GeoD: Consensus-based **Geo**desic **D**istributed Pose Graph Optimization

Eric Cristofalo, Eduardo Montijano, and Mac Schwager

***Abstract*—We present a consensus-based distributed pose graph optimization algorithm for obtaining an estimate of the 3D translation and rotation of each pose in a pose graph, given noisy relative measurements between poses. The algorithm, called GeoD, implements a continuous time distributed consensus protocol to minimize the geodesic pose graph error. GeoD is distributed over the pose graph itself, with a separate computation thread for each node in the graph, and messages are passed only between neighboring nodes in the graph. We leverage tools from Lyapunov theory and multi-agent consensus to prove the convergence of the algorithm. We identify two new consistency conditions sufficient for convergence: *pairwise consistency* of relative rotation measurements, and *minimal consistency* of relative translation measurements. GeoD incorporates a simple one step distributed initialization to satisfy both conditions. We demonstrate GeoD on simulated and real world SLAM datasets. We compare to a centralized pose graph optimizer with an optimality certificate (SE-Sync) and a Distributed Gauss-Seidel (DGS) method. On average, GeoD converges 20 times more quickly than DGS to a value with 3.4 times less error when compared to the global minimum provided by SE-Sync. GeoD scales more favorably with graph size than DGS, converging over 100 times faster on graphs larger than 1000 poses. Lastly, we test GeoD on a multi-UAV vision-based SLAM scenario, where the UAVs estimate their pose trajectories in a distributed manner using the relative poses extracted from their on board camera images. We show qualitative performance that is better than either the centralized SE-Sync or the distributed DGS methods.**

github.com/ericcristofalo/GeoD

***Index Terms*—Pose Graph Optimization, Multi-Robot Systems, Distributed Robot Systems, SLAM**

I. Introduction

In this paper, we propose a distributed algorithm for pose graph optimization called **Geo**desic **D**istributed Pose Graph Optimization, or GeoD. Our algorithm uses rotation errors computed from the geodesic distance on the manifold of rotations. This is in contrast to existing methods, which typically use an off-manifold rotation error known as the chordal distance. Our algorithm is based on multi-agent consensus, and is therefore distributed over the topology of the pose graph itself. Each pose in the pose graph is optimized through simple iterative update rules that implement a continuous time gradient descent on the geodesic pose graph error. Using tools from Lyapunov theory, we analytically prove that GeoD converges to a set of equilibria corresponding to local minima of the pose graph error. We show that GeoD gives fast and accurate empirical performance in comparisons with state-of-the-art centralized and distributed pose graph optimization algorithms.

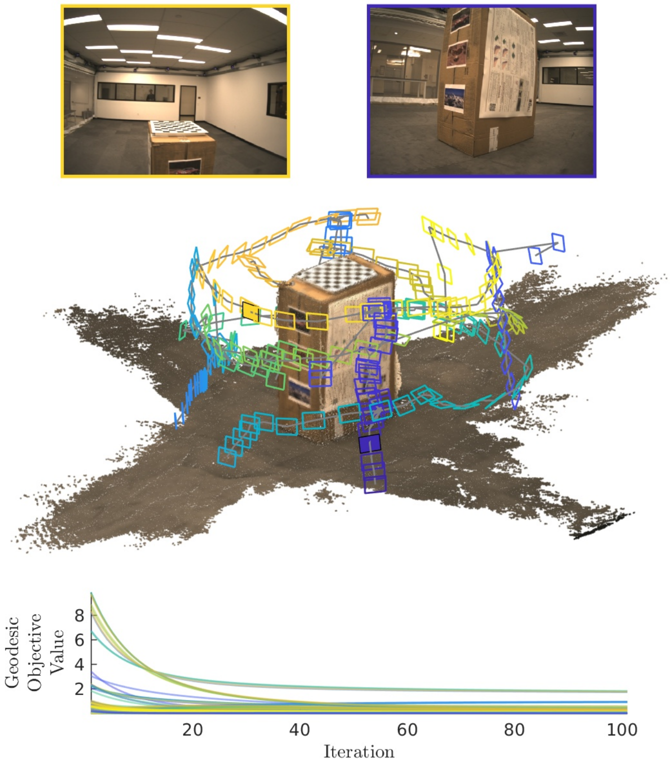


Fig. 1. We present GeoD, a consensus-based distributed $SE(3)$ pose graph optimization algorithm with provable convergence guarantees. This algorithm enables a group of robots, given noisy relative pose measurements, to reach agreement on the group's $SE(3)$ pose history in a distributed manner. The figure shows GeoD applied to a group of 7 quadrotor robots. (Top) Two images showing onboard views used to compute relative poses. (Middle) The reconstructed 3D map and pose history of all quadrotors. (Bottom) The geodesic objective value over time, showing convergence of the estimate for all quadrotor poses.

Our algorithm has the potential to enable systems with hundreds or thousands of robots to cooperatively maintain large-scale SLAM maps of an area. For example, a fleet of autonomous vehicles could use GeoD to maintain an up-to-date distributed SLAM map of an entire city, incorporating the sensor measurements of all vehicles, while distributing the storage and computation among the vehicles. Similarly, a fleet of delivery drones could use GeoD to cooperatively maintain an estimate of the locations of all drones in the area,

This work was supported by a NDSEG fellowship; Spanish projects PGC2018-098817-A-I00 (MINECO/FEDER), DGA T04-FSE; NSF grants CNS-1330008 and IIS-1646921; ONR grant N00014-18-1-2830 and ONRG-NICOP-grant N62909-19-1-2027; and the Ford-Stanford Alliance Program.

E. Cristofalo and M. Schwager are with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, USA (ecristof;schwager@stanford.edu)

E. Montijano is with Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, Zaragoza 50018, Spain (emonti@unizar.es).

while maintaining a 3D map of a neighborhood for making safe door-step deliveries. In such safety critical multi-robot systems, mathematical assurances that the algorithm will not diverge or oscillate in an unstable fashion are essential. GeoD provides such convergence assurances.

While the majority of pose graph optimization algorithms are centralized [1]–[3], GeoD is distributed, anytime, and consensus-based. Unlike other distributed pose graph optimization algorithms [4], GeoD is distributed over the pose graph itself, so there is a different computational thread responsible for optimizing each pose in the graph. Each thread only passes messages with other threads that are neighbors in the pose graph. This architecture is flexible in that the multiple threads can be partitioned among multiple robots as a distributed multi-robot SLAM back-end. Each robot implements its own threads and communicates over a wireless network to pass messages with threads owned by other robots. In the case of a single robot, our algorithm also can be implemented on a GPU to give a fast, parallelized SLAM back-end.

GeoD operates on the geodesic distance on $SO(3)$ (the Special Orthogonal group of 3D rotation matrices) as opposed to using convex chordal relaxations, which is the state-of-the-art in the literature [2]–[4]. We leverage tools from Lyapunov theory and multi-agent consensus to formally prove that GeoD will converge on an estimate. We introduce the concept of *measurement consistency* to define the conditions for convergence which have not previously been identified in the consensus-based pose estimation literature [5]–[7]. Given arbitrary noisy relative pose measurements, GeoD starts with a single-step distributed initialization round to satisfy these conditions for convergence.

We illustrate the advantages of GeoD by comparing it to state-of-the-art centralized (SE-Sync [3]) and distributed (DGS [4]) chordal relaxation approaches. On average over several SLAM datasets, GoeD converges to a solution more than $700\times$ faster than SE-Sync with only 32.5% error from the global minimum on chordal value — despite its gradient-like nature — compared to 110.2% error obtained by DGS. Our method requires similar computation time to DGS for small networks, but requires approximately $100\times$ less time for large networks that contain more than 1000 poses. We finally demonstrate our algorithm in hardware experiments with a multi-UAV network, where the camera-equipped drones optimize a SLAM map using noisy relative pose measurements extracted from shared image features (see Fig. 1).

This work builds upon an earlier conference version [8], which used the angle-axis rotation parameterization. The current paper presents a fundamentally new algorithm using rotation matrices, together with an entirely new mathematical analysis. We also present new performance comparisons with two state-of-the-art pose graph optimization algorithms, and give results from new hardware experiments. We briefly compare GeoD with the earlier iteration from [8] in Section VI and show that GeoD is advantageous in terms of both convergence speed as well as solution accuracy by over a factor of $5\times$.

The rest of this paper is organized as follows. We review the pose graph optimization literature in Section II. We present the problem formulation and overview our solution in Section III. We briefly review the group of 3D rotation matrices in Section IV. We prove convergence of GeoD in Section V. Lastly, we present comparisons to state-of-the-art centralized and distributed approaches on SLAM datasets and a multi-agent system experiment in Section VI. Finally, we draw conclusions in Section VII.

II. Related Work

Pose graph optimization refers to a standard class of optimization problems in which a set of SE(3) rigid body poses (usually corresponding to poses for a robot, or robots, over time) must be estimated based on noisy relative measurements between these poses. Each pose represents a node and each relative measurement represents an edge, hence the underlying structure is represented by a graph. Pose graph optimization is fundamental to Simultaneous Localization and Mapping (SLAM), as it typically forms the “back-end” of a SLAM system, while the “front-end” produces the edges in the pose graph by extracting relative pose estimates from raw sensor measurements [1], [9], [10]. While mostly studied in the context of SLAM, pose graph optimization can also be used as an underlying optimization model for multi-robot relative localization [11], trajectory planning [12], structure from motion [13], and a variety of other problems fundamental to robotics [14].

Pose graph optimization is a nonlinear, nonconvex, least squares optimization for a translation vector and rotation matrix corresponding to each pose in the network. Advances in bundle adjustment have led to very efficient gradient methods [15] or Levenberg-Marquardt algorithms [2] for pose graph optimization. Most recently, the authors of [3], [16] introduced a semi-definite relaxation to the pose graph optimization problem that can be efficiently solved using an algorithm called SE-Sync. This algorithm, under broad conditions, gives a certificate of optimality in the following sense: if the certificate is produced, the solution is globally optimal, if not, the solution may or may not be global optimal.

These previous pose graph optimization methods are centralized and do not immediately lend themselves to decentralization. Distributed pose graph optimization is important for applications where multiple robots must collaborate to estimate a common SLAM map. Distributed algorithms for pose graph optimization are also important for taking advantage of multiple cores on a single chip, parallelization in a GPU, or in maintaining very large SLAM maps over many machines in a data center, as in cloud robotics applications. There are numerous multi-robot SLAM solutions that consider the distributed maximum likelihood estimation problem for linear Gaussian systems [17] and 3D nonlinear systems [18]. There are also distributed Kalman filter-like approaches [19], filtering methods using vision-specific sensors [20], and distributed SLAM with cloud-based resources [21]. Like the centralized approaches, distributed pose graph optimizers similarly rely on a convex chordal relaxation, but employ either distributed Successive Over-Relaxation or Jacobi Over-Relaxation solvers [4]. Specifically, [4] presents a two-stage algorithm for Distributed

Gauss-Seidel (DGS) optimization that first distributes the chordal initialization of rotations, and then distributes a single Gauss-Newton step of the full pose graph optimization problem. In contrast, our method optimizes the full $SE(3)$ state in a single algorithm and does not require a chordal relaxation. A recent distributed pose graph solver [22] provides a certificate of optimality using a sparse semidefinite relaxation, similarly to SE-Sync. Finally, another recent solver is asynchronous and distributed [23], but requires the pose graph updates to be executed separately, considering a single robot at a given time. Differently, our algorithm is able to update all the graph poses simultaneously in the fashion of consensus-based estimation methods. In this paper we compare GeoD to SE-Sync [3], as a representative of a state-of-the-art centralized and certifiable pose graph solver, and DGS [4], as a representative of a state-of-the-art distributed pose graph solver.

Our algorithm is based on multi-agent consensus, which are a family of continuous time control algorithms for a group of computational agents to agree on a common variable of interest [24]. Consensus-based algorithms are advantageous because they only require communication among local neighbors in a graph. These algorithms are generally light-weight, simple to implement, and can come with formal guarantees of convergence. Our work builds upon consensus algorithms in the multi-robot control literature [25]–[27], where robots specifically reach consensus on their relative positions while sharing raw measurements.

Prior work on distributed consensus-based pose estimation can thus be split into work that considers either relative pose measurements corrupted by noise or not corrupted by noise. Without noise, classic consensus algorithms [24] are well-suited for translation-only estimation. A number of methods have been proposed for orientation synchronization without noise that consider the angle-axis representation [6], QR-factorization [28], synchronization on the manifold of $SO(3)$ [29], or rotation averaging [30]. A common-frame $SE(3)$ estimation method is proposed in [31], where all agents desire to reach consensus on a common frame given a relative $SE(3)$ pose from that frame.

Noisy relative measurements have largely been considered in the analysis of translation-only estimation, a form of distributed linear least squares estimation [32]. For example, [33] considers distance-only measurements while [34] considers bearing only measurements. Noisy relative translation measurements themselves are considered in [17], which discusses the distributed computation of the maximum likelihood translation estimate. Other methods solve for the centroid of the network and anchor-based translations simultaneously [35]. Asynchronous estimation is treated in [36] and 2D estimation with Gaussian noise is analyzed in [37]. There are even fewer solutions that consider distributed 3D pose estimation given noisy relative measurements. Most notably, the distributed gradient descent method in [38] sequentially considers three separate objectives: first a convex relaxation of the rotation-only objective, then a translation-only objective, and finally the full 3D pose objective. Recently, the distributed $SE(3)$ synchronization method in [7] guarantees convergence with an iterative algorithm that projects the rotation estimates at each step onto $SO(3)$. These works consider distributed estimation for 3D pose, but do not characterize and analyze the conditions on the noisy measurements for consensus-based estimation systems to converge as we do in Section V. Finally, the bulk of the existing methods are not deployed on real robotic networks in SLAM applications as our method is in Section VI.

III. Problem Formulation and Solution Overview

The goal of this paper is to compute the global translations and rotations of each pose in a pose graph to minimize a translation and rotation error with respect to the noisy relative measurements. Specifically, we aim to solve Problem 1,

Problem 1 (Pose Graph Optimization Problem)**.**

$$\underset{\substack{\mathbf{t}_1,\ldots,\mathbf{t}_n \\ \mathbf{R}_1,\ldots,\mathbf{R}_n}}{\text{minimize}} \sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i} \|\mathbf{t}_j - \mathbf{t}_i - \mathbf{R}_i\tilde{\mathbf{t}}_{ij}\|_2^2 + \|\log(\mathbf{R}_i^T\mathbf{R}_j\tilde{\mathbf{R}}_{ij}^T)^\vee\|_2^2 \,.$$

The output of the optimization are the n poses[1] that are labeled by the vertex set $\mathcal{V} = \{1,\ldots,n\}$. The 3D rigid body pose of $i \in \mathcal{V}$ is measured with respect to an arbitrary global coordinate frame as a tuple containing a translation vector and a rotation matrix, $\big(\mathbf{t}_i \in \mathbb{R}^3, \mathbf{R}_i \in SO(3)\big) \in SE(3)$, or the 3D Special Euclidean group. Here $SO(3) = \{\mathbf{R} \in \mathbb{R}^{3\times 3} \mid \mathbf{R}^T\mathbf{R} = \mathbf{I}_3,\ \det(\mathbf{R}) = +1\}$ is the 3D Special Orthogonal group where $\mathbf{I}_3$ is a 3×3 identity matrix and $\det(\cdot)$ is the determinant of a matrix.

The noisy relative measurements are defined over the undirected edge set $\mathcal{E}$. Each noisy measurement is represented the directed edge component (i, j) and is defined as the tuple $(\tilde{\mathbf{t}}_{ij}, \tilde{\mathbf{R}}_{ij}) \in SE(3)$. This notation indicates the pose of j measured in the coordinate frame of pose i. In an undirected graph, we assume there exists another unique noisy measurement in the opposite direction (of pose i in frame j) such that $\tilde{\mathbf{R}}_{ij}\tilde{\mathbf{R}}_{ji} \neq \mathbf{I}_3$, for example. The noisy measurements are assumed to be acquired prior to solving the problem and are not measured at each time-step of the solution as in a visual odometry (VO) or online SLAM framework. The vertex set and the edge set define the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Finally, each pose has a local neighborhood subgraph, $\mathcal{N}_i \subset \mathcal{V}$, that defines each pose's immediately connected neighbor labels.

The rotation error term in Problem 1 is the *geodesic distance* between the relative rotation $(\mathbf{R}_i^T\mathbf{R}_j)$ and the relative rotation measurement $(\tilde{\mathbf{R}}_{ij})$: $\|\log(\mathbf{R}_i^T\mathbf{R}_j\tilde{\mathbf{R}}_{ij}^T)^\vee\|_2$. Geodesics are distances along the manifold and require additional machinery from Lie group theory. We will briefly review concepts related to the Lie group structure of $SO(3)$ in Section IV. The use of the geodesic distance is in contrast to many recent works that instead consider the *chordal distance* [3], [4], [16]. The chordal distance, $\|\mathbf{R}_i^T\mathbf{R}_j - \tilde{\mathbf{R}}_{ij}\|_F$, computes the Frobenius matrix norm of the difference in rotation matrices. This objective is used in *chordal relaxation* methods — such as rotation initialization techniques [39] — that formulate a convex optimization to solve for the rotations embedded in a vector space ($\mathbb{R}^9$), and then projects the final result back to $SO(3)$. A great comparison of the geodesic and

[1] Each *pose* here can represent either i) the pose of one robot or multiple robots that are evolving over time (as in SLAM) or ii) the poses of a group of robots at a given time (as in multi-robot relative localization).

chordal distance for rotation averaging can be found in [30]. We present a detailed comparison to state-of-the-art chordal relaxation methods in Section VI.

Our distributed consensus-based $SE(3)$ method is conceptually similar to distributed formation control, where the global translation and rotation estimates of each individual pose i evolve according to rigid body kinematics,

$$\dot{\mathbf{t}}_i = \boldsymbol{\nu}_i\,, \tag{1a}$$

$$\dot{\mathbf{R}}_i = \mathbf{R}_i \boldsymbol{\omega}_i^\wedge\,. \tag{1b}$$

The *control inputs* to the kinematic system are the linear velocity $\boldsymbol{\nu}_i \in \mathbb{R}^3$ and angular velocity $\boldsymbol{\omega}_i \in \mathbb{R}^3$ for the rigid body pose. The *hat* operator $(\cdot)^\wedge$ maps a vector into its skew-symmetric matrix analog (more detail is provided in Section IV). In particular, we propose to use the following feedback input laws for pose graph optimization,

$$\boldsymbol{\nu}_i = \sum_{j\in\mathcal{N}_i} \mathbf{t}_j - \mathbf{t}_i - \mathbf{R}_i \tilde{\mathbf{t}}_{ij}\,, \tag{2a}$$

$$\boldsymbol{\omega}_i^\wedge = \sum_{j\in\mathcal{N}_i} \log(\mathbf{R}_i^T \mathbf{R}_j \tilde{\mathbf{R}}_{ij}^T)\,. \tag{2b}$$

Note that the control terms take the same form as the error terms in the objective of Problem 1. The full $SE(3)$ distributed estimation consists of executing the kinematics of system (1) with control inputs (2), i.e.,

$$\dot{\mathbf{t}}_i = \sum_{j\in\mathcal{N}_i} \mathbf{t}_j - \mathbf{t}_i - \mathbf{R}_i \tilde{\mathbf{t}}_{ij}\,, \tag{3a}$$

$$\dot{\mathbf{R}}_i = \mathbf{R}_i \left(\sum_{j\in\mathcal{N}_i} \log(\mathbf{R}_i^T \mathbf{R}_j \tilde{\mathbf{R}}_{ij}^T) \right)\,. \tag{3b}$$

In the rest of this paper, we refer to system (3) as the **Geo**desic **D**istributed pose graph optimization system, or GeoD. GeoD is nonconvex for the same reasons as in Problem 1. We also highlight that even though system (3) looks like the first-order dynamics of a physical rigid body, we are using it to describe the nonlinear rate of change of our $SE(3)$ pose estimation. Lastly, the two expressions in (3) are assumed to run simultaneously, which is a notable difference compared to [38] which sequentially considers rotation, translation, and then the full $SE(3)$ problem.

It is worth noting that the estimation scheme we propose is similar to other distributed consensus-based algorithms. In particular, it is inspired by previous 3D distributed pose estimation work [8] and by 3D formation control [27]. Note that there are three major differences. First, with respect to [8], we use the rotation matrix parameterization of rotations instead of the angle-axis. This allows us to converge to a solution that is closer to the global minimum as we demonstrate in Section VI-A1. Second, the measurements we consider are corrupted by noise and no longer encode a *desired formation* as in the formation control literature. Consensus-based estimators with perturbed measurements are not guaranteed to converge to a stable equilibria set in general [32]. We show conditions on $(\tilde{\mathbf{t}}_{ij}, \tilde{\mathbf{R}}_{ij})$ for convergence of this system in Section V. Third, the measurements are observed once at the beginning of the estimation and not at each step in the estimation. This way, the noise is not fed back into the system at each time step.

Finally, we define a few additional quantities to support the development in the next sections. The graph Laplacian $\mathbf{L} \in \mathbb{R}^{n\times n}$ for an undirected graph $\mathcal{G}$ is constructed such that each $(i,j)^{\text{th}}$ entry is l_{ij} and

$$l_{ij} = \begin{cases} |\mathcal{N}_i|\,, & \text{if } i = j \\ -1\,, & \text{if } i \neq j \text{ and } (i,j) \in \mathcal{E} \\ 0\,, & \text{otherwise} \end{cases}.$$

It is well known that for undirected graphs $\mathbf{L}$ has eigenvalues $\lambda_1 = 0 \leq \lambda_2 \leq \ldots \leq \lambda_n$ [24]. The eigenvector associated to λ_1 is in the set of vectors with equal entries, i.e., $\boldsymbol{\lambda}_1 \in \mathbb{1}_n = \{\mathbf{a} \in \mathbb{R}^n \mid \mathbf{a} = \alpha \mathbf{1}_n,\ \alpha \in \mathbb{R}\}$ [40], where $\mathbf{1}_n$ represents the n-dimensional column vector of ones. This vector is then a left and right eigenvector of $\mathbf{L}$ — as graph $\mathcal{G}$ is undirected — and the null space of $\mathbf{L}$ is $\text{Null}(\mathbf{L}) = \mathbb{1}_n$. Lastly, the full rotation estimates lie in $SO(3)^n = \{\mathbf{R}_1, \ldots, \mathbf{R}_n \mid \mathbf{R}_i \in SO(3), \forall i \in \mathcal{V}\}$.

IV. The $SO(3)$ Matrix Lie Group

We make extensive use of the *Lie group* $SO(3)$ in the analysis of GeoD and therefore we review several relevant properties, as well as introduce a new useful result. Both $SO(3)$ and $SE(3)$ are Lie groups which are simultaneously groups and differentiable manifolds [41]. Every Lie group has an associated *Lie algebra* which is defined as the *tangent space* of the Lie group at identity element of the group. Lie algebra often take non-trivial forms but have associated vector spaces.

Recall that $SO(3)$ is the set of valid 3D rotation matrices. The Lie algebra of $SO(3)$ is $so(3) = \{\mathbf{S} \in \mathbb{R}^{3\times 3} \mid \mathbf{S}^T = -\mathbf{S}\}$ which is the set of skew-symmetric 3×3 matrices. Define the *vee* operator $(\cdot)^\vee : so(3) \to \mathbb{R}^3$ as the map between the Lie algebra and the associated vector space. The inverse operation is the *hat* operator $(\cdot)^\wedge : \mathbb{R}^3 \to so(3)$. Elements of the Lie algebra are mapped back to the group $SO(3)$ using the *exponential map* $\exp(\mathbf{x}^\wedge) : so(3) \to SO(3)$, which is given in closed form by the Rodrigues' rotation formula,

$$\exp(\mathbf{x}^\wedge) = \mathbf{I}_3 + \sin(\theta)\left(\frac{\mathbf{x}}{\theta}\right)^\wedge + (1-\cos(\theta))\left(\frac{\mathbf{x}}{\theta}\right)^{\wedge 2},$$

where $\mathbf{x} \in \mathbb{R}^3$, $\mathbf{x}^\wedge \in so(3)$, and $\theta = \|\mathbf{x}\|_2$. The inverse mapping is the *logarithmic map* $\log(\mathbf{R}) : SO(3) \to so(3)$,

$$\log(\mathbf{R}) = \begin{cases} \frac{\theta}{2\sin(\theta)}(\mathbf{R} - \mathbf{R}^T)\,, & \text{if } \theta \neq 0 \\ 0\,, & \text{if } \theta = 0 \end{cases}, \tag{4}$$

where,

$$\theta = \arccos\left(\frac{1}{2}(\text{tr}(\mathbf{R}) - 1)\right)\,, \tag{5}$$

and $\text{tr}(\cdot)$ returns the trace of a matrix. The logarithmic map, or sometimes called the matrix logarithm, is defined for $|\theta| < \pi$ to ensure the mapping is bijective. The *angle-axis* representation of the rotation matrix $\mathbf{R}$ is defined by the corresponding element of the vector space $\mathbf{x} = \theta\mathbf{u} = \log(\mathbf{R})^\vee$, where $\theta = \|\mathbf{x}\|_2 = \|\log(\mathbf{R})^\vee\|_2$ is the magnitude of the rotation about a unit axis $\mathbf{u} = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}$. This is the same θ defined in (5) and corresponds to the geodesic distance along the manifold from the identity element to $\mathbf{R}$. Conversely, we can write the rotation matrix as a function of the angle-axis, i.e., $\mathbf{R} = \exp(\mathbf{x}^\wedge)$.

In this paper, we are primarily interested in expressions that are functions of Lie algebra (see (3b)) and thus state some general properties of the matrix logarithm on real-valued matrices. From properties of the matrix exponential, we know that $\exp(\mathbf{B}\mathbf{A}\mathbf{B}^{-1}) = \mathbf{B}\exp(\mathbf{A})\mathbf{B}^{-1}$ for any $\mathbf{A} \in \mathbb{R}^{d\times d}$ and invertible $\mathbf{B} \in \mathbb{R}^{d\times d}$. If the logarithmic map of this generic $\mathbf{A}$ exists such that $\log(\exp(\mathbf{A})) = \exp(\log(\mathbf{A})) = \mathbf{A}$, then we can also write $\log(\mathbf{B}\mathbf{A}\mathbf{B}^{-1}) = \mathbf{B}\log(\mathbf{A})\mathbf{B}^{-1}$. For rotation matrices $\mathbf{R}, \mathbf{Q} \in SO(d)$, the equivalent expression is,

$$\log(\mathbf{Q}\mathbf{R}\mathbf{Q}^T) = \mathbf{Q}\log(\mathbf{R})\mathbf{Q}^T\,. \tag{6}$$

Note that in general $\log(\mathbf{R}^T\mathbf{Q}) \neq \log(\mathbf{R}) - \log(\mathbf{Q})$, however this statement is indeed true for rotations in $SO(2)$ [42]. Additionally, $\log(\mathbf{R}^T) = \log(\mathbf{R})^T = -\log(\mathbf{R})$. Finally, we state the following lemma for several well-known results involving skew-symmetric matrices [43], [44].

Lemma 1. *It follows that for* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$*, and* $\mathbf{A} \in \mathbb{R}^{3\times 3}$ *that,*

$$\mathbf{x}^T\mathbf{y} = \frac{1}{2}\mathrm{tr}\left(\mathbf{x}^{\wedge T}\mathbf{y}^\wedge\right)\,, \tag{7}$$

$$\mathrm{tr}(\mathbf{A}\mathbf{x}^\wedge) = -\mathbf{x}^T(\mathbf{A} - \mathbf{A}^T)^\vee\,. \tag{8}$$

The rate of change of a rotation matrix is well known given a vector of rotation rates $\boldsymbol{\omega} \in \mathbb{R}^3$: $\dot{\mathbf{R}} = \mathbf{R}\boldsymbol{\omega}^\wedge$ and $\dot{\mathbf{R}} = \boldsymbol{\omega}^\wedge\mathbf{R}$ for a body-frame $\boldsymbol{\omega}$ and global-frame $\boldsymbol{\omega}$, respectively. Unfortunately, finding a closed-form expression for the time rate of change of $\log(\mathbf{R})$ is challenging because it requires differentiating an infinite series representation of the matrix logarithm (See 3.3 of [45]) or, for $SO(3)$, differentiating (4) with respect to time — leading to a cumbersome expression in both cases. Below we state a theorem that will be central to the Lyapunov analysis of our algorithm in Section V-A. The theorem allows us to compactly express the time derivative of the squared geodesic distance, without explicitly requiring the time derivative of the matrix logarithm itself.

Theorem 1. *For a rotation matrix* $\mathbf{R} \in SO(3)$*, if* $\|\log(\mathbf{R})^\vee\|_2 < \pi$*, then,*

$$\frac{d}{dt}\frac{1}{2}\|\log(\mathbf{R})^\vee\|_2^2 = \log(\mathbf{R})^{\vee T}(\mathbf{R}^T\dot{\mathbf{R}})^\vee\,. \tag{9}$$

Proof. See Appendix A. □

V. Analysis

In this section we prove that the dynamics described by the system (3) will converge to an equilibrium if the noisy measurements $(\tilde{\mathbf{t}}_{ij}, \tilde{\mathbf{R}}_{ij})$ are *consistent*. Before we begin the analysis, we introduce our taxonomy of measurement consistency with three definitions.

Definition 1 (Global consistency)**.** *The measurements are globally consistent if* $\tilde{\mathbf{R}}_{ij} = \tilde{\mathbf{R}}_{i\ell}\tilde{\mathbf{R}}_{\ell j}$ *and* $\tilde{\mathbf{t}}_{ij} = \tilde{\mathbf{t}}_{i\ell} + \tilde{\mathbf{R}}_{i\ell}\tilde{\mathbf{t}}_{\ell j}$*, for all* $i, \ell, j \in \mathcal{V}$*. In other words, the product of relative pose measurements along any cycle in the graph must return the identity element.*

Definition 2 (Pairwise consistency)**.** *The measurements are pairwise consistent if* $\tilde{\mathbf{R}}_{ij} = \tilde{\mathbf{R}}_{ji}^T$ *and* $\tilde{\mathbf{t}}_{ij} = \tilde{\mathbf{R}}_{ij}\tilde{\mathbf{t}}_{ji}$*,* $\forall\ (i,j) \in \mathcal{E}$*.*

Definition 3 (Minimal consistency)**.** *The measurements are minimally consistent if* $\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i}\log(\tilde{\mathbf{R}}_{ij})^\vee = \mathbf{0}_3$ *and* $\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i}\tilde{\mathbf{t}}_{ij} = \mathbf{0}_3$*.*

Global consistency is the most restrictive form of consistency. It often appears in formation control literature where the desired formations are assumed to be provided to the system as inputs and represent a valid, desired formation [27]. Global consistency is almost always violated with real-world noisy relative measurements. Pairwise consistency is less restrictive than global consistency because it only requires consistency among each undirected edge as opposed to all cycles in the graph. This condition is easy to satisfy in practice with a distributed system because the raw measurements can be shared and averaged along each edge in a single step (see Section VI for more details). Minimal consistency is physically the least restrictive definition as it encompasses both global and pairwise consistency. However, achieving minimal consistency within a distributed network is more complicated than pairwise consistency.

In Section V-A, we first prove that *pairwise consistency* is a sufficient condition for convergence of the rotation-only estimation system (3b) using LaSalle's Invariance Principle. We then show in Section V-B that *minimal consistency* is a sufficient condition for the equilibria of the coupled translation system (3a) to be input-to-state stable. Note that this system is a function of the time-varying rotation estimates. Finally, we show that the coupled system, with rotations and translations simultaneously evolving, converges to an equilibrium. Note that equilibria of (3) are local minima of the pose graph objective function in Problem 1.

A. Equilibria of the $SO(3)$ Rotation Matrix Estimate

We now show that pairwise consistent relative rotation measurements, i.e., $\tilde{\mathbf{R}}_{ij} = \tilde{\mathbf{R}}_{ji}^T,\ \forall (i,j) \in \mathcal{E}$, implies the convergence of the rotation estimate (3b), i.e., $\dot{\mathbf{R}}_i = \mathbf{0}_{3\times3},\ \forall i \in \mathcal{V}$. For the analysis, let $\mathbf{R}_{ij} = \mathbf{R}_i^T\mathbf{R}_j$ represent the rotation product and $\boldsymbol{\omega}_{ij}^\wedge$ the *relative control components*,

$$\begin{aligned}\boldsymbol{\omega}_{ij}^\wedge &= \log(\mathbf{R}_i^T\mathbf{R}_j\tilde{\mathbf{R}}_{ij}^T) = \log(\mathbf{R}_{ij}\tilde{\mathbf{R}}_{ij}^T)\\ \boldsymbol{\omega}_{ji}^\wedge &= \log(\mathbf{R}_j^T\mathbf{R}_i\tilde{\mathbf{R}}_{ji}^T) = \log(\mathbf{R}_{ij}^T\tilde{\mathbf{R}}_{ij})\,.\end{aligned} \tag{10}$$

so that $\boldsymbol{\omega}_i^\wedge = \sum_{j\in\mathcal{N}_i}\boldsymbol{\omega}_{ij}^\wedge$; noting that $\boldsymbol{\omega}_{ij}^\wedge \neq -\boldsymbol{\omega}_{ji}^\wedge$.

We introduce the following Lemmas to assist in the rotation estimate convergence result in Theorem 2 that use properties of the matrix logarithm and the relative control components from (10). For clarity, the proofs of these helper Lemmas are shown in Appendix B.

Lemma 2. *For any* $\mathbf{Q} \in \{\mathbf{R}_{ij}, \tilde{\mathbf{R}}_{ij}\}$*, if the rotations measurements are pairwise consistent, then,*

$$\mathbf{Q}^T\boldsymbol{\omega}_{ij}^\wedge\mathbf{Q} = -\boldsymbol{\omega}_{ji}^\wedge\,. \tag{11}$$

Lemma 3. *For a pair of relative control components $\boldsymbol{\omega}_{ij}$ and $\boldsymbol{\omega}_{ik}$ as well as rotation matrix $\tilde{\mathbf{R}}_{ij}$, the following are equivalent,*

$$\boldsymbol{\omega}_{ij}^T \left(\tilde{\mathbf{R}}_{ij}\boldsymbol{\omega}_{ik}^{\wedge}\tilde{\mathbf{R}}_{ij}^T\right)^{\vee} = -\boldsymbol{\omega}_{ji}^T\boldsymbol{\omega}_{ik} \,. \tag{12}$$

Lemma 4. *For a pair of relative control components $\boldsymbol{\omega}_{ij}$ and $\boldsymbol{\omega}_{ik}$ as well as rotation matrices $\mathbf{R}_{ij}$ and $\tilde{\mathbf{R}}_{ij}$, the following dot products are equivalent,*

$$\boldsymbol{\omega}_{ij}^T \left(\tilde{\mathbf{R}}_{ij}\mathbf{R}_{ij}^T\boldsymbol{\omega}_{ik}^{\wedge}\mathbf{R}_{ij}\tilde{\mathbf{R}}_{ij}^T\right)^{\vee} = \boldsymbol{\omega}_{ij}^T\boldsymbol{\omega}_{ik} \,. \tag{13}$$

Lemma 5. *For $\bar{\mathbf{R}} = \mathbf{R}_{ij}\tilde{\mathbf{R}}_{ij}^T$, the product $\bar{\mathbf{R}}^T\dot{\bar{\mathbf{R}}}$ is,*

$$\bar{\mathbf{R}}^T\dot{\bar{\mathbf{R}}} = -\tilde{\mathbf{R}}_{ij}\mathbf{R}_{ij}^T\boldsymbol{\omega}_i^{\wedge}\mathbf{R}_{ij}\tilde{\mathbf{R}}_{ij}^T + \tilde{\mathbf{R}}_{ij}\boldsymbol{\omega}_j^{\wedge}\tilde{\mathbf{R}}_{ij}^T \in so(3) \,. \tag{14}$$

Theorem 2 (Rotation estimation convergence)**.** *If, $\forall(i,j) \in \mathcal{E}$, the rotation measurements $\tilde{\mathbf{R}}_{ij}$ and $\tilde{\mathbf{R}}_{ji}$ are pairwise consistent and the initial rotation estimates $\mathbf{R}_i$ lie in $\mathcal{B}_{\frac{\pi}{2}-\varepsilon} = \{\mathbf{R}_1, \dots, \mathbf{R}_n \mid \|\log(\mathbf{R}_i^T\mathbf{R}_j\tilde{\mathbf{R}}_{ij}^T)\|_2 \leq (\frac{\pi}{2}-\varepsilon), \forall(i,j) \in \mathcal{E}\} \subset SO(3)^n$, where $\varepsilon \in \mathbb{R}_+$ is an arbitrarily small nonnegative number, then the estimates evolving according to* (3b) *will converge to the set of equilibria $\mathbb{E}_{\mathbf{R}} = \{\mathbf{R}_1, \dots, \mathbf{R}_n \mid \dot{\mathbf{R}}_i = \mathbf{R}_i\boldsymbol{\omega}_i^{\wedge} = \mathbf{R}_i\sum_{j\in\mathcal{N}_i}\boldsymbol{\omega}_{ij}^{\wedge} = \mathbf{0}_{3\times3}, \forall i \in \mathcal{V}\}$.*

Proof. First, the set $\mathcal{B}_{\frac{\pi}{2}-\varepsilon}$ is positively invariant under (3b) according on the analysis in [6] (Theorem 10) which leverages properties from [30]. Our controller differs from [6] due to the rotation measurement offset terms, however the analysis that compares the gradient and the covariant derivative on $SO(3)$ holds for our controller as well.

Define a Lyapunov-like function $V : SO(3)^n \to \mathbb{R}$ that is a function of the agent's rotation matrices:

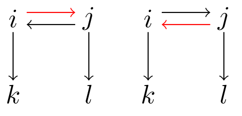
$$V(\mathbf{R}_1, \dots, \mathbf{R}_n) = \sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i}\frac{1}{2}\|\log(\mathbf{R}_i^T\mathbf{R}_j\tilde{\mathbf{R}}_{ij}^T)^{\vee}\|_2^2. \tag{15}$$

This is a sum of squared relative control components from (10) over all directed edges in the graph. The time derivative of V is written using the time derivative of the squared geodesic distance from Theorem 1 as well as the result from Lemma 5,

$$\begin{aligned}\dot{V} &= \sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i}\frac{1}{2}\frac{d}{dt}\|\log(\mathbf{R}_{ij}\tilde{\mathbf{R}}_{ij}^T)^{\vee}\|_2^2 \\ &= \sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i}\log(\mathbf{R}_{ij}\tilde{\mathbf{R}}_{ij}^T)^{\vee T}\left((\mathbf{R}_{ij}\tilde{\mathbf{R}}_{ij}^T)^T\frac{d}{dt}(\mathbf{R}_{ij}\tilde{\mathbf{R}}_{ij}^T)\right)^{\vee} \\ &= \sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i}\boldsymbol{\omega}_{ij}^T\left(-\tilde{\mathbf{R}}_{ij}\mathbf{R}_{ij}^T\boldsymbol{\omega}_i^{\wedge}\mathbf{R}_{ij}\tilde{\mathbf{R}}_{ij}^T + \tilde{\mathbf{R}}_{ij}\boldsymbol{\omega}_j^{\wedge}\tilde{\mathbf{R}}_{ij}^T\right)^{\vee} \,.\end{aligned} \tag{16}$$

Each control term $\boldsymbol{\omega}_i^{\wedge}$ is now replaced by the definition of the rotational control input in (2b),

$$\begin{aligned}\dot{V} = &\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i}\boldsymbol{\omega}_{ij}^T\Bigg(-\tilde{\mathbf{R}}_{ij}\mathbf{R}_{ij}^T\Bigg(\sum_{k\in\mathcal{N}_i}\boldsymbol{\omega}_{ik}^{\wedge}\Bigg)\mathbf{R}_{ij}\tilde{\mathbf{R}}_{ij}^T \\ &\qquad + \tilde{\mathbf{R}}_{ij}\Bigg(\sum_{l\in\mathcal{N}_j}\boldsymbol{\omega}_{jl}^{\wedge}\Bigg)\tilde{\mathbf{R}}_{ij}^T\Bigg)^{\vee} \\ = &\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i}\Bigg(\sum_{k\in\mathcal{N}_i}-\boldsymbol{\omega}_{ij}^T\left(\tilde{\mathbf{R}}_{ij}\mathbf{R}_{ij}^T\boldsymbol{\omega}_{ik}^{\wedge}\mathbf{R}_{ij}\tilde{\mathbf{R}}_{ij}^T\right)^{\vee} \\ &\qquad + \sum_{l\in\mathcal{N}_j}\boldsymbol{\omega}_{ij}^T\left(\tilde{\mathbf{R}}_{ij}\boldsymbol{\omega}_{jl}^{\wedge}\tilde{\mathbf{R}}_{ij}^T\right)^{\vee}\Bigg) \,.\end{aligned} \tag{17}$$



(a) (b)

Fig. 2. Example of the directed edges represented in the sums of (18). Fig. 2(a) represents the directed edge (i, j) and the neighbors of i and j. For this edge, the neighbors of i show up in the product with $\boldsymbol{\omega}_{ij}$ (first component of sum in (18)). Fig. 2(b) represents the directed edge (j, i) and the same neighbors as before. Similarly, the neighbors of i show up in the product with $\boldsymbol{\omega}_{ji}$ (now the second component of sum in (18)). Therefore, when considering one directed edge (i, j), we can instead count the neighbors of i twice and leave the neighbors of j for the opposite edge.

And finally, we use the dot product equivalence results from Lemmas 3 and 4 to arrive at the following,

$$\dot{V} = \sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i}\left(-\sum_{k\in\mathcal{N}_i}\boldsymbol{\omega}_{ij}^T\boldsymbol{\omega}_{ik} - \sum_{l\in\mathcal{N}_j}\boldsymbol{\omega}_{ji}^T\boldsymbol{\omega}_{jl}\right) . \tag{18}$$

Notice that for each directed edge (i, j), there is a sum over neighbors of i including $\boldsymbol{\omega}_{ij}$ and a sum over all neighbors of node j containing $\boldsymbol{\omega}_{ji}$. Since we are considering a directed edge, these neighbors will be counted again when we consider the directed edge (j, i) in the opposite direction — see the illustration of each edge's sums in Fig. 2. Thus, we can include these double counted edges for each directed edge,

$$\begin{aligned}\dot{V} &= \sum_{i=1}^{n}\sum_{j\in\mathcal{N}_i}\left(-2\sum_{k\in\mathcal{N}_i}\boldsymbol{\omega}_{ij}^T\boldsymbol{\omega}_{ik}\right) \\ &= -2\sum_{i=1}^{n}\sum_{j\in\mathcal{N}_i}\sum_{k\in\mathcal{N}_i}\boldsymbol{\omega}_{ij}^T\boldsymbol{\omega}_{ik} \,.\end{aligned} \tag{19}$$

Finally, we rearrange the two sums over neighbors of i into the product of the same sums over neighbors of i,

$$\begin{aligned}\dot{V} &= -2\sum_{i=1}^{n}\left(\sum_{j\in\mathcal{N}_i}\boldsymbol{\omega}_{ij}^T\right)\left(\sum_{j\in\mathcal{N}_i}\boldsymbol{\omega}_{ij}\right) \\ &= -2\sum_{i=1}^{n}\boldsymbol{\omega}_i^T\boldsymbol{\omega}_i \,.\end{aligned} \tag{20}$$

Therefore the set of rotation estimates where $\dot{V} = 0$ corresponds to the estimates where the control inputs $\boldsymbol{\omega}_i$ are all zero, i.e., the rotation estimate equilibria set $\mathbb{E}_{\mathbf{R}}$. □

B. Equilibria of the Coupled $SE(3)$ Translation Estimate

The coupled translation estimation given by system (3a),

$$\dot{\mathbf{t}}_i = \sum_{j\in\mathcal{N}_i}\mathbf{t}_j - \mathbf{t}_i - \mathbf{R}_i\tilde{\mathbf{t}}_{ij} \,, \tag{21}$$

where the $\mathbf{R}_i$ terms are the time-varying rotation estimates that were analyzed in the previous section. We consider the stacked version of system (21) and define vector $\mathbf{t} = [\mathbf{t}_1^T, \dots, \mathbf{t}_n^T]^T \in \mathbb{R}^{3n}$ such that the stacked dynamics of translation estimate are,

$$\dot{\mathbf{t}} = -(\mathbf{L} \otimes \mathbf{I}_3)\mathbf{t} - \boldsymbol{\delta}(t) \,, \tag{22}$$

where $\mathbf{L} \in \mathbb{R}^{n\times n}$ is the graph Laplacian, $\otimes$ represents the Kronecker product required for 3D estimates, and $\boldsymbol{\delta} \in \mathbb{R}^{3n}$ is the vector of desired offsets where each component is $\boldsymbol{\delta}_i(t) = \sum_{j\in\mathcal{N}_i} \mathbf{R}_i\tilde{\mathbf{t}}_{ij} \in \mathbb{R}^3$. Note that the desired offsets are functions of the rotation estimates. The equlibria set of this joint system is defined as $\mathbb{E}_{\mathbf{t}} = \{\mathbf{t} \in \mathbb{R}^{3n} \mid \dot{\mathbf{t}} = \mathbf{0}_{3n}\}$ such that a $\mathbf{t} \in \mathbb{E}_{\mathbf{t}}$ is an equilibrium point of (3a).

The analysis of this coupled estimation system follows from the coupled 3D estimation analysis: Corollary 8 from [8].

Corollary 1 (Translation estimation input-to-state stability)**.** *If the distributed pose estimate system* (1) *is controlled by* (2)*, the rotation measurements are minimally consistent, and the coupled rotation-translation offsets obey* $\lim_{t\to\infty} \boldsymbol{\delta}(t) = \boldsymbol{\delta}_\infty$ *with* $\boldsymbol{\delta}_\infty$ *minimally consistent, then,*

$$\lim_{t\to\infty} \mathbf{t} = \mathbf{t}_{eq}\,, \tag{23}$$

where $\mathbf{t}_{eq} \in \mathbb{E}_{\mathbf{t}}$ *is an equilibrium point of* (3a).

Proof. The proof directly follows Corollary 8 of [8]. In brief, the rotation estimates converge (Theorem 2) to constant values, thus the translation offsets $\boldsymbol{\delta}(t)$ can be set as minimally consistent (e.g., via pairwise averaging). Thus, the forced component of the translation estimation error dynamics vanishes asymptotically (Proposition 7 of [8]) and, from Lemma 4.6 of [46], the system is input-to-state stable. □

The result in Corollary 1 tells us that the translation and rotation estimation can be run concurrently. Previous distributed estimation works often consider these in separate, sequential algorithms, however this is not necessary with our method. Additionally, Corollary 1 requires that the coupled rotation-translation offsets tend towards a minimally consistent vector. How exactly that consistency is achieved is a design element of the algorithm and may be accomplished in many ways. We propose a number of options here and relate them to other algorithms in literature.

Since rotation is independent of translation, an intuitive technique is to average the initial pairwise rotation measurements between connected neighbors. The rotation measurements may be initialized at the beginning of the algorithm for each edge pair $(i, j) \in \mathcal{E}$ by,

$$\tilde{\mathbf{R}}'_{ij} = \tilde{\mathbf{R}}_{ij} \exp\left(\frac{1}{2}\log(\tilde{\mathbf{R}}_{ij}^T\tilde{\mathbf{R}}_{ji}^T)\right)\,, \tag{24}$$

such that $\tilde{\mathbf{R}}'_{ij}\tilde{\mathbf{R}}'_{ji} = \mathbf{I}_3$. This operation is not expensive as it involves one communication round between agent i and its immediate neighbors. Further, if the original graph is directed, it can be transformed into an undirected graph by inverting measurements for the opposite directions which will inherently constitute a pairwise consistent undirected edge.

For translation, one must consider the time-varying rotation estimate. One approach is to average the coupled translation terms between each update loop using,

$$\tilde{\mathbf{t}}'_{ij} = \frac{1}{2}(\tilde{\mathbf{t}}_{ij} - \mathbf{R}_{ij}^T\tilde{\mathbf{t}}_{ji})\,. \tag{25}$$

This operation only requires a similar one-hop communication at the beginning, however agent i can perform this operation without additional communication at all steps in the future. This is the technique used in Section VI.

In the algorithm by Tron et al. [38], the estimates are updated using the gradient of an objective similar to the one in Problem 1. Notably, these gradient-based updates naturally enforce pairwise consistency of the noisy measurements and thus satisfy our requirement for stability. Inspired by their method, this real-time consistency can be incorporated into a modified control law for the estimation system (1):

$$\boldsymbol{\nu}_i = \sum_{j\in\mathcal{N}_i} (\mathbf{t}_j - \mathbf{t}_i) + \frac{1}{2}\left(\mathbf{R}_j\tilde{\mathbf{t}}_{ji} - \mathbf{R}_i\tilde{\mathbf{t}}_{ij}\right)\,. \tag{26}$$

This type of pairwise averaging also appears in [7] (Algorithm 1) despite the different form of rotation. We also relate our findings to formation control literature such as [27], where they require global consistency on the desired poses for the proof of convergence of the coupled translation and rotation formation. The global consistency requirement is significantly more restrictive than our pairwise and minimal consistency. However, our method would also converge given globally consistent measurements if they were somehow provided.

VI. Empirical Results and Performance Comparisons

We present results from deploying GeoD[2] on a series of simulated and experimental pose graph datasets in Section VI-A. We also show results from a distributed multi-robot SLAM and 3D reconstruction scenario in Section VI-B, in which seven quadrotor robots share their camera images with their neighbors to estimate their relative poses along their trajectories. We run GeoD using the relative poses collected from these robots to estimate the group's poses while simultaneously building a 3D reconstruction of an unknown scene.

For all datasets, we compare GeoD's performance with the state-of-the-art centralized $SE(3)$ pose synchronization algorithm, *SE-Sync* [3], and a Distributed Gauss-Seidel algorithm, *DGS* [4]. Both methods solve a convex relaxation of a pose graph optimization problem that is similar to Problem 1, except using the chordal distance instead of the geodesic distance as we do. SE-Sync solves a semi-definite relaxation version of chordal problem while DGS solves a quadratic relaxation of the chordal problem in a distributed fashion. We implement the variant of DSG that uses Jacobi Over-Relaxation (JOR) [4]. SE-Sync provides a certificate of optimality if the final solution of the relaxed problem yields the same result as the original problem. In this way, we can compare our distributed method to the centralized, chordal global minimum. In this section, we denote this global minimum using the chordal distance with a *. We reiterate that this global minimum is for the chordal objective and not the geodesic objective shown in Problem 1. We report both objective values in this section for completeness.

In practice, we implement a discretized version of GeoD (3) with a time step of 0.05 seconds. The only requirement to use our method is the pairwise averaging condition. We enforce this by averaging the rotation measurement between

[2] github.com/ericcristofalo/GeoD

neighboring robots once using (24) before running GeoD, and then by enforcing (25) upon each time step. Convergence for the distributed methods is determined by the stopping condition for the difference between consecutive objectives values decreasing below 10^{-2}, an order of magnitude lower than the condition used in [4].

Although this paper provides convergence results using the distributed estimation scheme, the approach is not guaranteed to reach the global minimum due to the gradient-like nature of consensus methods. Therefore the initialization will impact the final solution. The initialization quality is not the focus of this paper, but an excellent discussion on initialization is provided in [39]. We initialize the pose estimates using simulated *GPS-like* measurements, assuming a robot has access to GPS and an inertial Measurement Unit (IMU), to provide a translation and rotation estimate. If such sensors are not available, we can initialize the poses by computing a minimum *spanning tree* through the graph that represents the chain of relative pose transformations from am arbitrary root node. There exists standard distributed algorithms to compute such a spanning tree that can be computed in a time on the order of the number of robots in the network [47]. Random or identity initialization will also work with GeoD.

A. Simulation Results

We demonstrate our distributed method with two sets of numerical simulations that illustrate

1) the performance compared to our previous consensus-based distributed algorithm [8] that uses the angle-axis rotation representation (Section VI-A1), and
2) the performance compared to state-of-the-art centralized [3] and distributed [4] methods that utilize chordal relaxations (Section VI-A2).

We consider seven network topologies: the *random*, *circular*, *grid*, *sphere*, *experiment*, *parking-garage*, and *cubicle* (Fig. 3). The *random* network's ground truth poses are incrementally generated by uniformly random sampling translations within a communication ball around a randomly selected node such that the graph is guaranteed to be connected. The ground truth rotation matrices are uniformly randomly sampled [48]. The *circular*, *grid*, and *sphere* networks have prescribed translations and topologies while the rotations are uniformly randomly sampled as well. We generate the noisy pairwise relative measurements for the first four networks by adding aggressive noise to ground truth relative poses. Translation noise is generated by a Gaussian distribution $\mathcal{N}(\mathbf{0}_3, \tau^2\mathbf{I}_3)$, where $\tau = 0.5$ meters, and the rotation noise is sampled in the vector space of $SO(3)$ and mapped back to the group using the exponential map: $\boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}_3, \kappa^2\mathbf{I}_3)$ and $\tilde{\mathbf{R}}_{ij} = \mathbf{R}_i^T\mathbf{R}_j \exp(\boldsymbol{\nu}^\wedge)$, where $\kappa = 0.524$ radians or 30 degrees. We evaluate the methods on the *experiment* data from Section VI-B as well for completeness. The *parking-garage* and *cubicle* networks are part of a SLAM benchmark dataset [3] that includes the noisy measurements and no ground truth positioning. These datasets illustrate the ability for GeoD to scale effective to very large pose graphs.

We initialize the estimates using the GPS-like initialization technique. The initial translation estimate is generated by corrupting the ground truth translation with additive zero-mean Guassian noise sampled from same distribution as the relative measurement noise defined above. The initial rotation estimate is generated in a similar manner to the relative rotation measurements using the same distribution: $\hat{\mathbf{R}}_i = \mathbf{R}_i \exp(\boldsymbol{\nu}^\wedge)$. These lead to exaggerated GPS-like initial conditions that may be much more accurate in real robotics scenarios.

1) Consensus-based Distributed Comparison: We first compare our distributed estimation technique to the only other state-of-the-art consensus-based approach that formally considers the noisy relative measurements in the analysis which uses the angle-axis representation [8]. We execute both methods on the sphere dataset with pairwise consistency in the measurements. We initialize both methods using the GPS technique as well. GeoD quickly converges to a geodesic value of 422.238 in 32 iterations (0.0089 seconds/agent) as calculated from the objective in Problem 1. The angle-axis approach achieves a geodesic value of 2237.248 in 281 iterations (0.0743 seconds/agent). The angle-axis method in [8] yields approximately $5\times$ greater pose estimation error and takes approximately $9\times$ longer than GeoD. We do not show plots of the resulting pose graphs due to space limitations, however the pose graph using [8] appears significantly farther form the ground truth poses than the pose graph from GeoD. This performance deficit is apparent in all other datasets as well, hence we do not consider the method from [8] in the more extensive performance comparisons below.

2) State-of-the-art Comparison: The data in Table I compares the chordal and geodesic values between all three methods, as well as the percent error of the chordal value with respect to the global minimum chordal value (denoted by a *). Table II compares the run times of each method, as well as the communication iterations to convergence for the distributed methods. SE-Sync is a centralized algorithm and does not involve communication rounds, hence the tables show no iterations for SE-Sync. All methods are initialized using the GPS-like initialization, except for the parking-garage and cubicle datasets which are initialized using the spanning-tree initialization. SE-Sync reaches the certified global minimum chordal value in all cases. We report the following average statistics over all datasets except for the cubicle since DSG runs out of memory and could not return a result. Compared to the global minimum, our distributed method converges to a value with an average error of 32.5%, while the DGS approach converges to a value with an average error of 110.2%. It is important to recall that our method does not operate on the chordal error directly, so it should not be expected to reach the global minimum chordal value. Despite this fact, it still outperforms DGS.

Our method is on average 717.27 times faster than SE-Sync. On the small datasets, DGS is about $2-4\times$ faster than GeoD. However, DGS requires $100\times$ more computation on the garage dataset time compared to our method and exceeds our computer's memory capacity in the cubicle case, failing to produce a result. We believe this is due to the pre-processing required to set up and store the DGS problem for such a large

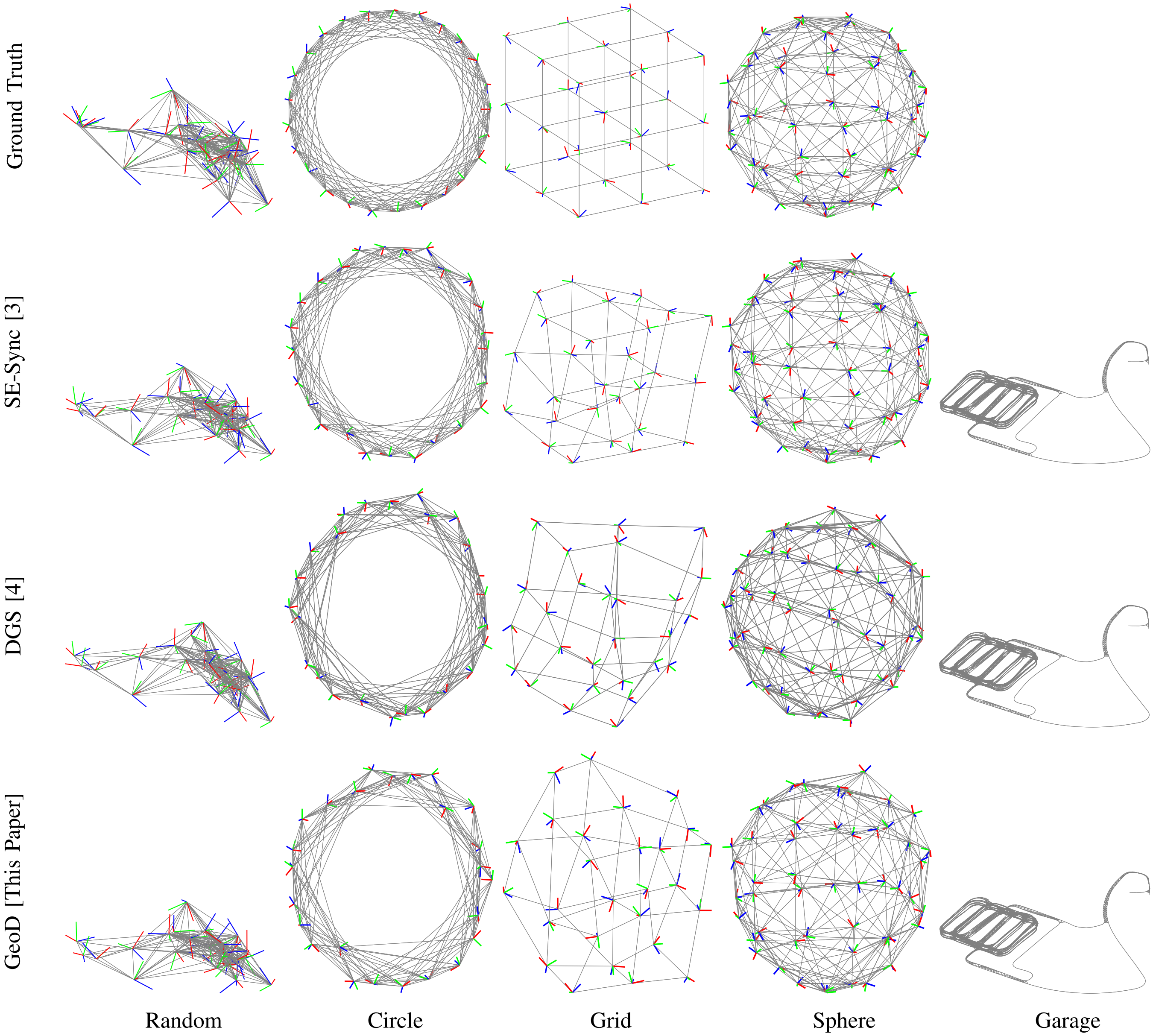


Fig. 3. Several pose graphs used in the numerical analysis are shown above; from left to right are the random, circular, grid, sphere, and parking-garage pose graphs. The ground truth network topologies are shown in the first row (the parking-garage does not include ground truth data). The global minimum solutions provided by SE-Sync are shown in the second row. The solutions obtained by DGS are shown in the third row. The solutions obtained by GeoD are shown in the fourth row. Our distributed solution reaches these results on average over 700 times faster than the time to compute the centralized solution and is approximately 100× faster than DGS on large networks like the parking-garage. Moreover, our method obtains a value that is on average 3.4× more accurate than DGS over all networks. The differences in value are hard to appreciate visually, but the grid is a good example where our approach yields a result closer to SE-Sync than DGS.

number of agents. In contrast, our method runs (3) for each pose, which is more efficient, and scales well with graphs of thousands of poses. Although our method does not reach the global minimum, it reaches qualitatively representative local minima as shown in Fig. 3 more quickly than with SE-Sync, and significantly more quickly than DGS for large networks.

B. Multi-robot Pose Estimation Experiment

We compare approaches in an experiment using real quadrotors with forward-facing cameras. The purpose of this experiment is to show that our algorithm can estimate the poses of multiple robots' trajectories using only raw images collected during flight. Our algorithm then estimates the poses of the quadrotors as they moved through the scene, informing each robot where it is and where its neighbors are throughout the flight. We fly seven autonomous quadrotors around an object of interest while simultaneously collecting images from their cameras (see Fig. 4). After collection, the quadrotors extact image features and match them with their neighboring robots using classic feature-based Structure from Motion (SfM) methods. We additionally use SfM to estimate the noisy relative poses between inter and intra quadrotor trajectories. This

TABLE I. Pose graph optimization objective value results. We compare our distributed consensus-based method, GeoD, to those from a state-of-the-art certifiable centralized method, SE-Sync, and a state-of-the-art distributed method, DGS. SE-Sync and DGS are convex relaxations that utilize the chordal distance instead of the geodesic distance, hence we report both values in this table. SE-Sync provides the certified global minimum chordal value which is indicated by the *. In all scenarios, our method reaches values that are closer to the global minimum compared to DGS, despite using different objective function. Note DGS runs out of memory on the large datasets with more than one thousand poses.

			SE-Sync [3]		DGS [4]			GeoD [This Paper]		
Scene	# Poses	# Measurements	Geodesic	Chordal*	Geodesic	Chordal	% Error from *	Geodesic	Chordal	% Error from *
random	25	390	284.205	407.277	349.717	505.180	24.0%	286.315	409.615	0.6%
circle	25	300	221.367	315.298	304.577	415.762	31.9%	239.830	332.776	5.5%
grid	27	108	58.479	85.774	88.417	123.244	43.7%	66.179	92.082	7.4%
sphere	50	544	387.896	558.386	496.650	688.637	23.3%	422.238	589.080	5.5%
experiment	217	508	78.056	99.635	296.637	435.768	337.4%	86.922	133.413	33.9%
garage	1,661	6,275	1.250	1.263	3.533	3.801	201.0%	3.055	3.056	142.1%
cubicle	5,750	16,869	644.799	717.126	—	—	—	1234.126	1324.659	84.7%

TABLE II. Pose graph optimization computation time results. We compare GeoD's timing performance to SE-Sync and DGS. In all cases, our method outperforms SE-Sync. For small networks less than 50 agents, DGS is only faster by a factor of $2-4\times$. For larger networks, GeoD easily outperforms DGS, indicating that it scales well with increasing number of poses. In fact, our DGS implementation runs out of memory on large networks with greater than one thousand poses, illustrating GeoD's ability to successfully scale and handle pose graphs with thousands of poses.

			SE-Sync [3]		DGS [4]		GeoD [This Paper]	
Scene	# Poses	# Measurements	Iterations	Time [s]	Iterations	Time/Agent [s]	Iterations	Time/Agent [s]
random	25	390	—	0.1750	9	0.0054	17	0.0189
circle	25	300	—	0.0975	15	0.0027	16	0.0044
grid	27	108	—	0.0593	10	0.0013	55	0.0068
sphere	50	544	—	0.1134	10	0.0020	32	0.0078
experiment	217	508	—	0.4836	141	0.0146	781	0.0609
garage	1,661	6,275	—	84.7133	265	2.6211	94	0.0204
cubicle	5,750	16,869	—	19.6280	—	—	5732	1.1467

information is the input to GeoD as the graph of noisy pose measurements. Here, we assume each quadrotor is responsible for computing the GeoD iterations (3) for all poses within its own trajectory.

We initialize the pose optimization by adding artificial noise to the poses obtained from an external motion capture system during their flight, to simulate the GPS-like initialization used throughout this work. In this experiment the parameters for the initial condition noise are $\tau = 0.1$ meters and $\kappa = 0.175$ radians or 10 degrees. GeoD reaches a final geodesic value of 86.922 (compared to the geodesic value associated to the chordal global minimum of 78.056) in 0.0609 seconds/agent (compared to 0.4836 seconds for SE-Sync). In contrast, the distributed DGS method reaches a final geodesic value of 296.637 in 0.0146 seconds/agent. The final pose graphs are visualized around the SfM 3D reconstruction in Fig. 4. The pose graphs are aligned to the reconstruction afterwards to illustrate the resulting scale of the solution. Interestingly, the converged solution of GeoD visually represents the ground truth trajectories better than SE-Sync. We suspect this indicates that the geodesic value captures important geometrical qualities.

VII. Conclusion

We have presented GeoD, a consensus-based distributed pose graph optimization algorithm with provable convergence guarantees. GeoD utilizes the geodesic distance on rotation matrices rather than the chordal distance and is guaranteed to converge when the noisy measurements along the graph's edges are constistent. We have leveraged tools from Lyapunov theory and multi-agent consensus to prove that pairwise consistent relative measurements are a sufficient condition for convergence of the entire $SE(3)$ estimator.

We have compared GeoD to state-of-the-art centralized and decentralized methods. Our gradient-based method reaches convergence in less time than the centralized method and reaches a more accurate solution that the distributed method. GeoD was additionally shown to scale better on pose graphs with large numbers of poses. We finally deployed our pose graph optimization method on real image data acquired from aerial drone trajectories. Our method allows the multi-UAV system to align their images and use this information to perform a vision-based 3D reconstruction.

We believe consensus-based estimators like GeoD are useful for their provable guarantees and their ease of implementation. In the future, we hope to find the conditions that ensure convergence to the global geodesic minimum and to deploy consensus-based methods as back-ends in an online distributed SLAM system.

Appendix

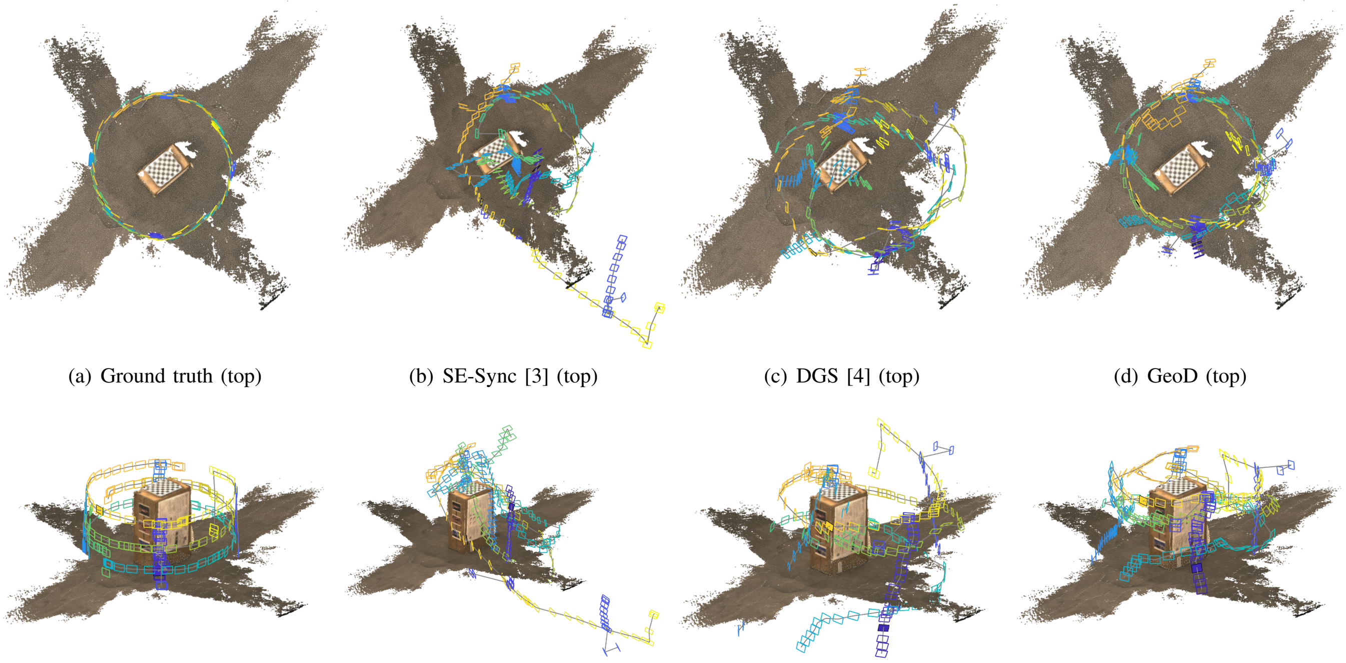
A. Proof of Theorem 1

Proof. From the definition of θ in (5), we write the time derivative of the squared geodesic distance as a function of θ and $\dot{\mathbf{R}}$,

$$\frac{d}{dt}\frac{1}{2}\|\log(\mathbf{R})^{\vee}\|_2^2 = \frac{d}{dt}\frac{1}{2}\theta^2 = \theta\dot{\theta} = \frac{-\theta\,\mathrm{tr}(\dot{\mathbf{R}})}{2\sin(\theta)}\,. \tag{27}$$

We premultiply the $\dot{\mathbf{R}}$ inside the trace by the identity matrix $\mathbf{R}\mathbf{R}^T$ such that $\mathbf{R}(\mathbf{R}^T\dot{\mathbf{R}})$ is a product of a rotation matrix, $\mathbf{R}$, and a skew-symmetric matrix, $(\mathbf{R}^T\dot{\mathbf{R}})$, such that,

$$\frac{d}{dt}\frac{1}{2}\|\log(\mathbf{R})^{\vee}\|_2^2 = \frac{-\theta\,\mathrm{tr}(\mathbf{R}(\mathbf{R}^T\dot{\mathbf{R}}))}{2\sin(\theta)}\,. \tag{28}$$



(a) Ground truth (top) (b) SE-Sync [3] (top) (c) DGS [4] (top) (d) GeoD (top)

(e) Ground truth (side) (f) SE-Sync [3] (side) (g) DGS [4] (side) (h) GeoD (side)

Fig. 4. We deploy GeoD on a team of vision-equipped quadrotors to compare their images with their neighbors, extract noisy relative pose measurements, and reach agreement on the quadrotors' poses in a distributed manner. Above are examples of the camera poses in the 3D reconstructed scene for the ground truth (a, e), SE-Sync (b, f), DGS (c, g), and GeoD (d, h). The figures in the top row and bottom row are viewed from an aerial view and side view, respectively. We aligned the resulting solutions to the 3D reconstruction by anchoring the first camera pose to the ground truth coordinate frame. Our method outperforms the DGS by converging to a solution with less value (86.922 compared to 296.637) and also qualitatively better represents the ground truth acquired via motion capture. Notice that the trajectories provided GeoD visually correspond to the ground truth, while the other methods return pose graphs that intersect the ground plane.

Note that $(\mathbf{R}^T\dot{\mathbf{R}}) \in so(3)$ due to the expression for the rate of change of a rotation matrix. Using Lemma 1, we find the resulting expression is a function of $\log(\mathbf{R})$ and thus arrive at the desired expression,

$$\begin{aligned}\frac{d}{dt}\frac{1}{2}\|\log(\mathbf{R})^\vee\|_2^2 &= \frac{\theta(\mathbf{R}^T\dot{\mathbf{R}})^{\vee T}(\mathbf{R}-\mathbf{R}^T)^\vee}{2\sin(\theta)}\\ &= (\mathbf{R}^T\dot{\mathbf{R}})^{\vee T}\left(\frac{\theta}{2\sin(\theta)}(\mathbf{R}-\mathbf{R}^T)\right)^\vee\\ &= (\mathbf{R}^T\dot{\mathbf{R}})^{\vee T}\log(\mathbf{R})^\vee\,.\end{aligned} \tag{29}$$

□

B. Proofs of Results from Section V-A

Proof of Lemma 2. From the properties of the matrix Logarithm (6), we know that the following is true for any $\mathbf{Q} \in \{\mathbf{R}_{ij}, \tilde{\mathbf{R}}_{ij}\}$ and $\boldsymbol{\omega}_{ij}$,

$$\begin{aligned}\mathbf{Q}^T\boldsymbol{\omega}_{ij}^\wedge\mathbf{Q} &= \mathbf{Q}^T\log(\mathbf{R}_{ij}\tilde{\mathbf{R}}_{ij}^T)\mathbf{Q}\\ &= \log(\mathbf{Q}^T\mathbf{R}_{ij}\tilde{\mathbf{R}}_{ij}^T\mathbf{Q})\,.\end{aligned} \tag{30}$$

The previous equation now takes the same form for each $\mathbf{Q} \in \{\mathbf{R}_{ij}, \tilde{\mathbf{R}}_{ij}\}$,

$$\begin{aligned}\log(\tilde{\mathbf{R}}_{ij}^T\mathbf{R}_{ij}\tilde{\mathbf{R}}_{ij}^T\tilde{\mathbf{R}}_{ij}) &= \log(\mathbf{R}_{ij}^T\mathbf{R}_{ij}\tilde{\mathbf{R}}_{ij}^T\mathbf{R}_{ij})\\ &= \log(\tilde{\mathbf{R}}_{ij}^T\mathbf{R}_{ij})\,.\end{aligned} \tag{31}$$

Finally, using the property of the matrix logarithm of a matrix inverse, we reach the desired result.

$$\begin{aligned}\log(\tilde{\mathbf{R}}_{ij}^T\mathbf{R}_{ij}) &= -\log(\mathbf{R}_{ij}^T\tilde{\mathbf{R}}_{ij})\\ &= -\log(\mathbf{R}_{ji}\tilde{\mathbf{R}}_{ji}^T)\\ &= -\boldsymbol{\omega}_{ji}^\wedge\,,\end{aligned} \tag{32}$$

and $\mathbf{Q}^T\boldsymbol{\omega}_{ij}^\wedge\mathbf{Q} = -\boldsymbol{\omega}_{ji}^\wedge$ as intended. □

Proof of Lemma 3. We reach the desired result using Lemma 2, (7), and the properties of the trace operator,

$$\begin{aligned}\boldsymbol{\omega}_{ij}^T\left(\tilde{\mathbf{R}}_{ij}\boldsymbol{\omega}_{ik}^\wedge\tilde{\mathbf{R}}_{ij}^T\right)^\vee &= \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\omega}_{ij}^{\wedge\,T}\tilde{\mathbf{R}}_{ij}\boldsymbol{\omega}_{ik}^\wedge\tilde{\mathbf{R}}_{ij}^T\right)\\ &= \frac{1}{2}\mathrm{tr}\left(\tilde{\mathbf{R}}_{ij}^T\boldsymbol{\omega}_{ij}^{\wedge\,T}\tilde{\mathbf{R}}_{ij}\boldsymbol{\omega}_{ik}^\wedge\right)\\ &= \frac{1}{2}\mathrm{tr}\left(-\boldsymbol{\omega}_{ji}^{\wedge\,T}\boldsymbol{\omega}_{ik}^\wedge\right)\\ &= -\boldsymbol{\omega}_{ji}^T\boldsymbol{\omega}_{ik}\,.\end{aligned} \tag{33}$$

□

Proof of Lemma 4. Similarly to Lemma 3, we reach the desired result using the previous properties of the matrix Logarithm,

$$\begin{aligned}
&\boldsymbol{\omega}_{ij}^{T}\left(\tilde{\mathbf{R}}_{ij}\mathbf{R}_{ij}^{T}\boldsymbol{\omega}_{ik}^{\wedge}\mathbf{R}_{ij}\tilde{\mathbf{R}}_{ij}^{T}\right)^{\vee} \\
&\qquad= \frac{1}{2}\mathrm{tr}\left({\boldsymbol{\omega}_{ij}^{\wedge}}^{T}\tilde{\mathbf{R}}_{ij}\mathbf{R}_{ij}^{T}\boldsymbol{\omega}_{ik}^{\wedge}\mathbf{R}_{ij}\tilde{\mathbf{R}}_{ij}^{T}\right) \\
&\qquad= \frac{1}{2}\mathrm{tr}\left(\mathbf{R}_{ij}\tilde{\mathbf{R}}_{ij}^{T}{\boldsymbol{\omega}_{ij}^{\wedge}}^{T}\tilde{\mathbf{R}}_{ij}\mathbf{R}_{ij}^{T}\boldsymbol{\omega}_{ik}^{\wedge}\right) \\
&\qquad= \frac{1}{2}\mathrm{tr}\left(-\mathbf{R}_{ij}{\boldsymbol{\omega}_{ji}^{\wedge}}^{T}\mathbf{R}_{ij}^{T}\boldsymbol{\omega}_{ik}^{\wedge}\right) \\
&\qquad= \frac{1}{2}\mathrm{tr}\left({\boldsymbol{\omega}_{ij}^{\wedge}}^{T}\boldsymbol{\omega}_{ik}^{\wedge}\right) \\
&\qquad= \boldsymbol{\omega}_{ij}^{T}\boldsymbol{\omega}_{ik}\,.
\end{aligned} \tag{34}$$

□

Proof of Lemma 5. The product $\bar{\mathbf{R}}^{T}\dot{\bar{\mathbf{R}}}$ is given by,

$$\begin{aligned}
\bar{\mathbf{R}}^{T}\dot{\bar{\mathbf{R}}} &= (\mathbf{R}_{i}^{T}\mathbf{R}_{j}\tilde{\mathbf{R}}_{ij}^{T})^{-1}\frac{d}{dt}(\mathbf{R}_{i}^{T}\mathbf{R}_{j}\tilde{\mathbf{R}}_{ij}^{T}) \\
&= \tilde{\mathbf{R}}_{ij}\mathbf{R}_{j}^{T}\mathbf{R}_{i}\left(\dot{\mathbf{R}}_{i}^{T}\mathbf{R}_{j}\tilde{\mathbf{R}}_{ij}^{T} + \mathbf{R}_{i}^{T}\dot{\mathbf{R}}_{j}\tilde{\mathbf{R}}_{ij}^{T}\right) \\
&= \tilde{\mathbf{R}}_{ij}\mathbf{R}_{ij}^{T}\dot{\mathbf{R}}_{i}^{T}\mathbf{R}_{j}\tilde{\mathbf{R}}_{ij}^{T} + \tilde{\mathbf{R}}_{ij}\mathbf{R}_{j}^{T}\dot{\mathbf{R}}_{j}\tilde{\mathbf{R}}_{ij}^{T}\,.
\end{aligned}$$

The time derivative of a rotation matrix $\mathbf{R}_i$ is defined by the rotation estimation system (3b) and substituting this into the previous equation yields the stated result,

$$\begin{aligned}
\dot{\boldsymbol{\omega}}_{ij}^{\wedge} &= \tilde{\mathbf{R}}_{ij}\mathbf{R}_{ij}^{T}\left(\mathbf{R}_{i}\boldsymbol{\omega}_{i}^{\wedge}\right)^{T}\mathbf{R}_{j}\tilde{\mathbf{R}}_{ij}^{T} + \tilde{\mathbf{R}}_{ij}\mathbf{R}_{j}^{T}\left(\mathbf{R}_{j}\boldsymbol{\omega}_{j}^{\wedge}\right)\tilde{\mathbf{R}}_{ij}^{T} \\
&= \tilde{\mathbf{R}}_{ij}\mathbf{R}_{ij}^{T}{\boldsymbol{\omega}_{i}^{\wedge}}^{T}\mathbf{R}_{ij}\tilde{\mathbf{R}}_{ij}^{T} + \tilde{\mathbf{R}}_{ij}\boldsymbol{\omega}_{j}^{\wedge}\tilde{\mathbf{R}}_{ij}^{T} \\
&= -\tilde{\mathbf{R}}_{ij}\mathbf{R}_{ij}^{T}\boldsymbol{\omega}_{i}^{\wedge}\mathbf{R}_{ij}\tilde{\mathbf{R}}_{ij}^{T} + \tilde{\mathbf{R}}_{ij}\boldsymbol{\omega}_{j}^{\wedge}\tilde{\mathbf{R}}_{ij}^{T}\,.
\end{aligned}$$

Additionally, this expression is indeed a skew-symmetric matrix since it is the sum of two elements in $so(3)$ from the property of logarithm products (6). □

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*, vol. 32, no. 6, pp. 1309–1332, 2016.

[2] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g 2 o: A general framework for graph optimization," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 3607–3613.

[3] 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 (IJRR)*, vol. 38, no. 2-3, pp. 95–125, 2016.

[4] S. Choudhary, L. Carlone, C. Nieto, J. Rogers, H. I. Christensen, and F. Dellaert, "Distributed mapping with privacy and communication constraints: Lightweight algorithms and object-based models," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 12, pp. 1286–1311, 2017.

[5] R. Tron and R. Vidal, "Distributed image-based 3-d localization of camera sensor networks," in *Proceedings of the 48th Conference on Decision and Control (CDC)*. IEEE, 2009, pp. 901–908.

[6] J. Thunberg, W. Song, E. Montijano, Y. Hong, and X. Hu, "Distributed attitude synchronization control of multi-agent systems with switching topologies," *Automatica*, vol. 50, no. 3, pp. 832–840, 2014.

[7] J. Thunberg, F. Bernard, and J. Gonçalves, "Distributed synchronization of euclidean transformations with guaranteed convergence," in *Proceedings of the 56th Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 3757–3762.

[8] E. Cristofalo, E. Montijano, and M. Schwager, "Consensus-based distributed 3d pose estimation with noisy relative measurements," in *Proceedings of the 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 2646–2653.

[9] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 834–849.

[10] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics (T-RO)*, vol. 31, no. 5, pp. 1147–1163, 2015.

[11] A. Martinelli, F. Pont, and R. Siegwart, "Multi-robot localization using relative observations," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*. IEEE, 2005, pp. 2797–2802.

[12] M. Toussaint, "Robot trajectory optimization using approximate inference," in *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, 2009, pp. 1049–1056.

[13] F. Dellaert, S. M. Seitz, C. E. Thorpe, and S. Thrun, "Structure from motion without correspondence," in *Proceedings Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2. IEEE, 2000, pp. 557–564.

[14] F. Dellaert, M. Kaess *et al.*, "Factor graphs for robot perception," *Foundations and Trends in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.

[15] 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.

[16] D. Rosen, L. Carlone, A. Bandeira, and J. Leonard, "A certifiably correct algorithm for synchronization over the special Euclidean group," in *International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2016.

[17] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *International Conference on Information Processing in Sensor Networks*. IEEE, 2005, pp. 63–70.

[18] V. Indelman, E. Nelson, N. Michael, and F. Dellaert, "Multi-robot pose graph localization and data association from unknown initial relative poses via expectation maximization," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 593–600.

[19] R. Olfati-Saber, "Distributed kalman filter with embedded consensus filters," in *Proceedings of the 44th Conference on Decision and Control (CDC)*. IEEE, 2005, pp. 8179–8184.

[20] T. Nägeli, C. Conte, A. Domahidi, M. Morari, and O. Hilliges, "Environment-independent formation flight for micro aerial vehicles," in *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 1141–1146.

[21] L. Riazuelo, J. Civera, and J. M. Montiel, "C2tam: A cloud framework for cooperative tracking and mapping," *Robotics and Autonomous Systems*, vol. 62, no. 4, pp. 401–413, 2014.

[22] Y. Tian, K. Khosoussi, D. M. Rosen, and J. P. How, "Distributed certifiably correct pose-graph optimization," *arXiv preprint arXiv:1911.03721*, 2020.

[23] Y. Tian, A. Koppel, A. S. Bedi, and J. P. How, "Asynchronous and parallel distributed pose graph optimization," *arXiv preprint arXiv:2003.03281*, 2020.

[24] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control (TAC)*, vol. 49, no. 9, pp. 1520–1533, 2004.

[25] E. Montijano, J. Thunberg, X. Hu, and C. Sagues, "Multi-robot distributed visual consensus using epipoles," in *Proceedings of the 50th Conference on Decision and Control (CDC) and European Control Conference (ECC)*. IEEE, 2011, pp. 2750–2755.

[26] Z. Wang and M. Schwager, "Multi-robot manipulation without communication," in *Distributed Autonomous Robotic Systems (DARS)*. Springer, 2016, pp. 135–149.

[27] E. Montijano, E. Cristofalo, D. Zhou, M. Schwager, and C. Sagüés, "Vision-based distributed formation control without an external positioning system," *IEEE Transactions on Robotics (T-RO)*, vol. 32, no. 2, pp. 339–351, 2016.

[28] J. Thunberg, J. Markdahl, and J. Goncalves, "Dynamic controllers for column synchronization of rotation matrices: a qr-factorization approach," *Automatica*, vol. 93, pp. 20–25, 2018.

[29] R. Tron, B. Afsari, and R. Vidal, "Intrinsic consensus on so (3) with almost-global convergence." in *Proceedings of the 51st Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 2052–2058.

[30] M. Moakher, "Means and averaging in the group of rotations," *SIAM journal on matrix analysis and applications*, vol. 24, no. 1, pp. 1–16, 2002.

[31] R. Tron, R. Vidal, and A. Terzis, "Distributed pose averaging in camera networks via consensus on se (3)," in *Second ACM/IEEE International Conference on Distributed Smart Cameras*, 2008, pp. 1–10.

[32] F. Garin and L. Schenato, "A survey on distributed estimation and control applications using linear consensus algorithms," in *Networked Control Systems*. Springer, 2010, pp. 75–107.

[33] B. D. Anderson, I. Shames, G. Mao, and B. Fidan, "Formal theory of noisy sensor network localization," *SIAM Journal on Discrete Mathematics*, vol. 24, no. 2, pp. 684–698, 2010.

[34] A. Savvides, W. L. Garber, R. L. Moses, and M. B. Srivastava, "An analysis of error inducing parameters in multihop sensor node localization," *Transactions on Mobile Computing*, vol. 4, no. 6, pp. 567–577, 2005.

[35] R. Aragues, L. Carlone, C. Sagues, and G. Calafiore, "Distributed centroid estimation from noisy relative measurements," *Systems & Control Letters*, vol. 61, no. 7, pp. 773–779, 2012.

[36] A. Carron, M. Todescato, R. Carli, and L. Schenato, "An asynchronous consensus-based algorithm for estimation from noisy relative measurements," *IEEE Transactions on Control of Network Systems (TCNS)*, vol. 1, no. 3, pp. 283–295, 2014.

[37] G. Piovan, I. Shames, B. Fidan, F. Bullo, and B. D. Anderson, "On frame and orientation localization for relative sensing networks," *Automatica*, vol. 49, no. 1, pp. 206–213, 2013.

[38] R. Tron and R. Vidal, "Distributed 3-d localization of camera sensor networks from 2-d image measurements," *IEEE Transactions on Automatic Control (TAC)*, vol. 59, no. 12, pp. 3325–3340, 2014.

[39] 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 *Proceedings of the International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4597–4604.

[40] F. R. Chung and F. C. Graham, *Spectral graph theory*. American Mathematical Society, 1997, no. 92.

[41] T. D. Barfoot, *State Estimation for Robotics*. Cambridge University Press, 2017.

[42] J. Thunberg, E. Montijano, and X. Hu, "Distributed attitude synchronization control," in *Proceedings of the 50th Conference on Decision and Control (CDC) and the European Control Conference (ECC)*. IEEE, 2011, pp. 1962–1967.

[43] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on se (3)," in *Proceedings of the 49th Conference on Decision and Control (CDC)*. IEEE, 2010, pp. 5420–5425.

[44] ——, "Control of complex maneuvers for a quadrotor uav using geometric methods on se (3)," *arXiv preprint arXiv:1003.2005*, 2010.

[45] L. Dieci, B. Morini, and A. Papini, "Computational techniques for real logarithms of matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 3, pp. 570–593, 1996.

[46] H. K. Khalil, *Nonlinear control*, 3rd ed. Pearson New York, 2015.

[47] B. Awerbuch, "Optimal distributed algorithms for minimum weight spanning tree, counting, leader election, and related problems," in *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, 1987, pp. 230–240.

[48] J. Arvo, "Fast random rotation matrices," in *Graphics Gems III (IBM Version)*. Elsevier, 1992, pp. 117–120.