

SIM-Sync: From Certifiably Optimal Synchronization over the 3D Similarity Group to Scene Reconstruction with Learned Depth

Xihang Yu*

College of Literature, Science, and the Arts, University of Michigan.

Heng Yang†

School of Engineering and Applied Sciences, Harvard University

Abstract

We present **SIM-Sync**, a *certifiably optimal* algorithm that estimates camera trajectory and 3D scene structure *directly from multiview image keypoints*. **SIM-Sync** fills the gap between pose graph optimization and bundle adjustment; the former admits efficient global optimization but requires relative pose measurements and the latter directly consumes image keypoints but is difficult to optimize globally (due to camera projective geometry).

The bridge to this gap is a *pretrained* depth prediction network. Given a graph with nodes representing monocular images taken at unknown camera poses and edges containing pairwise image keypoint correspondences, **SIM-Sync** first uses a pretrained depth prediction network to *lift* the 2D keypoints into 3D *scaled* point clouds, where the scaling of the per-image point cloud is unknown due to the scale ambiguity in monocular depth prediction. **SIM-Sync** then seeks to *synchronize* jointly the unknown camera poses and scaling factors (*i.e.*, over the 3D similarity group) by minimizing the sum of the Euclidean distances between edge-wise scaled point clouds. The **SIM-Sync** formulation, despite nonconvex, allows designing an efficient certifiably optimal solver that is almost identical to the **SE-Sync** algorithm. Particularly, after solving the translations in closed-form, the remaining optimization over the rotations and scales can be written as a *quadratically constrained quadratic program*, for which we apply Shor’s semidefinite relaxation. We show how to add scale regularization in the semidefinite program to prevent contraction of the estimated scales.

We demonstrate the tightness, robustness, and practical usefulness of **SIM-Sync** in both simulated and real experiments. In simulation, we show (i) **SIM-Sync** compares favorably with **SE-Sync** in scale-free synchronization, and (ii) **SIM-Sync** can be used together with robust estimators to tolerate a high amount of outliers. In real experiments, we show (a) **SIM-Sync** achieves similar performance as **Ceres** on bundle adjustment datasets, and (b) **SIM-Sync** performs on par with **ORB-SLAM3** on the TUM dataset with zero-shot depth prediction.¹

1 Introduction

3D scene reconstruction and camera trajectory estimation from image sequences remains one of the most fundamental and extensively studied problems in robotics and computer vision. At the heart of this problem lies (i) finding reliable *feature correspondences*, *e.g.*, keypoints, across images (sometimes referred to as data association in robotics), and (ii) *estimating camera poses* given the associated features. In this paper, we focus on the camera trajectory estimation problem and denote $x_i = (R_i, t_i) \in \text{SE}(3)$, $i = 1, \dots, N$ as the set of camera poses to be estimated.

*Work done during visit at the Harvard Computational Robotics Lab. Email: xihangyu@umich.edu

†Email: hankyang@seas.harvard.edu

¹Code available: <https://github.com/ComputationalRobotics/SIM-Sync>.

A long list of formulations and solutions have been developed for camera trajectory estimation, for which we refer to [5, Chapter 9] and [29, Chapter 11]. We motivate our work by describing two of the most popular formulations.

The first formulation, exemplified by *pose graph optimization* (PGO) [24] for simultaneous localization and mapping (SLAM) in robotics, first estimates *relative* camera poses from image features, *i.e.*, estimating $\tilde{x}_{ij} \approx x_i^{-1}x_j$ for $(i, j) \in \mathcal{E}$ ² with overlapping image features,³ and then solves an optimization problem that *synchronizes* $\{x_i\}_{i=1}^N$ from the relative measurements $\{\tilde{x}_{ij}\}_{(i,j) \in \mathcal{E}}$. The synchronization problem, despite nonconvex, has an objective function that is *polynomial* in the unknowns. Consequently, the seminal work SE-Sync [9, 10, 24] demonstrated efficiently solving the problem to *certifiable global optimality* using semidefinite programming (SDP) relaxations and customized low-rank SDP solvers. Holmes and Barfoot [14] derived similar SDP-based global optimality certificates for landmark-based SLAM, as long as the landmark measurements are in 3D, which preserves the polynomiality of the objective function.

The second formulation, exemplified by *bundle adjustment* (BA) [1, 25] for structure from motion (SfM) in computer vision, solves an optimization problem that jointly estimates camera poses and 3D keypoints by minimizing *geometric reprojection errors*. The key challenge to solve this formulation is that the objective function is no longer polynomial in the camera poses and 3D keypoints, but rather a sum of *rational* functions. Therefore, the most popular solution methods, *e.g.*, COLMAP [1], Ceres [2], and GTSAM [12], rely on gradient-based local optimization techniques and can be sensitive to the quality of initialization. It is, however, not impossible to solve this formulation to global optimality. For example, by replacing the geometric reprojection error with the *object space error*,⁴ [26] recovers polynomiality and designed a globally convergent algorithm, albeit not based on SDP relaxations. It may also be possible to apply the SDP relaxation hierarchy designed for rational function optimization [7] to the BA formulation. However, the SDP relaxation hierarchy in [7] scales poorly (and worse than its polynomial counterpart) and such an attempt has never been made in the literature.

In summary, given image feature correspondences, the BA-type formulation estimates camera poses in a single step by minimizing an objective function that is directly constructed from image keypoints.⁵ The PGO-type formulation, however, takes a two-step approach, where the first step estimates relative camera poses and the second step performs pose synchronization. The BA-type formulation is more straightforward than the PGO-type formulation,⁶ but more difficult to optimize globally. Therefore, we ask the question: *can we design a camera trajectory estimation formulation that (i) directly consumes image features and (ii) admits efficient global optimization?*

Contributions. Inspired by the recent trend in computer vision [17, 20, 21, 36] that leverages a *learned depth prediction network* [23] for bundle adjustment, we propose a *certifiably optimal* camera trajectory estimation algorithm that directly consumes image feature correspondences. The key insight is that, with the help of a pretrained depth prediction network, 2D keypoint correspondences can be effectively *lifted* to 3D keypoint correspondences, leading to a formulation whose objective function is again polynomial in the unknown camera poses. Yet different from the formulation in multiple point cloud registration (MPCR) [11, 15], the lifted 3D keypoints have

²In PGO, a pose graph is formulated, where the node set $\mathcal{V} = \{1, \dots, N\}$ includes the unknown absolute camera poses, and the edge set \mathcal{E} contains all pairs of nodes such that relative poses can be measured.

³For example, relative camera poses can be estimated using RANSAC [13] plus the five-point algorithm [22]. More generally, such relative poses can be estimated not only from cameras images, but also from other sensor modalities such as IMU, GPS, and LiDAR with suitable algorithms.

⁴The object space error is essentially the point-to-line distance between the 3D keypoint and the bearing vector emanating from the camera center to the 2D image keypoint. This error function has been used by multiple authors as an approximation for the geometric reprojection error [16].

⁵We remark that BA algorithms often estimate relative camera poses to initialize the joint optimization in camera poses and 3D keypoints. Therefore, one can also consider them as two-step approaches.

⁶The BA-type formulation only require camera images, while the PGO-type formulation typically requires additional sensors such as IMU.

an *unknown scaling factor* (per camera frame) due to the scale ambiguity of depth prediction. As a result, our formulation seeks to estimate, for each camera frame $i \in [N]$, both the camera pose $(R_i, t_i) \in \text{SE}(3)$ and the scaling factor $s_i > 0$, *i.e.*, an element $x_i = (s_i, R_i, t_i)$ in the *3D similarity group* $\text{SIM}(3)$. For this reason, we call our formulation **SIM-Sync**, which performs synchronization over $\text{SIM}(3)$ using pairwise image keypoint correspondences.⁷ A graphical illustration of **SIM-Sync** using the **TUM** dataset [28] as an example is provided in Fig. 1.

The nice property of our **SIM-Sync** formulation is that it allows designing a certifiably optimal solver in a way that is almost identical to **SE-Sync**. Specifically, we first solve the unknown translations as a function of the rotations and scales, arriving at an optimization problem whose objective is a *quartic* (*i.e.*, degree-four) polynomial in the rotations and scales. Then, by creating *scaled rotations* $\tilde{R}_i = s_i R_i, i = 1, \dots, N$, the quartic polynomial becomes *quadratic* and the problem becomes a quadratically constrained quadratic program (QCQP), for which we apply the standard Shor's semidefinite relaxation [4]. Given the same number of camera frames N , our **SIM-Sync** relaxation leads to an SDP having the same matrix size as that of **SE-Sync** ($3N \times 3N$), but with *fewer* linear equality constraints (due to fewer quadratic equality constraints on \tilde{R}_i). Moreover, we show that it is possible to *regularize* the scale estimation to be close to 1 by adding $\sum_{i=1}^N (s_i^2 - 1)^2$ into the objective, which can be conveniently handled by the SDP relaxation with small positive semidefinite variables. This regularization effectively prevents *contraction* (*i.e.*, s_i tends to zero) of the estimated camera trajectory in certain test cases (*e.g.*, the 3D grid graph).

We then conduct a suite of simulated and real experiments to investigate the empirical performance of **SIM-Sync**. Particularly, with simulations we show that (i) the **SIM-Sync** relaxation is almost always *exact* (tight) under small to medium noise corruption in the measurements, *i.e.*, globally optimal estimates can be computed and certified; (ii) **SIM-Sync** achieves similar estimation accuracy as **SE-Sync** (and **SE-Sync** refined by g2o); (iii) **SIM-Sync** can be robustified against 40–80% outlier correspondences by running GNC [33] and TEASER [34] to preprocess pairwise correspondences. With real experiments we show that (a) **SIM-Sync** compares favorably with Ceres on the BAL bundle adjustment dataset [1], and (b) **SIM-Sync** achieves similar performance as ORB-SLAM3 [8] on the **TUM** dataset [28] with zero-shot depth prediction.

Paper organization. We present the **SIM-Sync** formulation in Section 2, where we also derive the simplified QCQP formulation. We introduce the semidefinite relaxation and scale regularization in Section 3. We present experimental results in Section 4 and conclude in Section 5.

2 Problem Formulation

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $i \in \mathcal{V} = [N]$ is associated with an RGB image $I_i \in \mathbb{R}^{H \times W \times 3}$ and an unknown camera pose $(R_i, t_i) \in \text{SE}(3)$, and each edge $(i, j) \in \mathcal{E}$ contains a set of n_{ij} dense pixel-to-pixel correspondences $\mathcal{C}_{ij} = \{p_{i,k} \leftrightarrow p_{j,k}\}_{k=1}^{n_{ij}}$ with $p_{i,k} \in \mathbb{R}^2$ the k -th pixel location in image I_i and $p_{j,k} \in \mathbb{R}^2$ the k -th pixel location in image I_j . Assuming all the camera intrinsics $\{K_i\}_{i=1}^N$ are known, we can compute

$$\tilde{p}_{i,k} = K_i^{-1} \begin{bmatrix} p_{i,k}^x \\ p_{i,k}^y \\ 1 \end{bmatrix} \quad (1)$$

as the *bearing vector* normalized by the camera intrinsics. The third entry of $\tilde{p}_{i,k}$ is equal to 1.

Pretrained depth prediction. Suppose we are given a pretrained depth estimation network that, for each image I_i , produces a depth map. Let $d_{i,k} > 0$ be the predicted depth of $p_{i,k}$ and

⁷An earlier work [27] formulates a synchronization problem over $\text{SIM}(3)$ with relative pose measurements instead of direct image keypoints.

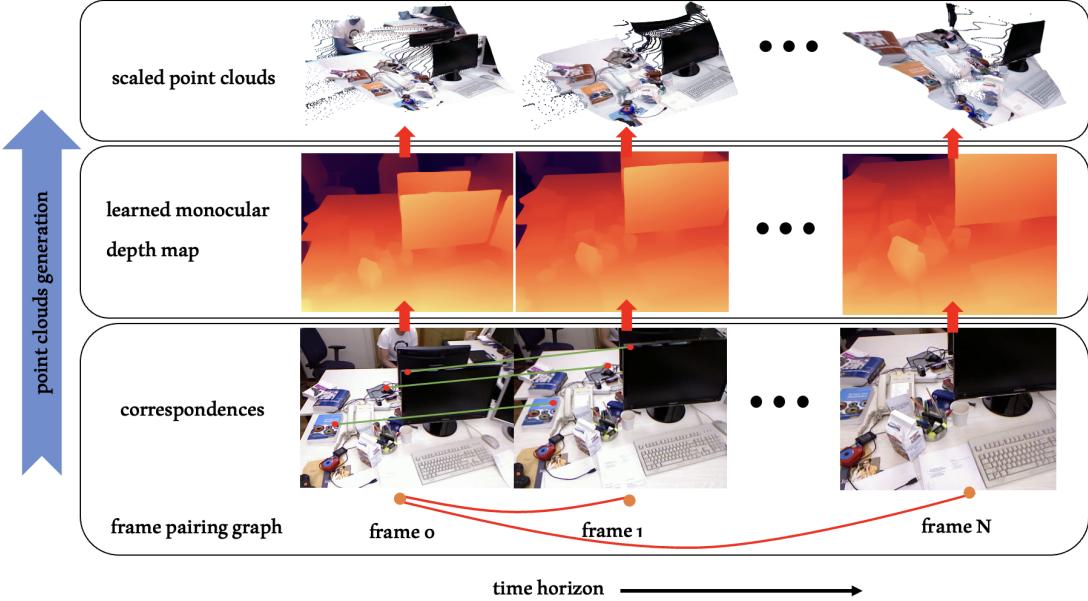


Figure 1: Illustration of **SIM-Sync** on the **TUM** dataset [28].

$s_i > 0$ be the unknown scale coefficient for image I_i .⁸ Consequently,

$$\hat{p}_{i,k} = s_i d_{i,k} \tilde{p}_{i,k} \quad (2)$$

corresponds to the 3D location of $p_{i,k}$ in the i -th camera frame. Effectively, with the pre-trained depth predictor, for every $(i, j) \in \mathcal{E}$, we have a pair of *scaled* point cloud measurements $\{d_{i,k} \tilde{p}_{i,k}\}_{k=1}^{n_{ij}}$ and $\{d_{j,k} \tilde{p}_{j,k}\}_{k=1}^{n_{ij}}$, as shown in Fig. 1 top panel.

The SIM-Sync formulation. We are interested in estimating the unknown camera poses and the per-image scale coefficients $\{x_i = (s_i, R_i, t_i)\}_{i=1}^N$. We formulate the following optimization

$$\min_{\substack{s_i > 0, R_i \in \text{SO}(3), t_i \in \mathbb{R}^3 \\ i=1, \dots, N}} \sum_{(i,j) \in \mathcal{E}} \sum_{k=1}^{n_{ij}} w_{ij,k} \left\| \left(R_i \underbrace{(s_i d_{i,k} \tilde{p}_{i,k}) + t_i}_{\hat{p}_{i,k}} \right) - \left(R_j \underbrace{(s_j d_{j,k} \tilde{p}_{j,k}) + t_j}_{\hat{p}_{j,k}} \right) \right\|^2 \quad (\text{SIM-Sync})$$

where the objective function seeks to minimize the 3D point-to-point distances because (R_i, t_i) transforms $\hat{p}_{i,k}$, and (R_j, t_j) transforms $\hat{p}_{j,k}$ into the same global coordinate frame. In (SIM-Sync), we include $w_{ij,k} > 0$ for generality: these known weights capture the potential uncertainty of the correspondences. Usually these weights are unknown and in our experiments we use GNC and TEASER to estimate them so that $w_{ij,k} = 1$ indicates inliers and $w_{ij,k} = 0$ indicates outliers.

Anchoring. Problem (SIM-Sync) is ill-defined. One can choose $s_i \rightarrow 0, \forall i = 1, \dots, N, t_1 = t_2 = \dots = t_N = \text{constant}$, and the objective of (SIM-Sync) can be set arbitrarily close to zero. To resolve this issue, we anchor the first frame and set $R_1 = \mathbf{I}_3, t_1 = \mathbf{0}, s_1 = 1$, which is common practice in many related pose graph estimation formulations [24].

⁸In practice, we use interpolation to obtain $d_{i,k}$ because the depth map is discretized. We also discard depth values that are too far away, which tend to be erroneous.

2.1 A QCQP Formulation

The (**SIM-Sync**) formulation is readily in the form of a polynomial optimization problem (POP). The objective function is a quartic polynomial, the constraint $s_i > 0$ is an affine polynomial inequality, and the constraint $R_i \in \text{SO}(3)$ is equivalent to a set of quadratic polynomial constraints [32]. Therefore, one can directly apply Lasserre's hierarchy of moment relaxations [18] to design convex SDP relaxations for (**SIM-Sync**). However, as noted in [32], a direct application of Lasserre's hierarchy often leads to SDPs beyond the scalability of current solvers. Therefore, in the following we will simplify (**SIM-Sync**) as a quadratically constrained quadratic problem (QCQP), in a way that is inspired by SE-Sync [24].

Our first step is to simplify the objective function in (**SIM-Sync**).

Proposition 1 (Simple Objective Function). *Let $t = [t_1^\top, \dots, t_N^\top]^\top \in \mathbb{R}^{3N}$ be the concatenation of translations, and $r = [\text{vec}(s_1 R_1)^\top, \dots, \text{vec}(s_N R_N)^\top]^\top \in \mathbb{R}^{9N}$ be the concatenation of (vectorized) scaled rotations, then the objective function of (**SIM-Sync**) can be written as*

$$L(t, r) = t^\top (Q_1 \otimes \mathbf{I}_3) t + 2r^\top (V \otimes \mathbf{I}_3) t + r^\top (Q_2 \otimes \mathbf{I}_3) r, \quad (3)$$

where Q_1, Q_2, V can be computed as follows

$$Q_1 = \sum_{(i,j) \in \mathcal{E}} (W_{ij} e_i^\top - W_{ij} e_j^\top)^\top (W_{ij} e_i^\top - W_{ij} e_j^\top) \in \mathbb{R}^{N \times N}, \quad (4)$$

$$Q_2 = \sum_{(i,j) \in \mathcal{E}} (e_i^\top \otimes P_i^\top - e_j^\top \otimes P_j^\top)^\top (e_i^\top \otimes P_i^\top - e_j^\top \otimes P_j^\top) \in \mathbb{R}^{3N \times 3N}, \quad (5)$$

$$V = \sum_{(i,j) \in \mathcal{E}} (e_i^\top \otimes P_i^\top - e_j^\top \otimes P_j^\top)^\top (W_{ij} e_i^\top - W_{ij} e_j^\top) \in \mathbb{R}^{3N \times N}, \quad (6)$$

with

$$P_i = [\sqrt{w_{ij,1}} d_{i,1} \tilde{p}_{i,1} \quad \cdots \quad \sqrt{w_{ij,n_{ij}}} d_{i,n_{ij}} \tilde{p}_{i,n_{ij}}] \in \mathbb{R}^{3 \times n_{ij}}, \quad i = 1, \dots, N, \quad (7)$$

$$W_{ij} = [\sqrt{w_{ij,1}}, \dots, \sqrt{w_{ij,n_{ij}}}]^\top \in \mathbb{R}^{n_{ij}}, \quad (i, j) \in \mathcal{E}, \quad (8)$$

and $e_i \in \mathbb{R}^N$ the all-zero vector except that the i -th entry is equal to 1.

The proof of Proposition 1 is in Appendix A.

Now that the objective $L(t, r)$ in (3) is quadratic in t , an unconstrained variable. Therefore, we can set the gradient of $L(t, r)$ w.r.t. t to zero and solve the optimal t in closed form.

Proposition 2 (Scaled-Rotation-Only Formulation). *Let $R = [s_1 R_1, \dots, s_N R_N] \in \mathbb{R}^{3 \times 3N}$ be the concatenation of (unvectorized) scaled rotations, then problem (**SIM-Sync**) is equivalent to the following optimization*

$$\rho^* = \min_R \text{tr}(QR^\top R), \quad (9)$$

where Q can be computed as

$$Q = A^\top Q_1 A + V A + A^\top V^\top + Q_2 \in \mathbb{S}^{3N}, \quad (10)$$

with

$$A = \begin{bmatrix} \mathbf{0}_{1 \times 3N} \\ -(\bar{Q}_1^\top \bar{Q}_1)^{-1} \bar{Q}_1^\top V^\top \end{bmatrix} \in \mathbb{R}^{N \times 3N}, \quad (11)$$

and $\bar{Q}_1 \in \mathbb{R}^{N \times (N-1)}$ includes the last $N-1$ columns of Q_1 . Moreover, denote the optimal solution of (9) as R^* , then the optimal translation to (**SIM-Sync**) can be recovered as

$$t^* = (A \otimes \mathbf{I}_3) \text{vec}(R^*). \quad (12)$$

The proof of Proposition 2 is in Appendix B.

Problem (9) is still a quartic POP because the variable R contains the product between scales and rotations. Our last step is to create new variables $\bar{R}_i = s_i R_i, i = 1, \dots, N$ so that problem (9) becomes a QCQP.

Proposition 3 (QCQP Formulation). *Let $\mathfrak{so}(3) \subset \mathbb{R}^{3 \times 3}$ be the set of matrices that can be written as the product between a nonnegative scalar and a 3×3 orthogonal matrix, i.e.,*

$$\mathfrak{so}(3) = \{\bar{R} \in \mathbb{R}^{3 \times 3} \mid \exists s \geq 0, R \in O(3) \text{ such that } \bar{R} = sR\}. \quad (13)$$

Then $\mathfrak{so}(3)$ can be described by the following quadratic constraints

$$\bar{R} = [c_1 \ c_2 \ c_3] \in \mathfrak{so}(3) \iff \begin{cases} c_1^\top c_1 = c_2^\top c_2 = c_3^\top c_3 \\ c_1^\top c_2 = c_2^\top c_3 = c_3^\top c_1 = 0 \end{cases}. \quad (14)$$

Consider the following quadratically constrained quadratic program (QCQP)

$$\rho_{\text{QCQP}}^* = \min_R \text{tr}(QR^\top R) \text{ subject to } R = [\bar{R}_1 \ \dots \ \bar{R}_N] \in \mathfrak{so}(3)^N, \quad (\text{QCQP})$$

and let $R^ = [\bar{R}_1^*, \dots, \bar{R}_N^*]^\top$ be a global optimizer. If*

$$\det \bar{R}_i^* > 0, i = 1, \dots, N, \quad (15)$$

then R^ is a global minimizer to problem (9) and hence also (SIM-Sync).*

The proof of Proposition 3 is in Appendix C.

With Proposition 3, we know that if we can solve (QCQP) to global optimality, then by checking the determinants of the optimal solution as in (15), we can certify its global optimality to the original (SIM-Sync) problem. In fact, as shown in SE-Sync [24], typically the relaxation from $SO(3)$ to $O(3)$ is tight because the set of rotations and the set of reflections in $O(3)$ are disjoint from each other. Therefore, we can almost expect $\rho_{\text{QCQP}}^* = \rho^*$ (as we also observe in experiments).

3 Semidefinite Relaxation

The previous section has reformulated (more precisely, relaxed) the (SIM-Sync) formulation as the compact (QCQP). We can now design the following semidefinite relaxation.

Proposition 4 (SDP Relaxation). *The following semidefinite program (SDP)*

$$f^* = \min_{X \in \mathbb{S}^{3N}} \text{tr}(QX) \text{ subject to } X = \begin{bmatrix} \alpha_1 \mathbf{I}_3 & \cdots & * \\ \vdots & \ddots & \vdots \\ * & \cdots & \alpha_N \mathbf{I}_3 \end{bmatrix} \succeq 0 \quad (16)$$

is a convex relaxation to (QCQP) and $f^ \leq \rho_{\text{QCQP}}^*$. Let X^* be a global minimizer of (16). If $\text{rank}(X^*) = 3$, then X^* can be factorized as $X^* = (R^*)^\top R^*$, where $R^* \in \mathfrak{so}(3)^N$ is a global optimizer to (QCQP).*

Note that $\alpha_1 = 1$ in (16) because we set $s_1 = 1$. To enforce the diagonal blocks of X in (16) to be scaled identity matrices, one just need to (i) set their off-diagonal entries as zero, and (ii) set their diagonal entries to be equal to each other. As a result, there are $5(N - 1) + 6$ linear equality constraints in (16), which is fewer than the $6N$ linear equality constraints in SE-Sync.

Suboptimality. In practice, checking the rank condition of the optimal solution of (16) can be sensitive to numerical thresholds. Therefore, we always generate a solution \hat{R} from X^* that is also feasible for problem (9) and evaluate the objective of (9) at \hat{R} , denoted as $\hat{\rho}$ and satisfies

$$f^* \leq \rho_{\text{QCQP}}^* \leq \rho^* \leq \hat{\rho}. \quad (17)$$

We then compute the *relative suboptimality*

$$\eta = \frac{\hat{\rho} - f^*}{1 + |f^*| + |\hat{\rho}|}. \quad (18)$$

Clearly, $\eta = 0$ certifies global optimality of the solution \hat{R} and tightness of the SDP relaxation.

Rounding. We perform the following procedure to round a feasible \hat{R} from X^* . First we compute the spectral decomposition of $X^* = \sum_{i=1}^{3N} \lambda_i u_i u_i^\top$. Then we assemble

$$U = [\sqrt{\lambda_1} u_1 \quad \sqrt{\lambda_2} u_2 \quad \sqrt{\lambda_3} u_3] = [U_1 \quad \cdots \quad U_N]^\top \in \mathbb{R}^{3N \times 3}, \quad (19)$$

where $\lambda_1, \lambda_2, \lambda_3$ are the three largest eigenvalues. Finally, we compute the scales and rotations

$$\hat{s}_1 = 1, \hat{s}_i = \|U_1^\top U_i\|_F / \sqrt{3}, i = 2, \dots, N, \quad (20)$$

$$\hat{R}_1 = \mathbf{I}_3, \hat{R}_i = \Pi_{\text{SO}(3)}(U_1^\top U_i / s_i), i = 2, \dots, N, \quad (21)$$

and assemble $\hat{R} = [\hat{s}_1 \hat{R}_1, \dots, \hat{s}_N \hat{R}_N]$, where $\Pi_{\text{SO}(3)}$ denotes the projection onto $\text{SO}(3)$.

3.1 Scale Regularization

Empirically, we find that for certain graph structures (shown in Section 4), the optimal scale estimation of (9) tends to become much smaller than 1 for $i = 2, \dots, N$, a phenomenon that we call *contraction*. This is undesired because the true scales are often close to 1. Therefore, we propose to regularize the (QCQP) and the SDP (16). Observe that, if the relaxation is tight, then $\alpha_i = s_i^2$ in (16) for $i = 1, \dots, N$. Therefore, by adding $(\text{tr}(X_{ii})/3 - 1)^2 = (\alpha_i - 1)^2$ (X_{ii} denotes the i -th diagonal block of X) into the objective of (16), we encourage the SDP to penalize $(s_i^2 - 1)^2$ and hence prevent the scale estimation from contracting. The scale-regularized problem, fortunately, is still an SDP.

Proposition 5 (Scale Regularization). *The following scale-regularized problem*

$$\min_{X \in \mathbb{S}^{3N}} \text{tr}(QX) + \lambda \left(\sum_{i=1}^N (\text{tr}(X_{ii})/3 - 1)^2 \right) \text{ subject to } X \text{ as in (16)} \quad (22)$$

for a given $\lambda > 0$, is equivalent to

$$\min_{X \in \mathbb{S}^{3N}} \text{tr}(QX) + \lambda \sum_{i=1}^N t_i \quad (23)$$

$$\text{subject to } X \text{ as in (16)} \quad (24)$$

$$\begin{bmatrix} 1 & \text{tr}(X_{ii})/3 - 1 \\ \text{tr}(X_{ii})/3 - 1 & t_i \end{bmatrix} \succeq 0, i = 1, \dots, N. \quad (25)$$

Proposition 5 is easy to verify because (25) implies $t_i \geq (\text{tr}(X_{ii})/3 - 1)^2$ by the Schur complement lemma, and the minimization in (23) will push $t_i = (\text{tr}(X_{ii})/3 - 1)^2$.

Both SDPs (16) and (23) are implemented in Python and solved with MOSEK by directly passing the problem data to the MOSEK Python interface.

4 Experiments

We test the performance of **SIM-Sync** in both simulated and real datasets. All experiments are conducted on a laptop equipped with an Intel 14-Core i7-12700H CPU and 32 GB memory.

In Section 4.1, we test **SIM-Sync** in simulated *scale-free synchronization* problems (*i.e.*, $s_i = 1, i = 1, \dots, N$) and compare its performance to **SE-Sync** and **SE-Sync+g2o**.

In Section 4.2, we test the scale regularization (23) and show that it effectively prevents contraction of the estimated pose graph.

In Section 4.3, we simulate outliers in the feature correspondences and demonstrate that **SIM-Sync** can be used together with robust estimators such as **GNC** and **TEASER**.

Finally, we test **SIM-Sync** in real datasets. Section 4.4 provides results of **SIM-Sync** on the **BAL** dataset [1] that is popular in computer vision for bundle adjustment. Section 4.5 provides results of **SIM-Sync** on the **TUM** dataset [28] that is popular in robotics for SLAM.

4.1 Scale-free Synchronization

Setup. We assume that the scaling factor is known, *i.e.*, $s_i = 1, i = 1, \dots, N$, which is a realistic assumption when the images are taken by RGB-D cameras or registered with LiDAR scanners. Consequently, we are only interested in estimating node-wise poses $(R_i, t_i), i \in \mathcal{V}$ given pairs of point cloud measurements over the edges \mathcal{E} . To simulate the pose graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we first simulate a random point cloud $P \in \mathbb{R}^{3 \times n}, n = 1000$ in the world frame. Each point in P follows a Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_3)$. We then simulate a trajectory of camera poses $(R_i, t_i), i = 1, \dots, N$ with $N = 50$ by following certain graph topologies, specifically, a circle, a grid, and a line, as commonly used in related works [15, 24]. Details for simulating the camera trajectories are as follows.

- Circle. The camera moves in a circle with a radius of 10 meters for one round in 50 steps, while facing the circle’s center.
- Grid. In a cube with a edge-length of 2 meters, the camera moves on the surface for 50 steps. The camera can only move one meter to an adjacent node at each step. The starting point is randomly chosen among all nodes. A typical example is as shown in Fig. 2.
- Line. The camera moves linearly for 3 meters in 50 discrete steps while facing the point cloud P at a distance of 10 meters from the line.

Given each camera pose $(R_i, t_i), i = 1, \dots, N$, we generate a noisy point cloud observation

$$P_i = R_i P + t_i + \epsilon_i \quad (26)$$

where $\epsilon_i \in \mathbb{R}^{3 \times n}$ are i.i.d. Gaussian noise vectors following $\mathcal{N}(0, \sigma^2 \mathbf{I}_3)$. We then simulate correspondences over each edge $(i, j) \in \mathcal{E}$ by subsampling P_i and P_j . To make the correspondences more realistic, we associate P_i and P_j as follows:

1. We first find a subset of indices $\mathcal{I}_{ij} \subseteq [n]$ such that points in \mathcal{I}_{ij} lie in both the field of view (FOV) of camera i and the FOV of camera j . FOV is set as 60 degrees for all experiments.
2. We then randomly select a subset $\mathcal{K}_{ij} \subseteq \mathcal{I}_{ij}$ with cardinality q , a random number between 10 and $|\mathcal{I}_{ij}|$, and let $\widehat{P}_i = \{p_{i,k} \in P_i \mid k \in \mathcal{K}_{ij}\}$ and $\widehat{P}_j = \{p_{j,k} \in P_j \mid k \in \mathcal{K}_{ij}\}$ be the final point cloud pairs on edge (i, j) .

We pass $(\widehat{P}_i, \widehat{P}_j)_{(i,j) \in \mathcal{E}}$ to **(SIM-Sync)** to estimate node-wise absolute poses.

Baselines. We compare **SIM-Sync** with **SE-Sync**. In order to use **SE-Sync**, we need to estimate relative poses among all the edges \mathcal{E} . This is done by running Arun’s method [3] on the point

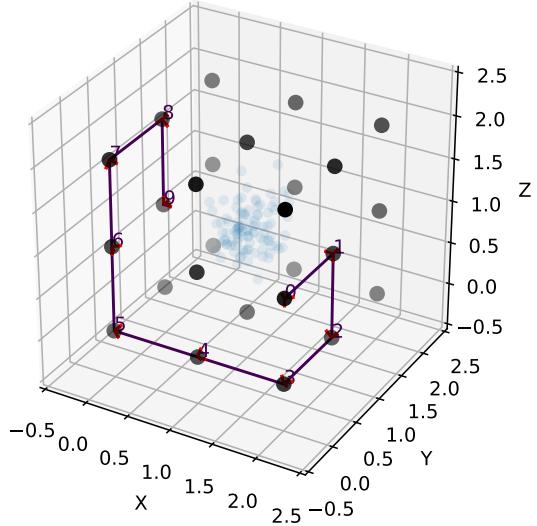


Figure 2: A sample camera trajectory in the Grid dataset.

cloud pairs $(\hat{P}_i, \hat{P}_j)_{(i,j) \in \mathcal{E}}$. **SE-Sync** also requires a covariance estimation of the relative pose. To do so, we compute the Cramer-Rao lower bound at Arun’s optimal solution and feed the covariance estimates to **SE-Sync**. We provide a detailed derivation of the covariance matrix in Appendix E.1. Note that **SE-Sync** assumes the rotational noise follows an isotropic Langevin distribution and it internally computes a Langevin approximation of the covariance matrix fed to it. Therefore, we also compare with **SE-Sync+g2o**, where the **SE-Sync** solution is used to initialize a local search using **g2o** with the Cramer-Rao lower bound.

Results. We choose $\sigma \in \{0.001, 0.01, 0.1, 1, 2, 3\}$ and at each noise level we run 20 Monte Carlo random tests. Fig. 3 shows the rotation errors and translation errors of **SIM-Sync** compared with **SE-Sync** and **SE-Sync+g2o**. In both the circle dataset and the grid dataset, **SIM-Sync** surpasses **SE-Sync** and **SE-Sync+g2o** by a (very) small margin, while in the line dataset, **SIM-Sync** and **SE-Sync** perform almost the same. Fig. 3 bottom row plots the relative suboptimality η (*cf.* (17)) of **SIM-Sync** and **SE-Sync**. We consider the relaxation is not tight if η exceeds 0.05 (the red horizontal dashed line). In the circle dataset and the line dataset, when $\sigma = 4$, **SE-Sync**’s relaxation becomes completely inexact, while **SIM-Sync** can still achieve tightness, although not always.

4.2 Scale Regularization

Setup. We study the effect of the number of poses N and the regularization factor λ on the performance of **SIM-Sync**. We use the same circle, grid, and line datasets in Section 4.1, where the scaling factor is unknown and randomly generated in $[0.9, 1.1]$. The noise level σ is fixed to 0.01. For each dataset, we generate $n = 100$ points, and vary the number of poses $N \in \{10, 50, 100, 200, 400\}$. The regularization factor λ is tested with $\{0, 1, 10, 100\}$ for the circle and line datasets, while an additional $\lambda = 200$ for the grid dataset. To ensure statistical significance, we perform 20 Monte Carlo simulations for each combination of N and λ .

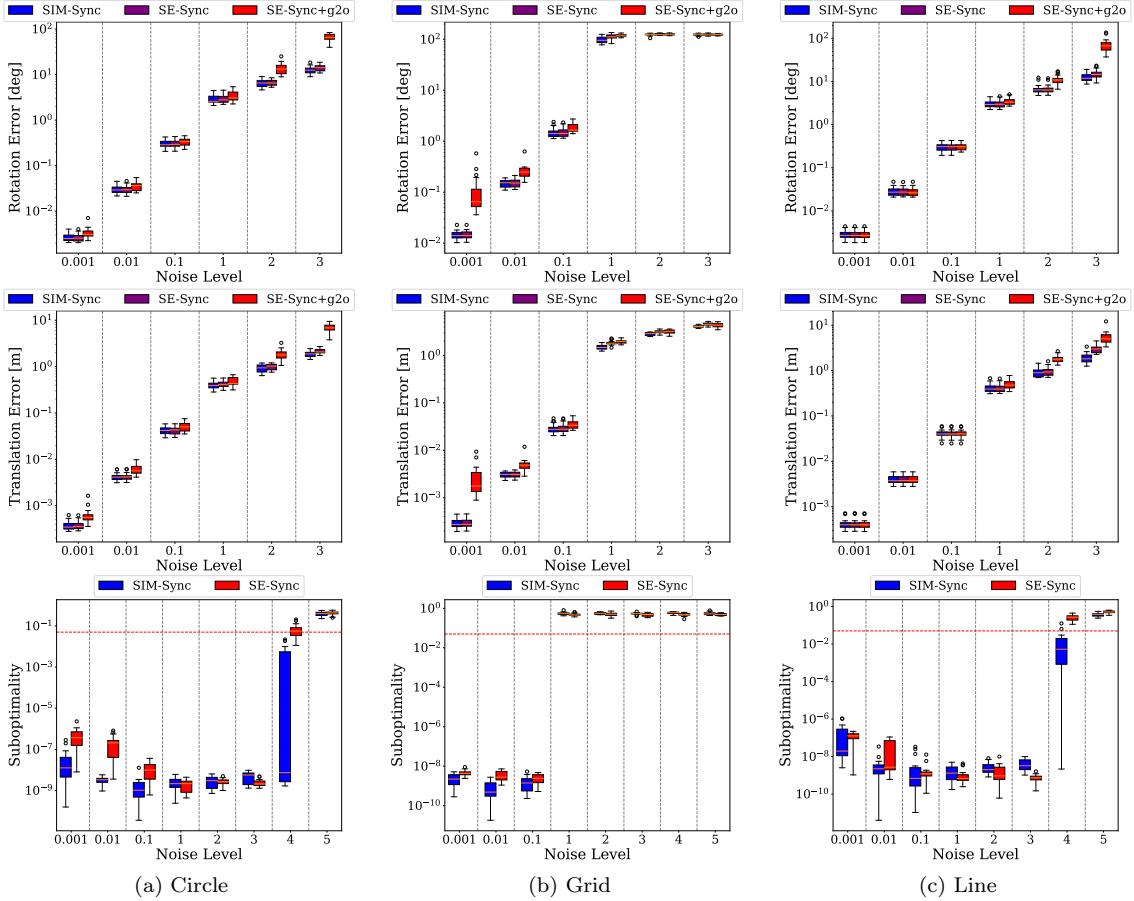


Figure 3: Rotation errors (top row), translation errors (middle row), and suboptimality (bottom row) of **SIM-Sync**, **SE-Sync** and **SE-Sync+g2o** in scale-free synchronization.

Results. Fig. 4(a)(c) plot the averaged scale estimation, rotation error, and translation error w.r.t. number of poses N on the circle dataset and the line dataset, with different colors representing different regularization factors λ . We observe that (i) as N increases, translation estimation gets slightly worse, but rotation estimation remains unaffected; (ii) the scale estimation does not contract a lot as N increases, even without scale regularization (*i.e.*, $\lambda = 0$). Fig. 4(b) shows the same results on the grid dataset, where we clearly observe contraction. Without regularization, the average scale decreases to 0.5 when $N = 400$, which also leads to poor translation estimation. With regularization $\lambda = 200$, however, we see that contraction is effectively prevented and the translation error also gets improved. This suggests that regularization improves the performance of **SIM-Sync** when N is large. It is interesting to see that rotation estimation is not affected by N and λ . This makes sense because scale and translation are coupled, while rotation is independent. We suspect that the circle graph and the line graph have a certain type of “*rigidity*” that makes them more robust to contraction, while the grid graph has weaker “*rigidity*” (*i.e.*, it is easier to bend and twist the trajectory in Fig. 2).

4.3 Outlier Rejection with Robust Estimators

Setup. We follow the same setup as in previous sections to generate the circle, grid, and line datasets with $N = 50$ poses. We sample the scale s_i uniformly from $[0.9, 1.1]$ and choose the

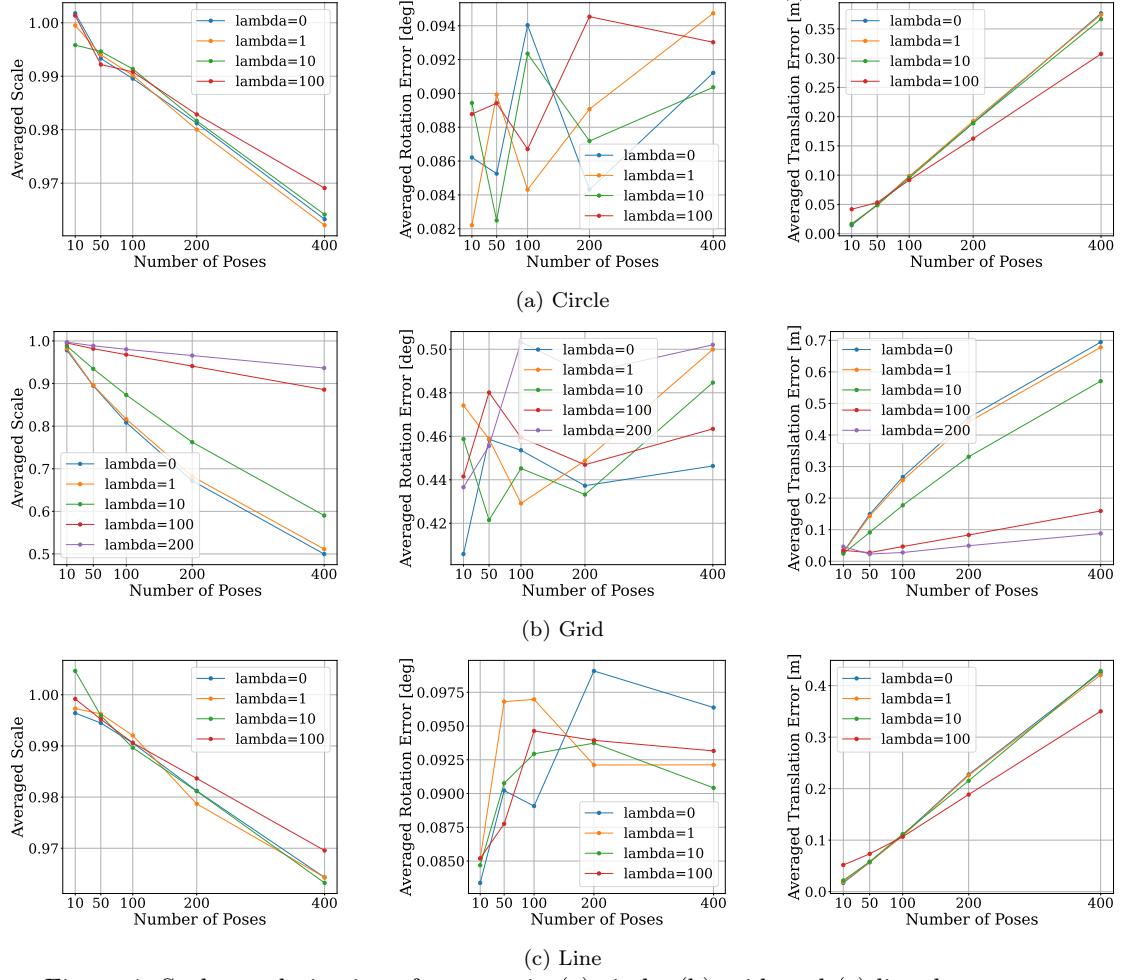


Figure 4: Scale regularization of SIM-Sync in (a) circle, (b) grid, and (c) line datasets.

noise $\sigma = 0.01$. To generate outliers, we randomly replace a fraction of the edge-wise point correspondences (\hat{P}_i, \hat{P}_j) with outliers generated from $\mathcal{N}(\mathbf{0}, \mathbf{I}_3)$. We sweep the outlier rate from 0 to 90% and perform 20 Monte Carlo simulations at each outlier rate.

Robustify SIM-Sync. We propose three ways to robustify SIM-Sync.

1. **SIM-Sync-GNC.** This approach directly wraps the SIM-Sync solver in the GNC framework [33] with a truncated least squares (TLS) robust cost function, as shown in the following optimization.

$$\min_{\substack{s_i > 0, R_i \in \text{SO}(3), t_i \in \mathbb{R}^3 \\ i=1, \dots, N}} \sum_{(i,j) \in \mathcal{E}} \sum_{k=1}^{n_{ij}} \min \left\{ \frac{\| (R_i(s_i d_{i,k} \tilde{p}_{i,k}) + t_i) - (R_j(s_j d_{j,k} \tilde{p}_{j,k}) + t_j) \|^2}{\beta^2}, 1 \right\}, \quad (27)$$

where β is set according to Appendix E.2. To solve (27), SIM-Sync-GNC starts with all weights $w_{ij,k} = 1$ in (SIM-Sync) and gradually sets some of the weights to zero based on the residuals.

2. **GNC+SIM-Sync.** This approach is a two-step algorithm. In step one, we solve the following

TLS scaled point cloud registration problem over each edge $(i, j) \in \mathcal{E}$

$$\min_{s_{ij} > 0, R_{ij} \in \text{SO}(3), t_{ij} \in \mathbb{R}^3} \sum_{k=1}^{n_{ij}} \min \left\{ \frac{\|s_{ij}R_{ij}p_{j,k} + t_{ij} - p_{i,k}\|^2}{\beta^2}, 1 \right\} \quad (28)$$

using GNC with a nonminimal solver that we develop, presented in Appendix D, based on Umeyama’s method [30]. β is set according to Appendix E.3.

In step two, we remove all outliers deemed by GNC and solve (**SIM-Sync**).

3. TEASER+SIM-Sync. Since problem (28) is exactly the problem solved by TEASER [34], we directly apply TEASER to solve (28) and then pass the inliers to (**SIM-Sync**).

Results. Fig. 5 shows the rotation, translation, and scale estimation errors w.r.t. outlier rates in the circle, grid, and line datasets. We first observe that (i) SIM-Sync-GNC fails at outlier rate 10%. This shows that GNC plus a nonminimal solver does not always work, especially when the model to be estimated is high-dimensional and the robust estimation problem is more combinatorial. We then observe that (ii) GNC+SIM-Sync is robust against 50% outliers. This shows that it is a better strategy to first apply GNC to low-dimensional robust fitting.⁹ Lastly, we observe that (iii) TEASER+SIM-Sync successfully handles outlier rates of 70% and 80%.

4.4 BAL Experiments

Setup. We test two sequences in the bundle adjustment dataset BAL [1]: the `dubrovnik-16-22106` sequence and the `ladyburg-318-41628` sequence. The former sequence consists of 16 poses with 22,106 points, and the latter sequence consists of 318 poses with 41,628 points. Both sequences provide pixel-wise correspondences for frame pairs, and these correspondences are contaminated with outliers. Since no images are provided, we cannot use a learned module to predict depth. Therefore, we use the z -component of the ground truth point position in camera frame i , which is computed by using the groundtruth camera pose to transform the point from the global frame to the i -th camera frame. Consequently, the scaling effect is not applicable, and we disable the scale prediction. To remove outliers, we use TEASER. We also test the performance of TEASER+SIM-Sync+GT, which uses ground truth poses to filter out outlier correspondences.

Baseline. We compare with two baselines TEASER+SE-Sync and Ceres. TEASER+SE-Sync first uses TEASER to estimate pair-wise relative poses and then feed them into SE-Sync, while Ceres directly optimizes reprojection errors of 3D keypoints.¹⁰ We initialize Ceres as follows: (i) camera intrinsics initialized as groundtruth; (ii) 3D keypoints initialized as groundtruth; (iii) the z -component of the camera poses are initialized using groundtruth; (iv) the other components of the camera poses are initialized to be zeros. We remark that this initialization strategy using groundtruth values is optimistic.¹¹ We also combine TEASER+SIM-Sync and TEASER+SE-Sync with Ceres, *i.e.*, we use the estimation from TEASER+SIM-Sync and TEASER+SE-Sync to initialize Ceres.

Results. Table 1 shows the quantitative results for the `dubrovnik-16-22106` sequence (the rotation error and the translation error are averaged over all the nodes). We can see that (i) in the ideal case where outliers are filtered, TEASER+SIM-Sync+GT achieves very accurate reconstruction; (ii) TEASER+SIM-Sync and TEASER+SE-Sync perform worse than TEASER+SIM-Sync+GT, but with the refinement of Ceres, the final results are accurate as well. In fact, they are better than using Ceres alone. Fig. 6 shows qualitative results of the reconstruction, where ICP is used to refine the

⁹Note that for high outlier rates, there are no data points for GNC+SIM-Sync because it fails and produces infinite values that are discarded from the plots.

¹⁰We use the official implementation http://ceres-solver.org/hnls_tutorial.html.

¹¹Without these groundtruth values as initialization it is difficult to get Ceres to work well.

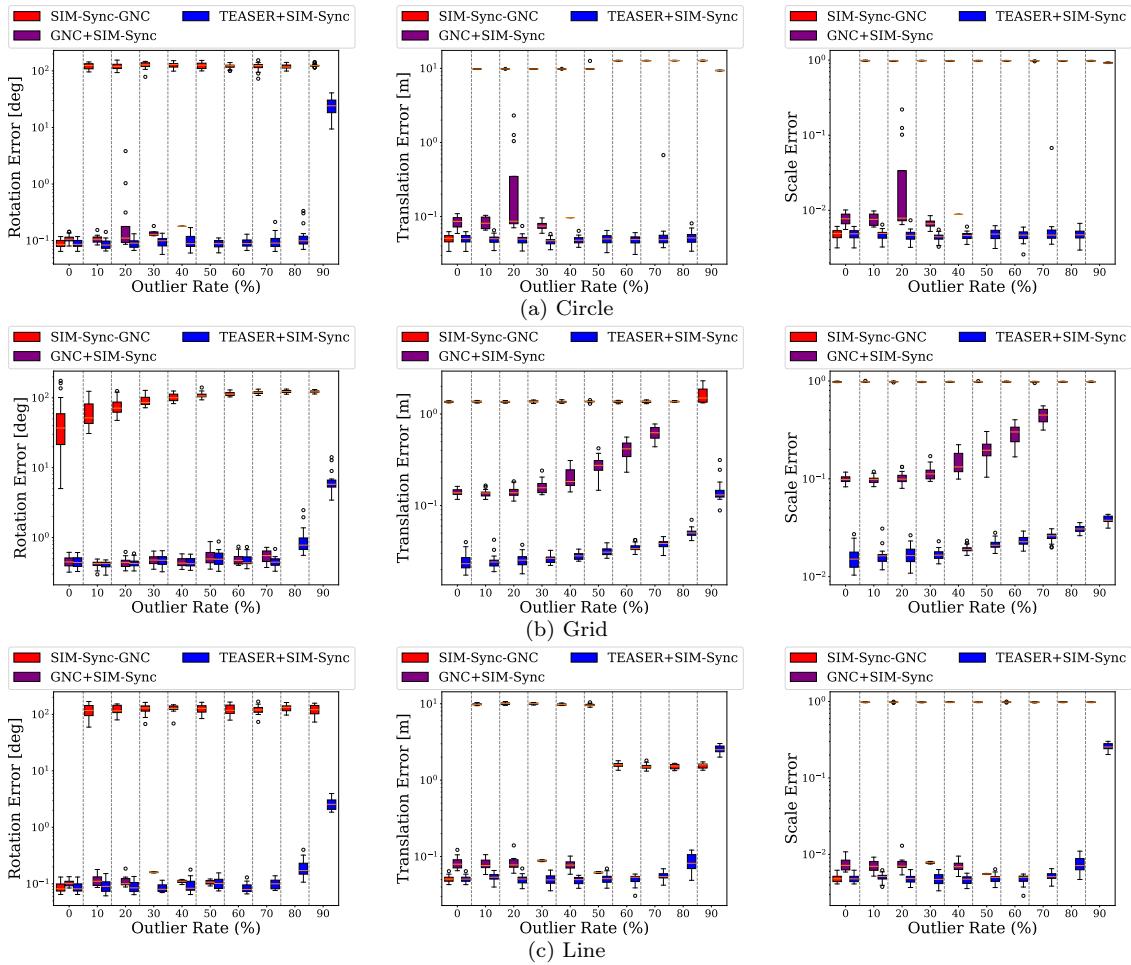


Figure 5: Robustness of SIM-Sync-GNC, GNC+SIM-Sync and TEASER+SIM-Sync in the (a) circle, (b) grid, and (c) line datasets.

reconstruction. Both TEASER+SIM-Sync and TEASER+SE-Sync did not use Ceres refinement. We see that TEASER+SIM-Sync already achieves good reconstruction without ICP and Ceres.

Table 2 shows the results for the `ladybug-318-41628` sequence. We observe that TEASER+SIM-Sync+GT still performs quite well. However, both TEASER+SIM-Sync and TEASER+SE-Sync fail to produce accurate pose estimation, though their results are better than Ceres. In fact we see that both TEASER+SIM-Sync and TEASER+SE-Sync lose tightness (suboptimality is around 10% and 20%). This suggests that the correspondences provided in this sequence is contaminated by a higher amount of outliers (compared with the `dubrovnik-16-22106` sequence) and the point clouds are also noisier. Fig. 7 shows the qualitative reconstruction results.

4.5 TUM Experiments

Setup. We test two sequences in the `TUM` dataset, the first 200 frames in the `freiburg1_xyz` sequence and the first 200 frames in the `freiburg2_xyz` sequence, respectively.¹² For TEASER+SIM-Sync, we use learned depth obtained from the MiDaS-v3 model [6, 23], with the largest 10% depth

¹²We discard the first 60 frames in `freiburg2_xyz` since the camera shakes and results in blurred images.

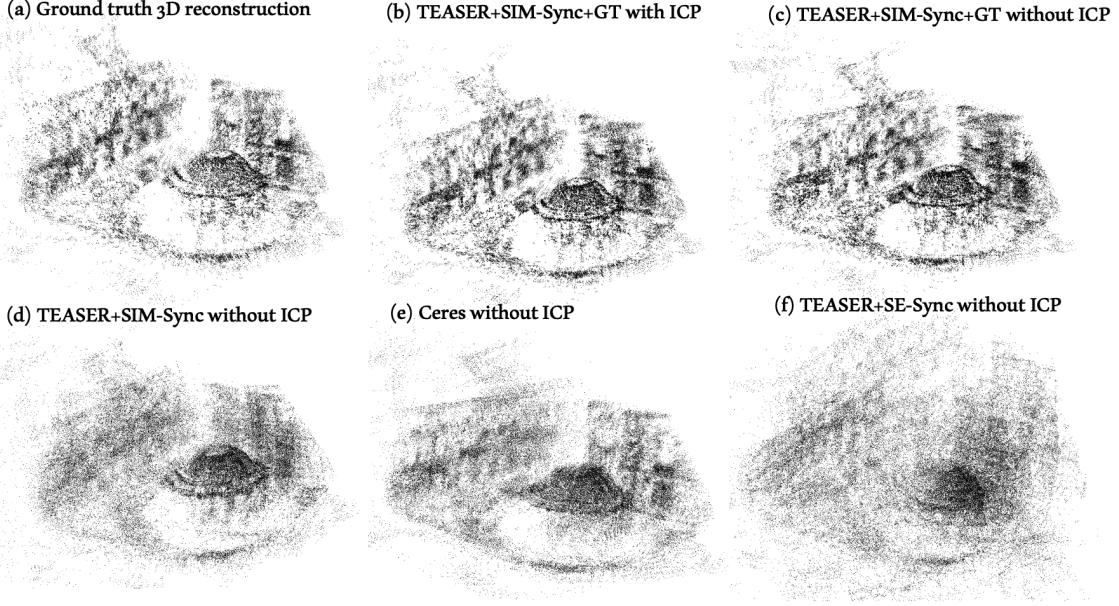


Figure 6: 3D reconstruction of a front view street scene from BAL `dubrovnik_16` sequence.

Table 1: Rotation error, translation error and suboptimality of TEASER+SIM-Sync+GT, TEASER+SIM-Sync, TEASER+SE-Sync and Ceres in BAL dataset `dubrovnik_16` sequence.

	TEASER+SIM-Sync+GT	TEASER+SIM-Sync w/o Ceres	TEASER+SIM-Sync w Ceres	TEASER+SE-Sync w/o Ceres	TEASER+SE-Sync w Ceres	Ceres
Rotation Error [deg]	0.615	8.041	2.031	16.287	2.009	2.837
Translation Error [m]	0.271	3.564	1.634	4.393	1.828	2.586
Suboptimality	1.578e-9	2.996e-9	/	9.651e-10	/	/

discarded. Note that MiDaS-v3 is not trained on the TUM dataset, and we directly use its default parameter configuration (*i.e.*, zero-shot). For TEASER+SIM-Sync+GTDepth, we use ground truth depth. We conduct two tests for TEASER+SIM-Sync/TEASER+SIM-Sync+GTDepth. In the first test, we run TEASER+SIM-Sync/TEASER+SIM-Sync+GTDepth on the *essential graph* (EG), which is a set of key frames and edges selected by the ORB-SLAM3 [8] algorithm. In the second test, we run TEASER+SIM-Sync/TEASER+SIM-Sync+GTDepth on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ generated as follows

$$\mathcal{E} = \{(i, j) \mid i = 1, 2, \dots, N - 2, \text{ and } j = i + 1 \text{ or } j = i + 2 \text{ or } (i = N - 1 \text{ and } j = N)\}, \quad (29)$$

i.e., we sample the frame pairs of neighboring 3 frames. We utilize SIFT [19] to get initial correspondences and then apply the learned CAPS descriptor [31] to sort the correspondences by the feature similarity between two points. We keep a maximum of 400 SIFT correspondences.

Baseline. We use the state-of-the-art visual SLAM algorithm ORB-SLAM3 [8] as a baseline. We use Monocular mode without IMU of ORB-SLAM3 and run on its default setting in the official script of running TUM dataset.¹³ We also compare with the stereo mode of ORB-SLAM3.

Results. We take a detour first to demonstrate how TEASER works as shown in Fig. 8. We randomly pick a pair of images from TEASER+SIM-Sync+GTDepth and TEASER+SIM-Sync. The red lines indicate outlier correspondences detected by TEASER while the green lines are inliers. We can see that TEASER performs quite well in classifying outliers from inliers.

Tables 3 and 4 show the quantitative results of all methods in the `freiburg1_xyz` and `freiburg2_xyz` sequences, respectively. We follow the standard evaluation protocol of visual

¹³https://github.com/UZ-SLAMLab/ORB_SLAM3

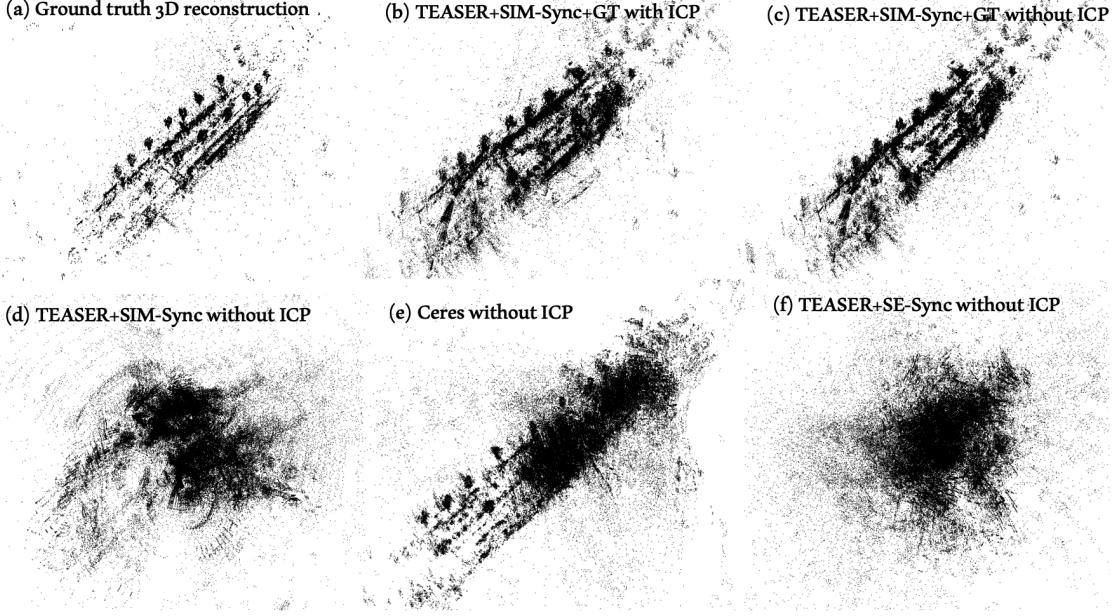


Figure 7: 3D reconstruction of a bird view street scene from `BAL ladyburg_318` dataset.

Table 2: Rotation error, translation error and suboptimality of `TEASER+SIM-Sync+GT`, `TEASER+SIM-Sync`, `TEASER+SE-Sync` and `Ceres` in `BAL` dataset `ladyburg_318` sequence.

	<code>TEASER+SIM-Sync+GT</code>	<code>TEASER+SIM-Sync</code> w/o Ceres	<code>TEASER+SIM-Sync</code> w Ceres	<code>TEASER+SE-Sync</code> w/o Ceres	<code>TEASER+SE-Sync</code> w Ceres	<code>Ceres</code>
Rotation Error [deg]	9.730	78.457	75.046	86.067	87.995	82.601
Translation Error [m]	0.334	2.299	3.307	2.600	5.858	4.776
Suboptimality	0.021	0.092	/	0.216	/	/

odometry for assessing pose accuracy, *i.e.*, Absolute Trajectory Error (ATE) and Relative Pose Error (RPE).¹⁴ Since the scale of the output of ORB-SLAM3 is unknown, we scale up the predicted translation to the scale of the groundtruth.¹⁵ We can see that `TEASER+SIM-Sync+GT` and `TEASER+SIM-Sync` with Essential Graph achieve comparable accuracy as ORB-SLAM3, while being simpler and more direct algorithms that also offer optimality guarantees. On the other hand, `TEASER+SIM-Sync+GT` and `TEASER+SIM-Sync` with the naive graph as in (29) show worse accuracy. This result implies that the essential graph is a better graph architecture than the naive graph in scene reconstruction.

We show the qualitative 3D reconstruction results of the `freiburg1_xyz` and `freiburg2_xyz` sequences in Fig. 9, using `TEASER+SIM-Sync` with learned depth. The reconstruction is formed by stacking the learned depth point clouds of all frames (after transformation to a common coordinate frame).

We can see that even with learned depth, the `TEASER+SIM-Sync` reconstruction achieves good accuracy.

¹⁴ATE quantifies the root-mean-square error between predicted camera positions and the groundtruth positions. RPE measures the relative pose disparity between pairs of adjacent frames, including both translation error (RPE-T) and rotational error (RPE-R).

¹⁵The factor is the median of norms of ground truth translation divided by the median of norms of predicted translation.

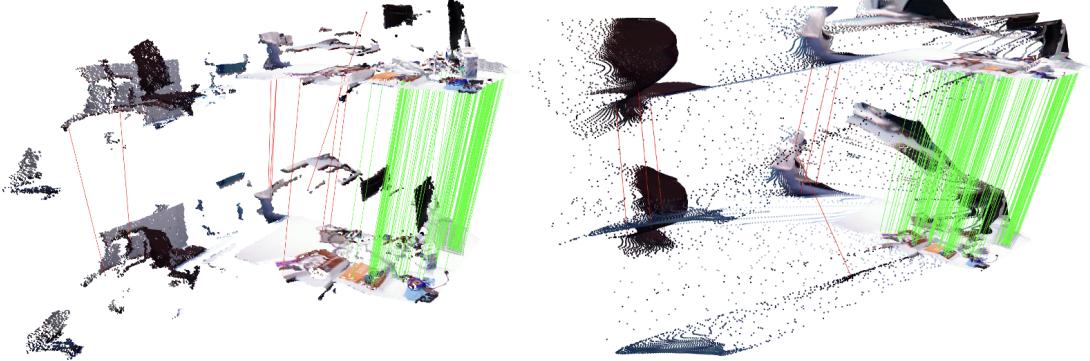


Figure 8: Illustration of outlier detection by TEASER. (Left: TEASER+SIM-Sync+GTDepth, Right: TEASER+SIM-Sync with learned depth)

Table 3: Rotation and translation error comparison for TEASER+SIM-Sync+GTDepth, TEASER+SIM-Sync and ORB-SLAM3 in the 200-pose `freiburg1_xyz` sequence of TUM. We set the regularization factor to $\lambda = 3$ for TEASER+SIM-Sync+GTDepth and TEASER+SIM-Sync without essential graph, and $\lambda = 0$ with essential graph (EG).

	TEASER+SIM-Sync+GTDepth		TEASER+SIM-Sync		ORB-SLAM3	ORB-SLAM3 (RGB-D)
	w/o EG	w EG	w/o EG	w EG		
ATE [m]	0.2453	0.1335	0.2003	0.0638	0.1081	0.0147
RPE Trans [m]	0.0167	0.0097	0.0163	0.0064	0.0068	0.0043
RPE Rot [deg]	1.2001	0.6476	1.3320	0.5659	0.5521	0.2813
Suboptimality	1.1966e-9	1.1966e-9	3.4336e-9	9.8312e-11	/	/

5 Conclusions

We introduced **SIM-Sync**, a certifiably optimal algorithm for camera trajectory estimation and scene reconstruction directly from image-level correspondences. With a pretrained depth prediction network, 2D image keypoints are lifted to 3D scaled point clouds, and **SIM-Sync** seeks to jointly synchronize camera poses and unknown (per-image) scaling factors to minimize the sum of Euclidean distances between matching points. By first developing a QCQP formulation and then applying semidefinite relaxation, **SIM-Sync** can achieve certifiable global optimality. We demonstrate the tightness, (outlier-)robustness, and practical usefulness of **SIM-Sync** in both simulated and real datasets. Future research aims to (i) speed up **SIM-Sync** by exploiting low-rankness of the optimal SDP solutions [24, 35], and (ii) leverage the 3D reconstruction from **SIM-Sync** to improve the imperfect depth prediction from the pretrained model.

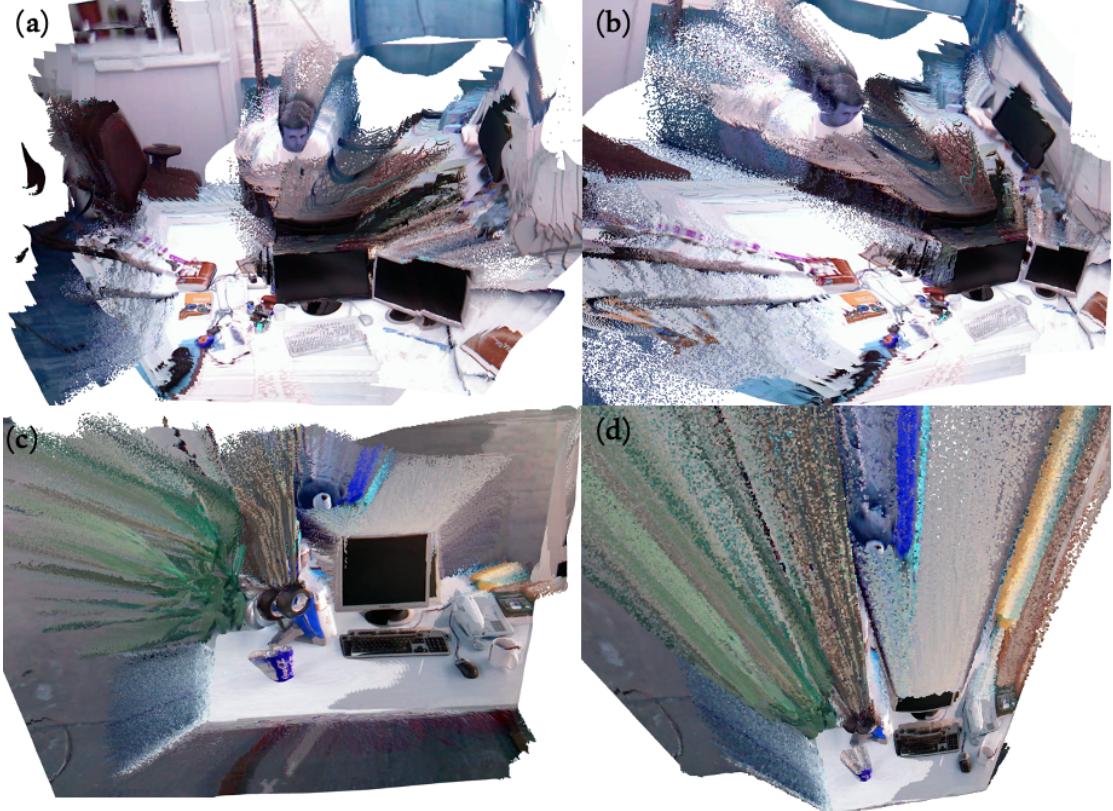


Figure 9: 3D reconstruction of **TEASER+SIM-Sync** with learned depth in **TUM**. (a) Forward view reconstruction and (b) Side view reconstruction in the **freiburg1_xyz** sequence. (c) Forward view reconstruction and (d) Bird view reconstruction in the **freiburg2_xyz** sequence.

Table 4: Rotation and translation error comparison for **TEASER+SIM-Sync+GTDepth**, **TEASER+SIM-Sync** and **ORB-SLAM3** in the 200-pose **freiburg2_xyz** sequence of **TUM**. Regularization factor same as Table 3.

	TEASER+SIM-Sync+GTDepth w/o EG	TEASER+SIM-Sync+GTDepth w EG	TEASER+SIM-Sync w/o EG	TEASER+SIM-Sync w EG	ORB-SLAM3	ORB-SLAM3 (RGB-D)
ATE [m]	0.2089	0.1237	0.3190	0.0358	0.0246	0.1023
RPE Trans [m]	0.0072	0.0017	0.0030	0.0012	0.0007	0.0017
RPE Rot [deg]	0.3586	0.2349	0.9982	0.2240	0.2155	0.1739
Suboptimality	9.8904e-10	1.7527e-9	2.7516e-9	6.1298e-10	/	/

Appendix

A Proof of Proposition 1

Proof. Let

$$P_i = \begin{bmatrix} \sqrt{w_{ij,1}} d_{i,1} \tilde{p}_{i,1} & \cdots & \sqrt{w_{ij,n_{ij}}} d_{i,n_{ij}} \tilde{p}_{i,n_{ij}} \end{bmatrix} \in \mathbb{R}^{3 \times n_{ij}}, \quad i = 1, \dots, N,$$

$$W_{ij} = [\sqrt{w_{ij,1}}, \dots, \sqrt{w_{ij,n_{ij}}}]^\top \in \mathbb{R}^{n_{ij}}, \quad (i, j) \in \mathcal{E},$$

For any t_i and $\text{vec}(s_i R_i)$, define the selection matrices

$$t_i = Z_i^t t, \quad \text{vec}(s_i R_i) = Z_i^R r, \quad Z_i^t \in \mathbb{R}^{3 \times 3N}, \quad Z_i^R \in \mathbb{R}^{9 \times 9N}.$$

We then write the objective in (SIM-Sync) as

$$\begin{aligned} L(t, r) &= \sum_{(i,j) \in \mathcal{E}} \| (s_i R_i) P_i + t_i W_{ij}^\top - (s_j R_j) P_j - t_j W_{ij}^\top \|_F^2 \\ &= \sum_{(i,j) \in \mathcal{E}} \| (P_i^\top \otimes \mathbf{I}_3) \text{vec}(s_i R_i) + (W_{ij} \otimes \mathbf{I}_3) t_i - (P_j^\top \otimes \mathbf{I}_3) \text{vec}(s_j R_j) - (W_{ij} \otimes \mathbf{I}_3) t_j \|_F^2 \\ &= \sum_{(i,j) \in \mathcal{E}} \left\| \underbrace{[(P_i^\top \otimes \mathbf{I}_3) Z_i^R - (P_j^\top \otimes \mathbf{I}_3) Z_j^R]}_{=: C_{ij}^R} r + \underbrace{[(W_{ij} \otimes \mathbf{I}_3) Z_i^t - (W_{ij} \otimes \mathbf{I}_3) Z_j^t]}_{=: C_{ij}^t} t \right\|^2 \end{aligned} \quad (\text{A30})$$

Note that Z_i^R and Z_i^t can actually be decomposed as

$$\begin{aligned} Z_i^R &= e_i^\top \otimes \mathbf{I}_3 \otimes \mathbf{I}_3 \\ Z_i^t &= e_i^\top \otimes \mathbf{I}_3 \end{aligned}$$

where $e_i \in \mathbb{R}^N$ is the basis vector in \mathbb{R}^N where the i^{th} entry is 1 while other entries are all 0's. Plug in C_{ij}^R and C_{ij}^t , we obtain:

$$\begin{aligned} C_{ij}^R &= (P_i^\top \otimes \mathbf{I}_3)(e_i^\top \otimes \mathbf{I}_3 \otimes \mathbf{I}_3) - (P_j^\top \otimes \mathbf{I}_3)(e_j^\top \otimes \mathbf{I}_3 \otimes \mathbf{I}_3) \\ &= (P_i^\top(e_i^\top \otimes \mathbf{I}_3)) \otimes \mathbf{I}_3 - (P_j^\top(e_j^\top \otimes \mathbf{I}_3)) \otimes \mathbf{I}_3 \\ &= ((1 \otimes P_i^\top)(e_i^\top \otimes \mathbf{I}_3)) \otimes \mathbf{I}_3 - ((1 \otimes P_j^\top)(e_j^\top \otimes \mathbf{I}_3)) \otimes \mathbf{I}_3 \\ &= (e_i^\top \otimes P_i^\top) \otimes \mathbf{I}_3 - (e_j^\top \otimes P_j^\top) \otimes \mathbf{I}_3 \\ &= (e_i^\top \otimes P_i^\top - e_j^\top \otimes P_j^\top) \otimes \mathbf{I}_3 \\ C_{ij}^t &= (W_{ij} \otimes \mathbf{I}_3)(e_i^\top \otimes \mathbf{I}_3) - (W_{ij} \otimes \mathbf{I}_3)(e_j^\top \otimes \mathbf{I}_3) \\ &= (W_{ij} e_i^\top) \otimes \mathbf{I}_3 - (W_{ij} e_j^\top) \otimes \mathbf{I}_3 \\ &= (W_{ij} e_i^\top - W_{ij} e_j^\top) \otimes \mathbf{I}_3 \end{aligned}$$

Plug C_{ij}^R and C_{ij}^t back to (A30), we then simplify the objective as:

$$\begin{aligned} L(t, r) &= \sum_{(i,j) \in \mathcal{E}} \| ((e_i^\top \otimes P_i^\top - e_j^\top \otimes P_j^\top) \otimes \mathbf{I}_3) r + ((W_{ij} e_i^\top - W_{ij} e_j^\top) \otimes \mathbf{I}_3) t \|_F^2 \\ &= \begin{bmatrix} t \\ r \end{bmatrix}^\top \left(\begin{bmatrix} Q_1 & V^\top \\ V & Q_2 \end{bmatrix} \otimes \mathbf{I}_3 \right) \begin{bmatrix} t \\ r \end{bmatrix} \\ &= t^\top (Q_1 \otimes \mathbf{I}_3) t + 2r^\top (V \otimes \mathbf{I}_3) t + r^\top (Q_2 \otimes \mathbf{I}_3) r \end{aligned} \quad (\text{A31})$$

where

$$\begin{aligned} Q_1 &= \sum_{(i,j) \in \mathcal{E}} (W_{ij}e_i^\top - W_{ij}e_j^\top)^\top (W_{ij}e_i^\top - W_{ij}e_j^\top) \in \mathbb{R}^{N \times N} \\ Q_2 &= \sum_{(i,j) \in \mathcal{E}} (e_i^\top \otimes P_i^\top - e_j^\top \otimes P_j^\top)^\top (e_i^\top \otimes P_i^\top - e_j^\top \otimes P_j^\top) \in \mathbb{R}^{3N \times 3N} \\ V &= \sum_{(i,j) \in \mathcal{E}} (e_i^\top \otimes P_i^\top - e_j^\top \otimes P_j^\top)^\top (W_{ij}e_i^\top - W_{ij}e_j^\top) \in \mathbb{R}^{3N \times N} \end{aligned} \quad (\text{A32})$$

concluding the proof. \square

B Proof of Proposition 2

Proof. To represent t as a function of r , simply take the gradient of (3) w.r.t. t and setting it to zero, we obtain

$$(Q_1 \otimes \mathbf{I}_3)t = (V \otimes \mathbf{I}_3)^\top r. \quad (\text{A33})$$

Proposition A6 (Laplacian Representation of Q_1). Q_1 can be represented as:

$$Q_1 = L(\mathcal{G}) \quad (\text{A34})$$

where $L(\mathcal{G})$ is the Laplacian of \mathcal{G} .

Proof. Note that \mathcal{G} is a weighted undirected graph. Calling $\delta(i)$ for the set of edges incident to a vertex v and $n_e = n_{ij}, w_{e,k} = w_{ij,k}$ for $e = (i, j)$, the Laplacian of \mathcal{G} is:

$$L(\mathcal{G})_{ij} = \begin{cases} \sum_{e \in \delta(i)} \sum_{k=1}^{n_e} \sqrt{w_{e,k}} & \text{if } i = j, \\ -\sum_{k=1}^{n_{ij}} \sqrt{w_{ij,k}} & \text{if } (i, j) \in \mathcal{E}, \\ 0 & \text{if } (i, j) \notin \mathcal{E}. \end{cases} \quad (\text{A35})$$

On the other hand, by expanding (A32) and compare it with (A35), we finish the proof. \square

Note that $\text{rank}(L(\mathcal{G})) = N - 1$, then $\text{rank}(Q_1 \otimes \mathbf{I}_3) = \text{rank}(L(\mathcal{G})) \text{rank}(\mathbf{I}_3)$ by Proposition A6. Thus $Q_1 \otimes \mathbf{I}_3$ is not invertible. As we fixed $s_1 = 1, R_1 = \mathbf{I}_3, t_1 = 0$, t has a unique solution. Calling $\bar{t} = [t_2; \dots; t_N] \in \mathbb{R}^{3N-3}$ and $\bar{Q}_1 = [c_2; \dots; c_N] \in \mathbb{R}^{N \times (N-1)}$ where c_i is the i th column of Q_1 . We have

$$(\bar{Q}_1 \otimes \mathbf{I}_3)\bar{t} = (V \otimes \mathbf{I}_3)^\top r \quad (\text{A36})$$

Since $\text{rank}(L(\mathcal{G})) = N - 1$ and $\sum_{i=1, \dots, N} c_i = \mathbf{0}_N$, then $\text{span}(c_1, \dots, c_n) = \text{span}(c_2, \dots, c_n)$ and $\text{rank}(\bar{Q}_1) = N - 1$ which implies that \bar{Q}_1 has full column rank. Hence, by taking inverse and rearrange, we obtain

$$\bar{t} = (\bar{A} \otimes \mathbf{I}_3)r \quad (\text{A37})$$

where

$$\bar{A} = -(\bar{Q}_1^\top \bar{Q}_1)^{-1} \bar{Q}_1^\top V^\top \quad (\text{A38})$$

Together with $t_1 = 0$,

$$t = (A \otimes \mathbf{I}_3)r \quad (\text{A39})$$

with

$$A = \begin{bmatrix} \mathbf{0}_{1 \times 3N} \\ -(Q_1^\top Q_1)^{-1} Q_1^\top V^\top \end{bmatrix} \in \mathbb{R}^{N \times 3N} \quad (\text{A40})$$

Now we have a closed form solution of t . Plug in the solution of t into (A31), we obtain:

$$L(r) = r^\top ((A^\top Q_1 A + VA + A^\top V^\top + Q_2) \otimes \mathbf{I}_3) r \quad (\text{A41})$$

Note that

$$r = \text{vec}(R) \quad (\text{A42})$$

Then $L(r)$ is equivalent to

$$L(R) = \text{vec}(R)^\top ((A^\top Q_1 A + VA + A^\top V^\top + Q_2) \otimes \mathbf{I}_3) \text{vec}(R) \quad (\text{A43})$$

Rewrite (A43) in a more compact matricized form gives:

$$\rho^* = \min_R \text{tr}(QR^\top R) \quad (\text{A44})$$

$$Q := A^\top Q_1 A + VA + A^\top V^\top + Q_2 \in \mathbb{S}^{3N}, \quad (\text{A45})$$

concluding the proof. \square

C Proof of Proposition 3

Proof. We first prove (14).

“ \implies ”: If $\bar{R} \in \mathfrak{so}(3)$, then $\bar{R} = sR$ for some $s \geq 0$ and $R \in O(3)$. Therefore,

$$\begin{bmatrix} c_1^\top c_1 & c_1^\top c_2 & c_1^\top c_3 \\ * & c_2^\top c_2 & c_2^\top c_3 \\ * & * & c_3^\top c_3 \end{bmatrix} = \bar{R}^\top \bar{R} = (sR)^\top (sR) = s^2 R^\top R = s^2 \mathbf{I}_3,$$

which shows the quadratic constraints in (14) must hold.

“ \impliedby ”: Suppose the quadratic constraints in (14) hold. (i) If $c_i^\top c_i = 0, i = 1, 2, 3$, then $\bar{R} = \mathbf{0} \in \mathfrak{so}(3)$ trivially holds. (ii) If $c_i^\top c_i = \alpha > 0, i = 1, 2, 3$, then \bar{R} can be written as

$$\bar{R} = \sqrt{\alpha} \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix},$$

where $u_i, i = 1, 2, 3$ are unit vectors. However,

$$c_i^\top c_j = 0 \iff \alpha u_i^\top u_j = 0 \iff u_i^\top u_j = 0, \quad \forall i \neq j,$$

which means (u_1, u_2, u_3) are orthogonal to each other and therefore $[u_1, u_2, u_3] \in O(3)$.

We then show that if an optimal solution of (QCQP) satisfies (15), then it must be an optimizer of (9). First note that (QCQP) is a relaxation of (9) and $\rho_{\text{QCQP}}^* \leq \rho^*$. This is because any feasible solution to (9) must also be feasible to (QCQP), due to the fact that any scaled rotation must also lie in $\mathfrak{so}(3)$ by its definition (13). However, if R^* satisfies (15), then we claim that R^* is also feasible to (9) (hence also optimal to (9)), *i.e.*, each \bar{R}_i^* can be written as a scaled rotation. In fact, $\bar{R}_i^* \in \mathfrak{so}(3)$ implies $\bar{R}_i^* = s_i^* R_i^*$ for some $s_i^* \geq 0$ and $R_i^* \in O(3)$. However,

$$\det \bar{R}_i^* = (s_i^*)^3 \det R_i^* > 0$$

implies $s_i^* \neq 0$ and $R_i^* \in SO(3)$, because any matrix in $O(3)$ is either a rotation (with +1 determinant) or a reflection (with -1 determinant). \square

D Weighted Scaled Point Cloud Registration

Consider the optimization problem where $\{(Y_i, X_i), i = 1, \dots, n\}$ are matching scaled point clouds and we seek the best similarity transformation between them

$$\min_{s>0, R \in \text{SO}(3), t \in \mathbb{R}^3} \sum_{i=1, \dots, n}^n w_i \|sRX_i + t - Y_i\|^2, \quad (\text{A46})$$

where $w_i, i = 1, \dots, n$ are known weights. We will show that problem (A46) admits a closed-form solution.

Let the objective function be

$$f = \sum_{i=1}^n w_i \|sRX_i + t - Y_i\|^2. \quad (\text{A47})$$

Firstly, take the derivative with respect to t :

$$\frac{\partial f}{\partial t} = 2 \sum_i w_i (sRX_i + t - Y_i) \quad (\text{A48})$$

Set it to zero, we obtain:

$$t = \mu_y - sR\mu_x \quad (\text{A49})$$

where

$$\mu_y = \frac{\sum_i w_i y_i}{\sum_i w_i} \quad (\text{A50})$$

$$\mu_x = \frac{\sum_i w_i x_i}{\sum_i w_i} \quad (\text{A51})$$

Substitute t with (A49) in original optimization, we obtain:

$$f = \sum_{i=1}^n w_i \|sR[X_i - \mu_x] - (Y_i - \mu_y)\|^2 \quad (\text{A52})$$

$$= \sum_{i=1}^n \|\sqrt{w_i} sR[X_i - \mu_x] - \sqrt{w_i} (Y_i - \mu_y)\|^2 \quad (\text{A53})$$

$$= \sum_{i=1}^n \|R[\sqrt{w_i} s(X_i - \mu_x)] - \sqrt{w_i} (Y_i - \mu_y)\|^2 \quad (\text{A54})$$

Let

$$A = [\sqrt{w_1}(Y_1 - \mu_y) \quad \cdots \quad \sqrt{w_n}(Y_n - \mu_y)] \in \mathbb{R}^{3 \times n}, \quad (\text{A55})$$

$$B = [\sqrt{w_1}(X_1 - \mu_x) \quad \cdots \quad \sqrt{w_n}(X_n - \mu_x)] \in \mathbb{R}^{3 \times n} \quad (\text{A56})$$

Then f is simplified as:

$$f = \|R \cdot (sB) - A\|^2 \quad (\text{A57})$$

Now solve R (taken s as known), the optimization is Wabha's problem. When $\text{rank}(AB^\top) \geq 2$, the optimal rotation matrix R can be uniquely determined as a function of s

$$g = \min_R \|R \cdot (sB) - A\|^2 = \|A\|^2 + s^2 \|B\|^2 - 2\text{tr}(sDS) \quad (\text{A58})$$

where AB^\top can be computed as UDV^\top by singular value decomposition and

$$S = \begin{cases} \mathbf{I}_3 & \text{if } \det(U)\det(V) = 1, \\ \text{diag}(1, \dots, 1, -1) & \text{if } \det(U)\det(V) = -1. \end{cases} \quad (\text{A59})$$

And the optimizer R is:

$$R = USV^\top \quad (\text{A60})$$

Now take derivative of g with respect to s , we get:

$$\frac{\partial g}{\partial s} = 2s \|B\|^2 - 2\text{tr}(DS) \quad (\text{A61})$$

To minimize g , we have

$$s = \frac{\text{tr}(DS)}{\|B\|^2}. \quad (\text{A62})$$

In summary, we first compute s by (A62), and then R according to (A60), finally t from (A49).

E Noise Analysis

There are several places that involve noise analysis in the paper. (i) **SE-Sync** needs uncertainty estimation of the solution returned by Arun's method. We provide analysis in Section E.1. (ii) In **SIM-Sync-GNC**, GNC needs a noise bound β (*cf.* (27)). We provide analysis in Section E.2. (iii) In **GNC+SIM-Sync** and **TEASER+SIM-Sync**, GNC and TEASER need edge-wise noise bounds (*cf.* (28)). We provide analysis in Section E.3.

E.1 Covariance Estimation of Arun's Method

Consider $(R_i, t_i), (R_j, t_j) \in \text{SE}(3)$ as camera poses in frame i and j respectively. For point clouds P in world frame, we generate point clouds P_i and P_j by corrupting noises $\epsilon_i \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_3)$ and $\epsilon_j \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_3)$ (assuming ϵ_i and ϵ_j are independent):

$$P_i = R_i P + t_i + \epsilon_i \quad (\text{A63})$$

$$P_j = R_j P + t_j + \epsilon_j \quad (\text{A64})$$

Remove variable P , we obtain:

$$P_i = R_i R_j^\top P_j + t_i - R_i R_j^\top t_j + \epsilon_i - R_i R_j^\top \epsilon_j \quad (\text{A65})$$

Reparametrize the variables:

$$R_{ij} = R_i R_j^\top \in \text{SO}(3) \quad (\text{A66})$$

$$t_{ij} = t_i - R_j R_j^\top t_j \in \mathbb{R}^3 \quad (\text{A67})$$

$$\epsilon_{ij} = \epsilon_i - R_i R_j^\top \epsilon_j \sim \mathcal{N}(0, 2\sigma^2 \mathbf{I}_3) \quad (\text{A68})$$

We obtain:

$$P_i = R_{ij} P_j + t_{ij} + \epsilon_{ij} \quad (\text{A69})$$

Arun's method estimates R_{ij} and t_{ij} . We want to estimate the uncertainty of the solution computed by Arun's method.

Our strategy is to firstly utilize Arun's method to find the optimal solution for noise free (A69), and then form a Maximum Likelihood Estimator by disturbing the rotation and translation around optimizer. Denote the optimizer of R_{ij} and t_{ij} in Arun's method as R_{ij}^* and t_{ij}^* . Then we rewrite (A69) as

$$P_i = R_{ij}^* \exp(\hat{\omega}_{ij}) P_j + t_{ij}^* + \delta_{ij} + \epsilon_{ij} \quad (\text{A70})$$

where $\hat{\omega}$ is a skew-symmetric matrix in the Lie algebra $\mathfrak{so}(3)$ and $\delta_{ij} \in \mathbb{R}^3$. In (A70), we reparameterize the rotation matrix by compositioning on a rotation action and a logarithm map onto the Tangent Space $T_{R_{ij}^*} \text{SO}(3)$. The exponential map is used to map $\hat{\omega}$ to a 3D rotation matrix R in the Lie group $\text{SO}(3)$:

$$R = \exp(\hat{\omega}) \quad (\text{A71})$$

The hat operator maps this vector to a skew-symmetric matrix $\hat{\omega}$ in $\mathfrak{so}(3)$, given by:

$$\hat{\omega} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (\text{A72})$$

Specifically, for the k -th measurement, (A70) is:

$$P_{ik} = R_{ij}^* \exp(\hat{\omega}_{ij}) P_{jk} + t_{ij}^* + \delta_{ij} + \epsilon_{ijk} \quad (\text{A73})$$

where $\epsilon_{ijk} \sim \mathcal{N}(0, \Sigma_k)$ and further assume that ϵ_{ijk} are i.i.d for $k = 1, \dots, n_{ij}$. With $P_{ik}, P_{jk}, R_{ij}^*, t_{ij}^*$ known for all $k = 1, \dots, n_{ij}$, rewrite (A70) as optimization problem on $x_{ij} = (\omega_{ij}, \delta_{ij})$:

$$(\omega_{ij}^*, \delta_{ij}^*) = \arg \min \sum_{k=1}^{n_{ij}} \left\| \underbrace{R_{ij}^* \exp(\hat{\omega}_{ij}) P_{jk} + t_{ij}^* + \delta_{ij} - P_{ik}}_{=:f(x_{ij})} \right\|_{\Sigma_k^{-1}} \quad (\text{A74})$$

Since $f(x_{ij})$ is nonlinear function of x_{ij} and the distribution of x_{ij} can be non-Gaussian and arbitrarily complex, we can only compute a Cramer-Rao lower bound on the posterior distribution by linearizing $f(x_{ij})$ around optimal x_{ij}^* . By the optimality of R_{ij}^*, t_{ij}^* , the optimizer is $x^* = (\omega_{ij}, \delta_{ij}) = 0$.

$$(\omega_{ij}^*, \delta_{ij}^*) = \arg \min \sum_{k=1}^{n_{ij}} \left\| \underbrace{R_{ij}^* \exp(\hat{\omega}_{ij}) P_{jk} + t_{ij}^* + \delta_{ij} - P_{ik}}_{=:f(x_{ij})} \right\|_{\Sigma_k^{-1}} \quad (\text{A75})$$

Note that

$$f(x_{ij}) = f(0) + \frac{df}{dx_{ij}} \Big|_{x_{ij}=0} (x_{ij} - 0) \quad (\text{A76})$$

$$= R_{ij}^* P_{jk} + t_{ij}^* + \frac{df}{dx_{ij}} \Big|_{x_{ij}=0} x_{ij} \quad (\text{A77})$$

where

$$\frac{df}{dx_{ij}} \Big|_{x_{ij}=0} = \begin{bmatrix} -R_{ij}^* \hat{P}_{jk} & \mathbf{I}_3 \end{bmatrix} \quad (\text{A78})$$

Plug $f(x_{ij})$ into (A74). We obtain:

$$(\omega_{ij}^*, \delta_{ij}^*) \approx \arg \min \sum_{k=1}^{n_{ij}} \left\| \underbrace{\frac{df}{dx_{ij}} \Big|_{x_{ij}=0}}_{=:H_k} x_{ij} - \underbrace{(P_{ik} - R_{ij}^* P_{jk} - t_{ij}^*)}_{=:y_k} \right\|_{\Sigma_k^{-1}} \quad (\text{A79})$$

$$= \arg \min \sum_{k=1}^{n_{ij}} \|H_k x_{ij} - \tilde{y}_k\|_{\Sigma_k^{-1}} \quad (\text{A80})$$

This forms a linear psedo-measurement equation:

$$y_k = H_k x_{ij} + \epsilon_k \quad (\text{A81})$$

with $\epsilon_k \sim \mathcal{N}(0, \Sigma_k) = \mathcal{N}(0, 2\sigma^2 \mathbf{I}_3)$. With the assumption that ϵ_{ijk} are i.i.d for $k = 1, \dots, n_{ij}$, we obtain the posterior covariance of x_{ij} :

$$\begin{aligned} \Sigma_{x_{ij}} &= \text{Cov}(\omega_{ij}, t_{ij}) \\ &= (\sum_{k=1}^{n_{ij}} H_k^T \Sigma_k^{-1} H_k)^{-1} \\ &= \sigma^2 (\sum_{k=1}^{n_{ij}} H_k^T H_k)^{-1} \end{aligned} \quad (\text{A82})$$

We feed the covariance matrix in (A82) to SE-Sync.

E.2 Noise Bound for SIM-Sync-GNC

Consider $(R_i, t_i), (R_j, t_j) \in \text{SE}(3)$ as camera poses in frame i and j respectively. For point cloud $P \in \mathbb{R}^3$ in world frame, we generate point clouds P_i and P_j by corrupting noises $\epsilon_i \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_3)$ and $\epsilon_j \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_3)$:

$$P_i = \frac{1}{s_i} (R_i P + t_i + \epsilon_i) \quad (\text{A83})$$

$$P_j = \frac{1}{s_j} (R_j P + t_j + \epsilon_j) \quad (\text{A84})$$

Remove variable P , we obtain:

$$s_i R_i^T P_i - R_i^T t_i - (s_j R_j^T P_j - R_j^T t_j) = R_i^T \epsilon_i - R_j^T \epsilon_j \quad (\text{A85})$$

$(R_i, t_i), (R_j, t_j) \in \text{SE}(3)$ are transformations from world frame to camera frame i and j . If we represent the above equation using $(R'_i, t'_i), (R'_j, t'_j) \in \text{SE}(3)$ that are transformations from camera frame i and j to world frame. We have:

$$s_i R'_i P_i + t'_i - (s_j R'_j P_j + t'_j) = R'_i \epsilon_i - R'_j \epsilon_j \sim \mathcal{N}(0, 2\sigma^2 \mathbf{I}_3) \quad (\text{A86})$$

Then we obtain:

$$\|s_i R'_i P_i + t'_i - (s_j R'_j P_j + t'_j)\|_{\frac{1}{2\sigma^2} \mathbf{I}_3}^2 \sim \chi_{n=3}^2 \quad (\text{A87})$$

For a probability value of 0.9999 with 3 degrees of freedom, a threshold value of 21.11 is computed from the Chi-square distribution table. Statistically, it suggests that we have 99.99% confidence that any samples that fall outside of this threshold are outliers. Together with covariance normalization factor $\sqrt{2\sigma^2}$, $\beta = \sqrt{21.11} \cdot \sqrt{2\sigma}$.

E.3 Noise Bound for TEASER and GNC

Consider $(R_i, t_i), (R_j, t_j) \in \text{SE}(3)$ as camera poses in frame i and j respectively. For point clouds P in world frame, we generate point clouds P_i and P_j by corrupting noises $\epsilon_i \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_3)$ and $\epsilon_j \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_3)$:

$$P_i = \frac{1}{s_i} (R_i P + t_i + \epsilon_i) \quad (\text{A88})$$

$$P_j = \frac{1}{s_j} (R_j P + t_j + \epsilon_j) \quad (\text{A89})$$

Remove variable P , we obtain:

$$P_i = \frac{s_j}{s_i} R_i R_j^\top P_j + \frac{1}{s_i} t_i - \frac{1}{s_i} R_i R_j^\top t_j + \frac{1}{s_i} (\epsilon_i - R_i R_j^\top \epsilon_j) \quad (\text{A90})$$

We obtain:

$$P_i = s_{ij} R_{ij} P_j + t_{ij} + \epsilon_{ij} \quad (\text{A91})$$

by reparametrizing the variables:

$$s_{ij} = \frac{s_j}{s_i} \quad (\text{A92})$$

$$R_{ij} = R_i R_j^\top \in \text{SO}(3) \quad (\text{A93})$$

$$t_{ij} = \frac{1}{s_i} t_i - \frac{1}{s_i} R_j R_j^\top t_j \in \mathbb{R}^3 \quad (\text{A94})$$

$$\epsilon_{ij} = \frac{1}{s_i} (\epsilon_i - R_i R_j^\top \epsilon_j) \sim \mathcal{N}(0, \frac{2\sigma^2}{s_i^2} \mathbf{I}_3) \quad (\text{A95})$$

Then we obtain:

$$\|s_{ij} R_{ij} P_j + t_{ij} - P_i\|_{\frac{s_i^2}{2\sigma^2} \mathbf{I}_3}^2 \sim \chi_{n=3}^2 \quad (\text{A96})$$

For a probability value of 0.9999 with 3 degrees of freedom, a threshold value of 21.11 is computed from the Chi-square distribution table. Statistically, it suggests that we have 99.99% confidence that any samples that fall outside of this threshold are outliers. Together with covariance normalization factor $\sqrt{\frac{2\sigma^2}{s_i^2}}$, $\beta = \sqrt{21.11} \cdot \frac{\sqrt{2}\sigma}{s_i}$.

References

- [1] Sameer Agarwal, Noah Snavely, Steven M Seitz, and Richard Szeliski. Bundle adjustment in the large. In *European Conf. on Computer Vision (ECCV)*, pages 29–42. Springer, 2010. [2](#), [3](#), [8](#), [12](#)
- [2] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. Ceres Solver, 3 2022. URL <https://github.com/ceres-solver/ceres-solver>. [2](#)
- [3] K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on pattern analysis and machine intelligence*, (5):698–700, 1987. [8](#)
- [4] Xiaowei Bao, Nikolaos V Sahinidis, and Mohit Tawarmalani. Semidefinite relaxations for quadratically constrained quadratic programming: A review and comparisons. *Mathematical programming*, 129:129–157, 2011. [3](#)

- [5] Timothy D Barfoot. *State estimation for robotics*. Cambridge University Press, 2017. [2](#)
- [6] Reiner Birk, Diana Wofk, and Matthias Müller. Midas v3.1 – a model zoo for robust monocular relative depth estimation. *arXiv preprint arXiv:2307.14460*, 2023. [13](#)
- [7] Florian Bugarin, Didier Henrion, and Jean Bernard Lasserre. Minimizing the sum of many rational functions. *Mathematical Programming Computation*, 8(1):83–111, 2016. [2](#)
- [8] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021. [3](#), [14](#)
- [9] Luca Carlone, David M Rosen, Giuseppe Calafiore, John J Leonard, and Frank Dellaert. Lagrangian duality in 3d slam: Verification techniques and optimal solutions. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 125–132. IEEE, 2015. [2](#)
- [10] Luca Carlone, Giuseppe C Calafiore, Carlo Tommolillo, and Frank Dellaert. Planar pose graph optimization: Duality, optimal solutions, and verification. *IEEE Trans. Robotics*, 32(3):545–565, 2016. [2](#)
- [11] Kunal N Chaudhury, Yuehaw Khoo, and Amit Singer. Global registration of multiple point clouds using semidefinite programming. *SIAM Journal on Optimization*, 25(1):468–501, 2015. [2](#)
- [12] Frank Dellaert and GTSAM Contributors. borglab/gtsam, May 2022. URL <https://github.com/borglab/gtsam>. [2](#)
- [13] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. [2](#)
- [14] Connor Holmes and Timothy D Barfoot. An efficient global optimality certificate for landmark-based slam. *IEEE Robotics and Automation Letters*, 8(3):1539–1546, 2023. [2](#)
- [15] José Pedro Iglesias, Carl Olsson, and Fredrik Kahl. Global optimality for point set registration using semidefinite programming. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 8287–8295, 2020. [2](#), [8](#)
- [16] Laurent Kneip, Hongdong Li, and Yongduek Seo. Upnp: An optimal o(n) solution to the absolute pose problem with universal applicability. In *European Conf. on Computer Vision (ECCV)*, pages 127–142. Springer, 2014. [2](#)
- [17] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1611–1621, 2021. [2](#)
- [18] Jean B Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on optimization*, 11(3):796–817, 2001. [5](#)
- [19] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999. [14](#)
- [20] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Transactions on Graphics (ToG)*, 39(4):71–1, 2020. [2](#)

- [21] Nathaniel Merrill, Patrick Geneva, and Saimouli Katragadda Chuchu Chen Guoquan Huang. Fast monocular visual-inertial initialization leveraging learned single-view depth. In *Robotics: Science and Systems (RSS)*, 2023. 2
- [22] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):756–770, 2004. 2
- [23] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022. 2, 13
- [24] David M Rosen, Luca Carlone, Afonso S Bandeira, and John J Leonard. Se-sync: A certifiably correct algorithm for synchronization over the special euclidean group. *The International Journal of Robotics Research*, 38(2-3):95–125, 2019. 2, 4, 5, 6, 8, 16
- [25] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2
- [26] Gerald Schweighofer and Axel Pinz. Fast and globally convergent structure and motion estimation for general camera models. In *British Machine Vision Conf. (BMVC)*, pages 147–156, 2006. 2
- [27] Hauke Strasdat, J Montiel, and Andrew J Davison. Scale drift-aware large scale monocular slam. *Robotics: science and Systems VI*, 2(3):7, 2010. 3
- [28] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012. 3, 4, 8
- [29] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022. 2
- [30] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(04):376–380, 1991. 12
- [31] Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning feature descriptors using camera pose supervision. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I* 16, pages 757–774. Springer, 2020. 14
- [32] Heng Yang and Luca Carlone. Certifiably optimal outlier-robust geometric perception: Semidefinite relaxations and scalable global optimization. *IEEE transactions on pattern analysis and machine intelligence*, 45(3):2816–2834, 2022. 5
- [33] Heng Yang, Pasquale Antonante, Vasileios Tzoumas, and Luca Carlone. Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection. *IEEE Robotics and Automation Letters*, 5(2):1127–1134, 2020. 3, 11
- [34] Heng Yang, Jingnan Shi, and Luca Carlone. Teaser: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 37(2):314–333, 2020. 3, 12
- [35] Heng Yang, Ling Liang, Luca Carlone, and Kim-Chuan Toh. An inexact projected gradient method with rounding and lifting by nonlinear programming for solving rank-one semidefinite relaxation of polynomial optimization. *Mathematical Programming*, 201(1-2):409–472, 2023. 16

- [36] Zhoutong Zhang, Forrester Cole, Zhengqi Li, Michael Rubinstein, Noah Snavely, and William T Freeman. Structure and motion from casual videos. In *European Conf. on Computer Vision (ECCV)*, pages 20–37. Springer, 2022. [2](#)