2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Pose Graph Optimization over Planar Unit Dual Quaternions: Improved Accuracy with Provably Convergent Riemannian Optimization

William D. Warke*, J. Humberto Ramos*, Prashant Ganesh†, Kevin M. Brink‡, Matthew T. Hale§

***Abstract*— It is common in pose graph optimization (PGO) algorithms to assume that noise in the translations and rotations of relative pose measurements is uncorrelated. However, existing work shows that in practice these measurements can be highly correlated, which leads to degradation in the accuracy of PGO solutions that rely on this assumption. Therefore, in this paper we develop a novel algorithm derived from a realistic, correlated model of relative pose uncertainty, and we quantify the resulting improvement in the accuracy of the solutions we obtain relative to state-of-the-art PGO algorithms. Our approach utilizes Riemannian optimization on the planar unit dual quaternion (PUDQ) manifold, and we prove that it converges to first-order stationary points of a Lie-theoretic maximum likelihood objective. Then we show experimentally that, compared to state-of-the-art PGO algorithms, this algorithm produces estimation errors that are lower by 10% to 25% across several orders of magnitude of correlated noise levels and graph sizes.**

I. Introduction

Pose graph optimization (PGO) algorithms aim to optimally reconstruct the trajectory of a mobile agent using a set of uncertain relative measurements that were collected enroute. PGO is a backend component for numerous applications in robotics and computer vision, including simultaneous localization and mapping (SLAM) [1], [2], bundle adjustment [3], structure from motion [4], and photogrammetry [5]. Additionally, a variety of related practical problems of interest [6]–[9] can be transformed into PGO problems, making it a versatile tool for optimization in these fields.

Some well-established PGO frameworks, such as g2o [10], GTSAM [11], and iSAM [12], have addressed the PGO problem using a mix of Euclidean and heuristic optimization techniques. More recently, algorithms based on Riemannian optimization, including SE-Sync [13], Cartan-Sync [14], and CPL-Sync [15], have demonstrated that, under certain conditions, the PGO problem admits a semidefinite relaxation whose solution approximates the solution of the original, unrelaxed problem. One condition assumed by the above algorithms (and others) is that uncertainties in position and orientation are modeled by isotropic (uncorrelated) noise.

*Department of Mechanical and Aerospace Engineering, University of Florida, FL, USA. Emails: {william.warke, jramoszuniga}@ufl.edu.

†Principal Research Engineer, EpiSci Science, Incorporated. Email: prashantganesh@episci.com.

‡Air Force Research Lab, Munitions Directorate at Eglin Air Force Base. Email: kevin.brink@us.af.mil.

§Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332, USA. Email: matthale@gatech.edu. This work was supported by AFRL under grant FA8651-22-F-1052, AFOSR under grant FA9550-19-1-0169, and ONR under grant N00014-21-1-2495. Distribution Statement A. Approved for public release: distribution is unlimited.

However, the isotropic noise assumption runs contrary to existing results on uncertainty representations for rigid motion groups, which mathematically encode PGO problems. Specifically, it was shown in 2D [16] and in 3D [17] that the propagation of uncertainty through compound rigid motions is best captured by a Lie-theoretic model [18], namely, a Gaussian distribution on the Lie algebra of a rigid motion group. In fact, the authors of [19] demonstrated that such a Lie-theoretic model accurately predicted the distribution of a compound rigid motion trajectory where traditional models failed. These Lie-theoretic models are inherently anisotropic, which suggests that a PGO algorithm that incorporates anisotropy may attain improved accuracy.

Therefore, in this paper, we formulate 2D PGO problems on the manifold of *planar unit dual quaternions* (PUDQs), which we use to explicitly incorporate anisotropy in uncertainty models. To solve such problems, we use a Riemannian trust region (RTR) algorithm, for which we derive global convergence guarantees. The contributions of this paper are:

- We present what is, to the best of our knowledge, the first provably convergent PGO algorithm that permits arbitrarily large, anisotropic uncertainties.
- We prove that the proposed algorithm converges to first-order critical points given *any* initialization.
- We show that the resulting pose estimates are always at least 10% more accurate than the state of the art and more than 25% more accurate on high-dimensional problems.

The closest related works are [20]–[22]. In [20], a unit dual quaternion approach to PGO was developed using heuristic optimization techniques without formal guarantees, whereas we employ provably convergent Riemannian-geometric techniques. The authors of [21] used a Lie-theoretic objective, but did not include convergence guarantees or quantify the accuracy of their solutions. The work in [22] uses a similar problem formulation to us, though that work was entirely empirical. We differ both by proving convergence and showing improvement in accuracy over a class of Riemannian algorithms that were not studied in [22].

The rest of the paper is organized as follows. Section II provides preliminaries, and Section III provides a formal problem statement. Section IV outlines the proposed algorithm, and Section V proves that it converges. Section VI contains numerical results, and Section VII concludes.

II. Preliminaries

In this section, we include mathematical preliminaries that are necessary for our PUDQ PGO problem formulation. For detailed derivations, see our technical report [23].

A. Planar unit dual quaternion construction

Similarly to [22], we construct the PUDQ manifold as a representation of planar rigid motion. Given an orthonormal basis $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$, a planar rigid motion is characterized by a translation, denoted $\mathbf{t} = t_x\mathbf{i} + t_y\mathbf{j}$, and a rotation about the $\mathbf{k}$ axis by an angle $\theta \in (-\pi, \pi]$.

The PUDQ parameterization of this motion is given by $\mathbf{x} = \mathbf{x}_r + \epsilon\mathbf{x}_d$, where ϵ is a *dual number* satisfying $\epsilon^2 = 0, \epsilon \neq 0$. The *real* and *dual* parts of $\mathbf{x}$, denoted here as $\mathbf{x}_r \in \mathbb{S}^1$ and $\mathbf{x}_d \in \mathbb{R}^2$, respectively, are $\mathbf{x}_r \triangleq \cos(\theta/2) + \sin(\theta/2)\,\mathbf{k}$ and $\mathbf{x}_d \triangleq \frac{1}{2}\mathbf{t} \otimes \mathbf{x}_r$, with "$\otimes$" denoting the Hamilton product [24] under the convention $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$. Applying the Hamilton product to two PUDQs, denoted $\mathbf{x}$ and $\mathbf{y}$, yields the composition operator "$\boxplus$", which can be expressed as

$$\mathbf{x} \boxplus \mathbf{y} = \underbrace{\begin{bmatrix} x_0 & -x_1 & 0 & 0 \\ x_1 & x_0 & 0 & 0 \\ x_2 & x_3 & x_0 & -x_1 \\ x_3 & -x_2 & x_1 & x_0 \end{bmatrix}}_{Q_L(\mathbf{x})} \underbrace{\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} y_0 & -y_1 & 0 & 0 \\ y_1 & y_0 & 0 & 0 \\ y_2 & -y_3 & y_0 & y_1 \\ y_3 & y_2 & -y_1 & y_0 \end{bmatrix}}_{Q_R(\mathbf{y})} \underbrace{\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}}_{\mathbf{x}}, \tag{1}$$

where $Q_L(\cdot)$ and $Q_R(\cdot)$ denote the left and right composition maps, respectively. From (1), we have the identity element $\mathbb{1} = [1, 0, 0, 0]^\top$ and inverse formula $\mathbf{x}^{-1} = [x_0, -x_1, -x_2, -x_3]^\top$. The set of PUDQs forms the smooth manifold $\mathcal{M} \triangleq \mathbb{S}^1 \rtimes \mathbb{R}^2 \subset \mathbb{R}^4$, which we embed in $\mathbb{R}^4$ as

$$\mathcal{M} \triangleq \left\{ \mathbf{x} \in \mathbb{R}^4 \mid h(\mathbf{x}) = \mathbf{x}^\top \tilde{P} \mathbf{x} - 1 = 0 \right\} \subset \mathbb{R}^4, \tag{2}$$

where $\tilde{P} \triangleq \mathrm{diag}(\{1, 1, 0, 0\})$ and $h(\mathbf{x})$ is the *defining function* [25] for $\mathcal{M}$. PGO algorithms optimize over N poses, so we extend (2) to the N-fold product manifold $\mathcal{M}^N \triangleq (\mathbb{S}^1 \rtimes \mathbb{R}^2)^N$. Below, we will use the operator $\mathrm{vec}(\cdot)$, where

$$\mathrm{vec}((\mathbf{x}_i)_{i=1}^N) \triangleq [\mathbf{x}_1^\top, \mathbf{x}_2^\top, \ldots, \mathbf{x}_N^\top]^\top,$$

with each $\mathbf{x}_i \in \mathcal{M}$. Since $(\mathbb{S}^1 \rtimes \mathbb{R}^2)^N \subset \mathbb{R}^{4\times N} \cong \mathbb{R}^{4N}$, we embed $\mathcal{M}^N$ in $\mathbb{R}^{4N}$. For $\mathcal{X}, \mathcal{Y} \in \mathcal{M}^N$, this embedding lets us write $\mathcal{X} = \mathrm{vec}((\mathbf{x}_i)_{i=1}^N)$ and $\mathcal{Y} = \mathrm{vec}((\mathbf{y}_i)_{i=1}^N)$, where $\mathbf{x}_i, \mathbf{y}_i \in \mathcal{M}$ for each i. This embedding also gives the identity $\mathbb{1}^N = \mathrm{vec}((\mathbb{1})_{i=1}^N)$, the inverse formula $\mathcal{X}^{-1} = \mathrm{vec}((\mathbf{x}_i^{-1})_{i=1}^N)$, and the product $\mathcal{X} \boxplus \mathcal{Y} = \mathrm{vec}((\mathbf{x}_i \boxplus \mathbf{y}_i)_{i=1}^N)$.

B. Logarithm and exponential maps

The smooth manifold $\mathcal{M}$ with the identity, inverse, and composition operator form a Lie group [18] whose Lie algebra is the *tangent space* at the identity element, denoted $\mathcal{T}_\mathbb{1}\mathcal{M}$. Given $\mathbf{x} \in \mathcal{M}$, the logarithm map at the identity element is $\mathrm{Log}_\mathbb{1} : \mathcal{M} \to \mathcal{T}_\mathbb{1}\mathcal{M}$, given by

$$\mathrm{Log}_\mathbb{1}(\mathbf{x}) = \frac{1}{\gamma(\mathbf{x})}[x_1,\ x_2,\ x_3]^\top, \tag{3}$$

with $\gamma(\mathbf{x}) \triangleq \mathrm{sinc}(\phi(\mathbf{x})) = \sin(\phi(\mathbf{x}))/\phi(\mathbf{x})$, where $\phi(\mathbf{x}) \triangleq \mathrm{wrap}(\arctan(x_1, x_0))$, $\arctan : \mathbb{S}^1 \to (-\pi, \pi]$ is the four-quadrant arctangent and

$$\mathrm{wrap}(\alpha) \triangleq \begin{cases} \alpha + \pi & \text{if } \alpha \leq -\pi/2 \\ \alpha - \pi & \text{if } \alpha > \pi/2 \\ \alpha & \text{otherwise.} \end{cases} \tag{4}$$

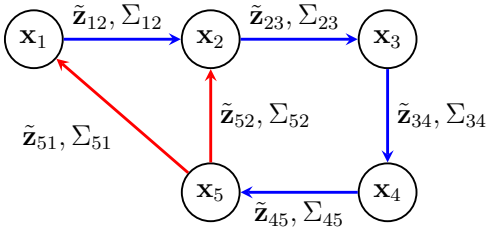

Fig. 1. A pose graph with $N = M = 5$, labeled with vertex poses $\mathbf{x}_i$, edge measurements $\tilde{\mathbf{z}}_{ij}$, and edge covariances Σ_{ij}. *Odometry* edges, shown in blue, connect neighboring vertices (i.e., $|j - i| = 1$). *Loop closure* edges, shown in red, connect any non-neighboring vertices (i.e., $|j - i| > 1$).

Here, $\phi : \mathcal{M} \to (-\pi/2, \pi/2]$ computes the half-angle of rotation about the $\mathbf{k}$-axis encoded by a point on $\mathcal{M}$. The half-angles $\phi + n\pi$ for all $n \in \mathbb{Z}$ encode the same rotation, so it is valid to wrap ϕ to $(-\pi/2, \pi/2]$ via (4). Moreover, (3) agrees with the standard unit dual quaternion logarithm constrained to the planar case, as we show in [23, Appendix B-E].

Given some $\mathbf{x}_t = [x_{t,1},\ x_{t,2},\ x_{t,3}]^\top \in \mathcal{T}_\mathbb{1}\mathcal{M}$, the exponential map at the identity, denoted $\mathrm{Exp}_\mathbb{1} : \mathcal{T}_\mathbb{1}\mathcal{M} \to \mathcal{M}$, is given by $\mathrm{Exp}_\mathbb{1}(\mathbf{x}_t) = \left[\cos(x_{t,1}), \gamma(\mathbf{x}_t)\,\mathbf{x}_t^\top\right]^\top$, where $\gamma(\mathbf{x}_t) \triangleq \mathrm{sinc}(x_{t,1})$ as above. For any $\mathbf{x}, \mathbf{y} \in \mathcal{M}$, we also have the point-wise logarithm map

$$\mathrm{Log}_\mathbf{x}(\mathbf{y}) = \mathbf{x} \boxplus [0, \mathrm{Log}_\mathbb{1}(\mathbf{x}^{-1} \boxplus \mathbf{y})^\top]^\top, \tag{5}$$

and, for $\mathbf{x} \in \mathcal{M}$, and some $\mathbf{y}_t \in \mathcal{T}_\mathbf{x}\mathcal{M}$, the point-wise exponential map

$$\mathrm{Exp}_\mathbf{x}(\mathbf{y}_t) = \mathbf{x} \boxplus \mathrm{Exp}_\mathbb{1}\left((\mathbf{x}^{-1} \boxplus \mathbf{y}_t)_{1:3}\right), \tag{6}$$

where $(\cdot)_{1:3}$ selects the last three entries of a vector. Furthermore, (5)-(6) extend to the product manifold $\mathcal{M}^N$, with $\mathrm{Log}_\mathcal{X}(\mathcal{Y}) = \mathrm{vec}\left((\mathrm{Log}_{\mathbf{x}_i}(\mathbf{y}_i))_{i=1}^N\right)$ for any $\mathcal{X}, \mathcal{Y} \in \mathcal{M}^N$, and, for any $\mathcal{Y}_t = \mathrm{vec}\left((\mathbf{y}_{t,i})_{i=1}^N\right) \in \mathcal{T}_\mathcal{X}\mathcal{M}^N$, the mapping

$$\mathrm{Exp}_\mathcal{X}(\mathcal{Y}_t) = \mathrm{vec}\left((\mathrm{Exp}_{\mathbf{x}_i}(\mathbf{y}_{t,i}))_{i=1}^N\right), \tag{7}$$

with $\mathrm{Log}_{\mathbf{x}_i}(\cdot)$ and $\mathrm{Exp}_{\mathbf{x}_i}(\cdot)$ given by (5) and (6).

C. Pose Graph Construction

We now address the construction of a pose graph, as exemplified in Figure 1. First, let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed graph with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$ of ordered pairs $(i, j) \in \mathcal{V} \times \mathcal{V}$. Letting $|\mathcal{V}| = N$, we define $\mathcal{X} = \mathrm{vec}((\mathbf{x}_i)_{i\in\mathcal{V}}) \in \mathcal{M}^N$ to be the vector of N poses to be estimated, with individual poses denoted $\mathbf{x}_i \in \mathcal{M}$. Then, letting $|\mathcal{E}| = M$, we define $\mathcal{Z} = \mathrm{vec}((\tilde{\mathbf{z}}_{ij})_{(i,j)\in\mathcal{E}}) \in \mathcal{M}^M$ to be the vector of M relative pose measurements, where $\tilde{\mathbf{z}}_{ij} \in \mathcal{M}$ encodes a measured transformation from $\mathbf{x}_i$ to $\mathbf{x}_j$, taken in the frame of $\mathbf{x}_i$. The noise covariance for $\tilde{\mathbf{z}}_{ij}$ is given by the matrix Σ_{ij}. The corresponding *pose graph* is then constructed by associating the vertex set $\mathcal{V}$ with $\mathcal{X}$, and the edge set $\mathcal{E}$ with $\mathcal{Z}$.

III. Problem Formulation

We now derive the problem to be solved. From the perspective of Bayesian inference, PGO algorithms aim to estimate the posterior distribution of poses that best fits a given dataset

of relative measurements made along a trajectory. Because a prior distribution is not always available, PGO is typically formulated as a *maximum likelihood estimation* (MLE) problem [1], and we use such a formulation here.

Motivated by [16], we utilize a Lie-theoretic measurement model for $\tilde{\mathbf{z}}_{ij}$ in which zero-mean Gaussian noise η_{ij} is mapped from $\mathcal{T}_{\mathbb{1}}\mathcal{M}$ to $\mathcal{M}$ via the exponential map, i.e.,

$$\tilde{\mathbf{z}}_{ij} = \mathbf{x}_i^{-1} \boxplus \mathbf{x}_j \boxplus \mathrm{Exp}_{\mathbb{1}}(\eta_{ij}), \tag{8}$$

with $\eta_{ij} \in \mathbb{R}^3$ and $\eta_{ij} \sim \mathcal{N}(0, \Sigma_{ij})$[1]. As noted in the Introduction, (8) gives a realistic model of compound, uncertain transformations. In [23, Appendix C], we show that (8) yields the MLE objective $\mathcal{F} : \mathcal{M} \to \mathbb{R}$, given by

$$\mathcal{F}(\mathcal{X}) = \frac{1}{2} \sum_{(i,j)\in\mathcal{E}} f_{ij}(\mathcal{X}), \tag{9}$$

where

$$f_{ij}(\mathcal{X}) \triangleq \|\mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j)\|^2_{\Omega_{ij}}. \tag{10}$$

Here, $\Omega_{ij} = \Sigma_{ij}^{-1}$ is the information matrix for edge (i, j), and $\mathbf{e}_{ij} : \mathcal{M} \times \mathcal{M} \to \mathcal{T}_{\mathbb{1}}\mathcal{M}$ is the *tangent* residual given by

$$\mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j) \triangleq \mathrm{Log}_{\mathbb{1}}(\mathbf{r}_{ij}(\mathbf{x}_i, \mathbf{x}_j)), \tag{11}$$

and $\mathbf{r}_{ij} : \mathcal{M} \times \mathcal{M} \to \mathcal{M}$ is the *manifold* residual, defined as $\mathbf{r}_{ij}(\mathbf{x}_i, \mathbf{x}_j) \triangleq \tilde{\mathbf{z}}_{ij}^{-1} \boxplus \mathbf{x}_i^{-1} \boxplus \mathbf{x}_j$.[2] In a geometric sense, $\mathbf{r}_{ij}$ encodes the geodesic along $\mathcal{M}$ from a measurement $\tilde{\mathbf{z}}_{ij}$ to the estimated relative transformation $\mathbf{x}_i^{-1} \boxplus \mathbf{x}_j$. The map $\mathbf{e}_{ij}$ then "unwraps" the geodesic to the Lie algebra. We note that $f_{ij}(\mathcal{X})$ in (10) is agnostic to the signs of $\mathbf{x}_i$ and $\mathbf{x}_j$, reflecting the fact that, for any $\mathbf{x} \in \mathcal{M}$, $\mathbf{x}$ and $-\mathbf{x}$ encode the same pose.

We now address *anchoring*, a problem that arises because the objective in (9) is invariant to a rigid transformation of all poses, i.e., $\mathcal{F}(\mathcal{X}) = \mathcal{F}(\mathcal{Y} \boxplus \mathcal{X}) = \mathcal{F}(\mathcal{X} \boxplus \mathcal{Y})$ for any $\mathcal{Y} \in \mathcal{M}^N$. To remedy this, one must "anchor" at least one vertex by fixing $\mathbf{x}_a$ for some $a \in \mathcal{V}$, so we assume that this has been done, as described in [1, Section IV-B]. Given this formulation, we now formally state the problem that we solve in the remainder of the paper.

Problem 1. Given a measurement set $\mathcal{Z} \in \mathcal{M}^M$, compute the *maximum likelihood estimate* $\mathcal{X}^\star \in \mathcal{M}^N$, where

$$\mathcal{X}^\star = \underset{\mathcal{X}\in\mathcal{M}^N}{\arg\min}\, \mathcal{F}(\mathcal{X}), \tag{12}$$

with $\mathcal{F}$ given by (9).

Problem 1 is a nonconvex, nonlinear least squares problem over a Riemannian manifold. In the following section, we employ Riemannian optimization techniques to solve (12).

IV. Algorithm Description

This section presents the method by which we solve Problem 1, starting with a brief description of the class of algorithms we employ. Trust-region methods [26] for optimization in $\mathbb{R}^n$ employ a local approximation of the objective function, called the *model*, about each iterate. The model is restricted to a neighborhood of the current iterate, called the *trust region*. At each iteration, a tentative update step is computed, and is accepted to compute the next iterate if the model sufficiently agrees with the objective. Riemannian trust region (RTR) methods [27, Chapter 7] generalize this idea to Riemannian manifolds, and we apply RTR to planar PGO on $\mathcal{M}^N$.

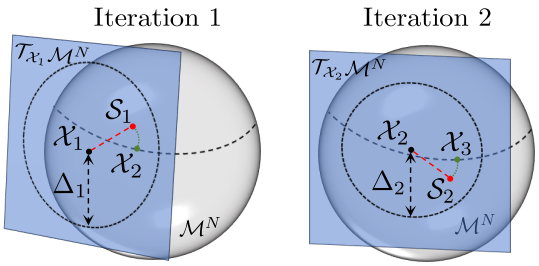

Fig. 2. An illustration of two iterations of the RTR algorithm. At each iteration, the algorithm computes a tangent step $\mathcal{S}_k \in \mathcal{T}_{\mathcal{X}_k}\mathcal{M}^N$, shown in red, within a trust region of radius Δ_k, which is indicated by the dotted circle shown in each tangent space. If the step is accepted (as defined in (17)), then the next iterate is computed as $\mathcal{X}_{k+1} = \mathrm{Exp}_{\mathcal{X}_k}(\mathcal{S}_k)$, which maps the step from the tangent space back to the manifold itself, as shown in green.

An illustration of the RTR algorithm is shown in Figure 2. At each iteration k, instead of approximating the objective $\mathcal{F}$, RTR computes an approximation of $\mathcal{F}$ in the tangent space at $\mathcal{X}_k$, called a *pullback*. The pullback is defined as $\hat{\mathcal{F}}_k \triangleq \mathcal{F} \circ \mathrm{Exp}_{\mathcal{X}_k}$.[3] The approximation takes the form of a second-order model $\hat{m}_k : \mathcal{T}_{\mathcal{X}_k} \to \mathbb{R}$, namely

$$\hat{m}_k(\mathcal{S}) \triangleq \mathcal{F}(\mathcal{X}_k) + \mathcal{S}^\top \mathrm{grad}\,\mathcal{F}(\mathcal{X}_k) + \frac{1}{2}\mathcal{S}^\top \mathcal{H}_k \mathcal{S}, \tag{13}$$

where $\mathcal{S} \in \mathcal{T}_{\mathcal{X}_k}\mathcal{M}^N$ is a tangent vector centered at $\mathcal{X}_k$, $\mathrm{grad}\,\mathcal{F} : \mathcal{M}^N \to \mathcal{T}_{\mathcal{X}_k}\mathcal{M}^N$ is the Riemannian gradient, and $\mathcal{H}_k : \mathcal{T}_{\mathcal{X}_k}\mathcal{M}^N \to \mathcal{T}_{\mathcal{X}_k}\mathcal{M}^N$ is a symmetric approximation of the Riemannian Hessian at $\mathcal{X}_k$. We include explicit forms for $\mathrm{grad}\,\mathcal{F}$ and our choice of $\mathcal{H}_k$ in Appendix B.

Our procedure corresponds to the RTR update given in [27]. The algorithm is initialized with $\mathcal{X}_0 \in \mathcal{M}^N$ and trust-region radius $\Delta_0 \in (0, \bar{\Delta}]$, where $\bar{\Delta} > 0$ is the user-specified maximum radius. At iteration k, the tentative step $\mathcal{S}_k$ is computed by solving the inner sub-problem

$$\mathcal{S}_k = \underset{\mathcal{S}\in\mathcal{T}_{\mathcal{X}_k}\mathcal{M}^N}{\arg\min}\, \hat{m}_k(\mathcal{S}) \text{ subject to } \|\mathcal{S}\|_2 \le \Delta_k, \tag{14}$$

where $\hat{m}_k$ is from (13). To solve (14), we employ the Steihaug-Toint truncated conjugate gradients (tCG) algorithm [30], [31], which offers unique benefits for trust-region sub-problems, including monotonic cost decrease and early termination (thereby approximating (14)) in the cases of negative curvature or trust region violation. Next, the *model agreement ratio*, denoted ρ_k, is computed as

$$\rho_k = \frac{\hat{\mathcal{F}}_k(\mathbf{0}) - \hat{\mathcal{F}}_k(\mathcal{S}_k)}{\hat{m}_k(\mathbf{0}) - \hat{m}_k(\mathcal{S}_k)}, \tag{15}$$

where $\mathbf{0} \in \mathbb{R}^{4N}$ is the zero vector. Intuitively, ρ_k quantifies

[1] We note that in practice, η_{ij} is unknown and only $\tilde{\mathbf{z}}_{ij}, \Sigma_{ij}$ are given.

[2] Henceforth, we simply write $\mathbf{e}_{ij} \triangleq \mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{r}_{ij} \triangleq \mathbf{r}_{ij}(\mathbf{x}_i, \mathbf{x}_j)$.

[3] The pullback can be defined using any retraction [28], [29]. We opt to use the exponential map since it is well-defined on $\mathcal{M}^N$ and simple to compute.

the level of agreement between the model and objective functions, which is used to update the trust-region radius Δ_k via

$$\Delta_{k+1} = \begin{cases} \frac{1}{4}\Delta_k & \text{if } \rho_k < \frac{1}{4} \\ \min\left\{2\Delta_k, \bar{\Delta}\right\} & \text{if } \rho_k > \frac{3}{4} \text{ and } \|\mathcal{S}_k\|_2 = \Delta_k \\ \Delta_k & \text{otherwise.} \end{cases} \tag{16}$$

The tentative step $\mathcal{S}_k$ is accepted to compute $\mathcal{X}_{k+1}$ only if ρ_k from (15) is greater than a user-defined model agreement threshold $\rho' \in (0, 1/4)$, i.e.,

$$\mathcal{X}_{k+1} = \begin{cases} \mathrm{Exp}_{\mathcal{X}_k}(\mathcal{S}_k) & \text{if } \rho_k > \rho' \\ \mathcal{X}_k & \text{otherwise.} \end{cases} \tag{17}$$

As summarized in Algorithm 1, the steps from (14)-(17) are repeated until the gradient norm crosses below a user-defined threshold ε_g, i.e., until $\|\mathrm{grad}\, \mathcal{F}(\mathcal{X}_k)\|_2 \leq \varepsilon_g$.

Algorithm 1: RTR for PUDQ PGO

Input: Edge measurement set $\mathcal{Z} \in \mathcal{M}^M$,
Maximum trust-region radius $\bar{\Delta} > 0$,
Model agreement threshold $\rho' \in (0, 1/4]$,
Gradient termination threshold $\varepsilon_g > 0$.
Initialize: $k \leftarrow 0$, $\mathcal{X}_0 \in \mathcal{M}^N$, $\Delta_0 \in \left(0, \bar{\Delta}\right]$
while $\|\mathrm{grad}\, \mathcal{F}(\mathcal{X}_k)\|_2 > \varepsilon_g$ **do**
  Compute $\mathcal{S}_k$ from (14) using tCG.
  Compute ρ_k using (15).
  Compute Δ_{k+1} using (16).
  Compute $\mathcal{X}_{k+1}$ using (17).
  $k \leftarrow k + 1$
end while
return $\mathcal{X}_k$

V. Convergence Analysis

In this section, we prove that Algorithm 1 is globally convergent. Specifically, given any initialization, it reaches a first-order critical point to within a user-specified tolerance in finite time. The authors of [29] proposed global rates of convergence for the RTR algorithm given a set of assumptions about the problem, so we treat these assumptions as sufficient conditions for convergence. For our proof, we will establish:

1) Lower-boundedness of $\mathcal{F}$ on $\mathcal{M}^N$.
2) Sufficient decrease in the model cost at each iteration.
3) A Lipschitz-type condition for gradients of pullbacks.
4) Radial linearity and boundedness of $\mathcal{H}_k$.

We will make each of these statements mathematically precise in the following analysis. Towards proving Condition 1, we first derive a lemma on continuity of $\mathcal{F}$.

Lemma 1. *The objective $\mathcal{F}$ is continuous on $\mathcal{M}^N$.*

Proof: By inspection of (3) and (9)-(11), and continuity of "$\boxplus$" from (1) as a linear map, it suffices to show that $\mathrm{Log}_{\mathbb{1}}$ is continuous on $\mathcal{M}$. While (3) and (4) contain discontinuities independently, we will show that their composition to form $\mathrm{Log}_{\mathbb{1}}$ does not. Let $\phi_1 \triangleq \arctan(r_1, r_0)$ (where $\mathbf{r}_{ij} = [r_0, r_1, r_2, r_3]^\top$ denotes the element-wise map), and let $\phi_2 \triangleq \mathrm{wrap}(\phi_1)$. Then, we have discontinuities in ϕ_1 at $(r_0, r_1) = (-1, 0)$, in $\mathrm{wrap}(\phi_1)$ at $\phi_1 = \pm\pi/2$, and in $(\gamma(\phi_2))^{-1}$ at $\phi_2 = \pm\pi$. We now observe that $\mathrm{wrap}(-\pi) = \mathrm{wrap}(\pi) = 0$, so $\lim_{(r_0,r_1)\to(-1,0)} \mathrm{wrap}(\phi_1) = 0$, thereby nullifying the discontinuities in ϕ_1. Next, $(\mathrm{sinc}(\phi_2))^{-1}$ is even and continuous on the domain $[-\pi/2, \pi/2]$, so $\lim_{\phi_2\to-\pi/2}(\gamma(\phi_2))^{-1} = \lim_{\phi_2\to\pi/2}(\gamma(\phi_2))^{-1} = \pi/2$, nullifying the discontinuities in ϕ_2. Finally, because $\lim_{\phi_2\to0}(\gamma(\phi_2))^{-1} = 1$ and, by (4), $\phi_2 \in (-\pi/2, -\pi/2]$, we conclude that $\mathrm{Log}_{\mathbb{1}}$ is continuous on $\mathcal{M}$, which implies that $\mathcal{F}$ is continuous on $\mathcal{M}^N$. ■

We now show compactness of sublevel sets of $\mathcal{F}$.

Theorem 1. *The μ-sublevel sets of $\mathcal{F}$, given by $\{\mathcal{X} \mid \mathcal{F}(\mathcal{X}) \leq \mu\}$, are compact.*

Proof: From (7), for every $\mathcal{X} \in \mathcal{M}^N$, $\mathrm{Exp}_{\mathcal{X}}$ is globally defined on $\mathcal{T}_{\mathcal{X}}\mathcal{M}^N$, which implies that $\mathcal{M}^N$ is geodesically complete. Therefore, the Hopf-Rinow Theorem [32] implies that closed and bounded subsets of $\mathcal{M}^N$ are compact, so it suffices to show that the sublevel sets are closed and bounded.

From (9)-(10), $\mathcal{F}(\mathcal{X}) \geq 0$ for all $\mathcal{X} \in \mathcal{M}^N$, which implies that the μ-sublevel sets of $\mathcal{F}$ are the preimages of the closed subsets $[0, \mu]$, i.e., μ-sublevel sets are of the form $\mathcal{F}^{-1}\left([0, \mu]\right)$. These sets are closed because $\mathcal{F}$ is continuous by Lemma 1.

Turning to boundedness of sublevel sets, (2) implies that $\mathcal{M}$ is unbounded, and therefore $\mathcal{M}^N$ is unbounded. Then, by [33, Theorem 1], the μ-sublevel sets are bounded if and only if $\mathcal{F}$ is coercive, i.e., for all $\mathcal{Y} \in \mathcal{M}^N$, every sequence $\{\mathcal{X}_l\}_{l\in\mathbb{N}} \subset \mathcal{M}^N$ such that $\lim_{l\to\infty} d_{\mathcal{M}^N}(\mathcal{X}_l, \mathcal{Y}) = \infty$ also satisfies $\lim_{l\to\infty} F(\mathcal{X}_l) = \infty$.[4] Therefore, it suffices to show that $\mathcal{F}$ is coercive, which we do next.

First, let $\mathcal{X}_l = \mathrm{vec}\left((\mathbf{x}_{l,i})_{i\in\mathcal{V}}\right)$ and $\mathcal{Y} = \mathrm{vec}\left((\mathbf{y}_i)_{i\in\mathcal{V}}\right)$, and observe from the definition of $d_{\mathcal{M}^N}(\mathcal{X}_l, \mathcal{Y})$ that

$$\lim_{d_{\mathcal{M}^N}(\mathcal{X}_l,\mathcal{Y})\to\infty} \max_{i\in\mathcal{V}} \|\mathrm{Log}_{\mathbb{1}}(\mathbf{x}_{l,i}^{-1} \boxplus \mathbf{y}_i)\|_2^2 = \infty.$$

We now rewrite $\|\mathrm{Log}_{\mathbb{1}}(\mathbf{x}_{l,i}^{-1} \boxplus \mathbf{y}_i)\|_2^2$ as

$$\|\mathrm{Log}_{\mathbb{1}}(\mathbf{x}_{l,i}^{-1} \boxplus \mathbf{y}_i)\|_2^2 = \gamma(\mathbf{x}_{l,i}^{-1} \boxplus \mathbf{y}_i)^{-2} \mathbf{x}_{l,i}^\top M_{LR}^-(\mathbf{y}_i) \mathbf{x}_{l,i},$$

where $M_{LR}^-(\mathbf{y}_i) \triangleq Q_{LR}^-(\mathbf{y}_i)^\top \mathrm{diag}\left(\{0, I_3\}\right) Q_{LR}^-(\mathbf{y}_i)$, with $Q_{LR}^-(\mathbf{y}_i)$ given in [23, Appendix A]. Since $\gamma(\mathbf{x}) \in [-\pi/2, \pi/2]$ for all $\mathbf{x} \in \mathcal{M}$, we have

$$\|\mathrm{Log}_{\mathbb{1}}(\mathbf{x}_{l,i}^{-1} \boxplus \mathbf{y}_i)\|_2^2 \leq (\pi^2/4) \lambda_{\max}(M_{LR}^-(\mathbf{y}_i)) \mathbf{x}_{l,i}^\top \mathbf{x}_{l,i}, \tag{18}$$

where $\lambda_{\max}(\cdot)$ denotes the maximum eigenvalue of a matrix. Since $\mathbf{y}_i$ is constant and $\lambda_{max}(M_{LR}^-(\mathbf{y}_i)) \geq 0$, (18) implies that $\lim_{\|\mathrm{Log}_{\mathbb{1}}(\mathbf{x}_{l,i}^{-1}\boxplus\mathbf{y}_i)\|_2^2\to\infty} (\mathbf{x}_{l,i}^\top \mathbf{x}_{l,i}) = \infty$. The first element of $\mathbf{x}_{l,i} \in \mathcal{M}$ is bounded by 1, so $\mathbf{x}_{l,i}^\top \mathbf{x}_{l,i} - 1 \leq \|\mathrm{Log}_{\mathbb{1}}(\mathbf{x}_{l,i})\|_2^2$. Therefore, $\lim_{(\mathbf{x}_{l,i}^\top \mathbf{x}_{l,i})\to\infty} \|\mathrm{Log}_{\mathbb{1}}(\mathbf{x}_{l,i})\|_2^2 = \infty$. Now, we note that for any $\mathbf{x}, \mathbf{y} \in \mathcal{M}$, we can write

$$\begin{aligned} \|\mathrm{Log}_{\mathbb{1}}(\mathbf{x} \boxplus \mathbf{y})\|_2^2 &= \gamma(\mathbf{x} \boxplus \mathbf{y})^{-1} \mathbf{y}^\top M_L(\mathbf{x}) \mathbf{y} \\ &= \gamma(\mathbf{x} \boxplus \mathbf{y})^{-1} \mathbf{x}^\top M_R(\mathbf{y}) \mathbf{x}, \end{aligned}$$

where $M_L(\mathbf{x}) \triangleq Q_L(\mathbf{x})^\top \mathrm{diag}\left(\{0, I_3\}\right) Q_L(\mathbf{x})$ and $M_R(\mathbf{y}) \triangleq Q_R(\mathbf{y})^\top \mathrm{diag}\left(\{0, I_3\}\right) Q_R(\mathbf{y})$. Because

[4] Here, $d_{\mathcal{M}^N}(\cdot, \cdot)$ is the geodesic distance on $\mathcal{M}^N$ defined in Appendix A.

$M_L(\cdot), M_R(\cdot) \succeq 0$, it holds that, for any $\mathbf{x}, \mathbf{y} \in \mathcal{M}$,

$$\lim_{\|\mathrm{Log}_{\mathbb{1}}(\mathbf{x} \boxplus \mathbf{y})\|_2^2 \to \infty} \max\left\{\|\mathrm{Log}_{\mathbb{1}}(\mathbf{x})\|_2^2, \|\mathrm{Log}_{\mathbb{1}}(\mathbf{y})\|_2^2\right\} = \infty. \tag{19}$$

We now observe that for any two vertices $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{M}$, with $i, j \in \mathcal{V}$ and $i > j$, it follows from connectedness of odometry edges in $\mathcal{E}$ that $\mathbf{x}_i = \mathbf{x}_j \boxplus \mathbf{c}_{ij}$, where

$$\mathbf{c}_{ij} \triangleq \tilde{\mathbf{z}}_{j(j+1)} \boxplus \mathbf{r}_{(j+1)(j+2)} \boxplus \cdots \boxplus \tilde{\mathbf{z}}_{(i-1)i} \boxplus \mathbf{r}_{(i-1)i}. \tag{20}$$

Equivalently, we have $\mathbf{x}_j = \mathbf{x}_i \boxplus \mathbf{c}_{ij}^{-1}$. Per Section III, we have fixed $\mathbf{x}_a$ for some $a \in \mathcal{V}$, and since $\mathrm{Log}_{\mathbb{1}}(\mathbf{x}^{-1}) = -\mathrm{Log}_{\mathbb{1}}(\mathbf{x})$, it holds that $\|\mathrm{Log}_{\mathbb{1}}(\mathbf{x}_{l,m})\|_2^2 = \|\mathrm{Log}_{\mathbb{1}}(\mathbf{c}_{am})\|_2^2$ for any $m \in \mathcal{V}$. Furthermore, because the $\tilde{\mathbf{z}}_{ij}$ terms in (20) are constant, applying (19) inductively yields, for any $m \in \mathcal{V}$,

$$\lim_{\|\mathrm{Log}_{\mathbb{1}}(\mathbf{x}_{l,m})\|_2^2 \to \infty} \max_{(i,j) \in \mathcal{E}} \|\mathrm{Log}_{\mathbb{1}}(\mathbf{r}_{ij})\|_2^2 = \infty.$$

From (10), $\lambda_{\min}(\Omega_{ij}) \|\mathrm{Log}_{\mathbb{1}}(\mathbf{r}_{ij})\|_2^2 \leq f_{ij}(\mathcal{X}_l)$, where $\lambda_{\min}(\cdot)$ is the minimum eigenvalue, and $\lambda_{\min}(\Omega_{ij}) > 0$ because $\Omega_{ij} = \Sigma_{ij}^{-1} \succ 0$. Then $\lim_{\|\mathrm{Log}_{\mathbb{1}}(\mathbf{r}_{ij})\|_2^2 \to \infty} f_{ij}(\mathcal{X}_l) = \infty$, and (9) gives $\lim_{f_{ij}(\mathcal{X}_l) \to \infty} \mathcal{F}(\mathcal{X}_l) = \infty$. Then $\mathcal{F}$ is coercive and the proof is complete. ■

Next, we show that the objective $\mathcal{F}$ satisfies Condition 1.

Lemma 2. *There exists $\mathcal{F}^\star \geq 0$ such that $\mathcal{F}(\mathcal{X}) \geq \mathcal{F}^\star$ for all $\mathcal{X} \in \mathcal{M}^N$.*

Proof: Lemma 1, Theorem 1, and the Weierstrass Theorem [34, Prop. A.8] imply the existence of a global minimizer $\mathcal{X}^\star \in \mathcal{M}^N$, which is the solution to Problem 1. Setting $\mathcal{F}^\star \triangleq \mathcal{F}(\mathcal{X}^\star)$ completes the proof. ■

We now show that Algorithm 1 satisfies Condition 2.

Lemma 3. *For all $\mathcal{X}_k$ computed by Algorithm 1 such that $\|\mathrm{grad}\,\mathcal{F}(\mathcal{X}_k)\|_2 > \varepsilon_g$, it holds that the step $\mathcal{S}_k$ satisfies*

$$\hat{m}_k(\mathbf{0}) - \hat{m}_k(\mathcal{S}_k) \geq \frac{1}{2} \min\{\Delta_k, 2\varepsilon_g\}\varepsilon_g. \tag{21}$$

Proof: By design, iterates of the tCG algorithm produce a strict, monotonic decrease of the model cost $\hat{m}_k$ [29]. For all k, the first tCG iterate is the Cauchy step, which satisfies (21) by definition and thus completes the proof. ■

The forthcoming analysis in Lemma 4, Theorem 2, and Lemma 5 addresses Condition 3, namely, Lipschitz continuity of the Riemannian gradient, $\mathrm{grad}\,\mathcal{F}$. First, we use Theorem 2 to prove its Lipschitz continuity on compact subsets of $\mathcal{M}^N$.

Theorem 2. *The Riemannian gradient, $\mathrm{grad}\,\mathcal{F}$, is L_g-Lipschitz continuous on any compact subset $\mathcal{K} \subset \mathcal{M}^N$. That is, there exists $L_g > 0$ such that for all $\mathcal{X}, \mathcal{Y} \in \mathcal{K}$ we have*

$$\|\mathcal{P}_{\mathcal{X} \to \mathcal{Y}} \mathrm{grad}\,\mathcal{F}(\mathcal{X}) - \mathrm{grad}\,\mathcal{F}(\mathcal{Y})\|_2 \leq L_g d_{\mathcal{M}^N}(\mathcal{X}, \mathcal{Y}), \tag{22}$$

where $\mathcal{P}_{\mathcal{X} \to \mathcal{Y}} : \mathcal{T}_{\mathcal{X}}\mathcal{M}^N \to \mathcal{T}_{\mathcal{Y}}\mathcal{M}^N$ is the parallel transport operator defined in Appendix A.

Proof: A necessary and sufficient condition for (22) is that, for all $\mathcal{X} \in \mathcal{K}$, the Riemannian Hessian, $\mathrm{Hess}\,\mathcal{F}$, has operator norm bounded by L_g, i.e.,

$$\sup_{\mathcal{V} \in \mathcal{T}_{\mathcal{X}}\mathcal{M}, \|\mathcal{V}\|_2 = 1} \|\mathrm{Hess}\,\mathcal{F}(\mathcal{X})[\mathcal{V}]\|_2 \leq L_g. \tag{23}$$

In our technical report [23, Appendices F-G], we derive $\mathrm{Hess}\,\mathcal{F}$ and derive a constant L_g for which (23) holds on any compact subset $\mathcal{K} \subset \mathcal{M}^N$, completing the proof. ■

To apply Theorem 2 to Algorithm 1, we must first show that the computed iterates remain within the $\mathcal{F}(\mathcal{X}_0)$-sublevel set for all k, which is accomplished by Lemma 4.

Lemma 4. *The objective $\mathcal{F}$ is monotonically decreasing with respect to the iterates of Algorithm 1. In particular, it holds that $\mathcal{F}(\mathcal{X}_k) \leq \mathcal{F}(\mathcal{X}_0)$ for all k.*

Proof: By (21), we have $\hat{m}_k(\mathbf{0}) - \hat{m}_k(\mathcal{S}_k) > 0$ for all k. If any $\mathcal{S}_k$ would yield an increase in $\mathcal{F}$, then $\mathcal{F}(\mathcal{X}_k) - \mathcal{F}(\mathrm{Exp}_{\mathcal{X}_k}(\mathcal{S}_k)) < 0$, and (15) implies $\rho_k < 0$. By (17), such an $\mathcal{S}_k$ is rejected and, therefore the condition $\mathcal{F}(\mathcal{X}_{k+1}) = \mathcal{F}(\mathcal{X}_k)$ is enforced in such cases. Thus, since it cannot occur that $\mathcal{F}(\mathcal{X}_{k+1}) > \mathcal{F}(\mathcal{X}_k)$, we see that $\mathcal{F}(\mathcal{X}_{k+1}) \leq \mathcal{F}(\mathcal{X}_k)$ for all k. By induction, $\mathcal{F}(\mathcal{X}_k) \leq \mathcal{F}(\mathcal{X}_0)$ for all k, completing the proof. ■

Now, Lemma 5 extends Theorem 2 to any $\mathcal{X}_k$ computed by Algorithm 1, which shows that Condition 3 is satisfied.

Lemma 5. *For all $\mathcal{X}_k$ computed by Algorithm 1, there exists $L_g \geq 0$ such that*

$$\left|\mathcal{F}\left(\mathrm{Exp}_{\mathcal{X}_k}(\mathcal{S})\right) - \left(\mathcal{F}(\mathcal{X}_k) + \mathcal{S}^\top \mathrm{grad}\,\mathcal{F}(\mathcal{X}_k)\right)\right| \leq \frac{L_g}{2}\|\mathcal{S}\|_2^2 \tag{24}$$

for all $\mathcal{S} \in \mathcal{T}_{\mathcal{X}_k}\mathcal{M}^N$ such that $\|\mathcal{S}\|_2 \leq \bar{\Delta}$ and for all k.

Proof: Let $M_{\mathcal{X}_0} \triangleq \{\mathcal{X} \mid \mathcal{F}(\mathcal{X}) \leq \mathcal{F}(\mathcal{X}_0)\}$ and set

$$\mathcal{K} \triangleq M_{\mathcal{X}_0} \cup \{\mathrm{Exp}_{\mathcal{X}}(\mathcal{S}) \mid \mathcal{X} \in M_{\mathcal{X}_0}, \|\mathcal{S}\|_2 \leq \bar{\Delta}\}. \tag{25}$$

Then Theorem 1 implies that $M_{\mathcal{X}_0}$ is compact, and therefore so is $\mathcal{K}$. Lemma 4 implies $\mathcal{X}_k \in M_{\mathcal{X}_0} \subset \mathcal{K}$ for all k. By Theorem 2, there exists $L_g > 0$ to which (22) applies for all $\mathcal{X}_k \in \mathcal{K}$. From [35, Lemma 2.1], we find that (22) implies (24), completing the proof. ■

Lemmas 6 and 7 address Condition 4, which pertains to properties of $\mathcal{H}_k$, the Riemannian Hessian approximation used in (14) and spelled out in Appendix B.

Lemma 6. *The operator $\mathcal{H}_k$ in 32 is radially linear, i.e., for all $\mathcal{S} \in \mathcal{T}_{\mathcal{X}_k}\mathcal{M}^N$ and all $\alpha \geq 0$, we have $\mathcal{H}_k[\alpha\mathcal{S}] = \alpha\mathcal{H}_k[\mathcal{S}]$.*

Proof: Equation (32) is linear by inspection. ■

Lemma 7. *The operator $\mathcal{H}_k$ in (32) is bounded for all $\mathcal{X}_k$ computed by Algorithm 1, i.e., there exists $\beta < \infty$ such that*

$$\max_{\mathcal{S}}\left\{\|\mathcal{H}_k\mathcal{S}\|_2 \mid \mathcal{S} \in \mathcal{T}_{\mathcal{X}_k}\mathcal{M}^N, \|\mathcal{S}\|_2 = 1\right\} \leq \beta. \tag{26}$$

Proof: First, $\|\mathcal{S}\|_2 = 1$ implies $\|\mathcal{H}_k\mathcal{S}\|_2 \leq \|\mathcal{H}_k\|_2$. Substituting (32), applying the triangle inequality, and using

the fact that $\lambda_{max}\left(\mathcal{P}_{\mathcal{X}}\right)=1$ yields

$$\left\|\mathcal{H}_k\right\|_2 \leq \sum_{(i,j)\in\mathcal{E}} \left\|\mathcal{P}_{\mathcal{X}}\mathcal{R}_{ij}\mathcal{P}_{\mathcal{X}}\right\|_2 \leq \sum_{(i,j)\in\mathcal{E}} \left\|\mathcal{R}_{ij}\right\|_2. \quad (27)$$

Since, by definition of $\|\cdot\|_2$ and $\|\cdot\|_F$ we have $\|\mathcal{R}_{ij}\|_2 \leq \|\mathcal{R}_{ij}\|_F$, we reach

$$\left\|\mathcal{R}_{ij}\right\|_2 \leq 4\left\|\mathcal{A}_{ij}\right\|_F\left\|\mathcal{B}_{ij}\right\|_F\left\|\Omega_{ij}\right\|_F. \quad (28)$$

Now, we set $\mathcal{K}$ as in (25) and apply the bounds derived in our technical report in [23, Appendix J] for $\|\mathcal{A}_{ij}\|_F$ and $\|\mathcal{B}_{ij}\|_F$ on compact subsets of $\mathcal{M}^N$. Since every term on the right-hand side of (28) is bounded, we see that the right-hand side of (27) is bounded, completes the proof. ■

Our convergence analysis culminates in Theorem 3.

Theorem 3. *Let $\varepsilon_g \leq {}^{\Delta_0}/_{\lambda_g}$ be given, where Δ_0 is from Section IV, $\lambda_g \triangleq {}^1/_4\min\left\{{}^1/_\beta, {}^1/_{2(L_g+\beta)}\right\}$, L_g is from (24), and β is from (26). Then, for any initialization $\mathcal{X}_0 \in \mathcal{M}^N$, Algorithm 1 produces an iterate $\mathcal{X}_k$ that satisfies $\|\mathrm{grad}\,\mathcal{F}(\mathcal{X}_k)\|_2 \leq \varepsilon_g$ in no more than K iterations, where*

$$K \leq \frac{\mathcal{F}\left(\mathcal{X}_0\right)-\mathcal{F}\left(\mathcal{X}^\star\right)}{\rho'\lambda_g}\frac{3}{\varepsilon_g^2}+\frac{1}{2}\log_2\left(\frac{\Delta_0}{\lambda_g\varepsilon_g}\right), \quad (29)$$

where ρ' is from (17) and $\mathcal{X}^\star$ is from Lemma 2.

Proof: Lemmas 2, 3, and 5-7 show the satisfaction of Conditions 1-4 in [29, Theorem 12], which immediately implies that (29) holds for Algorithm 1. ■

Theorem 3 gives provable convergence of Algorithm 1 to approximate first-order critical points of $\mathcal{F}$ *under any initialization* $\mathcal{X}_0$, and we note that the tolerance ε_g can be made to take arbitrary values by adjusting Δ_0.

VI. Experimental Results

In this section, we validate the accuracy of Algorithm 1 relative to the Riemannian PGO solvers SE-Sync [13] and Cartan-Sync [14]. Both yield a global minimizer identical to that computed by the class of Riemannian algorithms that use semidefinite relaxations (e.g., [15], [36]), so we omit additional comparisons to those algorithms.

Because an objective comparison necessitated the use of exact ground truth, we opted to adapt three synthetic PGO datasets with diverse vertex and edge counts. The first of these is Grid1000, which we synthesized[5] with $N = 1000$ vertices and $M = 1250$ edges. The remaining datasets are publicly available, and serve as common benchmarks for PGO evaluations, namely, (i) M3500 [37], with $N = 3500$, $M = 5598$, and (ii) City10000 [12], with $N = 10000$, $M = 20687$. To generate PGO trial datasets, we apply calibrated noise to the ground truth dataset for each graph. Each of these datasets, including ground truth, is available at `https://github.com/corelabuf/planar_pgo_datasets`.

[5]To synthesize the Grid1000 dataset, a ground truth trajectory is computed along a randomized grid resembling the Manhattan datasets created for [37]. Loop closure edges were selected at random, specifically, with 3.0% probability of an edge at Euclidean inter-pose distances of up to 2 meters.

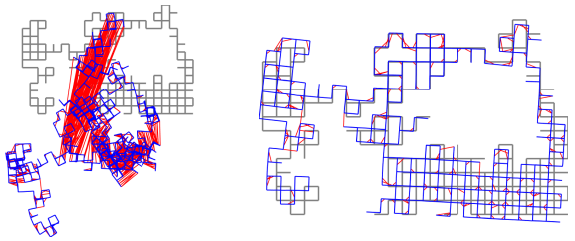

Fig. 3. (Left) The M3500 pose graph dataset, corrupted with Lie-theoretic noise. (Right) The estimated graph computed by Algorithm 1. Odometry edges are blue, loop closures are red, and ground truth is shown in gray.

A. PGO dataset generation

To generate a PGO dataset, the true edge measurements from each dataset are corrupted using the Lie-theoretic noise model from (8). The edge measurement noise covariance, Σ_{ij}, is computed as $\Sigma_{ij} \sim W_3\left(\sigma_w\Sigma_w, 10\right)$, where $W_d\left(V,n\right)$ is the Wishart distribution with dimension d, scale matrix V, and n degrees of freedom[6]. Here, σ_w is a variance tuning parameter, and Σ_w is given by $\Sigma_w \triangleq J_3 + \mathrm{diag}\left(\left[u_1,u_2,u_3\right]\right)$, where $J_3 \in \mathbb{R}^{3\times 3}$ is a matrix of ones and $u_i \sim \mathcal{U}_{(0,1]}$ are uniformly sampled on the interval $(0,1]$. This generates random, positive-definite, anisotropic covariance matrices with $\mathbb{E}[\Sigma_{ij}] = 10\sigma_w\Sigma_w$, which simulates relative pose covariances computed by a Lie-theoretic estimator. Using this approach, we generated 5 trial datasets per ground truth, for a total of 15. Figure 3 depicts an M3500 variant generated with $\sigma_w = 5.62\cdot 10^{-5}$ alongside the estimate computed by Algorithm 1.

B. Evaluation methodology

Solutions computed by each algorithm were evaluated using the root-mean-square relative pose error (RPE) metric. RPE measures total edge deformation with respect to the ground truth, and gives an objective performance metric for SLAM algorithms [39]. We denote $(\mathbf{x}_i^\diamond)_{i=1}^N$ to be the ground truth poses, and $(\hat{\mathbf{x}}_i)_{i=1}^N$ to be the poses estimated by a given algorithm. The Lie-theoretic RPE (RPE-L) is defined as

$$\text{RPE-L} \triangleq \sqrt{\frac{1}{M}\sum_{(i,j)\in\mathcal{E}}\left\|\mathrm{Log}_{\mathbb{1}}\left(\hat{\mathbf{z}}_{ij}^{-1}\boxplus\mathbf{z}_{ij}^\diamond\right)\right\|_2^2}, \quad (30)$$

where $\hat{\mathbf{z}}_{ij} \triangleq \hat{\mathbf{x}}_i^{-1}\boxplus\hat{\mathbf{x}}_j$ and $\mathbf{z}_{ij}^\diamond \triangleq \left(\mathbf{x}_i^\diamond\right)^{-1}\boxplus\mathbf{x}_j^\diamond$. Now, let $(\hat{\mathbf{t}}_i,\hat{\theta}_i)$ and $(\mathbf{t}_i^\diamond,\theta_i^\diamond)$ denote the translations and rotations corresponding to $\hat{\mathbf{x}}_i$ and $\mathbf{x}_i^\diamond$, respectively. The Euclidean RPE (RPE-E) is defined as

$$\text{RPE-E} \triangleq \sqrt{\frac{1}{M}\sum_{(i,j)\in\mathcal{E}}\left(\left\|\hat{\mathbf{t}}_{ij}-\mathbf{t}_{ij}^\diamond\right\|^2 + d(\hat{\theta}_{ij},\theta_{ij}^\diamond)^2\right)}, \quad (31)$$

where $\hat{\mathbf{t}}_{ij} \triangleq R^\top(\hat{\theta}_i)\left(\hat{\mathbf{t}}_j-\hat{\mathbf{t}}_i\right)$, $\mathbf{t}_{ij}^\diamond \triangleq R^\top\left(\theta_i^\diamond\right)\left(\mathbf{t}_j^\diamond-\mathbf{t}_i^\diamond\right)$, $d\left(\theta_1,\theta_2\right)$ is the minimal angle between θ_1 and θ_2, and

$$R\left(\theta\right) \triangleq \begin{bmatrix}\cos(\theta) & -\sin(\theta)\\ \sin(\theta) & \cos(\theta)\end{bmatrix}.$$

[6]The sample covariance matrix of a multivariate Gaussian random variable is Wishart-distributed [38], making it an apt choice for this application.

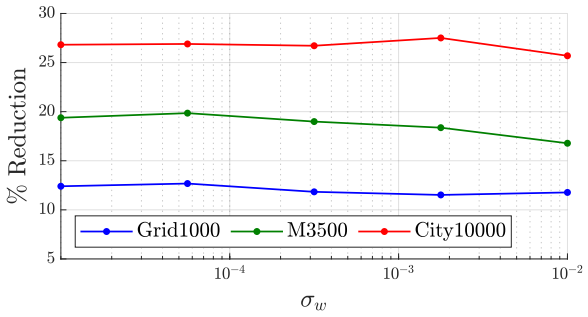

Fig. 4. Percent reduction in Lie-theoretic RPE for the solutions computed by Algorithm 1 relative to Cartan-Sync and SE-Sync. Reduction in Euclidean RPE was omitted due to it being indistinguishable from the Riemannian case. We see greater than 10% decrease for the Grid1000 dataset over the entire noise regime, and greater than 15% & 25% for the M3500 and City10000 datasets, respectively. In all cases, the improvement in accuracy attained by Algorithm 1 grows with the number of vertices and edges present in a graph.

For evaluation, the variance scaling parameter, σ_w, was varied from 10^{-5} to 10^{-2}, which equated to mean Euclidean covariances with standard deviations ranging from $7.26 \cdot 10^{-3}$ to $2.29 \cdot 10^{-1}$ meters for translations, and from $4.05 \cdot 10^{-1}$ to 12.81 degrees for rotations. We anchor $\mathbf{x}_1 \triangleq \mathbb{1}$ for all three algorithms. The initial iterate $\mathcal{X}_0$ is computed using the chordal relaxation [40] method; though not necessary for convergence of Algorithm 1, it is the default for both SE-Sync and Cartan-Sync, so we implement it to provide a fair comparison. Algorithm 1 was configured with parameters $\varepsilon_g = 10^{-2}$, $\Delta_0 = 100$, $\bar{\Delta} = 10^6$, $\rho' = 10^{-2}$, and the inner tCG algorithm was implemented with parameters $\kappa = 0.05$, $\theta = 0.25$, per the notation in [25, Section 6.5].

C. Evaluation results

Algorithm 1 converged to an approximate stationary point in all of the 15 pose graphs. The RPEs computed for each run according to (30) and (31) are included in Table I, alongside the percent reduction in RPE attained by Algorithm 1 for each run, which is plotted in Figure 4. SE-Sync and Cartan-Sync computed identical solutions for each dataset, and exhibited a notable estimation bias across the entire noise spectrum, owing to the assumption of isotropic noise and the resulting approximation error. As shown in Figure 4, Algorithm 1 demonstrated a consistent reduction in RPE. In fact, the gap in RPE increases with the number of vertices and edges in each graph, highlighting the scalability of our proposed solution.

VII. Conclusion

We presented a novel algorithm for planar PGO derived from a realistic, Lie-theoretic model for uncertainty in sensor measurements. The proposed algorithm was proven to converge in finite-time to approximate first-order stationary points under any initialization, while requiring no additional assumptions about the problem. Numerically, the proposed algorithm showed significantly improved accuracy over the state of the art, and future work will extend the algorithm to the 3D case and distributed/asynchronous implementations.

References

[1] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard. A tutorial on graph-based slam. *IEEE Intell. Transp. Syst. Mag.*, 2(4):31–43, 2010.

[2] N. Sünderhauf and P. Protzel. Towards a robust back-end for pose graph slam. In *Int. Conf. Robot. Autom. (ICRA)*, pp. 1254–1261. IEEE, 2012.

[3] D. Bender, M. Schikora, J. Sturm, and D. Cremers. A graph based bundle adjustment for ins-camera calibration. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, 40:39–44, 2013.

[4] M. Havlena, A. Torii, and T. Pajdla. Efficient structure from motion by graph optimization. In *European Conf. Comput. Vis. (ECCV)*, pp. 100–113. Springer, 2010.

[5] Q.-X. Huang, S. Flöry, N. Gelfand, M. Hofer, and H. Pottmann. Reassembling fractured objects by geometric matching. In *ACM SIGGRAPH 2006 Papers*, pp. 569–578, 2006.

[6] J. Thunberg, E. Montijano, and X. Hu. Distributed attitude synchronization control. In *Conf. Decis. Control (CDC) and European Control Conf.*, pp. 1962–1967. IEEE, 2011.

[7] R. Tron, B. Afsari, and R. Vidal. Intrinsic consensus on so (3) with almost-global convergence. In *Conf. Decis. Control (CDC)*, pp. 2052–2058. IEEE, 2012.

[8] J. Peters, D. Borra, B. Paden, and F. Bullo. Sensor network localization on the group of 3d displacements. *SIAM J. Control Optim.*, 2014.

[9] Y. Li, Y. Ushiku, and T. Harada. Pose graph optimization for unsupervised monocular visual odometry. In *Int. Conf. Robot. Autom. (ICRA)*, pp. 5439–5445. IEEE, 2019.

[10] G. Grisetti, R. Kümmerle, H. Strasdat, and K. Konolige. g2o: A general framework for (hyper) graph optimization. In *Int. Conf. Robot. Autom. (ICRA)*, pp. 9–13, 2011.

[11] F. Dellaert. Factor graphs and gtsam: A hands-on introduction. *Georgia Institute of Technology, Tech. Rep*, 2:4, 2012.

[12] M. Kaess, A. Ranganathan, and F. Dellaert. isam: Incremental smoothing and mapping. *IEEE Trans. Robot.*, 24(6):1365–1378, 2008.

[13] 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. *Int. J. Robot. Res.*, 38(2-3):95–125, 2019.

[14] J. Briales and J. Gonzalez-Jimenez. Cartan-sync: Fast and global se (d)-synchronization. *IEEE Robot. Autom. Lett.*, 2(4):2127–2134, 2017.

[15] T. Fan, H. Wang, M. Rubenstein, and T. Murphey. Efficient and guaranteed planar pose graph optimization using the complex number representation. In *Int. Conf. Intell. Robots Syst. (IROS)*, pp. 1904–1911. IEEE, 2019.

[16] A. W. Long, K. C. Wolfe, M. J. Mashner, G. S. Chirikjian, et al. The banana distribution is gaussian: A localization study with exponential coordinates. *Robotics: Science and Systems VIII*, 265:1, 2013.

[17] T. D. Barfoot and P. T. Furgale. Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Trans. Robot.*, 30(3):679–693, 2014.

[18] J. Sola, J. Deray, and D. Atchuthan. A micro lie theory for state estimation in robotics. *arXiv preprint arXiv:1812.01537*, 2018.

[19] D. O. Wheeler, D. P. Koch, J. S. Jackson, T. W. McLain, and R. W. Beard. Relative navigation: A keyframe-based approach for observable gps-degraded navigation. *IEEE Control Syst. Mag.*, 38(4):30–48, 2018.

[20] J. Cheng, J. Kim, Z. Jiang, and W. Che. Dual quaternion-based graphical slam. *Robotics and Autonomous Systems*, 77:15–24, 2016.

[21] F. Bai, T. Vidal-Calleja, and G. Grisetti. Sparse pose graph optimization in cycle space. *IEEE Trans. Robot.*, 37(5):1381–1400, 2021.

[22] K. Li, J. Cox, B. Noack, and U. D. Hanebeck. Improved pose graph optimization for planar motions using riemannian geometry on the manifold of dual quaternions. *IFAC-PapersOnLine*, 53(2):9541–9547, 2020.

[23] W. D. Warke, J. H. Ramos, P. Ganesh, K. M. Brink, and M. T. Hale. Technical report: Pose graph optimization over planar unit dual quaternions: Improved accuracy with provably convergent riemannian optimization. *arXiv preprint arXiv:2404.00010*, 2024.

[24] W. R. Hamilton. Xi. on quaternions; or on a new system of imaginaries in algebra. *Lond. Edinb. Dubl. Phil. Mag. J. Sci.*, 33(219):58–60, 1848.

[25] N. Boumal. *An introduction to optimization on smooth manifolds*. Cambridge University Press, 2023.

[26] J. Nocedal and S. J. Wright. Trust-region methods. *Numerical Optimization*, pp. 66–100, 2006.

[27] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2008.

[28] P.-A. Absil, C. G. Baker, and K. A. Gallivan. Trust-region methods on riemannian manifolds. *Found. Comput. Math.*, 7:303–330, 2007.

[29] N. Boumal, P.-A. Absil, and C. Cartis. Global rates of convergence for nonconvex optimization on manifolds. *IMA J. Numer. Anal.*, 39(1):1–33, 2019.

[30] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM J. Numer. Anal.*, 20(3):626–637, 1983.

[31] P. Toint. Towards an efficient sparsity exploiting newton method for minimization. In *Sparse matrices and their uses*, pp. 57–88. Academic press, 1981.

TABLE I

RESULTS OF THE 2D PGO DATASET EVALUATION. RPE AND PERCENT REDUCTION IN RPE ATTAINED BY ALGORITHM 1 ARE SHOWN ON THE RIGHT.

		SE-Sync [13]		Cartan-Sync [14]		Algorithm 1 [ours] (% Reduction)	
Dataset	σ_w	RPE-L	RPE-E	RPE-L	RPE-E	RPE-L	RPE-E
Grid1000	$1.0 \cdot 10^{-5}$	$6.2 \cdot 10^{-3}$	$1.2 \cdot 10^{-2}$	$6.2 \cdot 10^{-3}$	$1.2 \cdot 10^{-2}$	$5.4 \cdot 10^{-3}$ (–12.4%)	$1.1 \cdot 10^{-2}$ (–12.4%)
Grid1000	$5.6 \cdot 10^{-5}$	$1.5 \cdot 10^{-2}$	$2.9 \cdot 10^{-2}$	$1.5 \cdot 10^{-2}$	$2.9 \cdot 10^{-2}$	$1.3 \cdot 10^{-2}$ (–12.7%)	$2.6 \cdot 10^{-2}$ (–12.7%)
Grid1000	$3.2 \cdot 10^{-4}$	$3.5 \cdot 10^{-2}$	$7.1 \cdot 10^{-2}$	$3.5 \cdot 10^{-2}$	$7.1 \cdot 10^{-2}$	$3.1 \cdot 10^{-2}$ (–11.8%)	$6.2 \cdot 10^{-2}$ (–11.8%)
Grid1000	$1.8 \cdot 10^{-3}$	$7.9 \cdot 10^{-2}$	$1.6 \cdot 10^{-1}$	$7.9 \cdot 10^{-2}$	$1.6 \cdot 10^{-1}$	$7.0 \cdot 10^{-2}$ (–11.5%)	$1.4 \cdot 10^{-1}$ (–11.5%)
Grid1000	$1.0 \cdot 10^{-2}$	$1.9 \cdot 10^{-1}$	$3.9 \cdot 10^{-1}$	$1.9 \cdot 10^{-1}$	$3.9 \cdot 10^{-1}$	$1.7 \cdot 10^{-1}$ (–11.8%)	$3.4 \cdot 10^{-1}$ (–11.7%)
M3500	$1.0 \cdot 10^{-5}$	$5.4 \cdot 10^{-3}$	$1.1 \cdot 10^{-2}$	$5.4 \cdot 10^{-3}$	$1.1 \cdot 10^{-2}$	$4.4 \cdot 10^{-3}$ (–19.4%)	$8.7 \cdot 10^{-3}$ (–19.4%)
M3500	$5.6 \cdot 10^{-5}$	$1.3 \cdot 10^{-2}$	$2.6 \cdot 10^{-2}$	$1.3 \cdot 10^{-2}$	$2.6 \cdot 10^{-2}$	$1.0 \cdot 10^{-2}$ (–19.8%)	$2.1 \cdot 10^{-2}$ (–19.8%)
M3500	$3.2 \cdot 10^{-4}$	$3.1 \cdot 10^{-2}$	$6.2 \cdot 10^{-2}$	$3.1 \cdot 10^{-2}$	$6.2 \cdot 10^{-2}$	$2.5 \cdot 10^{-2}$ (–19.0%)	$5.0 \cdot 10^{-2}$ (–19.0%)
M3500	$1.8 \cdot 10^{-3}$	$7.4 \cdot 10^{-2}$	$1.5 \cdot 10^{-1}$	$7.4 \cdot 10^{-2}$	$1.5 \cdot 10^{-1}$	$6.0 \cdot 10^{-2}$ (–18.4%)	$1.2 \cdot 10^{-1}$ (–18.4%)
M3500	$1.0 \cdot 10^{-2}$	$1.7 \cdot 10^{-1}$	$3.4 \cdot 10^{-1}$	$1.7 \cdot 10^{-1}$	$3.4 \cdot 10^{-1}$	$1.4 \cdot 10^{-1}$ (–16.8%)	$2.9 \cdot 10^{-1}$ (–16.8%)
City10k	$1.0 \cdot 10^{-5}$	$4.9 \cdot 10^{-3}$	$9.7 \cdot 10^{-3}$	$4.9 \cdot 10^{-3}$	$9.7 \cdot 10^{-3}$	$3.6 \cdot 10^{-3}$ (–26.8%)	$7.1 \cdot 10^{-3}$ (–26.8%)
City10k	$5.6 \cdot 10^{-5}$	$1.2 \cdot 10^{-2}$	$2.3 \cdot 10^{-2}$	$1.2 \cdot 10^{-2}$	$2.3 \cdot 10^{-2}$	$8.5 \cdot 10^{-3}$ (–26.9%)	$1.7 \cdot 10^{-2}$ (–26.9%)
City10k	$3.2 \cdot 10^{-4}$	$2.8 \cdot 10^{-2}$	$5.5 \cdot 10^{-2}$	$2.8 \cdot 10^{-2}$	$5.5 \cdot 10^{-2}$	$2.0 \cdot 10^{-2}$ (–26.7%)	$4.0 \cdot 10^{-2}$ (–26.7%)
City10k	$1.8 \cdot 10^{-3}$	$6.6 \cdot 10^{-2}$	$1.3 \cdot 10^{-1}$	$6.6 \cdot 10^{-2}$	$1.3 \cdot 10^{-1}$	$4.8 \cdot 10^{-2}$ (–27.5%)	$9.5 \cdot 10^{-2}$ (–27.5%)
City10k	$1.0 \cdot 10^{-2}$	$1.6 \cdot 10^{-1}$	$3.1 \cdot 10^{-1}$	$1.6 \cdot 10^{-1}$	$3.1 \cdot 10^{-1}$	$1.2 \cdot 10^{-1}$ (–25.7%)	$2.3 \cdot 10^{-1}$ (–25.7%)

[32] C. Udriste. *Convex functions and optimization methods on Riemannian manifolds*, volume 297. Springer Science & Business Media, 2013.

[33] E. Celledoni, S. Eidnes, B. Owren, and T. Ringholm. Dissipative numerical schemes on riemannian manifolds with applications to gradient flows. *SIAM J. Sci. Comput.*, 40(6):A3789–A3806, 2018.

[34] D. P. Bertsekas. Nonlinear programming. *J. Oper. Res. Soc.*, 48(3):334–334, 1997.

[35] G. C. Bento, O. P. Ferreira, and J. G. Melo. Iteration-complexity of gradient, subgradient and proximal point methods on riemannian manifolds. *J. Optim. Theory Appl.*, 173(2):548–562, 2017.

[36] Y. Tian, K. Khosoussi, D. M. Rosen, and J. P. How. Distributed certifiably correct pose-graph optimization. *IEEE Trans. Robot.*, 37(6):2137–2156, 2021.

[37] E. Olson, J. Leonard, and S. Teller. Fast iterative alignment of pose graphs with poor initial estimates. In *Int. Conf. Robot. Autom. (ICRA)*, pp. 2262–2269. IEEE, 2006.

[38] C. Chatfield. *Introduction to multivariate analysis*. Routledge, 2018.

[39] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner. On measuring the accuracy of slam algorithms. *Autonomous Robots*, 27:387–407, 2009.

[40] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *Conf. Comput. Vis. Pattern Recognit.*, pp. 1–8. IEEE, 2007.

VIII. APPENDIX

A. Riemannian geometry of $\mathcal{M}$ and $\mathcal{M}^N$

Here, we briefly overview the Riemannian geometry involved with our formulation, which we cover in detail in our technical report [23, Appendix B]. Firstly, we employ an embedding of $\mathcal{M}$ in the ambient Euclidean space $\mathbb{R}^4$ with the inner product $\langle u, w\rangle = u^\top w$ and associated norm $\|u\|_2 = \sqrt{u^\top u}$ for all $u, w \in \mathbb{R}^4$. Thus, $\mathcal{M}$ inherits the metric $g_{\mathbf{x}}(u, w) = \langle u, w\rangle_{\mathbf{x}} \triangleq u^\top w$ and norm $\|u\|_{\mathbf{x}} \triangleq \|u\|_2$ for all $\mathbf{x} \in \mathcal{M}$ and $u, w \in \mathcal{T}_{\mathbf{x}}\mathcal{M}$. The product manifold $\mathcal{M}^N$ admits the metric $g_{\mathcal{X}}(\mathcal{U}, \mathcal{W}) = \sum_{i=1}^N g_{\mathbf{x}_i}(u_i, w_i) = \mathcal{U}^\top \mathcal{W}$, and norm $\|\mathcal{U}\|_{\mathcal{X}} \triangleq \|\mathcal{U}\|_2$ for all $\mathcal{X} \in \mathcal{M}^N$ and $\mathcal{U}, \mathcal{W} \in \mathcal{T}_{\mathcal{X}}\mathcal{M}^N$.

The geodesic distance metric measures the lengths of minimal curves between points. On $\mathcal{M}$, it is given by $d_{\mathcal{M}}(\mathbf{x}, \mathbf{y}) = \|\mathrm{Log}_{\mathbb{1}}(\mathbf{x}^{-1} \boxplus \mathbf{y})\|_2$ for $\mathbf{x}, \mathbf{y} \in \mathcal{M}$. On $\mathcal{M}^N$, it is given by $d_{\mathcal{M}^N}(\mathcal{X}, \mathcal{Y}) = (\sum_{i=1}^N \|\mathrm{Log}_{\mathbb{1}}(\mathbf{x}_i^{-1} \boxplus \mathbf{y}_i)\|_2^2)^{1/2}$ for $\mathcal{X} = \mathrm{vec}((\mathbf{x}_i)_{i=1}^N) \in \mathcal{M}^N$, and $\mathcal{Y} = \mathrm{vec}((\mathbf{y}_i)_{i=1}^N) \in \mathcal{M}^N$.

The parallel transport operator maps tangent vectors between tangent spaces. On $\mathcal{M}$, $\mathcal{P}_{\mathbf{x}\to\mathbf{y}} : \mathcal{T}_{\mathbf{x}}\mathcal{M} \to \mathcal{T}_{\mathbf{y}}\mathcal{M}$ denotes the parallel transport from $\mathcal{T}_{\mathbf{x}}\mathcal{M}$ to $\mathcal{T}_{\mathbf{y}}\mathcal{M}$ for any $\mathbf{x}, \mathbf{y} \in \mathcal{M}$. For $u_{\mathbf{y}} \in \mathcal{T}_{\mathbf{y}}\mathcal{M}$, it is given by $\mathcal{P}_{\mathbf{x}\to\mathbf{y}}(u_{\mathbf{y}}) = \mathbf{x} \boxplus (\mathbf{y}^{-1} \boxplus u_{\mathbf{y}})$. Extending this definition to $\mathcal{M}^N$ yields $\mathcal{P}_{\mathcal{X}\to\mathcal{Y}}(\mathcal{U}_{\mathcal{Y}}) = \mathrm{vec}((\mathbf{x}_i \boxplus (\mathbf{y}_i^{-1} \boxplus u_i))_{i=1}^N)$ for $\mathcal{U}_{\mathcal{Y}} = \mathrm{vec}((u_i)_{i=1}^N) \in \mathcal{T}_{\mathcal{Y}}\mathcal{M}^N$, $\mathcal{X} = \mathrm{vec}((\mathbf{x}_i)_{i=1}^N) \in \mathcal{M}^N$, and $\mathcal{Y} = \mathrm{vec}((\mathbf{y}_i)_{i=1}^N) \in \mathcal{M}^N$.

B. Riemannian gradient and Hessian approximation

The Riemannian gradient at $\mathcal{X} \in \mathcal{M}^N$, denoted $\mathrm{grad}\, \mathcal{F}(\mathcal{X})$, is the orthogonal projection of the Euclidean gradient, denoted $\partial \bar{\mathcal{F}}(\mathcal{X})$, onto $\mathcal{T}_{\mathcal{X}}\mathcal{M}^N$. The Euclidean gradient is given by $\partial \bar{\mathcal{F}}(\mathcal{X}) = \sum_{(i,j)\in\mathcal{E}} \mathbf{g}_{ij}(\mathbf{x}_i, \mathbf{x}_j)$, where $\mathbf{g}_{ij}(\mathbf{x}_i, \mathbf{x}_j) = [g_{ij,1}^\top, g_{ij,2}^\top, \ldots, g_{ij,N}^\top]^\top$ and

$$g_{ij,k} = \begin{cases} \mathcal{A}_{ij}^\top \Omega_{ij} \mathbf{e}_{ij} & i = k, \\ \mathcal{B}_{ij}^\top \Omega_{ij} \mathbf{e}_{ij} & j = k, \\ \mathbf{0}_{4\times 1} & \text{otherwise}, \end{cases}$$

with Jacobians $\mathcal{A}_{ij} \triangleq \partial \mathbf{e}_{ij}/\partial \mathbf{x}_i$ and $\mathcal{B}_{ij} \triangleq \partial \mathbf{e}_{ij}/\partial \mathbf{x}_j$. The orthogonal projector onto $\mathcal{T}_{\mathcal{X}}\mathcal{M}^N$, denoted $\mathcal{P}_{\mathcal{X}}$, is given by $\mathcal{P}_{\mathcal{X}} = \mathrm{diag}(\{\mathcal{P}_{\mathbf{x}_i}\}_{i=1}^N)$, with $\mathcal{P}_{\mathbf{x}_i} = I_4 - \tilde{P}\mathbf{x}_i\mathbf{x}_i^\top\tilde{P}$, where $I_4 \in \mathbb{R}^{4\times 4}$ is an identity matrix. This gives $\mathrm{grad}\, \mathcal{F}(\mathcal{X}) = \mathcal{P}_{\mathcal{X}} \partial \bar{\mathcal{F}}(\mathcal{X})$. See our technical report for derivations of $\mathcal{A}_{ij}, \mathcal{B}_{ij}$ [23, Appendix H] and $\mathcal{P}_{\mathbf{x}}, \mathcal{P}_{\mathcal{X}}$ [23, Appendix B].

In (13), $\mathcal{H}_k : \mathcal{T}_{\mathcal{X}_k}\mathcal{M}^N \to \mathcal{T}_{\mathcal{X}_k}\mathcal{M}^N$ is the Riemannian Gauss-Newton (RGN) approximation of the Riemannian Hessian at $\mathcal{X}_k$, which we derive in [23, Appendix F] to be

$$\mathcal{H}_k = \sum_{(i,j)\in\mathcal{E}} \mathcal{P}_{\mathcal{X}_k} \mathcal{R}_{ij} \mathcal{P}_{\mathcal{X}_k}, \tag{32}$$

where $\mathcal{R}_{ij} \in \mathbb{R}^{4N\times 4N}$ has only four nonzero blocks. Denoting block indices $\mathtt{i} \triangleq 4i{+}1 : 4i{+}4$ and $\mathtt{j} \triangleq 4j{+}1 : 4j{+}4$, they are given by $\mathcal{R}_{ij[\mathtt{i},\mathtt{i}]} = \mathcal{A}_{ij}^\top \Omega_{ij} \mathcal{A}_{ij}$, $\mathcal{R}_{ij[\mathtt{i},\mathtt{j}]} = \mathcal{A}_{ij}^\top \Omega_{ij} \mathcal{B}_{ij}$, $\mathcal{R}_{ij[\mathtt{j},\mathtt{i}]} = \mathcal{B}_{ij}^\top \Omega_{ij} \mathcal{A}_{ij}$, and $\mathcal{R}_{ij[\mathtt{j},\mathtt{j}]} = \mathcal{B}_{ij}^\top \Omega_{ij} \mathcal{B}_{ij}$.