \in \mathrm{OB}(r_{i+1}, n)$
9:     **end if**
10: **end for**
11: **return** $Y_i^* \in \mathrm{OB}(r_k, n)$

Algorithm 2 The rounding procedure for solutions of (r-SDP)

1: **Input**: An optimum $Y^* \in \mathrm{OB}(r, n)$ to (r-SDP)
2: Assign $\hat{z} = \begin{bmatrix} \hat{z}_1 & \cdots & \hat{z}_n \end{bmatrix}^\top \in \mathbb{C}^n$ to be the left-singular vector of Y^* that is associated with the greatest singular value
3: **for** $i = 1 \to n$ **do**
4:     $\hat{z}_i = \dfrac{\hat{z}_i}{|\hat{z}_i|}$
5: **end for**
6: **return** $\hat{z} \in \mathbb{C}_1^n$

Algorithm 3 The CPL-SLAM algorithm

1: **Input**: Integers $2 \leq r_0 < r_1 < \cdots < r_k \leq n + 1$; an initial iterate $z_0 \in \mathbb{C}_1^n$
2: Implement Algorithm 1 to compute an optimum $Y^* \in \mathrm{OB}(r, n)$
3: Implement Algorithm 2 to compute rotational states $\hat{z} \in \mathbb{C}_1^n$
4: Implement Eq. (13) to compute translational states $\hat{c} \in \mathbb{C}^n$
5: **return** $\hat{z} \in \mathbb{C}_1^n$ and $\hat{c} \in \mathbb{C}^n$

---

a rank-deficient second-order critical point is found, and such a method to solve semidefinite programming is referred as the Riemannian staircase optimization (Algorithm 1) [43], [54]. In [33], [38], [54], the Riemannian staircase optimization has been used to solve a number of semidefinite relaxations of synchronization problems. In addition, it is known that the RTR method solves (r-SDP) locally in polynomial time [43, Proposition 3]. In contrast to using interior point methods to solve (SDP) directly, the Riemannian staircase optimization using the RTR method is empirically orders of magnitude faster in solving large-scale smooth semidefinite programming.

As shown in Algorithm 2, the solution rounding of an optimum of $Y^* \in \mathrm{OB}(r, n)$ of (r-SDP) is simply to assign $\hat{z} = \begin{bmatrix} \hat{z}_1 & \cdots & \hat{z}_n \end{bmatrix} \in \mathbb{C}^n$ to be the left-singular vector of Y^* that is associated with the greatest singular value, and then normalize each z_i to get $\hat{z} \in \mathbb{C}_1^n$. The solution rounding algorithm is exact if $\mathrm{rank}(Y^*) = 1$. Moreover, it should be noted that the solution rounding algorithm can recover the global optimum $\hat{z}^* \in \mathbb{C}_1^n$ from Y^* as long as the exactness of (SDP) holds and $X = Y^* Y^{*H}$ solves (SDP).

From algorithms of Riemannian staircase optimization (Algorithm 1) and solution rounding (Algorithm 2), the proposed CPL-SLAM algorithm for planar graph-based SLAM is as shown in Algorithm 3, which follows a similar procedure to SE-Sync [33]. It should be noted that Lemmas 1 to 3 can be used to certify the global optimality of the solution, and Propositions 1 to 3 indicate that the CPL-SLAM algorithm is expected to retrieve the globally optimal solution to the planar graph-based SLAM as long as the noise magnitude is below a certain threshold. Therefore, it can be concluded that CPL-SLAM is certifiably correct.

Similar to SE-Sync [33] and Cartan-Sync [34], CPL-SLAM uses the trust-region method on Riemannian manifolds that relies on the truncated conjugated gradient (TCG) method to evaluate the descent direction [53]. The TCG method iteratively solves linear equations and improves the solution to necessary accuracy within finite iterations, which is usually faster than direct methods. In general, the TCG method needs

a preconditioner to accelerate the convergence. Even though the choice of preconditioner for graph-based SLAM without landmarks is immediate [33], there is still a lack of a suitable preconditioner for graph-based SLAM with landmarks. To address this issue, we also propose a preconditioner for graph-based SLAM with landmarks in Appendix C.

Even though the positive semidefinite matrix is not explicitly formed in CPL-SLAM or SE-Sync to solve planar graph-based SLAM, it can be seen that CPL-SLAM using the complex number representation results in semidefinite relaxations of smaller size than [30], [33]. In the semidefinite relaxation of CPL-SLAM, the $n \times n$ complex positive semidefinite matrix $X \in \mathbb{H}^n$ can be parameterized with n^2 real numbers, whereas the semidefinite relaxation in [33] using the matrix representation needs $2n^2 + n$ real numbers to parameterize the $2n \times 2n$ real positive semidefinite matrix, and that in [30] needs $4n^2$ real numbers to parameterize the $2n \times 2n$ complex positive semidefinite matrix.

It is obvious that the complex number representation is more concise than the matrix representation, and as a result, CPL-SLAM roughly requires half as much storage space as SE-Sync [33]. More importantly, as is discussed in Section VII, from both theoretical and empirical perspectives, the conciseness of the complex number representation reduces the computational cost a lot and renders the semidefinite relaxation much tighter, and thus, the resulting CPL-SLAM algorithm is much more efficient in numerical computation and much more robust to measurement noise than [33].

In contrast to the works of [30], [33], [35], CPL-SLAM is more general and more scalable. As mentioned before, CPL-SLAM is more efficient, tighter and more robust than SE-Sync [33]. Even though we use the same complex number representation as [30], our formulation is simpler and only depends on rotational states $z \in \mathbb{C}_1^n$, whereas [30] involves both translational and rotational states $c \in \mathbb{C}^n$ and $z \in \mathbb{C}_1^n$.

Moreover, [30] mainly focuses on the optimality verification of planar pose graph optimization, whereas ours not only works on optimality verification and obtains stronger theoretical results, but also presents more scalable algorithms to solve planar graph-based SLAM. In [35], the authors use bounded sum of squares programming to solve planar graph-based SLAM. Even though [35] always attains the globally optimal solution regardless of the measurement noise, it relies on sparse sum of squares programming, which, to our knowledge, has limited scalability for large-scale problems. As a result, CPL-SLAM can be expected to outperform [35] by several orders of magnitude in terms of computational time. Last but not least, except for [35], the works of [30], [33] are designed for planar pose graph optimization or angular synchronization, whereas ours considers the planar graph-based SLAM that has both poses and landmarks.

VII. THE RESULTS OF DATASET EXPERIMENTS

In this section, we implement CPL-SLAM on the simulated Tree datasets, simulated City datasets and a suite of large-scale 2D SLAM benchmark datasets with and without landmarks [30], [33], [55]. We compare CPL-SLAM with the popular state-of-the-art SE-Sync [33] and Powell's Dog-Leg method (PDL-GN) [21], [47]. Even though the original algorithms of SE-Sync [33] are not designed for problems with landmarks, we extend SE-Sync following a similar procedure as CPL-SLAM. For the linear solvers to compute a descent direction, CPL-SLAM and SE-Sync [33] use the indirect and iterative truncated conjugate gradient method, whereas PDL-GN [21], [47] uses the sparse direct method. The C++ code of CPL-SLAM is available at https://github.com/MurpheyLab/CPL-SLAM.

All the experiments have been performed on a laptop with an Intel i7-8750H CPU and 32GB of RAM running Ubuntu 18.04 and using g++ 7.8 as C++ compiler. We have done the computation on a single core of CPU. For all the experiments, we choose the initial rank to be $r_{SE} = 3$ and $r_{CPL} = 2$ for SE-Sync and CPL-SLAM, respectively, since we find that $r_{SE} = 3$ and $r_{CPL} = 2$ are in general good enough for SE-Sync and CPL-SLAM to solve planar graph-based SLAM given the noise levels in robotics and computer vision applications.

A. Tree Datasets

In this subsection, we evaluate the performance of CPL-SLAM, SE-Sync and PDL-GN on the simulated Tree datasets that are similar to tree10000 (Fig. 6k). A Tree dataset is consisted of 25×25 square grids in which each grid has side length of 1 m, and a robot trajectory of n poses along the rectilinear path of the square grid, and n' trees (landmarks) that are randomly distributed in the centre of some square grids. Odometric pose-pose measurements are available between each pair of sequential poses along the robot trajectory, whereas pose-landmark measurements between poses and trees that are close to each other are available with a probability of p_L; the pose-pose measurements $\tilde{\rho}_{ij} = (\tilde{c}_{ij}, \tilde{z}_{ij})$ and pose-landmark measurements $\tilde{s}_{ij}$ are generated from the noise models of Eq. (10). In our experiments, we investigate the performance

of these algorithms by varying each parameter individually and the default values for these parameters are chosen to be $n = 5000$, $n' = 250$, $p_L = 0.2$, $\tilde{c}_{ij}$ with an expected translational root-mean-squared error (RMSE) of $\sigma_t = 0.05$ m, $\tilde{z}_{ij}$ with an expected angular RMSE of $\sigma_R = 0.015\pi$ rad, and $\tilde{s}_{ij}$ with an expected positional RMSE of $\sigma_l = 0.05$ m.

For all the Tree datasets tested, CPL-SLAM, SE-Sync and PDL-GN all converge to the global optima when using the chordal initialization. As is shown in Fig. 1, it can be seen that CPL-SLAM is around $4 \sim 5$ times faster than SE-Sync and PDL-GN, whereas SE-Sync and PDL-GN are roughly as fast as each other.

The speed-up of CPL-SLAM over SE-Sync [33] in planar graph-based SLAM can be explained from several perspectives. 1) CPL-SLAM is more efficient for the objective and gradient evaluation, e.g., if the rank is $r_{SE} = 3$ and $r_{CPL} = 2$, CPL-SLAM only needs $\frac{1}{2} \sim \frac{2}{3}$ and $\frac{1}{4} \sim \frac{2}{3}$ operations of SE-Sync to evaluate the objective and gradient, respectively. 2) CPL-SLAM is more efficient in terms of the projection or retraction onto the manifold than SE-Sync – the projection map of CPL-SLAM is just to normalize n vectors, whereas that of SE-Sync has to compute n singular value decompositions, which is much more time consuming. 3) CPL-SLAM is more efficient for chordal initialization and solution rounding. 4) As a result of the conciseness of the complex number representation, the preconditioner used in CPL-SLAM has a better approximation the Hessian matrix than SE-Sync, and thus, has a faster convergence of the truncated conjugate gradient method that the Riemannian trust region method implements to evaluate the descent direction. Therefore, CPL-SLAM should be theoretically more efficient than SE-Sync, which is further confirmed by the results of the experiments.

Similar to [24], [27], [56], PDL-GN uses the Gauss-Newton method and might not perform well if there are large residues of the measurements and strong nonlinearities of the objective function [21], [47], whereas CPL-SLAM uses the exact Hessian to compute the Newton direction, and thus, is expected to converge faster and have better efficiency. In addition, as mentioned before, when evaluating the descent direction, PDL-GN factorizes sparse matrices to solve linear equations. In contrast, CPL-SLAM makes use of the truncated conjugate gradient method as the linear solver, which might also improve the overall efficiency of CPL-SLAM. On the other hand, since the choice of linear solvers is critical for the efficiency of optimizers, there is a possibility to improve PDL-GN's efficiency if the truncated conjugate gradient method is used.

The performance of CPL-SLAM, SE-Sync and PDL-GN is also evaluated if they are not well initialized. When the odometric initialization is used, it can be seen from Fig. 2 that CPL-SLAM and SE-Sync converge to the global optima in spite of the poor initial guess, whereas PDL-GN gets stuck at the local optima and has much greater objective values.

As mentioned before, the convergence of CPL-SLAM and SE-Sync to global optima does not rely on initial guess since CPL-SLAM and SE-Sync essentially solve the semidefinite relaxation of graph-based SLAM and are guaranteed to attain the globally optimal solution as long as the magnitude of measurement noise is below a certain threshold. As a compar-
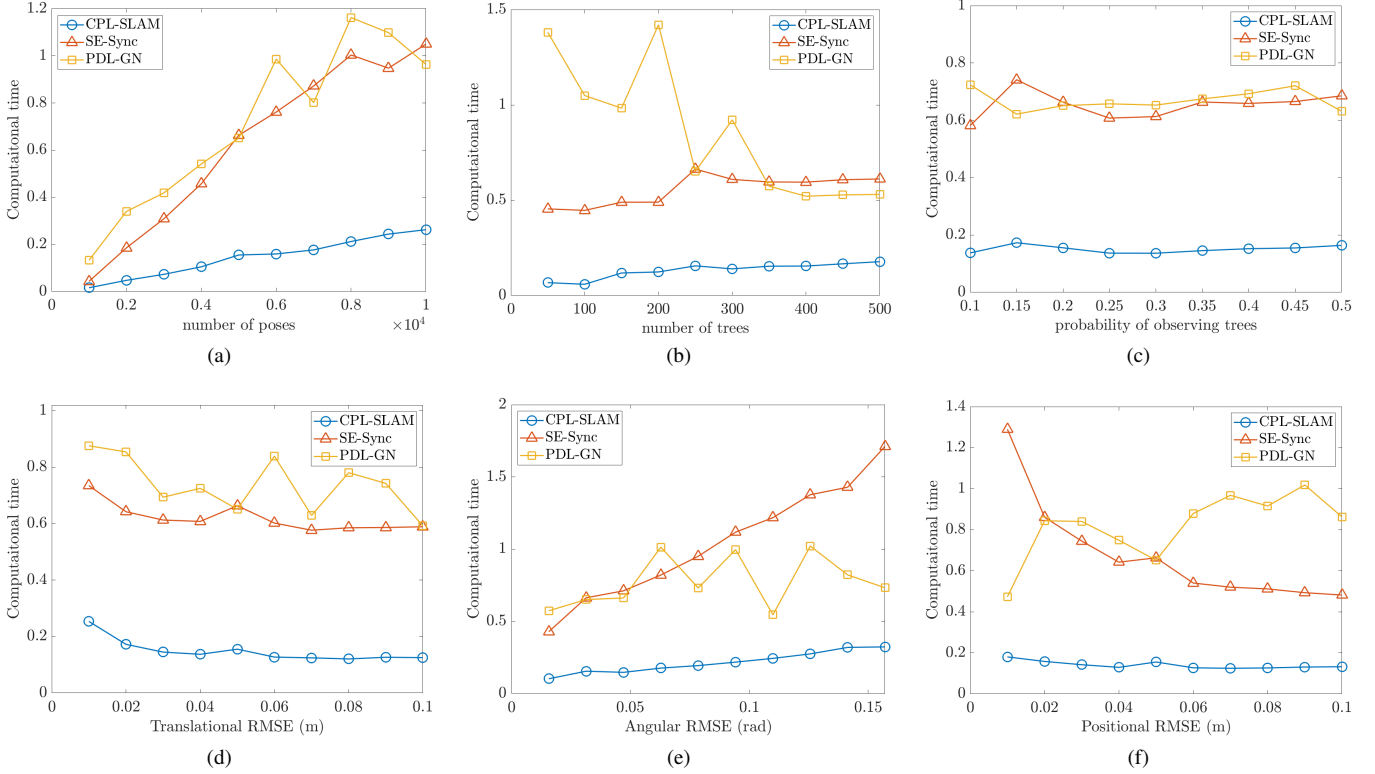
Fig. 1: The computational time of CPL-SLAM, SE-Sync and PDL-GN on the Tree datasets with varying each parameter individually while keeping the other parameters to be default values. The chordal initialization is used for all the tests. The results of each varying parameter are the number of poses n in (a), the number of trees n' in (b), the probability of observing trees p_L in (c), translational RMSEs of σ_t in (d), angular RMSEs of σ_R in (e) and positional RMSEs of σ_l in (f). The default values are $n = 5000$, $n' = 250$, $p_L = 0.2$, $\sigma_t = 0.05$ m, $\sigma_R = 0.015\pi$ rad and $\sigma_l = 0.05$ m. For all the Tree datasets tested, it can be seen that CPL-SLAM is around $4 \sim 5$ times faster than SE-Sync and PDL-GN, whereas SE-Sync and PDL-GN are roughly as fast as each other.

ison, PDL-GN is a local search method whose performance is closely related with quality of initial guess, and thus the global optimum convergence of PDL-GN is usually not guaranteed even with low measurement noise.

B. City Datasets

In this subsection, we evaluate the tightness of CPL-SLAM on a series of simulated City datasets that are similar to city10000 (Fig. 6b) but with high measurement noise. As a basis for comparison, we also evaluate the tightness of SE-Sync using the matrix representation [33]. In general, CPL-SLAM and SE-Sync are said to be tight if the globally optimal solution is exactly recovered from the semidefinite relaxation, or equivalently, there is no suboptimality gap between the rounded solution and the relaxed solution.

In our experiments, a City dataset consists of 25×25 square grids in which each grid has side length of 1 m, a robot trajectory of $n = 3000$ poses along the rectilinear path of the grid, odometric measurements that are available between sequential poses along the robot trajectory, and loop-closure measurements that are available at random between non-sequential poses with a probability $p_C = 0.1$. The odometric and loop-closure measurements are generated from noise

models of Eqs. (10a) and (10b), and the default translational weight factor is $\tau_{ij} = 88.89$ that corresponds to an expected translational RMSE of $\sigma_t = 0.15$ m and the default rotational weight factor is $\kappa_{ij} = 40.53$ that corresponds to an expected angular RMSE of $\sigma_R = 0.05\pi$ rad. For the datasets, we vary translational and rotational measurement weight factors τ_{ij} and κ_{ij} individually that correspond to translational and angular RMSEs of $\sigma_t = 0.1 \sim 0.3$ m and $\sigma_R = 0.03\pi \sim 0.15\pi$ rad, respectively, while keeping the other weight factor as the default value.

The results of CPL-SLAM and SE-Sync on the simulated City datasets with high translational and rotational measurement noise are in Figs. 3 and 4, respectively. For each translational and angular RMSE, we calculate the successful rates of exact recovery from the semidefinite relaxation (Fig. 3a and Fig. 4a), the relative suboptimality bounds between rounded and relaxed solutions (Fig. 3b and Fig. 4b), and the objective values of rounded and relaxed solutions (Fig. 3c and Fig. 4c) statistically from 50 randomly generated City datasets, in which we assume the globally optimal solution is exactly recovered if the relative suboptimality bound is less than 1×10^{-6}. From Fig. 3, it can be seen that CPL-SLAM holds the tightness on all the datasets with translational RMSEs
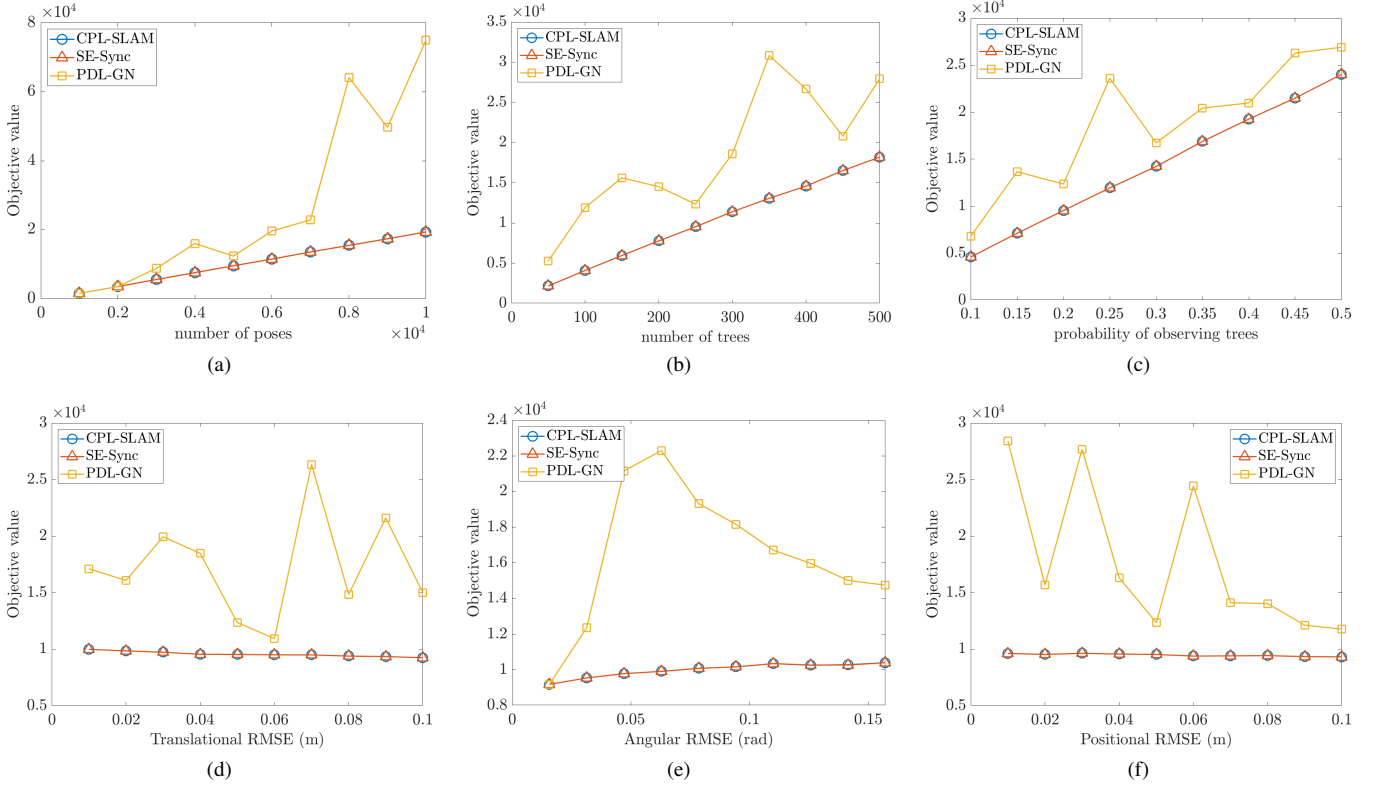
Fig. 2: The objective of CPL-SLAM, SE-Sync and PDL-GN on the Tree datasets using the odometric initialization. In the experiments, each parameter is varied individually while the other parameters are kept to be default values. The results of each varying parameter are the number of poses n in (a), the number of trees n' in (b), the probability of observing trees p_L in (c), translational RMSEs of σ_t in (d), angular RMSEs of σ_R in (e) and positional RMSEs of σ_l in (f). The default values are $n = 5000$, $n' = 250$, $p_L = 0.2$, $\sigma_t = 0.05$ m, $\sigma_R = 0.015\pi$ rad and $\sigma_l = 0.05$ m. For all the Tree datasets tested, it can be seen that CPL-SLAM and SE-Sync converge to global optima despite poor initialization, whereas PDL-GN gets stuck at local optima.

of $\sigma_t = 0.1 \sim 0.3$ m, whereas SE-Sync fails on some of the datasets. From Fig. 4, it can be seen that when the angular RMSE is small, i.e., approximately less than 0.15 rad, both CPL-SLAM and SE-Sync exactly recover the globally optimal solution from the semidefinite relaxation, and as angular RMSE increases and is greater than 0.15 rad, CPL-SLAM and SE-Sync begin to fail. In spite of this, we find that CPL-SLAM has a much higher successful rate of exact recovery from the semidefinite relaxation (Fig. 3a and Fig. 4a) and orders of magnitude smaller relative suboptimality bounds (Fig. 3b and Fig. 4b). Furthermore, for the objective value, CPL-SLAM has greater lower bound from the relaxed solution but lower upper bound from the rounded solution in scenarios of high measurement noise (Fig. 3c and Fig. 4c). All of these results indicate that CPL-SLAM has a tighter semidefinite relaxation using the complex number representation than SE-Sync using the matrix representation, and thus, is more robust to translational and rotational measurement noise.

In Fig. 3, it is interesting to see that SE-Sync fails on datasets with small translational measurement noise but works on datasets with large translational measurement noise. Even though there is lack of formal analysis, we guess this is because the tightness of the semidefinite relaxation in SE-

Sync, in addition to the magnitude of measurement noise, is also related with the ratio τ_{ij}/κ_{ij} of translational weight factors τ_{ij} and rotational weight factors κ_{ij}, i.e., when τ_{ij}/κ_{ij} increases, the semidefinite relaxation in SE-Sync tends to be relatively more sensitive to measurement nose.

It is obvious that the improved tightness and robustness of CPL-SLAM over SE-Sync in planar graph-based is associated with the more concise representation of complex numbers over matrices in the semidefinite relaxation, for which a theoretically complete analysis similar to [57] is left as future work. In spite of this, we present one possible reason that might help explain the improved tightness of CPL-SLAM. The semidefinite matrix resulting from the solution to planar graph-based SLAM using the matrix representation should take the form $X_R = \begin{bmatrix} X_{R_{ij}} \end{bmatrix} \in \mathbb{R}^{2n \times 2n}$ in which each (i, j)-th block $X_{R_{ij}}$ has the algebraic structure $X_{R_{ij}} = \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \in \mathbb{R}^{2 \times 2}$, and SE-Sync drops such an algebraic structure in the semidefinite relaxation. Even though it is possible for SE-Sync to keep this algebraic structure by either reformulating the associated data matrix or adding numbers of extra linear constraints, substantial computational efforts are required for both options. In comparison, CPL-SLAM preserves the algebraic structure
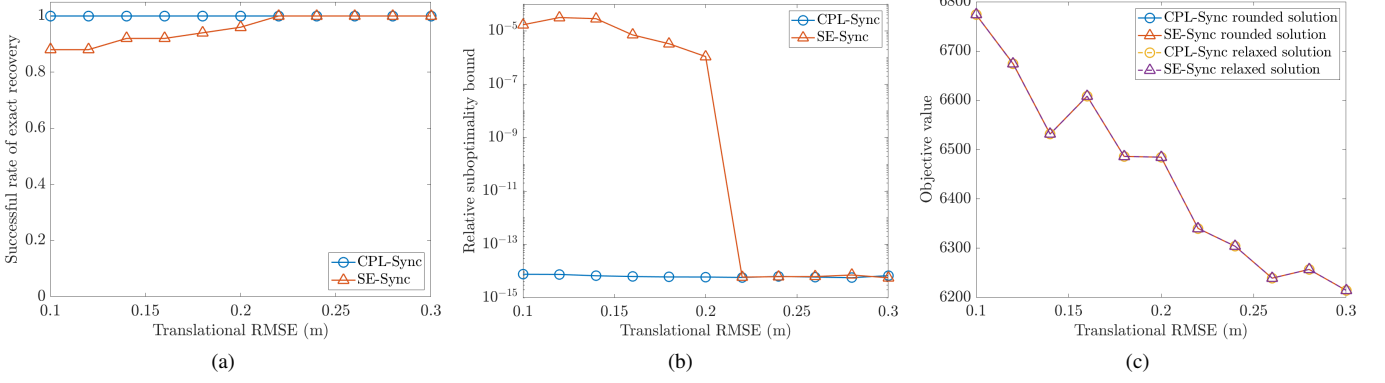
Fig. 3: The comparisons of CPL-SLAM and SE-Sync on the City datasets with high translational measurement noise with $n = 3000$, $p_C = 0.1$, $\kappa_{ij} = 40.53$ corresponding to angular RSME of $\sigma_R = 0.05\pi$ rad and varying τ_{ij} corresponding to different translational RSMEs of $\sigma_t = 0.1 \sim 0.3$ m. The results are (a) successful rates of exact recovery from the semidefinite relaxation, (b) relative suboptimality bounds between rounded and relaxed solutions, and (c) objective values of rounded and relaxed solutions. For all the datasets with different σ_t, CPL-SLAM has a tighter semidefinite relaxation and is more robust to translational measurement noise.
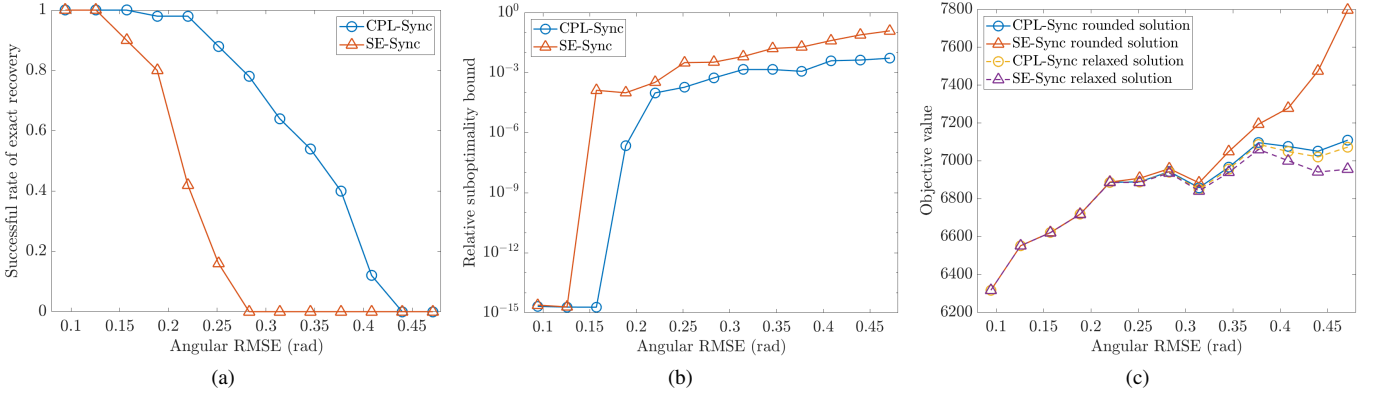

Fig. 4: The comparisons of CPL-SLAM and SE-Sync on the City datasets with high rotational measurement noise with $n = 3000$, $p_C = 0.1$, $\tau_{ij} = 88.89$ corresponding to translational RSME of $\sigma_t = 0.15$ m and varying κ_{ij} corresponding to different angular RSMEs of $\sigma_R = 0.03\pi \sim 0.15\pi$ rad. The results are (a) successful rates of exact recovery from the semidefinite relaxation, (b) relative suboptimality bounds between rounded and relaxed solutions, and (c) objective values of rounded and relaxed solutions. For all the datasets with different σ_R, CPL-SLAM has a tighter semidefinite relaxation and is more robust to rotational measurement noise.

of $X_{R_{ij}}$ as complex numbers in the semidefinite relaxation without having to reformulate the data matrix or introduce any extra constraints. As said before, the explanations given above are still hypotheses and need to be proved. However, we can still conclude from the Figs. 3 and 4 that the semidefinite relaxation in CPL-SLAM using the complex number representation is tighter than that in SE-Sync using the matrix representation, which further suggests that CPL-SLAM is more robust to measurement noise than SE-Sync.

C. SLAM Benchmark Datasets

In this subsection, we implement CPL-SLAM, SE-Sync and PDL-GN on a variety of 2D SLAM benchmark datasets with and without landmarks. In these datasets, city10000, M3500, M3500-a, M3500-b, M3500-c and tree10000 are simulated benchmark datasets while the others, i.e., ais2klinik, CSAIL,

intel, FR-079, MIT and victoria-park, are real-world datasets. In addition, tree10000 and victoria-park have positions of observed landmarks involved. The chordal initialization [58] is used for all the benchmark datasets tested.

For all the 2D SLAM benchmark datasets, CPL-SLAM, SE-Sync and PDL-GN converge to the globally optimal solution. The results are shown in Table I, in which n is the number of unknown poses and n' is the number of observed landmarks, m is the number of pose-pose measurements and m' is the number of pose-landmark measurements, f^* is the globally optimal objective value, and the total time accounts for all the time taken to solve graph-based SLAM and the RTR time only accounts for the time taken by the RTR method to solve Riemannian staircase optimization. A specific comparison of SE-Sync and CPL-SLAM is further shown in Fig. 5. From Table I and Fig. 5, it can be seen that CLP-Sync is significantly

TABLE I: Results of the 2D SLAM Benchmark datasets

| Dataset	$n + n'$	$m + m'$	f^*	PDL-GN [21], [47]	SE-Sync [33]		CPL-SLAM [ours]	
				Total time (s)	RTR time (s)	Total time (s)	RTR time (s)	Total time (s)
ais2klinik	15115	16727	1.885×10^2	3.2×10^0	2.6×10^0	2.7×10^0	1.0×10^0	1.2×10^0
city10000	10000	20687	6.386×10^2	1.8×10^0	8.6×10^{-1}	1.2×10^0	5.2×10^{-1}	5.4×10^{-1}
CSAIL	1045	1172	3.170×10^1	2.6×10^{-2}	5.0×10^{-3}	1.4×10^{-2}	1.0×10^{-3}	5.0×10^{-3}
intel	1728	2512	5.236×10^1	1.3×10^{-1}	3.8×10^{-2}	6.1×10^{-2}	1.7×10^{-2}	2.6×10^{-2}
M3500	3500	5453	1.939×10^2	3.3×10^{-1}	1.5×10^{-1}	2.2×10^{-1}	7.4×10^{-2}	9.8×10^{-2}
M3500-a	3500	5453	1.598×10^3	4.1×10^{-1}	1.6×10^{-1}	2.3×10^{-1}	8.0×10^{-2}	1.0×10^{-1}
M3500-b	3500	5453	3.676×10^3	1.6×10^0	5.3×10^{-1}	5.9×10^{-1}	2.6×10^{-1}	2.8×10^{-1}
M3500-c	3500	5453	4.574×10^3	2.4×10^0	7.5×10^{-1}	8.2×10^{-1}	3.7×10^{-1}	4.0×10^{-1}
FR-079	989	1217	2.859×10^1	3.9×10^{-2}	5.9×10^{-3}	1.8×10^{-2}	1.7×10^{-3}	5.5×10^{-3}
MIT	808	827	6.115×10^1	7.4×10^{-2}	1.4×10^{-2}	2.1×10^{-2}	4.7×10^{-3}	7.1×10^{-3}
tree10000	10100	14442	6.035×10^2	6.9×10^{-1}	5.4×10^{-1}	5.9×10^{-1}	1.3×10^{-1}	2.3×10^{-1}
victoria-park	7120	10608	4.660×10^2	2.1×10^0	5.9×10^{-1}	6.2×10^{-1}	1.4×10^{-1}	2.0×10^{-1}
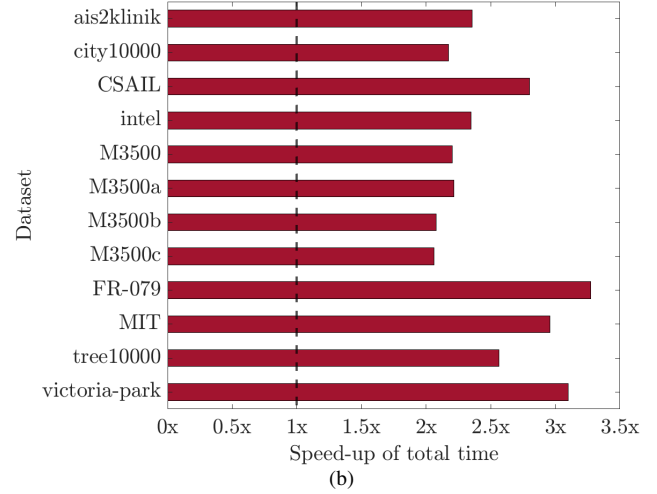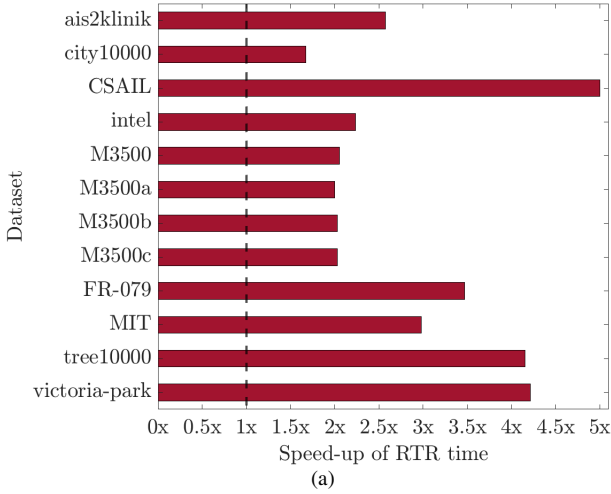
Fig. 5: The speed-up of CPL-SLAM over SE-Sync on 2D SLAM benchmark datasets. The results are (a) the speed-up of RTR time of CPL-SLAM over SE-Sync and (b) the speed-up of total time of CPL-SLAM over SE-Sync. CPL-SLAM is on average 2.87 and 2.51 times faster than SE-Sync for RTR time and total time, respectively.

faster than both SE-Sync and PDL-GN on all the SLAM benchmark datasets, in which CPL-SLAM outperforms PDL-GN by a factor of 5.53 on average for the overall computation, and outperforms SE-Sync by a factor of 2.87 and 2.51 on average for the computation of the RTR method and the overall computation, respectively. In particular, CPL-SLAM obtains a further improved performance of the RTR method over SE-Sync on the datasets with landmarks, and we think it is due to the conciseness of the complex number representation whose resulting preconditioner accelerates the truncated conjugate gradient method that is used in the RTR method to evaluate the descent direction.

The globally optimal results of CPL-SLAM on these 2D SLAM benchmark datasets are as shown in Fig. 6. It should be noted that M3500-a, M3500-b and M3500-c in Fig. 6f-Fig. 6h respectively have extra Gaussian noise with standard deviation 0.1 rad, 0.2 rad and 0.3 rad added to the rotational measurements of M3500 [30] in Fig. 6e, which indicates that CPL-SLAM can tolerate noisy measurements that are orders

of magnitude greater than real-world SLAM applications. For tree10000 in Fig. 6k and victoria-park in Fig. 6l with landmarks, we denote the positions of landmarks with red "+".

VIII. CONCLUSION

In this paper, we have presented CPL-SLAM that is a certifiably correct algorithm for planar graph-based SLAM using the complex number representation. By leveraging the complex semidefinite programming and Riemannian staircase optimization on complex oblique manifolds, CPL-SLAM is applicable to planar graph-based SLAM with and without landmarks. In addition, even though CPL-SLAM essentially solves the complex semidefinite relaxation, we prove that CPL-SLAM exactly retrieves the globally optimal solution to planar graph-based SLAM as long as the noise magnitude is below a certain threshold.

CPL-SLAM is compared with the state-of-the-art methods SE-Sync [33] and Powell's Dog-Leg [21], [47] on the simulated Tree datasets, the simulated City datasets and numerous
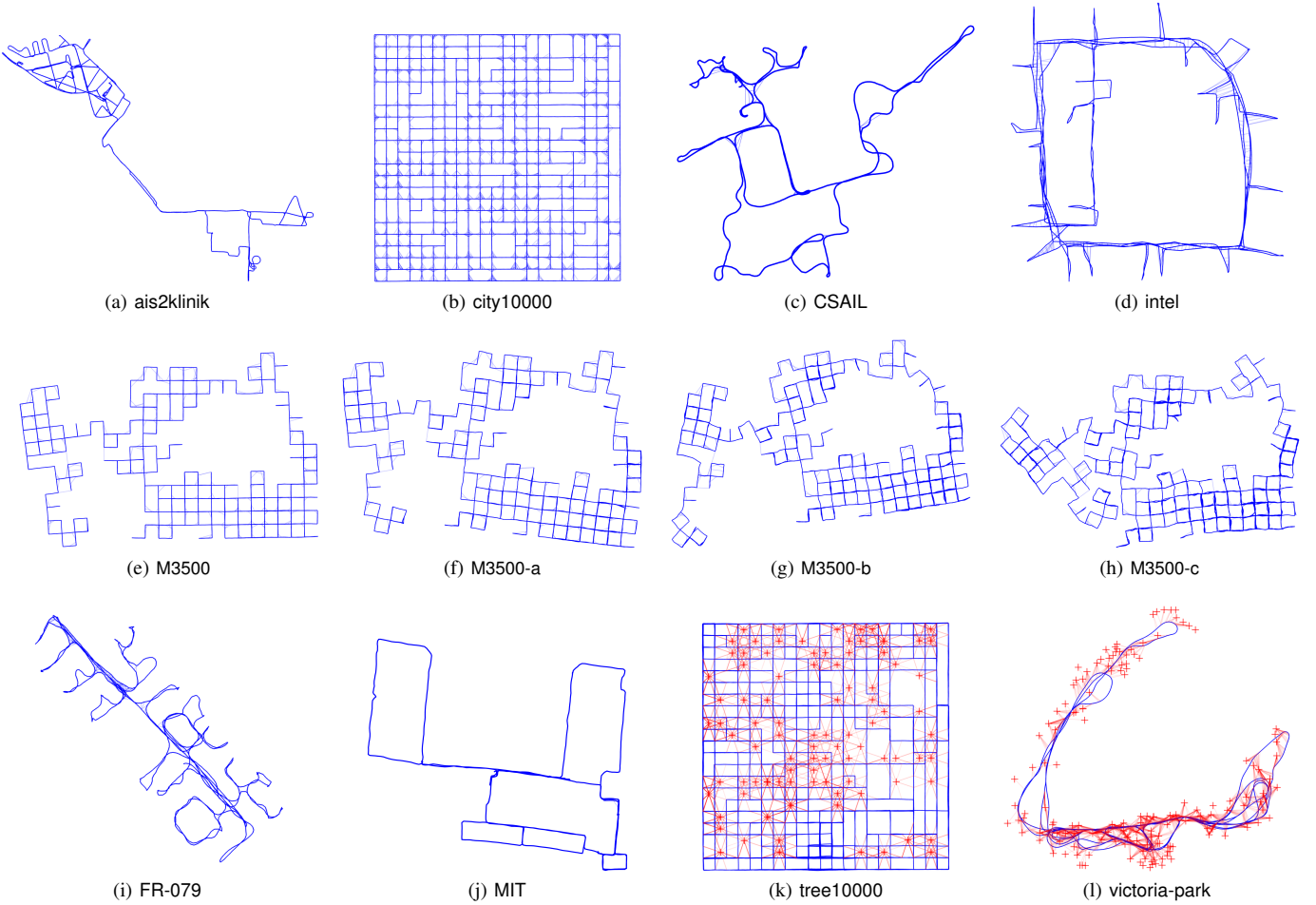
Fig. 6: The globally optimal results of CPL-SLAM on 2D SLAM benchmark datasets. It should be noted that CPL-SLAM still obtains global optima on M3500-a, M3500-b and M3500-c in (f)-(g), which respectively has large extra noise with standard deviations of 0.1 rad, 0.2 rad and 0.3 rad added to the rotational measurements of M3500 in (e). For tree10000 in (k) and victoria-park in (l) with landmarks, we denote the positions of landmarks with red "$+$".

large 2D simulated and real-world SLAM benchmark datasets in terms of scalability and robustness. The results of the data experiments indicate that CPL-SLAM is capable of solving planar graph-based SLAM certifiably, and more importantly, is more efficient in numerical computation and more robust to measurement noise. As a result, we expect that CPL-SLAM outperforms existing state-of-the-art methods to planar graph-based SLAM.

There is still great potential for improvements of CPL-SLAM in several aspects. A fully distributed extension of CPL-SLAM is definitely beneficial to multi-robot simultaneous localization and mapping. In spite of being able to tolerate large measurement noise, CPL-SLAM still needs to enhance its robustness to measurement outliers. At last, it is currently assumed that the positions of landmarks are fully known in CPL-SLAM, and we hope that in the future CPL-SLAM can handle range-only and bearing-only measurements of landmarks, which is another important extension.

APPENDIX A. THE DERIVATION OF (QP)

In this section, we derive (QP) following a similar procedure of [33, Appendix B] even though ours uses the complex number representation and has landmarks involved.

It is straightforward to rewrite (MLE) as

$$\min_{\substack{s \in \mathbb{C}^n, \\ c \in \mathbb{C}^n, \, z \in \mathbb{C}_1^n}} \left\| B \begin{bmatrix} s \\ c \\ z \end{bmatrix} \right\|_2^2 \tag{19}$$

in which

$$B \triangleq \begin{bmatrix} B_1 & B_2 \\ \mathbf{0} & B_3 \end{bmatrix} \in \mathbb{C}^{(2m+m') \times (2n+n')}.$$

Here $B_1 \in \mathbb{R}^{(m+m')\times(n+n')}$, $B_2 \in \mathbb{R}^{(m+m')\times n}$ and $B_3 \in \mathbb{C}^{m\times n}$ are given as

$$[B_1]_{ek} = \begin{cases} \sqrt{\nu_{ik}}, & e = (i,k) \in \overrightarrow{\mathcal{E}'}, \\ -\sqrt{\nu_{kj}}, & e = (k,j) \in \overrightarrow{\mathcal{E}'}, \\ \sqrt{\tau_{ik}}, & e = (i,k) \in \overrightarrow{\mathcal{E}}, \\ -\sqrt{\tau_{kj}}, & e = (k,j) \in \overrightarrow{\mathcal{E}}, \\ 0, & \text{otherwise}, \end{cases} \tag{20a}$$

$$[B_2]_{ek} = \begin{cases} -\sqrt{\nu_{kj}}\tilde{s}_{kj}, & e = (k,j) \in \overrightarrow{\mathcal{E}'}, \\ -\sqrt{\tau_{kj}}\tilde{c}_{kj}, & e = (k,j) \in \overrightarrow{\mathcal{E}}, \\ 0, & \text{otherwise}, \end{cases} \tag{20b}$$

and

$$[B_3]_{ek} = \begin{cases} -\sqrt{\kappa_{kj}}\tilde{z}_{kj}, & e = (k,j) \in \overrightarrow{\mathcal{E}}, \\ \sqrt{\kappa_{ik}}, & e = (i,k) \in \overrightarrow{\mathcal{E}}, \\ 0, & \text{otherwise}, \end{cases} \tag{20c}$$

respectively. Since (P) is also equivalent to (12), it can be concluded that

$$B_1^H B_1 = \Lambda, \tag{21a}$$
$$B_1^H B_2 = \widetilde{\Theta}, \tag{21b}$$
$$B_2^H B_2 = \widetilde{\Sigma}^z, \tag{21c}$$
$$B_3^H B_3 = L(\widetilde{G}^z) \tag{21d}$$

in which Λ, $\widetilde{\Theta}$, $\widetilde{\Sigma}$ and $L(\widetilde{G}^z)$ are as defined in (12). If we let $\widetilde{M}^\sigma \triangleq \widetilde{\Sigma}^z - \widetilde{\Theta}^H \Lambda^\dagger \widetilde{\Theta}$, then from Eqs. (21a) to (21d), we obtain

$$\begin{aligned} \widetilde{M}^\sigma &= B_2^H B_2 - B_2^H B_1 (B_1^H B_1)^\dagger B_1^H B_2 \\ &= B_2^H \left(\mathbf{I} - B_1 (B_1^H B_1)^\dagger B_1^H\right) B_2, \end{aligned} \tag{22}$$

in which B_1, B_2 and B_3 are defined as Eqs. (20a) to (20c). It should be noted that we might rewrite B_1 and B_2 as

$$B_1 = \Omega^{\frac{1}{2}} A^\top, \qquad B_2 = \Omega^{\frac{1}{2}} \widetilde{T}, \tag{23}$$

in which $A \triangleq A(\overrightarrow{G})$ and $\widetilde{T}$ are given by Eqs. (17) and (16), respectively. Substituting Eq. (23) into Eq. (22), we obtain

$$\begin{aligned} \widetilde{M}^\sigma &= B_2^H \left(\mathbf{I} - B_1 (B_1^H B_1)^\dagger B_1^H\right) B_2 \\ &= \widetilde{T}^H \Omega^{\frac{1}{2}} \Pi \Omega^{\frac{1}{2}} \widetilde{T}, \end{aligned} \tag{24}$$

in which

$$\Pi = \mathbf{I} - \Omega^{\frac{1}{2}} A^\top \left(A\Omega A^\top\right)^\dagger A\Omega^{\frac{1}{2}} \in \mathbb{R}^{(m+m')\times(m+m')}.$$

As a result, it can be concluded that

$$\widetilde{M} = L(\widetilde{G}^z) + \widetilde{M}^\sigma = L(\widetilde{G}^z) + \widetilde{T}^H \Omega^{\frac{1}{2}} \Pi \Omega^{\frac{1}{2}} \widetilde{T}.$$

Furthermore, it is known that $X^\top (XX^\top)^\dagger = X^\dagger$ for any matrix X, then we further obtain

$$\begin{aligned} \Pi &= \mathbf{I} - \Omega^{\frac{1}{2}} A^\top \left(A\Omega A^\top\right)^\dagger A\Omega^{\frac{1}{2}} \\ &= \mathbf{I} - \left(A\Omega^{\frac{1}{2}}\right)^\dagger A\Omega^{\frac{1}{2}}, \end{aligned} \tag{25}$$

which according to [59, Chapter 5.13] is the matrix of orthogonal projection operator $\pi : \mathbb{C}^{m+m'} \to \ker(A(\overrightarrow{G})\Omega^{\frac{1}{2}})$ onto the kernel space of $A\Omega^{\frac{1}{2}}$. In addition, similar to [33, Appendix B.2], it is possible to further decompose Π in terms of sparse matrices and their inverse for efficient computation even though Π is in general a dense matrix.

APPENDIX B. PROOFS OF THE LEMMAS AND PROPOSITIONS IN SECTION V

In this section, we present proofs of the lemmas and propositions in Section V. These proofs draw heavily on [51] and follows a similar procedure to that of [33, Appendix C] and [38, Section 4.3].

B.1. Proof of Lemma 1

It is known that the unconstrained Euclidean gradient of $F(z) \triangleq z^H \widetilde{M} z$ is $\nabla F(z) = 2\widetilde{M} z$, and thus, if we let $S(z) \triangleq \widetilde{M} - \Re\{\text{ddiag}(\widetilde{M} zz^H)\}$, the Riemannian gradient is

$$\begin{aligned} \text{grad}\, F(z) &= \text{proj}_x(\nabla F(z)) \\ &= 2(\widetilde{M} - \Re\{\text{ddiag}(\widetilde{M} zz^H)\})z \\ &= 2S(z)z, \end{aligned} \tag{26}$$

in which the linear projection operator $\text{proj}_z : \mathbb{C}^n \to T_z\mathbb{C}_1^n$ is defined to be

$$\text{proj}_z u = u - \Re\{\text{ddiag}(uz^H)\}z.$$

In addition, it should be noted that we have assumed that $\mathbb{C}_1^n$ is a Riemannian submanifold of Euclidean space, then the Riemannian Hessian is

$$\text{Hess}\, F(z)[\dot{z}] = \text{proj}_z \text{D}\,\text{grad}\, F(z)[\dot{z}] = \text{proj}_z 2S(z)\dot{z}, \tag{27}$$

in which $\text{D}\,\text{grad}\, F(z)[\dot{z}]$ is the direction derivative of $\text{grad}\, F(z)$ along direction $\dot{z}$. From Eq. (27), we obtain

$$\langle \text{Hess}\, F(z)[\dot{z}], \dot{z}\rangle = 2\langle S(z)\dot{z}, \dot{z}\rangle.$$

Moreover, according to [51, Chapter 5], if $\exp_z : T_z\mathbb{C}_1^n \to \mathbb{C}_1^n$ is the exponential map at $z \in \mathbb{C}_1^n$, we obtain

$$\left.\frac{\text{d}}{\text{d}t} F \circ \exp_z(t\dot{z})\right|_{t=0} = \langle \text{grad}\, F(z), \dot{z}\rangle$$

and

$$\left.\frac{\text{d}^2}{\text{d}t^2} F \circ \exp_z(t\dot{z})\right|_{t=0} = \langle \text{Hess}\, F(z)[\dot{z}], \dot{z}\rangle.$$

Therefore, if $\hat{z} \in \mathbb{C}_1^n$ is a local optimum for Eq. (QP) and $\hat{S} = S(\hat{z})$, it is required that $\hat{S}\hat{z} = 0$ and $\langle \dot{z}, \hat{S}\dot{z}\rangle \geq 0$ for all $\dot{z} \in T_x\mathbb{C}_1^n$, which completes the proof.

B.2. Proof of Lemma 2

It should be noted that (1) to (5) in Lemma 2 are KKT conditions of (SDP), which proves the necessity. Since the identity matrix $\mathbf{I} \in \mathbb{C}^{n\times n}$ is strictly feasible to Lemma 2, the Slater's condition is satisfied, which proves the sufficiency. In addition, it should be noted that the Slater's condition also holds for the dual of (SDP). If $\text{rank}(\hat{S}) = n - 1$, according to [60, Theorem 6], $\hat{S}$ is dual nondegenerate. Moreover, by complementary slackness, $\hat{S}$ is also optimal for the dual of (SDP), which, as a result of [60, Theorem 10], implies that $\hat{X}$ is unique. If $\text{rank}(\hat{S}) = n - 1$, it can be concluded that $\hat{X}$ has rank one from $\hat{S}\hat{X} = \mathbf{0}$.

B.3. Proof of Lemma 3

Since $\hat{z} \in \mathbb{C}_1^n$ is a first-order critical point and $\hat{S} \succeq 0$, we conclude that $\hat{z}$ is a second-order critical point from Lemma 1. Also it can be checked that $\hat{X} = \hat{z}^H \hat{z} \in \mathbb{H}^n$ satisfies (1) to (5) in Lemma 2, thus, $\hat{z}$ solves (QP), and $\hat{X}$ solves (SDP) and is the unique global optimum for (SDP) if $\mathrm{rank}(\hat{S}) = n - 1$.

B.4. Proof of Proposition 1

In order to prove Proposition 1, we need Propositions 4 and 5 as follows.

Proposition 4. If $\underline{M} \in \mathbb{H}^n$ is data matrix of the form Eq. (18) that is constructed with the true (latent) relative measurements, and $\underline{z} \in \mathbb{C}_1^n$ is the true (latent) value of rotational states z, then $\underline{M}\,\underline{z} = \mathbf{0}$ and $\lambda_2(\underline{M}) > 0$.

Proof. For consistency, we assume that (P) and (QP) are formulated with the true (latent) relative measurements. Let $\underline{s} \in \mathbb{C}^{n'}$ and $\underline{c} \in \mathbb{C}^n$ be the true (latent) value of landmark positions and translational states c, respectively, then $\underline{\xi} = \begin{bmatrix} \underline{s}^\top & \underline{c}^\top & \underline{z}^\top \end{bmatrix}^\top \in \mathbb{C}^{n'} \times \mathbb{C}^n \times \mathbb{C}_1^n$ solves (P), and the optimal objective value is 0. Since (QP) is equivalent to (P), it can be concluded that $\underline{z} \in \mathbb{C}_1^n$ solves (QP), and the optimal objective value of (QP) is 0 as well. Furthermore, since $\underline{M} \succeq 0$, we obtain $\underline{M}\,\underline{z} = \mathbf{0}$. Let $\Xi \triangleq \mathrm{diag}\{\underline{z}_1, \cdots, \underline{z}_n\} \in \mathbb{C}^{n \times n}$ and $L(W^z) \in \mathbb{R}^{n \times n}$ be the Laplacian such that

$$[L(W^z)]_{ij} \triangleq \begin{cases} \displaystyle\sum_{(i,k) \in \mathcal{E}} \kappa_{ik}, & i = j, \\ -\kappa_{ij}, & (i,j) \in \mathcal{E}, \\ 0, & \text{otherwise}, \end{cases}$$

we obtain $L(\underline{G}^z) = \Xi L(W^z)\Xi^H$. It should be noted that G is assumed to be connected, and as a result, $\lambda_2(L(\underline{G}^z)) > 0$ and $L(\underline{G}^z)\underline{z} = \mathbf{0}$. Furthermore, it is by the definition of $\underline{M}$ or $\widetilde{M}$ in Eq. (14) that

$$\underline{M} = L(\underline{G}^z) + \underline{M}^\sigma, \tag{28}$$

in which $\underline{M}^\sigma = \underline{\Sigma}^z - \underline{\Theta}^H \Lambda^\dagger \underline{\Theta}$. From Eqs. (21a) to (21c) and (22), we obtain that $\underline{M}^\sigma$ is the Schur complement of

$$\begin{bmatrix} B_1^H B_1 & B_1^H B_2 \\ B_2^H B_1 & B_2^H B_2 \end{bmatrix} = \begin{bmatrix} B_1^H \\ B_2^H \end{bmatrix} \begin{bmatrix} B_1 & B_2 \end{bmatrix} \succeq 0,$$

which suggests that $\underline{M}^\sigma \succeq 0$ and $\lambda_1(\underline{M}^\sigma) \geq 0$. As a result of Eq. (28), $\lambda_2(L(\underline{G}^z)) > 0$ and $\lambda_1(\underline{M}^\sigma) \geq 0$, we obtain

$$\lambda_2(\underline{M}) \geq \lambda_2(L(\underline{G}^z)) + \lambda_1(\underline{M}^\sigma) > 0,$$

which completes the proof. $\square$

Proposition 5. If $\underline{z} \in \mathbb{C}_1^n$ is the true (latent) value of $z \in \mathbb{C}_1^n$, and $\hat{z}$ solves (QP), and $d(\underline{z}, \hat{z}) \triangleq \min_{\theta \in \mathbb{R}} \|\hat{z} - e^{\mathbf{i}\theta}\underline{z}\|$, then we obtain

$$d(\underline{z}, \hat{z}) \leq 2\sqrt{\frac{n\|\widetilde{M} - \underline{M}\|_2}{\lambda_2(\underline{M})}} \tag{29}$$

Proof. If we define $\Delta M \triangleq \widetilde{M} - \underline{M} \in \mathbb{H}^n$ to be the perturbation matrix, then

$$\underline{z}^H \widetilde{M}\,\underline{z} = \underline{z}^H \underline{M}\,\underline{z} + \underline{z}^H \Delta M\underline{z} = \underline{z}^H \Delta M\underline{z} \leq n\|\Delta M\|_2, \tag{30}$$

in which, according to Proposition 4, $\underline{z}^H \underline{M}\,\underline{z} = 0$. In addition, it should be noted that

$$\underline{z}^H \widetilde{M}\,\underline{z} \geq \hat{z}^H \widetilde{M}\,\hat{z} \tag{31}$$

and

$$\hat{z}^H \widetilde{M}\,\hat{z} = \hat{z}^H \underline{M}\hat{z} + \hat{z}^H \Delta M\hat{z} \geq \hat{z}^H \underline{M}\hat{z} - n\|\Delta M\|_2. \tag{32}$$

From Eqs. (30) to (32), we obtain

$$2n\|\Delta M\|_2 \geq \hat{z}^H \underline{M}\hat{z} \tag{33}$$

From Proposition 4, we obtain

$$\hat{z}^H \underline{M}\hat{z} = (\hat{z} - \frac{1}{n}\underline{z}^H \hat{z}\underline{z})^H \underline{M}(\hat{z} - \frac{1}{n}\underline{z}^H \hat{z}\underline{z}) \\ \geq \lambda_2(\underline{M})\|\hat{z} - \frac{1}{n}\underline{z}^H \hat{z}\underline{z}\|^2, \tag{34}$$

in which the equality "=" uses $\underline{M}\,\underline{z} = \mathbf{0}$ and the inequality "$\geq$" uses $\lambda_2(\underline{M}) > \lambda_1(\underline{M}) = 0$ and $\underline{z}^H(\hat{z} - \frac{1}{n}\underline{z}^H \hat{z}\underline{z}) = 0$. Furthermore, an algebraic manipulation indicates that

$$\|\hat{z} - \frac{1}{n}\underline{z}^H \hat{z}\underline{z}\|^2 = n - \frac{1}{n}|\underline{z}^H \hat{z}|^2. \tag{35}$$

From Eqs. (34) and (35), we obtain

$$\hat{z}^H \underline{M}\hat{z} \geq \lambda_2(\underline{M})\|\hat{z} - \frac{1}{n}\underline{z}^H \hat{z}\underline{z}\|^2 \\ = \frac{1}{n}\lambda_2(\underline{M})(n^2 - |\underline{z}^H \hat{z}|^2) \\ \geq \lambda_2(\underline{M})(n - |\underline{z}^H \hat{z}|), \tag{36}$$

in which the last inequality "$\geq$" uses the Cauchy-Schwarz inequality

$$|\underline{z}^H \hat{z}| \leq \|\underline{z}\| \cdot \|\hat{z}\| = n.$$

Substituting Eq. (36) into Eq. (33) and simplifying the resulting equation, we obtain

$$n - |\underline{z}^H \hat{z}| \leq \frac{2n\|\Delta M\|_2}{\lambda_2(\underline{M})}. \tag{37}$$

In addition, from [38, Eq. (4.1)], it is known that $d(\underline{z}, \hat{z}) = \sqrt{2n - 2|\underline{z}^H \hat{z}|}$, and then from Eq. (37), we further obtain Eq. (29), which completes the proof. $\square$

To prove Proposition 1, we first decompose $\hat{S} = \widetilde{M} - \Re(\mathrm{ddiag}(\widetilde{M}\hat{z}^H \hat{z}))$ as follows:

$$\hat{S} = \widetilde{M} - \Re(\mathrm{ddiag}(\widetilde{M}\hat{z}^H \hat{z})) \\ = \underline{M} + \Delta M - \\ \Re\left\{\mathrm{ddiag}\left((\underline{M} + \Delta M)(\underline{z} + \Delta z)(\underline{z} + \Delta z)^H\right)\right\} \\ = \underline{M} + \Delta M - \Re\{\mathrm{ddiag}(\underline{M}\Delta z z^H + \underline{M}\Delta z \Delta z^H + \\ \underbrace{\Delta M(\underline{z} + \Delta z)(\underline{z} + \Delta z)^H)\}}_{\Delta S},$$

in which $\underline{z} \in \mathbb{C}_1^n$ is the true (latent) value of $z \in \mathbb{C}_1^n$ such that $\underline{M}\,\underline{z} = \mathbf{0}$, and $\hat{z}$ solves (QP), and $\Delta z \triangleq \hat{z} - \underline{z}$. In addition, we assume $\|\hat{z} - \underline{z}\| = d(\hat{z}, \underline{z}) \triangleq \min_{\theta \in \mathbb{R}} \|\hat{z} - e^{\mathbf{i}\theta}\underline{z}\|$. It is obvious that $\|\Delta S\|_2 \to 0$ as long as $\|\Delta M\|_2 \to 0$ and $\|\Delta z\| \to 0$, and by Proposition 5, we obtain $\|\Delta z\| \to 0$ as long as $\|\Delta M\|_2 \to 0$.

As a result, from continuity, there exists some $\gamma > 0$ such that $\|\Delta S\|_2 < \lambda_2(\underline{M})$ as long as $\|\Delta M\|_2 < \gamma$. Then we obtain

$$\lambda_i(\hat{S}) \geq \lambda_i(\underline{M}) - \|\Delta S\|_2 > \lambda_i(\underline{M}) - \lambda_2(\underline{M}) \geq 0$$

for all $i \geq 2$, which implies that $\hat{S}$ at least has $n-1$ positive eigenvalues. In addition, by Lemma 1, we obtain $\hat{S}\hat{z} = \mathbf{0}$, from which it can be concluded that $\hat{S} \succeq 0$ and $\text{rank}(\hat{S}) = n-1$. Furthermore, Lemma 3 guarantees that $\hat{X} = \hat{z}\hat{z}^H \in \mathbb{H}^n$ is the unique optimum of (SDP) if $\hat{S} \succeq 0$ and $\text{rank}(\hat{S}) = n-1$.

APPENDIX C. THE PRECONDITIONER FOR THE TRUNCATED CONJUGATE GRADIENT METHOD IN CPL-SLAM

In this section, we propose a preconditioner for the truncated conjugate gradient method in CPL-SLAM that targets for planar graph-based SLAM with landmarks.

Similar to [33], [34], [61], instead of factorizing the Riemannian Hessian matrix $\text{Hess}\,F(z) \in \mathbb{C}^{n \times n}$ in Eq. (27) to explicitly evaluate the descent direction, the Riemannian trust region (RTR) method in CPL-SLAM leverages the truncated conjugated gradient (TCG) method to approximate the descent direction with necessary accuracy. Though the TCG method is guaranteed to converge to the true solution within finite iterations, the rates of the convergence is closely related with the preconditioner $\text{Precon}(\text{Hess}\,F(z)) \in \mathbb{C}^{n \times n}$ that is used to approximate $\text{Hess}\,F(z)$ and iteratively solve

$$\text{Precon}(\text{Hess}\,F(z)) \cdot a = b$$

to evaluate the descent direction, in which $a,\, b \in \mathbb{C}^n$.

In graph-based SLAM, several preconditioners have been proposed for the TCG method [34], [61]. For CPL-SLAM, an immediate choice of the preconditioner $\text{Precon}(\text{Hess}\,F(z))$ is $L(\widetilde{G}^z) + \widetilde{\Sigma}^z$, which is the submatrix of $\widetilde{\Gamma}$ in Eq. (11) that corresponds to the rotational states,[3] and such a preconditioner works well for planar graph-based SLAM without landmarks. However, the preconditioner of $L(\widetilde{G}^z) + \widetilde{\Sigma}^z$ suffers slow convergence for planar graph-based SLAM with landmarks since the submatrix $L(\widetilde{G}^z) + \widetilde{\Sigma}^z$ loses information of pose-landmark measurements and results in a bad approximation of $\text{Hess}\,F(z)$.

In contrast to $L(\widetilde{G}^z) + \widetilde{\Sigma}^z$ that only captures the information of pose-pose measurements, the matrix

$$\widetilde{M} = L(\widetilde{G}^z) + \widetilde{T}^H \Omega^{\frac{1}{2}} \Pi \Omega^{\frac{1}{2}} \widetilde{T} \in \mathbb{C}^{n \times n} \qquad (38)$$

in (QP) implicitly but properly keeps the information of both pose-pose measurements and pose-landmark measurements. However, there is no exact expression of $\widetilde{M}$ and we need to evaluate the equation above to factorize $\widetilde{M}$. Furthermore, since $\widetilde{M}$ can be a dense matrix, the resulting factorization of $\widetilde{M}$ might be inefficient to solve

$$\widetilde{M} \cdot a = b, \qquad (39)$$

which affects the performance of the TCG method. As a result, we need some other methods rather than evaluate and factorize $\widetilde{M}$ explicitly to solve Eq. (39).

[3]A similar preconditioner is also used in SE-Sync [61].

It should be noted that the solution to

$$\min_{x \in \mathbb{C}^n} \frac{1}{2}\langle x, \widetilde{M}x \rangle - \langle x, b \rangle, \qquad (40)$$

is also a solution to Eq. (39). From Eqs. (11) to (14), it is straightforward to show that Eq. (40) is equivalent to

$$\min_{x \in \mathbb{C}^n,\, x' \in \mathbb{C}^{n+n'}} \frac{1}{2}\langle \begin{bmatrix} x' \\ x \end{bmatrix}, \widetilde{\Gamma} \begin{bmatrix} x' \\ x \end{bmatrix} \rangle - \langle \begin{bmatrix} x' \\ x \end{bmatrix}, \begin{bmatrix} \mathbf{0} \\ b \end{bmatrix} \rangle, \qquad (41)$$

and the solution $\begin{bmatrix} a' \\ a \end{bmatrix} \in \mathbb{C}^{2n+n'}$ to Eq. (41) can be computed in closed form as

$$\widetilde{\Gamma} \begin{bmatrix} a' \\ a \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ b \end{bmatrix}, \qquad (42)$$

or equivalently,

$$\begin{bmatrix} a' \\ a \end{bmatrix} = \widetilde{\Gamma}^\dagger \begin{bmatrix} \mathbf{0} \\ b \end{bmatrix}. \qquad (43)$$

It is by definition that $a \in \mathbb{C}^n$ in Eqs. (42) and (43) is also a solution to Eqs. (39) and (40). Therefore, we might factorize $\widetilde{\Gamma}$ and solve Eqs. (42) and (43) instead so as to solve Eq. (39). In Eqs. (14), (39) and (40), we construct the dense data matrix $\widetilde{M}$ using Eq. (13) to reduce the dimension of the optimization problem with no information loss. In Eqs. (42) and (43), on the other hand, we essentially reverse the operation of Eq. (13) to recover the sparse matrix $\widetilde{\Gamma}$ by augmenting the dimension. Since $\widetilde{\Gamma}$ is a sparse matrix whose exact expression requires no extra computation, it is more efficient to exploit the sparsity of $\widetilde{\Gamma}$ to solve Eqs. (42) and (43) than factorize the dense data matrix $\widetilde{M}$ to solve Eqs. (39) and (40).

APPENDIX D. THE COMPLEX OBLIQUE MANIFOLD

In this section, we give a brief review of the Riemannian geometric concepts and operators of the complex oblique manifold that are used in this paper. A detailed introduction to Riemannian geometry and optimization can be found in [51].

In Riemannian geometry, the complex oblique manifold

$$\text{OB}(r, n) \triangleq \{Y \in \mathbb{C}^{n \times r} | \text{ddiag}(YY^H) = \mathbf{I}\}$$

is a smooth and compact complex matrix manifold, whose tangent space at $Y \in \text{OB}(r, n)$ is

$$T_Y \text{OB}(r, n) \triangleq \{U \in \mathbb{C}^{n \times r} | \Re\{\text{ddiag}(UY^H)\} = \mathbf{O}\}.$$

For any $U \in \mathbb{C}^{n \times r}$, we define the projection operator $\text{proj}_Y : \mathbb{C}^{n \times r} \to T_Y \text{OB}(r, n)$ to be

$$\text{proj}_Y U \triangleq U - \Re\{\text{ddiag}(UY^H)\}Y.$$

7 For a smooth function $F : \text{OB}(r, n) \to \mathbb{R}$, it is by definition that the Riemannian gradient for $F(Y)$ is

$$\text{grad}\,F(Y) \triangleq \nabla F(Y) - \Re\{\text{ddiag}(\nabla F(Y)Y^H)\}Y.$$

Since we have assumed that the complex oblique manifold is a Riemannian submanifold of Euclidean space, then for any tangent vector $\dot{Y} \in T_Y \text{OB}(r, n)$, the Riemannian Hessian for $F(Y)$ can be computed as

$$\text{Hess}\,F(Y)[\dot{Y}] \triangleq \text{proj}_Y \text{Dgrad}\,F(Y)[\dot{Y}],$$

in which $\text{D}\,\text{grad}\,F(z)[\dot{z}]$ is the direction derivative of $\text{grad}\,F(z)$ along direction $\dot{z}$.
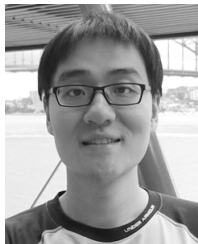
REFERENCES

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, 2016.

[2] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT press, 2005.

[3] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual SLAM algorithms: A survey from 2010 to 2016," *IPSJ Transactions on Computer Vision and Applications*, vol. 9, no. 1, p. 16, 2017.

[4] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3354–3361.

[5] A. Kleiner, J. Prediger, and B. Nebel, "Rfid technology-based exploration and slam for search and rescue," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 4054–4059.

[6] J. C. Kinsey, R. M. Eustice, and L. L. Whitcomb, "A survey of underwater vehicle navigation: Recent advances and new challenges," in *IFAC Conference of Manoeuvering and Control of Marine Craft*, vol. 88, 2006, pp. 1–12.

[7] J. Dong, J. G. Burnham, B. Boots, G. Rains, and F. Dellaert, "4d crop monitoring: Spatio-temporal reconstruction for agriculture," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[8] O. Vysotska and C. Stachniss, "Exploiting building information from publicly available maps in graph-based SLAM," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 4511–4516.

[9] J. Polvi, T. Taketomi, G. Yamamoto, A. Dey, C. Sandor, and H. Kato, "SlidAR: A 3d positioning method for SLAM-based handheld augmented reality," *Computers & Graphics*, vol. 55, pp. 33–43, 2016.

[10] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intelligent Transportation Systems Magazine*, 2010.

[11] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.

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

[13] T. Duckett, S. Marsland, and J. Shapiro, "Fast, on-line learning of globally consistent maps," *Autonomous Robots*, vol. 12, no. 3, pp. 287–300, 2002.

[14] U. Frese, P. Larsson, and T. Duckett, "A multilevel relaxation algorithm for simultaneous localization and mapping," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 196–207, 2005.

[15] E. Olson, J. Leonard, and S. Teller, "Fast iterative alignment of pose graphs with poor initial estimates," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE, 2006, pp. 2262–2269.

[16] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard, "A tree parameterization for efficiently computing maximum likelihood maps using gradient descent." in *Robotics: Science and Systems*, vol. 3, 2007, p. 9.

[17] T. Fan and T. Murphey, "Generalized proximal methods for pose graph optimization," in *International Symposium on Robotics Research (ISRR)*, 2019.

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

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

[20] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the bayes tree," *The International Journal of Robotics Research*.

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

[22] S. Huang, Y. Lai, U. Frese, and G. Dissanayake, "How far is slam from a linear least squares problem?" in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 3011–3016.

[23] H. Wang, G. Hu, S. Huang, and G. Dissanayake, "On the structure of nonlinearities in pose graph slam," *Robotics: Science and Systems VIII*, p. 425, 2013.

[24] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g^{2}o: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation*.

[25] L. Carlone and A. Censi, "From angular manifolds to the integer lattice: Guaranteed orientation estimation with application to pose graph optimization," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 475–492, 2014.

[26] L. Carlone, R. Aragues, J. A. Castellanos, and B. Bona, "A fast and accurate approximation for planar pose graph optimization," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 965–987, 2014.

[27] K. Khosoussi, S. Huang, and G. Dissanyake, "Exploiting the separable structure of slam," in *Robotics: Science and systems*, 2015.

[28] M. Liu, S. Huang, G. Dissanayake, and H. Wang, "A convex optimization based approach for pose slam problems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

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

[30] L. Carlone, G. C. Calafiore, C. Tommolillo, and F. Dellaert, "Planar pose graph optimization: Duality, optimal solutions, and verification," *IEEE Transactions on Robotics*, 2016.

[31] L. Carlone, D. M. Rosen, G. Calafiore, J. J. Leonard, and F. Dellaert, "Lagrangian duality in 3D SLAM: Verification techniques and optimal solutions," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[32] J. Briales and J. Gonzalez-Jimenez, "Fast global optimality verification in 3d slam," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[33] D. M. Rosen, L. Carlone, A. S. Bandeira, and J. J. Leonard, "SE-Sync: A certifiably correct algorithm for synchronization over the special Euclidean group," *The International Journal of Robotics Research*, 2019.

[34] J. Briales and J. Gonzalez-Jimenez, "Cartan-sync: Fast and global SE(d)-synchronization," *IEEE Robotics and Automation Letters*, 2017.

[35] J. G. Mangelson, J. Liu, R. M. Eustice, and R. Vasudevan, "Guaranteed globally optimal planar pose graph and landmark slam via sparse-bounded sums-of-squares programming," *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[36] A. Singer, "Angular synchronization by eigenvectors and semidefinite programming," *Applied and computational harmonic analysis*, vol. 30, no. 1, pp. 20–36, 2011.

[37] A. Singer and Y. Shkolnisky, "Three-dimensional structure determination from common lines in cryo-em by eigenvectors and semidefinite programming," *SIAM journal on imaging sciences*, vol. 4, no. 2, pp. 543–572, 2011.

[38] A. S. Bandeira, N. Boumal, and A. Singer, "Tightness of the maximum likelihood semidefinite relaxation for angular synchronization," *Mathematical Programming*, vol. 163, no. 1-2, pp. 145–167, 2017.

[39] N. Boumal, "Nonconvex phase synchronization," *SIAM Journal on Optimization*, vol. 26, no. 4, pp. 2355–2377, 2016.

[40] A. Eriksson, C. Olsson, F. Kahl, and T.-J. Chin, "Rotation averaging and strong duality," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 127–135.

[41] M. Betke and L. Gurvits, "Mobile robot localization using landmarks," *IEEE transactions on robotics and automation*, vol. 13, no. 2, pp. 251–263, 1997.

[42] A. S. Bandeira, "A note on probably certifiably correct algorithms," *Comptes Rendus Mathematique*, vol. 354, no. 3, pp. 329–333, 2016.

[43] N. Boumal, V. Voroninski, and A. Bandeira, "The non-convex Burer-Monteiro approach works on smooth semidefinite programs," in *Advances in Neural Information Processing Systems*, 2016.

[44] G. S. Chirikjian, *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*. Springer Science & Business Media, 2011, vol. 2.

[45] J. M. Selig, *Geometric fundamentals of robotics*. Springer Science & Business Media, 2004.

[46] T. Fan, H. Wang, M. Rubenstein, and T. Murphey, "Efficient and guaranteed planar pose graph optimization using the complex number representation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[47] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," Georgia Institute of Technology, Tech. Rep., 2012.

[48] C. Khatri and K. Mardia, "The von Mises-Fisher matrix distribution in orientation statistics," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 95–106, 1977.

[49] L. Carlone, "A convergence analysis for pose graph optimization via Gauss-Newton methods," in *2013 IEEE International Conference on Robotics and Automation*, 2013.

[50] J. Gallier, "The Schur complement and symmetric positive semidefinite (and definite) matrices," 2010.

[51] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[52] P.-A. Absil and K. A. Gallivan, "Joint diagonalization on the oblique manifold for independent component analysis," in *IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, 2006.

[53] P.-A. Absil, C. G. Baker, and K. A. Gallivan, "Trust-region methods on riemannian manifolds," *Foundations of Computational Mathematics*, vol. 7, no. 3, pp. 303–330, 2007.

[54] N. Boumal, "A Riemannian low-rank method for optimization over semidefinite matrices with block-diagonal constraints," *arXiv preprint arXiv:1506.00575*, 2015.

[55] Y. Latif, C. Cadena, and J. Neira, "Robust graph slam back-ends: A comparative analysis."

[56] S. M. Nasiri, R. Hosseini, and H. Moradi, "A recursive least square method for 3d pose graph optimization problem," *arXiv preprint arXiv:1806.00281*, 2018.

[57] R. Tron, D. M. Rosen, and L. Carlone, "On the inclusion of determinant constraints in Lagrangian duality for 3d slam," *Robot. Sci. Syst. Work.The Probl. Mob. Sensors*, 2015.

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

[59] C. D. Meyer, *Matrix analysis and applied linear algebra*. SIAM, 2000, vol. 71.

[60] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton, "Complementarity and nondegeneracy in semidefinite programming," *Mathematical programming*, 1997.

[61] D. Rosen and L. Carlone, "Computational enhancements for certifiably correct slam," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.
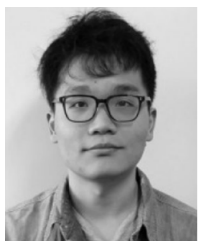
Michael Rubenstein received the Ph.D. degree in computer science from the University of Southern California, Los Angeles, CA, USA, in 2009. He is currently an Assistant Professor working on swarm robotics and control with the Department of Computer Science as well as the Department of Mechanical Engineering, Northwestern University, Evanston, IL, USA. His research interests include robot swarms and multirobot systems.

Todd Murphey received his B.S. degree in mathematics from the University of Arizona and the Ph.D. degree in Control and Dynamical Systems from the California Institute of Technology. He is a Professor of Mechanical Engineering at Northwestern University. His laboratory is part of the Neuroscience and Robotics Laboratory, and his research interests include robotics, control, computational methods for biomechanical systems, and computational neuroscience. Honors include the National Science Foundation CAREER award in 2006, membership in the 2014-2015 DARPA/IDA Defense Science Study Group, and Northwesterns Professorship of Teaching Excellence. He was a Senior Editor of the IEEE Transactions on Robotics.

Taosha Fan received his B.E. degree in automotive engineering from Tongji University, Shanghai, China, in 2013, and M.S. degrees in mechanical engineering and mathematics from Johns Hopkins University, Baltimore, MD, USA, in 2015. He is currently a Ph.D. candidate in mechanical engineering at Northwestern University, Evanston, IL, USA. His research interests lie at the intersection of robotic control, simulation and estimation.

Hanlin Wang received the B.E. degree in mechanical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2015. He is currently working toward the Ph.D. degree in computer science with the Department of Computer Science, Northwestern University, Evanston, IL, USA. His research interests include swarm systems, planning and scheduling, and distributed computation systems.