

TEASER: Fast and Certifiable Point Cloud Registration

Heng Yang, Jingnan Shi, Luca Carlone

Abstract—We propose the first fast and certifiable algorithm for the registration of two sets of 3D points in the presence of large amounts of outlier correspondences. A *certifiable algorithm* is one that attempts to solve an intractable optimization problem (e.g., robust estimation with outliers) and provides readily checkable conditions to verify if the returned solution is optimal (e.g., if the algorithm produced the most accurate estimate in the face of outliers) or bound its sub-optimality or accuracy.

Towards this goal, we first reformulate the registration problem using a *Truncated Least Squares* (TLS) cost that makes the estimation insensitive to a large fraction of spurious correspondences. Then, we provide a general graph-theoretic framework to decouple scale, rotation, and translation estimation, which allows solving in cascade for the three transformations. Despite the fact that each subproblem (scale, rotation, and translation estimation) is still non-convex and combinatorial in nature, we show that (i) TLS scale and (component-wise) translation estimation can be solved in polynomial time via an *adaptive voting* scheme, (ii) TLS rotation estimation can be relaxed to a semidefinite program (SDP) and the relaxation is tight, even in the presence of extreme outlier rates. We name the resulting algorithm TEASER (*Truncated least squares Estimation And SEmidefinite Relaxation*). While solving large SDP relaxations is typically slow, we develop a second certifiable algorithm, named TEASER++, that circumvents the need to solve an SDP and runs in *milliseconds*.

For both algorithms, we provide theoretical bounds on the estimation errors, which are the first of their kind for robust registration problems. Moreover, we test their performance on standard benchmarks, object detection datasets, and the 3DMatch scan matching dataset, and show that (i) both algorithms dominate the state of the art (e.g., RANSAC, branch-&-bound, heuristics) and are robust to more than 99% outliers, (ii) TEASER++ can run in milliseconds and it is currently the fastest robust registration algorithm, (iii) TEASER++ is so robust it can also solve problems without correspondences (e.g., hypothesizing all-to-all correspondences) where it largely outperforms ICP. We release a fast open-source C++ implementation of TEASER++.

Index Terms—3D registration, scan matching, point cloud alignment, robust estimation, certifiable algorithms, outliers-robust estimation, object pose estimation, 3D robot vision.

SUPPLEMENTARY MATERIAL

- Video: <https://youtu.be/xib1RSU0eeQ>
- Code: <https://github.com/MIT-SPARK/TEASER-plusplus>

I. INTRODUCTION

Point cloud registration (also known as *scan matching* or *point cloud alignment*) is a fundamental problem in robotics and computer vision and consists in finding the best transformation (rotation, translation, and potentially scale) that aligns two point clouds. It finds applications in motion estimation

H. Yang, J. Shi, and L. Carlone are with the Laboratory for Information & Decision Systems (LIDS), Massachusetts Institute of Technology, Cambridge, MA 02139, USA, Email: {hankyang, jinshi, lcarlone}@mit.edu

This work was partially funded by ARL DCIST CRA W911NF-17-2-0181, ONR RAIDER N00014-18-1-2828, Lincoln Laboratory “Resilient Perception in Degraded Environments”, and the Google Daydream Research Program.

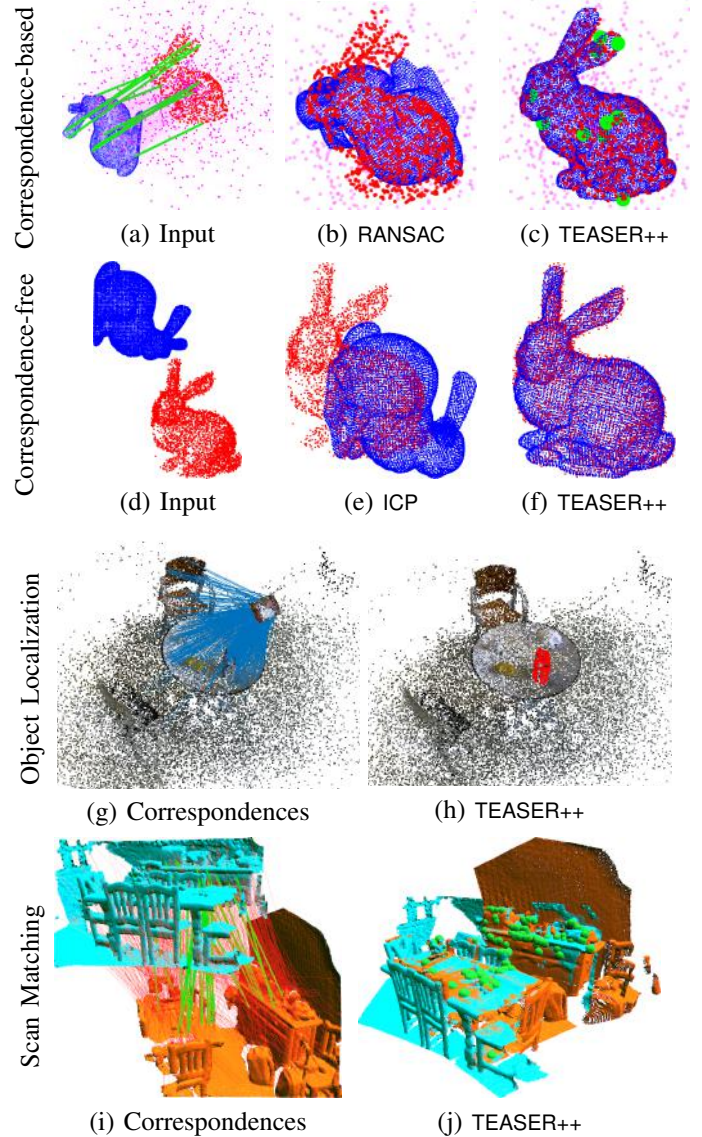


Fig. 1. We address 3D registration in the realistic case where many point-to-point correspondences are outliers due to incorrect matching. (a) Bunny dataset (source cloud in blue and target cloud in red) with 95% outliers (shown as magenta points) and 5% inliers (shown as green lines). Existing algorithms, such as RANSAC (b), can produce incorrect estimates without notice even after running for 10,000 iterations. Our certifiable algorithm, TEASER, largely outperforms the state of the art in terms of robustness and accuracy, and a fast implementation, named TEASER++ (c), computes accurate estimates in milliseconds even with extreme outlier rates and finds the small set of inliers (shown as green dots). The unprecedented robustness of TEASER++ enables the solution of correspondence-free registration problems (d), where ICP (e) fails without a good initial guess, while TEASER++ (f) succeeds without requiring an initial guess. We test our approach in challenging object localization (g-h) and scan matching (i-j) RGB-D datasets, using both traditional features (i.e., FPFH [1]) and deep-learned features (i.e., 3DSmoothNet [2]).

and 3D reconstruction [3], [4], [5], [6], object recognition and localization [7], [8], [9], [10], panorama stitching [11], and medical imaging [12], [13], to name a few.

When the ground-truth correspondences between the point clouds are known and the points are affected by zero-mean Gaussian noise, the registration problem can be readily solved, since elegant closed-form solutions [14], [15] exist for the case of isotropic noise, and recently proposed convex relaxations [16] are empirically tight even in the presence of large anisotropic noise. In practice, however, the correspondences are either unknown, or contain many outliers, leading these solvers to produce poor estimates. Large outlier rates are typical of 3D keypoint detection and matching [17], [18].

Commonly used approaches for registration with unknown or uncertain correspondences either rely on the availability of an initial guess for the unknown transformation (e.g., the Iterative Closest Point, ICP [19]), or implicitly assume the presence of a small set of outliers (e.g., RANSAC [20])¹. These algorithms may fail without notice (Fig. 1 (b),(e)) and may return estimates that are arbitrarily far from the ground-truth transformation. In general, the literature is divided between *heuristics*, that are fast but brittle, and *global methods* that are guaranteed to be robust, but run in worst-case exponential time (e.g., branch-&-bound methods, such as Go-ICP [22]).

This paper is motivated by the goal of designing an approach that (i) can solve registration globally (without relying on an initial guess), (ii) can tolerate extreme amounts of outliers (e.g., when 99% of the correspondences are outliers), (iii) runs in polynomial time and is fast in practice (i.e., can operate in real-time on a robot), and (iv) provides formal performance guarantees. In particular, we look for a *posteriori* guarantees, e.g., conditions that one can check after executing the algorithm to assess the quality of the estimate. This leads to the notion of *certifiable algorithms*, i.e., an algorithm that attempts to solve an intractable problem and provides checkable conditions on whether it succeeded. The interested reader can find a broader discussion in Appendix A.

Contribution. This paper proposes the first certifiable algorithm for 3D registration with outliers. Our first contribution (presented in Section IV) is to reformulate the registration problem using a *Truncated Least Squares* (TLS) cost that is insensitive to a large fraction of spurious data.

The second contribution (Section V) is a general framework to decouple scale, rotation, and translation estimation. The idea of decoupling rotation and translation has appeared in related work, e.g., [23], [24], [21]. The novelty of our proposal is threefold: (i) we develop invariant measurements to estimate the scale ([24], [21] assume the scale is given), (ii) we make the decoupling formal within the framework of *unknown-but-bounded* noise [25], [26], and (iii) we provide a general graph-theoretic framework to derive these invariant measurements. The graph-theoretic interpretation allows pruning a large amount of outliers by finding a *maximum clique* of the graph defined by the invariant measurements (Section VI).

The decoupling allows solving in cascade for scale, rotation, and translation. However, each subproblem is still combinatorial in nature. Our third contribution is to show that (i) in the scalar case TLS estimation can be solved exactly in

polynomial time using an *adaptive voting* scheme, and this enables efficient estimation of the scale and the (component-wise) translation (Section VII); (ii) we can formulate a tight *semidefinite programming* (SDP) relaxation to estimate the rotation and establish *a posteriori* conditions to check the quality of the relaxation (Section VIII). We remark that the rotation subproblem addressed in this paper is in itself a foundational problem in vision (where it is known as *rotation search* [27]) and aerospace (where it is known as the *Wahba problem* [28]). Our SDP relaxation is the first certifiable algorithm for robust rotation search.

Our forth contribution (Section IX) is a set of theoretical results certifying the quality of the solution returned by our algorithm, named *Truncated least squares Estimation And Semidefinite Relaxation* (TEASER). In the noiseless case, we provide easy-to-check conditions under which TEASER recovers the true transformation between the point clouds in the presence of outliers. In the noisy case, we provide bounds on the distance between the ground-truth transformation and TEASER’s estimate. To the best of our knowledge these are the first non-asymptotic error bounds for geometric estimation problems with outliers, while the literature on robust estimation in statistics (e.g., [29]) typically studies simpler problems in Euclidean space and focuses on asymptotic bounds.

Our fifth contribution (Section X) is to implement a fast version of TEASER, named TEASER++, that circumvents the need to solve a large SDP (computable in polynomial time, but typically slow using currently available SDP solvers). We leverage the theoretical results discussed in the previous paragraph to certify TEASER++’s performance. We release a fast open-source C++ implementation of TEASER++.

Our last contribution (Section XI) is an extensive evaluation in both standard benchmarks and on real datasets for object detection [30] and scan matching [31]. In particular, we show that (i) both TEASER and TEASER++ dominate the state of the art (e.g., RANSAC, branch-&-bound, heuristics) and are robust to more than 99% outliers, (ii) TEASER++ can run in milliseconds and it is currently the fastest robust registration algorithm, (iii) TEASER++ is so robust it can also solve problems without correspondences (e.g., hypothesizing all-to-all correspondences) where it largely outperforms ICP, and (iv) TEASER++ can boost registration performance when combined with deep-learned keypoint detection and matching.

Novelty with respect to [32], [33]. In our previous works, we introduced TEASER [32] and the quaternion-based relaxation of the rotation subproblem [33] (named QUASAR). The present manuscript brings TEASER to maturity by (i) providing explicit theoretical results on TEASER’s performance (Section IX), (ii) providing a fast optimality certification method (Section VIII-C), (iii) developing a fast algorithm, TEASER++, that circumvents the need to solve a large SDP, while still being certifiable (Section X), and (iv) reporting a more comprehensive experimental evaluation, including real tests on the 3DMatch dataset and examples of registration without correspondences (Sections XI-C and XI-E). These are major improvements both on the theoretical side ([32], [33] only certified performance on each subproblem, rather than end-to-end, and required solving a large SDP) and on

¹RANSAC’s runtime grows exponentially with the outlier ratio [21] and it typically performs poorly with large outlier rates (see Sections II and XI).

the practical side (TEASER++ is more than three orders of magnitude faster than our proposal in [32]).

II. RELATED WORK

There are two popular paradigms for the registration of 3D point clouds: *Correspondence-based* and *Simultaneous Pose and Correspondence* (i.e., *correspondence-free*) methods.

A. Correspondence-based Methods

Correspondence-based methods first detect and match 3D keypoints between point clouds using local [17], [34], [18], [1] or global [7], [35] descriptors, and then use an estimator to infer the transformation from these putative correspondences. 3D keypoint matching is known to be less accurate compared to 2D counterparts like SIFT and ORB, thus causing much higher outlier rates, e.g., having 95% spurious correspondences is considered common [21]. Therefore, a robust backend that can deal with extreme outlier rates is highly desirable.

Registration without Outliers. Horn [14] and Arun [15] show that optimal solutions (in the maximum likelihood sense) for scale, rotation, and translation can be computed in closed form when the correspondences are exactly known and the points are affected by isotropic zero-mean Gaussian noise. Olsson *et al.* [36] propose a method based on Branch-&-bound (BnB) that is globally optimal and allows point-to-point, point-to-line, and point-to-plane correspondences. Briaies and Gonzalez-Jimenez [16] propose a semidefinite relaxation to solve registration problems with anisotropic Gaussian noise.

If the two sets of points only differ by an unknown rotation (i.e., we do not attempt to estimate the scale and translation), we obtain a simplified version of the registration problem that is known as *rotation search* in computer vision [27], or *Wahba problem* in aerospace [28]. In aerospace, the vector observations are typically the directions to visible stars observed by sensors onboard the satellite. Closed-form solutions to the Wahba problem are known using both quaternion [14], [37] and rotation matrix [38], [39], [15], [40], [41] representations. The Wahba problem is also related to the well-known *Orthogonal Procrustes* problem [42] where one searches for orthogonal matrices (rather than rotations), for which a closed-form solution also exists [43]. The computer vision community has investigated the rotation search problem in the context of point cloud registration [19], [22], image stitching [11], motion estimation and 3D reconstruction [4], [5]. In particular, the closed-form solutions from Horn [14] and Arun *et al.* [15] can be used for (outlier-free) rotation search with isotropic Gaussian noise. Cheng and Crassidis [44] develop a local optimization algorithm for the case of anisotropic Gaussian noise. Ahmed *et al.* [45] develop an SDP relaxation for the case with bounded noise and no outliers. All these methods are known to perform poorly in the presence of outliers.

Robust Registration. Probably the most widely used robust registration approach is based on RANSAC [20], [46], which has enabled several early applications in vision and robotics [47], [48]. Despite its efficiency in the low-noise and low-outlier regime, RANSAC exhibits slow convergence and low accuracy with large outlier rates [21], where it becomes harder to

sample a “good” consensus set. Other approaches resort to *M-estimation*, which replaces the least squares objective function with robust costs that are less sensitive to outliers [49], [50], [51]. Zhou *et al.* [52] propose *Fast Global Registration* (FGR) that uses the Geman-McClure cost function and leverages graduated non-convexity to solve the resulting non-convex optimization. Since graduated non-convexity has to be solved in discrete steps, FGR does not guarantee global optimality [21]. Indeed, FGR tends to fail when the outlier ratio is high (>80%), as we show in Section XI. Parra and Chin [21] propose a *Guaranteed Outlier REmoval* (GORE) technique, that uses geometric operations to significantly reduce the amount of outlier correspondences before passing them to the optimization backend. GORE has been shown to be robust to 95% spurious correspondences [21]. In an independent effort, Parra *et al.* [53] find pairwise-consistent correspondences in 3D registration using a practical maximum clique (PMC) algorithm. While similar in spirit to our original proposal [32], both GORE and PMC do not estimate the *scale* of the registration and are typically slower since they rely on BnB (see Algorithm 2 in [21]). Yang and Carlone [32] propose the first certifiable algorithm for robust registration, which however requires solving a large-scale SDP hence being limited to small problems.

Similarly to the registration problem, rotation search with outliers has also been investigated in the computer vision community. Local techniques for robust rotation search are again based on RANSAC or M-estimation, but are brittle and do not provide performance guarantees. Global methods (that guarantee to compute globally optimal solutions) are based on *Consensus Maximization* [54], [55], [56], [57] and *branch-and-bound* (BnB) [58]. Hartley and Kahl [27] first proposed using BnB for rotation search, and Bazin *et al.* [59] adopted consensus maximization to extend their BnB algorithm with a robust formulation. BnB is guaranteed to return the globally optimal solution, but it runs in exponential time in the worst case. Another class of global methods for Consensus Maximization enumerates all possible subsets of measurements with size no larger than the problem dimension (3 for rotation search) to analytically compute candidate solutions, and then verify global optimality using computational geometry [60], [61]. These methods still require exhaustive enumeration in the worst case. In [33], we propose the first quaternion-based certifiable algorithm for robust rotation search (reviewed in Section VIII), which however requires solving a large SDP.

B. Simultaneous Pose and Correspondence Methods

Simultaneous Pose and Correspondence (SPC) methods alternate between finding the correspondences and computing the best transformation given the correspondences.

Local Methods. The *Iterative Closest Point* (ICP) algorithm [19] is considered a milestone in point cloud registration. However, ICP is prone to converge to *local minima* and it only performs well given a good initial guess. Multiple variants of ICP [62], [63], [64], [65], [66], [67], [68] have proposed to use robust cost functions to improve convergence. Probabilistic interpretations have also been proposed to improve ICP convergence, for instance interpreting the registration problem

as a minimization of the Kullback-Leibler divergence between two Mixture Models [69], [70], [71], [72]. Clark *et al.* [73] represent point clouds as continuous functions and align these functions using Riemannian optimization. All these methods rely on local search, do not provide global optimality guarantees, and typically fail unless a good initial guess is provided.

Global Methods. Global SPC approaches compute a globally optimal solution without initial guesses, and are usually based on BnB, which at each iteration divides the parameter space into multiple sub-domains (branch) and computes the bounds of the objective function for each sub-domain (bound). A series of geometric techniques have been proposed to improve the bounding tightness [27], [74], [22], [75], [76] and increase the search speed [22], [77]. However, the runtime of BnB increases exponentially with the size of the point cloud and it can be made worse by the explosion of the number of local minima resulting from high outlier ratios [21]. Global SPC registration can be also formulated as a mixed-integer program [78], though the runtime remains exponential.

Deep Learning Methods. The success of deep learning on 3D point clouds (e.g., PointNet [79] and DGCNN [80]) opens new opportunities for learning point cloud registration from data. Deep learning methods first learn to embed 3D points clouds in a high-dimensional feature space, then learn to detect and match keypoints to generate correspondences, after which optimization over the space of rigid transformations is performed to search the best alignment. PointNetLK [81] uses PointNet to learn feature representations and then iteratively search for the best rigid transformation to align the features representations, instead of the 3D coordinates. DCP [82] uses DGCNN features for correspondence matching and Horn’s method for registration in an end-to-end fashion (Horn’s method is differentiable). PRNet [83] extends DCP to aligning partially overlapping point clouds. Scan2CAD [84] and its improvement [85] apply similar deep learning pipelines to align CAD models to RGB-D scans. 3DSmoothNet [2] uses a siamese deep learning architecture to establish keypoint correspondences between two point clouds. In Section XI-E we show that (i) our approach provides a robust back-end for deep-learned keypoint matching algorithms (we use [2]), and (ii) that current deep learning approaches still struggle to produce acceptable outlier rates in real problems.

Remark 1 (Reconciling Correspondence-based and SPC Methods). *Correspondence-based and SPC methods are tightly coupled. First of all, approaches like ICP alternate guessing the correspondences and solving a correspondence-base problem. More importantly, one can always reformulate an SPC problem as a correspondence-based problem by hypothesizing all-to-all correspondences, i.e., associating each point in the first point cloud to all the points in the second. While to the best of our knowledge, no previous work pursued this formulation since it leads to an extreme number of outliers, we show that our approach can indeed solve the SPC problem thanks to its unprecedented robustness to outliers.*

III. NOTATIONS AND PRELIMINARIES

Scalars, Vectors, Matrices. We use lowercase characters (e.g., s) to denote real scalars, bold lowercase characters (e.g., \mathbf{v}) for real vectors, and bold uppercase characters (e.g., \mathbf{M}) for real matrices. \mathbf{M}_{ij} denotes the i -th row and j -th column *scalar* entry of matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$, and $[\mathbf{M}]_{ij,d}$ (or simply $[\mathbf{M}]_{ij}$ when d is clear from the context) denotes the i -th row and j -th column $d \times d$ *block* of the matrix $\mathbf{M} \in \mathbb{R}^{md \times nd}$. \mathbf{I}_d is the identity matrix of size d . We use “ \otimes ” to denote the Kronecker product and $\det(\cdot)$ for the matrix determinant. The 2-norm of a vector is denoted as $\|\cdot\|$. The Frobenious norm of a matrix is denoted as $\|\cdot\|_F$. For a symmetric matrix \mathbf{M} of size $n \times n$, we use $\lambda_1 \leq \dots \leq \lambda_n$ to denote its real eigenvalues.

Sets. We use calligraphic fonts to denote sets (e.g., \mathcal{S}). We use \mathcal{S}^n (resp. \mathcal{S}^n) to denote the group of real symmetric (resp. skew-symmetric) matrices with size $n \times n$. $\mathcal{S}_+^n \doteq \{\mathbf{M} \in \mathcal{S}^n : \mathbf{M} \succeq 0\}$ denotes the set of $n \times n$ symmetric *positive semidefinite* matrices. $\text{SO}(d) \doteq \{\mathbf{R} \in \mathbb{R}^{d \times d} : \mathbf{R}^T \mathbf{R} = \mathbf{I}_d, \det(\mathbf{R}) = +1\}$ denotes the d -dimensional *special orthogonal group*, while $\mathcal{U}^{d-1} = \{\mathbf{u} \in \mathbb{R}^d : \|\mathbf{u}\| = 1\}$ denotes the d -dimensional unit sphere.

Quaternions. Unit quaternions are an alternative representation for a 3D rotation $\mathbf{R} \in \text{SO}(3)$. We denote a unit quaternion as a unit-norm column vector $\mathbf{q} = [\mathbf{v}^T \ s]^T \in \mathcal{U}^3$, where $\mathbf{v} \in \mathbb{R}^3$ is the *vector part* of the quaternion and the last element s is the *scalar part*. We also use $\mathbf{q} = [q_1 \ q_2 \ q_3 \ q_4]^T$ to denote the four entries of the quaternion. Each quaternion represents a 3D rotation and the composition of two rotations \mathbf{q}_a and \mathbf{q}_b can be computed using the *quaternion product* $\mathbf{q}_c = \mathbf{q}_a \circ \mathbf{q}_b$:

$$\mathbf{q}_c = \mathbf{q}_a \circ \mathbf{q}_b = \mathbf{\Omega}_1(\mathbf{q}_a)\mathbf{q}_b = \mathbf{\Omega}_2(\mathbf{q}_b)\mathbf{q}_a, \quad (1)$$

where $\mathbf{\Omega}_1(\mathbf{q})$ and $\mathbf{\Omega}_2(\mathbf{q})$ are defined as follows:

$$\mathbf{\Omega}_1(\mathbf{q}) = \begin{bmatrix} q_4 & -q_3 & q_2 & q_1 \\ q_3 & q_4 & -q_1 & q_2 \\ -q_2 & q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix}, \quad \mathbf{\Omega}_2(\mathbf{q}) = \begin{bmatrix} q_4 & q_3 & -q_2 & q_1 \\ -q_3 & q_4 & q_1 & q_2 \\ q_2 & -q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix}. \quad (2)$$

The inverse of a quaternion $\mathbf{q} = [\mathbf{v}^T \ s]^T$ is defined as:

$$\mathbf{q}^{-1} = \begin{bmatrix} -\mathbf{v} \\ s \end{bmatrix}, \quad (3)$$

where one simply reverses the sign of the vector part. The rotation of a vector $\mathbf{a} \in \mathbb{R}^3$ can be expressed in terms of quaternion product. Formally, if \mathbf{R} is the (unique) rotation matrix corresponding to a unit quaternion \mathbf{q} , then:

$$\begin{bmatrix} \mathbf{R}\mathbf{a} \\ 0 \end{bmatrix} = \mathbf{q} \circ \hat{\mathbf{a}} \circ \mathbf{q}^{-1}, \quad (4)$$

where $\hat{\mathbf{a}} = [\mathbf{a}^T \ 0]^T$ is the homogenization of \mathbf{a} , obtained by augmenting \mathbf{a} with an extra entry equal to zero. The set of unit quaternions, i.e., the 4-dimensional unit sphere \mathcal{U}^3 , is a *double cover* of $\text{SO}(3)$ since \mathbf{q} and $-\mathbf{q}$ represent the same rotation (this fact can be easily seen by examining eq. (4)).

IV. ROBUST REGISTRATION WITH TRUNCATED LEAST SQUARES COST

In the robust registration problem, we are given two 3D point clouds $\mathcal{A} = \{\mathbf{a}_i\}_{i=1}^N$ and $\mathcal{B} = \{\mathbf{b}_i\}_{i=1}^N$, with $\mathbf{a}_i, \mathbf{b}_i \in \mathbb{R}^3$. We consider a *correspondence-based* setup, where we are

given putative correspondences $(\mathbf{a}_i, \mathbf{b}_i), i = 1, \dots, N$, that obey the following generative model:

$$\mathbf{b}_i = s^\circ \mathbf{R}^\circ \mathbf{a}_i + \mathbf{t}^\circ + \mathbf{o}_i + \boldsymbol{\epsilon}_i, \quad (5)$$

where $s^\circ > 0$, $\mathbf{R}^\circ \in \text{SO}(3)$, and $\mathbf{t}^\circ \in \mathbb{R}^3$ are the unknown (to-be-computed) scale, rotation, and translation, $\boldsymbol{\epsilon}_i$ models the measurement noise, and \mathbf{o}_i is a vector of zeros if the pair $(\mathbf{a}_i, \mathbf{b}_i)$ is an *inlier*, or a vector of arbitrary numbers for *outlier correspondences*. In words, if the i -th correspondence $(\mathbf{a}_i, \mathbf{b}_i)$ is an inlier correspondence, \mathbf{b}_i corresponds to a 3D transformation of \mathbf{a}_i (plus noise $\boldsymbol{\epsilon}_i$), while if $(\mathbf{a}_i, \mathbf{b}_i)$ is an outlier correspondence, \mathbf{b}_i is just an arbitrary vector.

Registration without Outliers. When $\boldsymbol{\epsilon}_i$ is a zero-mean Gaussian noise with isotropic covariance $\sigma_i^2 \mathbf{I}_3$, and all the correspondences are correct (i.e., $\mathbf{o}_i = \mathbf{0}, \forall i$), the Maximum Likelihood estimator of $(s^\circ, \mathbf{R}^\circ, \mathbf{t}^\circ)$ can be computed by solving the following nonlinear least squares problem:

$$\min_{s>0, \mathbf{R} \in \text{SO}(3), \mathbf{t} \in \mathbb{R}^3} \sum_{i=1}^N \frac{1}{\sigma_i^2} \|\mathbf{b}_i - s\mathbf{R}\mathbf{a}_i - \mathbf{t}\|^2 \quad (6)$$

Although (6) is a non-convex problem, due to the non-convexity of the set $\text{SO}(3)$, its optimal solution can be computed in closed form by decoupling the estimation of the scale, rotation, and translation, using Horn's [14] or Arun's method [15]. A key contribution of the present paper is to provide a way to decouple scale, rotation, and translation in the more challenging case with outliers.

In practice, a large fraction of the correspondences are *outliers*, due to incorrect keypoint matching. Despite the elegance of the closed-form solutions [14], [15], they are not robust to outliers, and a single "bad" outlier can compromise the correctness of the resulting estimate. Hence, we propose a *truncated least squares registration* formulation that can tolerate extreme amounts of spurious data.

Truncated Least Squares Registration. We depart from the Gaussian noise model and assume the noise is *unknown but bounded* [25]. Formally, we assume the *inlier* noise $\boldsymbol{\epsilon}_i$ in (5) is such that $\|\boldsymbol{\epsilon}_i\| \leq \beta_i$, where β_i is a given bound.

Then we adopt the following *Truncated Least Squares* (TLS) *Registration* formulation:

$$\min_{s>0, \mathbf{R} \in \text{SO}(3), \mathbf{t} \in \mathbb{R}^3} \sum_{i=1}^N \min \left(\frac{1}{\beta_i^2} \|\mathbf{b}_i - s\mathbf{R}\mathbf{a}_i - \mathbf{t}\|^2, \bar{c}^2 \right), \quad (7)$$

which computes a least-squares solution of measurements with small residuals ($\frac{1}{\beta_i^2} \|\mathbf{b}_i - s\mathbf{R}\mathbf{a}_i - \mathbf{t}\|^2 \leq \bar{c}^2$), while discarding measurements with large residuals (when $\frac{1}{\beta_i^2} \|\mathbf{b}_i - s\mathbf{R}\mathbf{a}_i - \mathbf{t}\|^2 > \bar{c}^2$ the i -th summand becomes a constant and does not influence the optimization). Note that one can always divide each summand in (7) by \bar{c}^2 : therefore, one can safely assume \bar{c}^2 to be 1. For the sake of generality, in the following we keep \bar{c}^2 since it provides a more direct "knob" to be stricter or more lenient towards potential outliers.

The noise bound β_i is fairly easy to set in practice and can be understood as a "3-sigma" noise bound or as the maximum error we expect from an inlier. The interested reader can find a more formal discussion on how to set β_i and \bar{c} in Appendix B. We remark that while we assume to have a bound on the

maximum error we expect from the inliers (β_i), we do not make assumptions on the generative model for the outliers, which is typically unknown in practice.

Remark 2 (TLS vs. Consensus Maximization). TLS estimation is related to Consensus Maximization [54], a popular robust estimation approach in computer vision. Consensus Maximization looks for an estimate that maximizes the number of inliers, while TLS simultaneously computes a least squares estimate for the inliers. The two methods are not guaranteed to produce the same choice of inliers in general, since TLS also penalizes inliers with large errors. Appendix C provides a toy example to illustrate the potential mismatch between the two techniques and provides necessary conditions under which the two formulations find the same set of inliers.

Despite being insensitive to outlier correspondences, the truncated least squares formulation (7) is much more challenging to solve globally, compared to the outlier-free case (6). This is the case even in simpler estimation problems where the feasible set of the unknowns is convex, as stated below.

Remark 3 (Hardness of TLS Estimation [86], [87]). The minimization of a sum of truncated convex functions, $\sum_{i=1}^N \min(f_i(\mathbf{x}), \bar{c}^2)$, over a convex feasible set $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$, is NP-hard in the dimension d [86], [87]. A simple exhaustive search can obtain a globally optimal solution in exponential time $O(2^N)$ [87].

Remark 3 states the NP-hardness of TLS estimation over a convex feasible set. The TLS problem (7) is even more challenging due to the non-convexity of $\text{SO}(3)$. While problem (7) is hard to solve directly, in the next section, we show how to decouple the estimation of scale, rotation, and translation using *invariant measurements*.

V. DECOUPLING SCALE, ROTATION, AND TRANSLATION ESTIMATION

We propose a general approach to decouple the estimation of scale, translation, and rotation in problem (7). The key insight is that we can reformulate the measurements (5) to obtain quantities that are invariant to a subset of the transformations (scaling, rotation, translation).

A. Translation Invariant Measurements (TIMs)

While the absolute positions of the points in \mathcal{B} depend on the translation \mathbf{t} , the relative positions are invariant to \mathbf{t} . Mathematically, given two points \mathbf{b}_i and \mathbf{b}_j from (5), the relative position of these two points is:

$$\mathbf{b}_j - \mathbf{b}_i = s\mathbf{R}(\mathbf{a}_j - \mathbf{a}_i) + (\mathbf{o}_j - \mathbf{o}_i) + (\boldsymbol{\epsilon}_j - \boldsymbol{\epsilon}_i), \quad (8)$$

where the translation \mathbf{t} cancels out in the subtraction. Therefore, we can obtain a *Translation Invariant Measurement* (TIM) by computing $\bar{\mathbf{a}}_{ij} \doteq \mathbf{a}_j - \mathbf{a}_i$ and $\bar{\mathbf{b}}_{ij} \doteq \mathbf{b}_j - \mathbf{b}_i$, and the TIM satisfies the following generative model:

$$\bar{\mathbf{b}}_{ij} = s\mathbf{R}\bar{\mathbf{a}}_{ij} + \mathbf{o}_{ij} + \boldsymbol{\epsilon}_{ij}, \quad (\text{TIM})$$

where $\mathbf{o}_{ij} \doteq \mathbf{o}_j - \mathbf{o}_i$ is zero if *both* the i -th and the j -th measurements are inliers (or arbitrary otherwise), while $\boldsymbol{\epsilon}_{ij} \doteq$

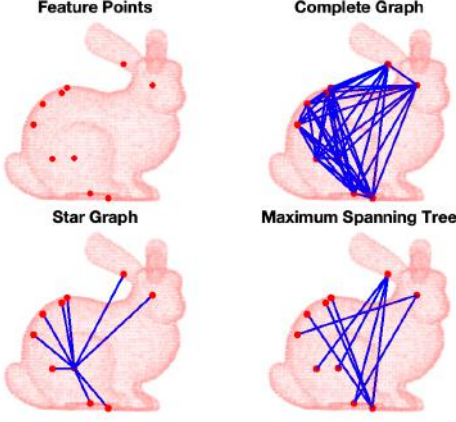


Fig. 2. Graph topologies for generating TIMs in the Bunny dataset [88].

$\epsilon_j - \epsilon_i$ is the measurement noise. It is easy to see that if $\|\epsilon_i\| \leq \beta_i$ and $\|\epsilon_j\| \leq \beta_j$, then $\|\epsilon_{ij}\| \leq \beta_i + \beta_j \doteq \delta_{ij}$.

The advantage of the TIMs in eq. (TIM) is that their generative model only depends on two unknowns, s and \mathbf{R} . The number of TIMs is upper-bounded by $\binom{N}{2} = N(N-1)/2$, where pairwise relative measurements between all pairs of points are computed. Theorem 4 below connects the TIMs with the topology of a graph defined over the 3D points.

Theorem 4 (Translation Invariant Measurements). Define the vectors $\mathbf{a} \in \mathbb{R}^{3N}$ (resp. $\mathbf{b} \in \mathbb{R}^{3N}$), obtained by concatenating all vectors \mathbf{a}_i (resp. \mathbf{b}_i) in a single column vector. Moreover, define an arbitrary graph \mathcal{G} with nodes $\{1, \dots, N\}$ and an arbitrary set of edges \mathcal{E} . Then, the vectors $\bar{\mathbf{a}} = (\mathbf{A} \otimes \mathbf{I}_3)\mathbf{a}$ and $\bar{\mathbf{b}} = (\mathbf{A} \otimes \mathbf{I}_3)\mathbf{b}$ are TIMs, where $\mathbf{A} \in \mathbb{R}^{|\mathcal{E}| \times N}$ is the incidence matrix of \mathcal{G} , and \otimes is the Kronecker product.

A proof of the theorem is given in Appendix D. Three potential graph topologies for generating TIMs on the Bunny dataset [88] are illustrated in Fig. 2.

B. Translation and Rotation Invariant Measurements (TRIMs)

While the relative locations of pairs of points (TIMs) still depend on the rotation \mathbf{R} , their distances are invariant to both \mathbf{R} and \mathbf{t} . Therefore, to build rotation invariant measurements, we compute the norm of each TIM vector:

$$\|\bar{\mathbf{b}}_{ij}\| = \|s\mathbf{R}\bar{\mathbf{a}}_{ij} + \mathbf{o}_{ij} + \epsilon_{ij}\|. \quad (9)$$

We now note that for the inliers ($\mathbf{o}_{ij} = \mathbf{0}$) it holds (using $\|\epsilon_{ij}\| \leq \delta_{ij}$ and the triangle inequality):

$$\|s\mathbf{R}\bar{\mathbf{a}}_{ij}\| - \delta_{ij} \leq \|s\mathbf{R}\bar{\mathbf{a}}_{ij} + \epsilon_{ij}\| \leq \|s\mathbf{R}\bar{\mathbf{a}}_{ij}\| + \delta_{ij}, \quad (10)$$

hence we can write (9) equivalently as:

$$\|\bar{\mathbf{b}}_{ij}\| = \|s\mathbf{R}\bar{\mathbf{a}}_{ij}\| + \tilde{o}_{ij} + \tilde{\epsilon}_{ij}, \quad (11)$$

with $|\tilde{\epsilon}_{ij}| \leq \delta_{ij}$, and $\tilde{o}_{ij} = 0$ if both i and j are inliers or is an arbitrary scalar otherwise. Recalling that the norm is rotation invariant and that $s > 0$, and dividing both sides of (11) by $\|\bar{\mathbf{a}}_{ij}\|$, we obtain new measurements $s_{ij} \doteq \frac{\|\bar{\mathbf{b}}_{ij}\|}{\|\bar{\mathbf{a}}_{ij}\|}$:

$$s_{ij} = s + o_{ij}^s + \epsilon_{ij}^s, \quad (\text{TRIM})$$

where $\epsilon_{ij}^s \doteq \frac{\tilde{\epsilon}_{ij}}{\|\bar{\mathbf{a}}_{ij}\|}$, and $o_{ij}^s \doteq \frac{\tilde{o}_{ij}}{\|\bar{\mathbf{a}}_{ij}\|}$. It is easy to see that $|\epsilon_{ij}^s| \leq \delta_{ij}/\|\bar{\mathbf{a}}_{ij}\|$ since $|\tilde{\epsilon}_{ij}| \leq \delta_{ij}$. We define $\alpha_{ij} \doteq \delta_{ij}/\|\bar{\mathbf{a}}_{ij}\|$.

Eq. (TRIM) describes a *Translation and Rotation Invariant Measurement* (TRIM) whose generative model is only function of the unknown scale s .

Remark 5 (Novelty of Invariant Measurements). Ideas similar to the translation invariant measurements (TIMs) have been used in recent work [77], [21], [24], [89], [90] while (i) the graph-theoretic interpretation of Theorem 4 is novel and generalizes previously proposed methods, and (ii) the notion of translation and rotation invariant measurements (TRIMs) is completely new. We also remark that while related work uses invariant measurements to filter-out outliers [21] or to speed up BnB [77], [24], we show that they also allow decoupling the estimation of scale, rotation, and translation.

A summary table of the invariant measurements and the corresponding noise bounds is given in Appendix E.

VI. TRUNCATED LEAST SQUARES ESTIMATION AND SEMIDEFINITE RELAXATION (TEASER): OVERVIEW

We propose a decoupled approach to solve in cascade for the scale, the rotation, and the translation in (7). The approach, named *Truncated least squares Estimation And SEMidefinite Relaxation* (TEASER), works as follows:

- 1) we use the TRIMs to estimate the scale \hat{s}
- 2) we use \hat{s} and the TIMs to estimate the rotation $\hat{\mathbf{R}}$
- 3) we use \hat{s} and $\hat{\mathbf{R}}$ to estimate the translation $\hat{\mathbf{t}}$ from $(\mathbf{a}_i, \mathbf{b}_i)$ in the original TLS problem (7).

We state each subproblem in the following subsections.

A. Robust Scale Estimation

The generative model (TRIM) describes linear *scalar* measurements s_{ij} of the unknown scale s , affected by bounded noise $|\epsilon_{ij}^s| \leq \alpha_{ij}$ including potential outliers (when $o_{ij}^s \neq 0$). Again, we estimate the scale given the measurements s_{ij} and the bounds α_{ij} using a TLS estimator:

$$\hat{s} = \arg \min_s \sum_{k=1}^K \min \left(\frac{(s - s_k)^2}{\alpha_k^2}, \bar{c}^2 \right), \quad (12)$$

where for simplicity we numbered the invariant measurements from 1 to $K = |\mathcal{E}|$ and adopted the notation s_k instead of s_{ij} . Section VII shows that (12) can be solved exactly and in polynomial time via *adaptive voting* (Algorithm 2).

B. Robust Rotation Estimation

Given the scale estimate \hat{s} produced by the scale estimation, the generative model (TIM) describes measurements $\bar{\mathbf{b}}_{ij}$ affected by bounded noise $\|\epsilon_{ij}\| \leq \delta_{ij}$ including potential outliers (when $\mathbf{o}_{ij} \neq \mathbf{0}$). Again, we compute \mathbf{R} from the estimated scale \hat{s} , the TIM measurements $(\bar{\mathbf{a}}_{ij}, \bar{\mathbf{b}}_{ij})$ and the bounds δ_{ij} using a TLS estimator:

$$\hat{\mathbf{R}} = \arg \min_{\mathbf{R} \in \text{SO}(3)} \sum_{k=1}^K \min \left(\frac{\|\bar{\mathbf{b}}_k - \hat{s}\mathbf{R}\bar{\mathbf{a}}_k\|^2}{\delta_k^2}, \bar{c}^2 \right), \quad (13)$$

where for simplicity we numbered the measurements from 1 to $K = |\mathcal{E}|$ and adopted the notation $\bar{\mathbf{a}}_k, \bar{\mathbf{b}}_k$ instead of $\bar{\mathbf{a}}_{ij}, \bar{\mathbf{b}}_{ij}$. Problem (13) is known as the *Robust Wahba* or *Robust*

Rotation Search problem [33]. Section VIII shows that (13) can be solved exactly and in polynomial time (in practical problems) via a *tight* semidefinite relaxation.

C. Robust Component-wise Translation Estimation

After obtaining the scale and rotation estimates \hat{s} and $\hat{\mathbf{R}}$ by solving (12)-(13), we can substitute them back into problem (7) to estimate the translation \mathbf{t} . Although (7) operates on the ℓ_2 norm of the vector, we propose to solve for the translation component-wise, i.e., we compute the entries t_1, t_2, t_3 of \mathbf{t} independently:

$$\hat{t}_j = \arg \min_{t_j} \sum_{i=1}^N \min \left(\frac{(t_j - [\mathbf{b}_i - \hat{s}\hat{\mathbf{R}}\mathbf{a}_i]_j)^2}{\beta_i^2}, \bar{c}^2 \right), \quad (14)$$

for $j = 1, 2, 3$, and where $[\cdot]_j$ denotes the j -th entry of a vector. Since $\mathbf{b}_i - \hat{s}\hat{\mathbf{R}}\mathbf{a}_i$ is a known vector in this stage, it is easy to see that (14) is a scalar TLS problem. Therefore, similarly to (12), Section VII shows that (14) can be solved exactly and in polynomial time via *adaptive voting* (Algorithm 2).

D. Boosting Performance: Max Clique Inlier Selection (MCIS)

While in principle we could simply execute the cascade of scale, rotation, and translation estimation described above, our graph-theoretic interpretation of Theorem 4 affords further opportunities to prune outliers.

Consider the TRIMs as edges in the complete graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ (where the vertices \mathcal{V} are the 3D points and the edge set \mathcal{E} induces the TIMs and TRIMs per Theorem 4). After solving the scale estimation problem (12) (discussed in Section VII), we can prune the edges (i, j) in the graph whose associated TRIM s_{ij} have been classified as outliers by the TLS formulation. This allows us to obtain a pruned graph $\mathcal{G}'(\mathcal{V}, \mathcal{E}')$, with $\mathcal{E}' \subseteq \mathcal{E}$, where gross outliers are discarded. The following result ensures that inliers form a clique in the graph $\mathcal{G}'(\mathcal{V}, \mathcal{E}')$ enabling an even more substantial rejection of outliers.

Theorem 6 (Maximal Clique Inlier Selection). *Edges corresponding to inlier TIMs form a clique in \mathcal{E}' , and there is at least one maximal clique in \mathcal{E}' that contains all the inliers.*

A proof of Theorem 6 is presented in Appendix F. Theorem 6 allows us to prune outliers by finding the maximal cliques of $\mathcal{G}'(\mathcal{V}, \mathcal{E}')$. Although finding the maximal cliques of a graph takes exponential time in general, there exist efficient approximation algorithms that scale to graphs with millions of nodes [91], [92], [93]. Under high outlier rates, the graph $\mathcal{G}'(\mathcal{V}, \mathcal{E}')$ is sparse and the maximal clique problem can be solved quickly in practice [94]. Therefore, in this paper, after performing scale estimation and removing the corresponding gross outliers, we compute the maximal clique with largest cardinality, i.e., the *maximum clique*, as the inlier set to pass to rotation estimation. Section XI-A shows that this method drastically reduces the number of outliers.²

E. Pseudocode of TEASER

The pseudocode of TEASER is summarized in Algorithm 1.

² A straightforward extension of this idea would be to explore multiple cliques until a large set of inliers is found.

Algorithm 1: Truncated least squares Estimation And SEmidefinite Relaxation (TEASER).

```

1 Input: points  $(\mathbf{a}_i, \mathbf{b}_i)$  and bounds  $\beta_i$  ( $i = 1, \dots, N$ ),
   threshold  $\bar{c}^2$  (default:  $\bar{c}^2 = 1$ ), graph edges  $\mathcal{E}$  (default:  $\mathcal{E}$ 
   describes the complete graph);
2 Output:  $\hat{s}, \hat{\mathbf{R}}, \hat{\mathbf{t}}$ ;
3 % Compute TIM and TRIM
4  $\bar{\mathbf{b}}_{ij} = \mathbf{b}_j - \mathbf{b}_i$ ,  $\bar{\mathbf{a}}_{ij} = \mathbf{a}_j - \mathbf{a}_i$ ,  $\delta_{ij} = \beta_i + \beta_j$ ,  $\forall (i, j) \in \mathcal{E}$ 
5  $s_{ij} = \frac{\|\bar{\mathbf{b}}_{ij}\|}{\|\bar{\mathbf{a}}_{ij}\|}$ ,  $\alpha_{ij} = \frac{\delta_{ij}}{\|\bar{\mathbf{a}}_{ij}\|}$ ,  $\forall (i, j) \in \mathcal{E}$ 
6 % Decoupled estimation of  $s, \mathbf{R}, \mathbf{t}$ 
7  $\hat{s} = \text{estimate\_s}(\{s_{ij}, \alpha_{ij} : \forall (i, j) \in \mathcal{E}\}, \bar{c}^2)$ 
8  $\mathcal{E} = \text{maxClique}(\mathcal{E})$  % remove gross outliers
9  $\hat{\mathbf{R}} = \text{estimate\_R}(\{\bar{\mathbf{a}}_{ij}, \bar{\mathbf{b}}_{ij}, \delta_{ij} : \forall (i, j) \in \mathcal{E}\}, \bar{c}^2, \hat{s})$ 
10  $\hat{\mathbf{t}} = \text{estimate\_t}(\{\mathbf{a}_i, \mathbf{b}_i, \beta_i : i = 1 \dots, N\}, \bar{c}^2, \hat{s}, \hat{\mathbf{R}})$ 
11 return:  $\hat{s}, \hat{\mathbf{R}}, \hat{\mathbf{t}}$ 

```

The following Sections VII-VIII describe how to implement the functions in lines 7, 9, 10 of Algorithm 1. In particular, we show how to obtain global and robust estimates of scale (`estimate_s`) and translation (`estimate_t`) in Section VII, and rotation (`estimate_R`) in Section VIII.

VII. ROBUST SCALE AND TRANSLATION ESTIMATION: ADAPTIVE VOTING

In this section, we propose an *adaptive voting* algorithm to solve *exactly* the robust scale estimation and the robust component-wise translation estimation.

A. Adaptive Voting for Scalar TLS Estimation

Both the scale estimation (12) and the component-wise translation estimation (14) resort to finding a TLS estimate of an unknown *scalar* given a set of outlier-corrupted measurements. Using the notation for scale estimation (12), the following theorem shows that one can solve *scalar* TLS estimation in polynomial time by a simple enumeration.

Theorem 7 (Optimal Scalar TLS Estimation). *Consider the scalar TLS problem in (12). For a given $s \in \mathbb{R}$, define the consensus set of s as $\mathcal{I}(s) = \{k : \frac{(s - s_k)^2}{\alpha_k^2} \leq \bar{c}^2\}$. Then, for any $s \in \mathbb{R}$, there are at most $2K - 1$ different non-empty consensus sets. If we name these sets $\mathcal{I}_1, \dots, \mathcal{I}_{2K-1}$, then the solution of (12) can be computed by enumeration as:*

$$\hat{s} = \arg \min \left\{ f_s(\hat{s}_i) : \hat{s}_i = \left(\sum_{k \in \mathcal{I}_i} \frac{1}{\alpha_k^2} \right)^{-1} \sum_{k \in \mathcal{I}_i} \frac{s_k}{\alpha_k^2}, \forall i \right\}, \quad (15)$$

where $f_s(\cdot)$ is the objective function of (12).

Theorem 7, whose proof is given in Appendix G, is based on the insight that the consensus set can only change at the boundaries of the intervals $[s_k - \alpha_k \bar{c}, s_k + \alpha_k \bar{c}]$ (Fig. 3(a)) and there are at most $2K$ such boundaries. The theorem also suggests a straightforward *adaptive voting* algorithm to solve (12), with pseudocode given in Algorithm 2. The algorithm first builds the boundaries of the intervals shown in Fig. 3(a) (line 4). Then, for each interval, it evaluates the consensus set (line 12, see also Fig. 3(b)). Since the consensus

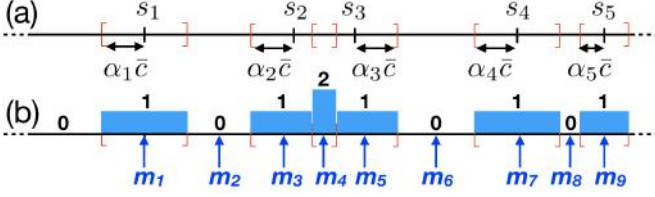


Fig. 3. (a) confidence interval for each measurement s_k (every s in the k -th interval satisfies $\frac{(s-s_k)^2}{\alpha_k^2 \bar{c}^2} \leq \bar{c}^2$; (b) cardinality of the consensus set for every s and middle-points m_i for each interval with constant consensus set.

Algorithm 2: Adaptive Voting.

```

1 Input:  $s_k, \alpha_k, \bar{c}$ ;
2 Output:  $\hat{s}$ , scale estimate solving (12);
3 % Define boundaries and sort
4  $v = \text{sort}([s_1 - \alpha_1 \bar{c}, s_1 + \alpha_1 \bar{c}, \dots, s_K - \alpha_K \bar{c}, s_K + \alpha_K \bar{c}])$ 
5 % Compute middle points
6  $m_i = \frac{v_i + v_{i+1}}{2}$  for  $i = 1, \dots, 2K - 1$ 
7 % Voting
8 for  $i = 1, \dots, 2K - 1$  do
9    $\mathcal{I}_i = \emptyset$ 
10  for  $k = 1, \dots, K$  do
11    if  $m_i \in [s_k - \alpha_k \bar{c}, s_k + \alpha_k \bar{c}]$  then
12       $\mathcal{I}_i = \mathcal{I}_i \cup \{k\}$  % add to consensus set
13    end
14  end
15 end
16 % Enumerate consensus sets and return best
17 return:  $\arg \min \left\{ f_s(\hat{s}_i) : \hat{s}_i = \left( \sum_{k \in \mathcal{I}_i} \frac{1}{\alpha_k^2} \right)^{-1} \sum_{k \in \mathcal{I}_i} \frac{s_k}{\alpha_k^2}, \forall i \right\}$ 

```

set does not change within an interval, we compute it at the interval centers (line 6, see also Fig. 3(b)). Finally, the cost of each consensus set is computed and the smallest cost is returned as optimal solution (line 17).

Remark 8 (Adaptive Voting). *The adaptive voting algorithm generalizes the histogram voting method of Scaramuzza [95] (i) to adaptively adjust the bin size in order to obtain an optimal solution and (ii) to solve a TLS (rather than Consensus Maximization) formulation. Adaptive voting can be also used for Consensus Maximization, by simply returning the largest consensus set \mathcal{I}_i in Algorithm 2. We also refer the interested reader to the paper of Liu and Jiang [86], who recently developed a similar algorithm in an independent effort and provide a generalization to 2D TLS estimation problems.*

In summary, the function `estimate_s` in Algorithm 1 calls Algorithm 2 to compute the optimal scale estimate \hat{s} , and the function `estimate_t` in Algorithm 1 calls Algorithm 2 three times (one for each entry of \mathbf{t}) and returns the translation estimate $\hat{\mathbf{t}} = [\hat{t}_1 \ \hat{t}_2 \ \hat{t}_3]^\top$.

VIII. ROBUST ROTATION ESTIMATION: SEMIDEFINITE RELAXATION AND FAST CERTIFICATES

This section describes how to compute an optimal solution to problem (13) or certify that a given rotation estimate is globally optimal. TLS estimation is NP-hard according to Remark 3 and the robust rotation estimation (13) has the

additional complexity of involving a non-convex domain (*i.e.*, $\text{SO}(3)$). This section provides a surprising result: we can compute globally optimal solutions to (3) (or certify that a given estimate is optimal) in polynomial time in virtually all practical problems using a *tight* semidefinite (SDP) relaxation.³

We achieve this goal in three steps. First, we show that the robust rotation estimation problem (13) can be reformulated as a *Quadratically Constrained Quadratic Program* (QCQP) by adopting a quaternion formulation and using a technique we name *binary cloning* (Section VIII-A). Second, we show how to obtain a *semidefinite relaxation* and relax the non-convex QCQP into a convex SDP with *redundant constraints* (Section VIII-B). The SDP relaxation enables solving (13) with global optimality guarantees in polynomial time. Third, we show that the relaxation also enables a fast certification algorithm, which, given a rotation guess, can test if the rotation is the optimal solution to (13) (Section VIII-C). The latter will be instrumental in developing a fast and certifiable registration approach that circumvents the time-consuming task of solving a large-scale SDP. The derivation in Sections VIII-A and VIII-B is borrowed from our previous work [33], while the fast certification in Section VIII-C has not been presented before.

A. Robust Rotation Estimation as a QCQP

This section rewrites problem (4) as a Quadratically Constrained Quadratic Program (QCQP). Using the quaternion preliminaries introduced in Section III –and in particular eq. (4)– it is easy to rewrite (13) using unit quaternions:

$$\min_{\mathbf{q} \in \mathcal{U}^3} \sum_{k=1}^K \min \left(\frac{\|\hat{\mathbf{b}}_k - \mathbf{q} \circ \hat{\mathbf{a}}_k \circ \mathbf{q}^{-1}\|^2}{\delta_k^2}, \bar{c}^2 \right), \quad (16)$$

where we defined $\hat{\mathbf{a}}_k \doteq [\hat{s} \hat{\mathbf{a}}_k^\top \ 0]^\top$ and $\hat{\mathbf{b}}_k \doteq [\hat{s} \hat{\mathbf{b}}_k^\top \ 0]^\top$, and “ \circ ” denotes the quaternion product. The main advantage of using (16) is that we replaced the set $\text{SO}(3)$ with a simpler set, the 4-dimensional unit sphere \mathcal{U}^3 .

From TLS to Mixed-Integer Programming. Problem (16) is hard to solve globally, due to the non-convexity of both the cost function and the domain \mathcal{U}^3 . As a first step towards obtaining a QCQP, we expose the non-convexity of the cost by rewriting the TLS cost using binary variables. In particular, we rewrite the inner “min” in (16) using the following property, that holds for any pair of scalars x and y :

$$\min(x, y) = \min_{\theta \in \{+1, -1\}} \frac{1 + \theta}{2} x + \frac{1 - \theta}{2} y. \quad (17)$$

Eq. (17) can be verified to be true by inspection: the right-hand-side returns x (with minimizer $\theta = +1$) if $x < y$, and y (with minimizer $\theta = -1$) if $x > y$. This enables us to rewrite problem (16) as a mixed-integer program including the quaternion \mathbf{q} and binary variables θ_k , $i = 1, \dots, K$:

$$\min_{\substack{\mathbf{q} \in \mathcal{U}^3 \\ \theta_k \in \{\pm 1\}}} \sum_{k=1}^K \frac{1 + \theta_k}{2} \frac{\|\hat{\mathbf{b}}_k - \mathbf{q} \circ \hat{\mathbf{a}}_k \circ \mathbf{q}^{-1}\|^2}{\delta_k^2} + \frac{1 - \theta_k}{2} \bar{c}^2. \quad (18)$$

³In other words, while the NP-hardness implies the presence of worst-case instances that are not solvable in polynomial time, these instances are not observed to frequently occur in practice.

The reformulation is related to the Black-Rangarajan duality between robust estimation and line processes [50]: the TLS cost is an extreme case of robust function that results in a binary line process. Intuitively, the binary variables $\{\theta_k\}_{k=1}^N$ in problem (18) decide whether a given measurement k is an inlier ($\theta_k = +1$) or an outlier ($\theta_k = -1$).

From Mixed-Integer to Quaternions. Now we convert the mixed-integer program (18) to an optimization over $K + 1$ quaternions. The intuition is that, if we define extra quaternions $\mathbf{q}_k \doteq \theta_k \mathbf{q}$, we can rewrite (18) as a function of \mathbf{q} and \mathbf{q}_k ($k = 1, \dots, K$). This is a key step towards getting a quadratic cost (Proposition 10) and is formalized as follows.

Proposition 9 (Binary Cloning). *The mixed-integer program (18) is equivalent (in the sense that they admit the same optimal solution \mathbf{q}) to the following optimization:*

$$\min_{\substack{\mathbf{q} \in \mathcal{U}^3 \\ \mathbf{q}_k = \{\pm \mathbf{q}\}}} \sum_{k=1}^K \frac{\|\hat{\mathbf{b}}_k - \mathbf{q} \circ \hat{\mathbf{a}}_k \circ \mathbf{q}^{-1} + \mathbf{q}^\top \mathbf{q}_k \hat{\mathbf{b}}_k - \mathbf{q} \circ \hat{\mathbf{a}}_k \circ \mathbf{q}_k^{-1}\|^2}{4\delta_k^2} + \frac{1 - \mathbf{q}^\top \mathbf{q}_k}{2} \bar{c}^2. \quad (19)$$

which involves $K + 1$ quaternions (\mathbf{q} and \mathbf{q}_k , $i = 1, \dots, K$).

While a formal proof is given in Appendix H, it is fairly easy to see that if $\mathbf{q}_k = \{\pm \mathbf{q}\}$, or equivalently, $\mathbf{q}_k = \theta_k \mathbf{q}$ with $\theta_k \in \{\pm 1\}$, then $\mathbf{q}_k^\top \mathbf{q} = \theta_k$, and $\mathbf{q} \circ \hat{\mathbf{a}}_k \circ \mathbf{q}_k^{-1} = \theta_k (\mathbf{q} \circ \hat{\mathbf{a}}_k \circ \mathbf{q}^{-1})$ which exposes the relation between (18) and (19). We dubbed the re-parametrization (19) *binary cloning* since now we created a “clone” \mathbf{q}_k for each measurement, such that $\mathbf{q}_k = \mathbf{q}$ for inliers (recall $\mathbf{q}_k = \theta_k \mathbf{q}$) and $\mathbf{q}_k = -\mathbf{q}$ for outliers.

From Quaternions to QCQP. We conclude this section by showing that (19) can be actually written as a QCQP. This observation is non-trivial since (19) has a *quartic* cost and $\mathbf{q}_k = \{\pm \mathbf{q}\}$ is not in the form of a quadratic constraint. The re-formulation as a QCQP is given in the following.

Proposition 10 (Binary Cloning as a QCQP). *Define a single column vector $\mathbf{x} = [\mathbf{q}^\top \mathbf{q}_1^\top \dots \mathbf{q}_K^\top]^\top$ stacking all variables in Problem (19). Then, Problem (19) is equivalent (in the sense that they admit the same optimal solution \mathbf{q}) to the following Quadratically-Constrained Quadratic Program:*

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^{4(K+1)}} \quad & \mathbf{x}^\top \mathbf{Q} \mathbf{x} \\ \text{s.t.} \quad & \mathbf{x}_q^\top \mathbf{x}_q = 1 \\ & \mathbf{x}_{q_k} \mathbf{x}_{q_k}^\top = \mathbf{x}_q \mathbf{x}_q^\top, \forall i = 1, \dots, K \end{aligned} \quad (20)$$

where $\mathbf{Q} \in \mathcal{S}^{4(K+1)}$ is a known symmetric matrix that depends on the TIM measurements $\hat{\mathbf{a}}_k$ and $\hat{\mathbf{b}}_k$ (the explicit expression is given in Appendix I), and the notation \mathbf{x}_q (resp. \mathbf{x}_{q_k}) denotes the 4D subvector of \mathbf{x} corresponding to \mathbf{q} (resp. \mathbf{q}_k).

A complete proof of Proposition 10 is given in Appendix I. Intuitively, (i) we developed the squares in the cost function (19), (ii) we used the properties of unit quaternions (Section III) to simplify the expression to a quadratic cost, and (iii) we adopted the more compact notation afforded by the vector \mathbf{x} to obtain (20).

B. Semidefinite Relaxation

Problem (20) writes the TLS rotation estimation problem (13) as a QCQP. Problem (20) is still a non-convex

problem (quadratic equality constraints are non-convex). Here we develop a *tight* convex semidefinite programming (SDP) relaxation for problem (20).

The crux of the relaxation consists in rewriting problem (20) as a function of the following matrix:

$$\mathbf{Z} = \mathbf{x} \mathbf{x}^\top = \begin{bmatrix} \mathbf{q} \mathbf{q}^\top & \mathbf{q} \mathbf{q}_1^\top & \dots & \mathbf{q} \mathbf{q}_K^\top \\ \star & \mathbf{q}_1 \mathbf{q}_1^\top & \dots & \mathbf{q}_1 \mathbf{q}_K^\top \\ \vdots & \vdots & \ddots & \vdots \\ \star & \star & \dots & \mathbf{q}_K \mathbf{q}_K^\top \end{bmatrix} \in \mathcal{S}_+^{4(K+1)}. \quad (21)$$

For this purpose we note that the objective function of (20) is a linear function of \mathbf{Z} :

$$\mathbf{x}^\top \mathbf{Q} \mathbf{x} = \text{tr}(\mathbf{Q} \mathbf{x} \mathbf{x}^\top) = \text{tr}(\mathbf{Q} \mathbf{Z}), \quad (22)$$

and that $\mathbf{x}_q \mathbf{x}_q^\top = [\mathbf{Z}]_{00}$, where $[\mathbf{Z}]_{00}$ denotes the top-left 4×4 diagonal block of \mathbf{Z} , and $\mathbf{x}_{q_k} \mathbf{x}_{q_k}^\top = [\mathbf{Z}]_{kk}$, where $[\mathbf{Z}]_{kk}$ denotes the k -th diagonal block of \mathbf{Z} (we number the row and column block index from 0 to K for notation convenience). Since any matrix in the form $\mathbf{Z} = \mathbf{x} \mathbf{x}^\top$ is a positive-semidefinite rank-1 matrix, we obtain:

Proposition 11 (Matrix Formulation of Binary Cloning). *Problem (20) is equivalent (in the sense that optimal solutions of a problem can be mapped to optimal solutions of the other) to the following non-convex rank-constrained program:*

$$\begin{aligned} \min_{\mathbf{Z} \succeq 0} \quad & \text{tr}(\mathbf{Q} \mathbf{Z}) \\ \text{s.t.} \quad & \text{tr}([\mathbf{Z}]_{00}) = 1 \\ & [\mathbf{Z}]_{ii} = [\mathbf{Z}]_{00}, \forall i = 1, \dots, K \\ & \text{rank}(\mathbf{Z}) = 1. \end{aligned} \quad (23)$$

The proof is given in Appendix J. At this point we are ready to develop our SDP relaxation by dropping the non-convex rank-1 constraint and adding redundant constraints.

Proposition 12 (SDP Relaxation with Redundant Constraints). *The following SDP is a convex relaxation of (23):*

$$\begin{aligned} \min_{\mathbf{Z} \succeq 0} \quad & \text{tr}(\mathbf{Q} \mathbf{Z}) \\ \text{s.t.} \quad & \text{tr}([\mathbf{Z}]_{00}) = 1 \\ & [\mathbf{Z}]_{kk} = [\mathbf{Z}]_{00}, \forall k = 1, \dots, K \\ & [\mathbf{Z}]_{ij} = [\mathbf{Z}]_{ij}^\top, \forall 0 \leq i < j \leq K. \end{aligned} \quad (24)$$

The redundant constraints in the last row of eq. (24) enforce all the off-diagonal 4×4 blocks of \mathbf{Z} to be symmetric, which is a redundant constraint for (20), since:

$$[\mathbf{Z}]_{ij} = \mathbf{q}_i \mathbf{q}_j^\top = (\theta_i \mathbf{q})(\theta_j \mathbf{q})^\top = \theta_i \theta_j \mathbf{q} \mathbf{q}^\top, \quad (25)$$

which is indeed a signed copy of $\mathbf{q} \mathbf{q}^\top$ and must be symmetric. In [33], we named the relaxation (24) QUASAR (*QUAternion-based Semidefinite relAXation for Robust alignment*). Appendix K provides a different approach to develop the convex relaxation using Lagrangian duality theory.

The following theorem provides readily checkable *a posteriori* conditions under which (24) computes an optimal solution to robust rotation estimation.

Theorem 13 (Global Optimality Guarantee). *Let \mathbf{Z}^* be the optimal solution of (24). If $\text{rank}(\mathbf{Z}^*) = 1$, then the SDP relaxation is said to be tight, i.e.,:*

- 1) *the optimal cost of (24) matches the optimal cost of the QCQP (20),*
- 2) *\mathbf{Z}^* can be written as $\mathbf{Z}^* = (\mathbf{x}^*)(\mathbf{x}^*)^\top$, where $\mathbf{x}^* = [(\mathbf{q}^*)^\top, (\mathbf{q}_1^*)^\top, \dots, (\mathbf{q}_K^*)^\top]^\top$, with $\mathbf{q}_k^* = \theta_k^* \mathbf{q}^*$, $\theta_k^* \in \{\pm 1\}$,*
- 3) *$\pm \mathbf{x}^*$ are the two unique global minimizers of the original QCQP (20).*

A formal proof is given in Appendix K. Theorem 13 states that if the SDP (24) admits a rank-1 solution, then one can extract the unique global minimizer (remember: \mathbf{q} and $-\mathbf{q}$ represent the same rotation) of the non-convex QCQP (20) from the rank-1 decomposition of the SDP solution.

In practice, we observe that (24) always returns a rank-1 solution: in Section XI, we empirically demonstrate the tightness of (24) in the face of noise and extreme outlier rates ($> 95\%$ [33]), and show its superior performance compared to a similar SDP relaxation using a rotation matrix parametrization (see eq. (13) in Section V.B of [32]). As it is often the case, we can only partially justify with theoretical results the phenomenal performance observed in practice. The interested reader can find a formal proof of tightness in the low-noise and outlier-free case in Theorem 7 and Proposition 8 of [33]

C. Fast Global Optimality Certification

Despite the superior robustness and (*a posteriori*) global optimality guarantees, the approach outlined in the previous section requires solving the large-scale SDP (24), which is known to be computationally expensive (e.g., solving the SDP (24) for $K = 100$ takes about 1200 seconds with MOSEK [96]). On the other hand, there exist fast heuristics that compute high-quality solutions to the non-convex TLS rotation problem (13) in milliseconds. For example, the *graduated non-convexity* (GNC) approach in [97] computes globally optimal TLS solutions with high probability when the outlier ratio is below 90%, while not being certifiable.

Therefore, in this section, we ask the question: *given an estimate from a fast heuristic, such as GNC,⁴ can we certify the optimality of the estimate or reject it as suboptimal?* In other words: can we make GNC certifiable? The answer is yes: we can leverage the insights of Section VIII-B to obtain a fast optimality certification algorithm that does not require solving a large SDP. The pseudocode is presented in Algorithm 3 and the following theorem proves its soundness.

Theorem 14 (Optimality Certification). *Given a feasible (but not necessarily optimal) solution $(\hat{\mathbf{q}}, \hat{\theta}_1, \dots, \hat{\theta}_K)$ of the TLS rotation estimation problem (16), denote with $\hat{\mathbf{x}} = [\hat{\mathbf{q}}^\top, \hat{\theta}_1 \hat{\mathbf{q}}^\top, \dots, \hat{\theta}_K \hat{\mathbf{q}}^\top]^\top$, and $\hat{\mu} = \hat{\mathbf{x}}^\top \mathbf{Q} \hat{\mathbf{x}}$, the corresponding solution and cost of the QCQP (20). Then, Algorithm 3 produces a sub-optimality bound η that satisfies:*

$$\frac{\hat{\mu} - \mu^*}{\hat{\mu}} \leq \eta, \quad (26)$$

⁴Our certification approach also applies to RANSAC, which however has the downside of being non-deterministic and typically less accurate than GNC.

Algorithm 3: Optimality Certification.

```

1 Input: a feasible solution  $(\hat{\mathbf{q}}, \hat{\theta}_1, \dots, \hat{\theta}_K)$  attaining cost
    $\hat{\mu}$  in the TLS rotation estimation problem (16);
   homogeneous and scaled TIMs  $(\hat{\mathbf{a}}_k, \hat{\mathbf{b}}_k)$ , noise bounds  $\delta_k$ 
   and  $\bar{c}^2$ ; maximum number of iterations  $T$  (default
    $T = 200$ ); desired relative sub-optimality gap  $\bar{\eta}$  (default
    $\bar{\eta} = 0.1\%$ )
2 Output: best relative suboptimality gap  $\eta$ 
3 % Compute data matrix  $\mathbf{Q}$  (Proposition 10)
4  $\mathbf{Q} = \text{get\_Q}(\hat{\mathbf{a}}_k, \hat{\mathbf{b}}_k, \delta_k, \bar{c}^2)$ 
5 % Compute  $\bar{\mathbf{x}}, \bar{\mathbf{Q}}$  (Appendix L)
6  $\bar{\mathbf{x}} = [1, \hat{\theta}_1, \dots, \hat{\theta}_K]^\top \otimes [0 \ 0 \ 0 \ 1]^\top$ ,  $\bar{\mathbf{Q}} = \hat{\Omega}_q^\top \mathbf{Q} \hat{\Omega}_q$ 
7 % Compute initial guess  $\bar{\mathbf{M}}_0 \in \bar{\mathcal{L}}$  for dual certificate
8  $\bar{\mathbf{M}}_0 = \bar{\mathbf{Q}} - \hat{\mu} \mathbf{J} + \text{get\_Delta0}(\hat{\mathbf{a}}_k, \hat{\mathbf{b}}_k, \delta_k, \bar{c}^2, \bar{\mathbf{Q}}, \bar{\mathbf{x}})$ 
9 % Alternating projections
10  $\bar{\mathbf{M}}_{\bar{\mathcal{L}}}^{(0)} = \bar{\mathbf{M}}_{\bar{\mathcal{L}}}^{(1)} = \bar{\mathbf{M}}_0$ ,  $\eta = +\infty$ 
11 for  $t = 1, \dots, T$  do
12   % Compute minimum eigenvalue
13    $\lambda_1^{(t)} = \text{get\_min\_eig}(\bar{\mathbf{M}}_{\bar{\mathcal{L}}}^{(t)})$ 
14   % Compute relative suboptimality bound
15    $\eta^{(t)} = \frac{|\lambda_1^{(t)}|(K+1)}{\hat{\mu}}$ 
16   if  $\eta^{(t)} < \eta$  then
17      $\eta = \eta^{(t)}$ 
18   end
19   if  $\eta < \bar{\eta}$  then
20     break
21   end
22   % Nesterov-style acceleration
23    $\gamma = \frac{t-2}{t+1}$ 
24    $\mathbf{V}_{\bar{\mathcal{L}}}^{(t)} = \bar{\mathbf{M}}_{\bar{\mathcal{L}}}^{(t)} + \gamma(\bar{\mathbf{M}}_{\bar{\mathcal{L}}}^{(t)} - \bar{\mathbf{M}}_{\bar{\mathcal{L}}}^{(t-1)})$ 
25   % Project to positive semidefinite cone  $\mathcal{S}_+$ 
26    $\mathbf{V}_{\mathcal{S}_+}^{(t)} = \Pi_{\mathcal{S}_+}(\mathbf{V}_{\bar{\mathcal{L}}}^{(t)})$ 
27   % Project to affine subspace  $\bar{\mathcal{L}}$ 
28    $\bar{\mathbf{M}}_{\bar{\mathcal{L}}}^{(t+1)} = \Pi_{\bar{\mathcal{L}}}(\mathbf{V}_{\mathcal{S}_+}^{(t)})$ 
29 end
30 return:  $\eta$ 

```

where μ^* is the (unknown) global minimum of the non-convex TLS rotation estimation problem (16) and the QCQP (20). Moreover, when the relaxation (24) is tight and the parameter γ in line 23 is set to zero, and if $\hat{\mathbf{q}} = \mathbf{q}^*$, $\hat{\theta}_k = \theta_k^*$ is a global minimizer of the TLS problem (16), then the sub-optimality bound $\eta^{(t)}$ (line 15) asymptotically converges to zero.

Although a complete derivation of Algorithm 3 and the proof of Theorem 14 is postponed to Appendix L, now we describe our key intuitions. The first insight is that, given a candidate solution to (16), whenever the SDP relaxation (24) is tight, we can compute a certificate of optimality from the Lagrangian dual problem [98, Section 5].

The second insight is that Lagrangian duality shows that the dual certificate is a matrix at the intersection between the positive semidefinite cone \mathcal{S}_+ and an affine subspace $\bar{\mathcal{L}}$ (whose expression is derived in Appendix L). Algorithm 3 looks for such a matrix ($\bar{\mathbf{M}}_{\bar{\mathcal{L}}}^{(t+1)}$ in line (28)) using the method of *alternating projections to convex sets* [99], [100] (line 26

and line 28), starting from a cleverly chosen initial guess (line 8). Since the alternating projections method guarantees convergence to a point in $\mathcal{S}_+ \cap \mathcal{M}$ from any initial guess whenever the intersection is non-empty, Algorithm 3 guarantees to asymptotically certify global optimality when provided with an optimal solution, whenever the relaxation (24) is tight.

The third insight is that even when the provided solution is not globally optimal and the relaxation is not tight, Algorithm 3 can still compute a *sub-optimality bound* from the minimum eigenvalue of the dual certificate matrix (line 15), which is informative of the quality of the candidate solution. To the best of our knowledge, this is the first result that can assess the sub-optimality of outlier-robust estimation beyond the instances commonly investigated in statistics [29].

Our fourth insight is algorithmic: to improve the convergence speed of the alternating projection method, line 23 implements a Nesterov-style [101] acceleration which empirically converges much faster than the vanilla alternating projections ($\gamma = 0$). Finally, to further speed up computation, Appendix M shows that the projections on lines 26–28 of Algorithm 3 can be computed in closed form, and Appendix U provides the expression of the initial guess in line 8. Section XI shows the effectiveness of Algorithm 3 in certifying the solutions from the GNC approach [97].

IX. PERFORMANCE GUARANTEES

This section establishes theoretical guarantees for TEASER. While Theorems 7, 13, and 14 establish when we obtain a globally optimal solution from optimization problems involved in TEASER, this section uses these theorems to bound the distance between the estimate produced by TEASER and the ground truth. Section IX-A investigates the noiseless (but outlier-corrupted) case, while Section IX-B considers the general case with noise and outliers.

A. Exact Recovery with Outliers and Noiseless Inliers

We start by analyzing the case in which the inliers are noiseless since it allows deriving stronger performance guarantees and highlights the fundamental difference the outlier generation mechanism can make. As we show below, in a noiseless case with randomly outliers generated, TEASER is able to recover the ground-truth transformation from only 3 inliers (and an arbitrary number of outliers!). On the other hand, when the outlier generation is adversarial, TEASER (and any other approach) requires that more than 50% of the measurements are inliers in order to retrieve the ground truth.

Theorem 15 (Estimation Contract with Noiseless Inliers and Random Outliers). *Assume (i) the set of correspondences contains at least 3 noiseless non-collinear and distinct⁵ inliers, where for an inlier i , it holds $\mathbf{b}_i = \mathbf{s}^\circ \mathbf{R}^\circ \mathbf{a}_i + \mathbf{t}^\circ$, and $(\mathbf{s}^\circ, \mathbf{R}^\circ, \mathbf{t}^\circ)$ denotes the ground-truth transformation we want to recover. Assume (ii) the outliers are in generic position (e.g., they are perturbed by random noise). If (iii) the inliers computed by TEASER (in each subproblem) have zero residual error for the corresponding subproblem, and (iv) the*

rotation subproblem produces a valid certificate (in the sense of Theorems 13 and 14), then with probability 1 the output $(\hat{\mathbf{s}}, \hat{\mathbf{R}}, \hat{\mathbf{t}})$ of TEASER exactly recovers the ground truth, i.e., $\hat{\mathbf{s}} = \mathbf{s}^\circ, \hat{\mathbf{R}} = \mathbf{R}^\circ, \hat{\mathbf{t}} = \mathbf{t}^\circ$.

A proof of Theorem 15 is given in Appendix N. Conditions (iii) and (iv) can be readily checked using the solution computed by TEASER, in the spirit of certifiable algorithms. Assumptions (i) and (ii) only ensure that the problem is well-posed. In particular assumption (i) is the same assumption required to compute a unique transformation without outliers [14]. Assumption (ii) is trickier since it requires *assuming* the outliers to be “generic” rather than adversarial: intuitively, if we allow an adversary to perturb an arbitrary number of points, s/he can create an identical copy of the points in \mathcal{A} with an arbitrary transformation and induce *any* algorithm into producing an incorrect estimate. Clearly, in such a setup outlier rejection is ill-posed and no algorithm can recover the correct solution.⁶

This fundamental limitation is captured by the following theorem, which focuses on the case with adversarial outliers.

Theorem 16 (Estimation Contract with Noiseless Inliers and Adversarial Outliers). *Assume (i) the set of correspondences is such that the number of inliers N_{in} and the number of outliers N_{out} satisfy $N_{\text{in}} \geq N_{\text{out}} + 3$ and no three inliers are collinear. If (ii) the inliers computed by TEASER (in each subproblem) have zero residual error for the corresponding subproblem, and (iii) the rotation subproblem produces a valid certificate (in the sense of Theorems 13 and 14), and (iv) TEASER returns a consensus set that satisfies (i), then the output $(\hat{\mathbf{s}}, \hat{\mathbf{R}}, \hat{\mathbf{t}})$ of TEASER exactly recovers the ground truth transformation, i.e., $\hat{\mathbf{s}} = \mathbf{s}^\circ, \hat{\mathbf{R}} = \mathbf{R}^\circ, \hat{\mathbf{t}} = \mathbf{t}^\circ$.*

A proof of Theorem 16 is given in Appendix O. We observe that now Theorem 16 provides an even clearer “contract” for TEASER: as long as the number of inliers is sufficiently larger than the number of outliers, TEASER provides readily-checkable conditions on whether it recovered the ground-truth transformation. If the percentage of inliers is below 50% (i.e., $N_{\text{in}} \leq N_{\text{out}}$), TEASER will attempt to retrieve the transformation that is consistent with the largest set of inliers (in the proof, we show that under the conditions of the theorem, TEASER produces a maximum consensus solution).

To the best of our knowledge, Theorems 15–16 provide the first exact recovery results for registration with outliers.

B. Approximate Recovery with Outliers and Noisy Inliers

The case with noisy inliers is strictly more challenging than the noise-free case. Intuitively, the larger the noise, the more blurred is the boundary between inliers and outliers. From the theoretical standpoint, this translates into weaker guarantees.

Theorem 17 (Estimation Contract with Noisy Inliers and Adversarial Outliers). *Assume (i) the set of correspondences contains at least 4 non-coplanar inliers, where for any inlier i ,*

⁶In such a case, it might still be feasible to enumerate all potential solutions, but the optimal solution of Consensus Maximization or the robust estimator (7) can no longer be guaranteed to be close to the ground truth.

⁵Distinct in the sense that no two points occupy the same 3D location.

$\|b_i - s^\circ R^\circ a_i - t^\circ\| \leq \beta$, and $(s^\circ, R^\circ, t^\circ)$ denotes the ground-truth transformation. Assume (ii) the inliers belong to the maximum consensus set in each subproblem⁷, and (iii) the second largest consensus set is “sufficiently smaller” than the maximum consensus set (as formalized in Lemma 23). If (iv) the rotation subproblem produces a valid certificate (Theorems 13-14), the output $(\hat{s}, \hat{R}, \hat{t})$ of TEASER satisfies:

$$|\hat{s} - s^\circ| \leq 2 \max_{ij} \alpha_{ij}, \quad (27)$$

$$\|s^\circ R^\circ - \hat{s} \hat{R}\|_F \leq 2\sqrt{3} \frac{\max_{ij} \alpha_{ij}}{\min_{ijk} \sigma_{\min}(U_{ijk})} \quad (28)$$

$$\|\hat{t} - t^\circ\| \leq (9 + 3\sqrt{3})\beta, \quad (29)$$

where $\sigma_{\min}(\cdot)$ denotes the smallest singular value of a matrix, and U_{ijk} is the 3×3 matrix $U_{ijk} = \begin{bmatrix} \bar{a}_{ij} & \bar{a}_{ih} & \bar{a}_{ik} \\ \|\bar{a}_{ij}\| & \|\bar{a}_{ih}\| & \|\bar{a}_{ik}\| \end{bmatrix}$. Moreover, $\sigma_{\min}(U_{ijk})$ is nonzero as long as (every 4 points a_i chosen among) the inliers are not coplanar.

A proof of Theorem 17 is given in Appendix P. To the best of our knowledge, Theorem (17) provides the first performance bounds for noisy registration problems with outliers. Condition (i) is similar to the standard assumption that guarantees the existence of a unique alignment in the outlier-free case, with the exception that we now require 4 non-coplanar inliers. While conditions (ii)-(iii) seem different and less intuitive than the ones in Theorem 16, they are a generalization of conditions (i) and (ii) in Theorem 16. Condition (ii) in Theorem 17 ensures that the inliers form a large consensus set (the same role played by condition (i) in Theorem 16). Condition (iii) in Theorem 17 ensures that the largest consensus set cannot be confused with other large consensus sets (the assumption of zero residual played a similar role in Theorem 16). These conditions essentially limit the amount of “damage” that outliers can do by avoiding that they form large sets of mutually consistent measurements.

Remark 18 (Interpretation of the Bounds). The bounds (27)-(29) have a natural geometric interpretation. First of all, we recall from Section V that we defined $\alpha_{ij} = \delta_{ij}/\|\bar{a}_{ij}\|$. Hence α_{ij} can be understood as (the inverse of) a signal to noise ratio: δ_{ij} measures the amount of noise, while $\|\bar{a}_{ij}\| = \|a_j - a_i\|$ measures the size of the point cloud (in terms of distance between two points). Therefore, the scale estimate will worsen when the noise becomes large when compared to the size of the point cloud. The parameter α_{ij} also influences the rotation bound (28), which, however, also depends on the smallest singular value of the matrix $\sigma_{\min}(U_{ijk})$. The latter measures how far is the point cloud from degenerate configurations. Finally, the translation bound does not depend on the scale of the point cloud, since for a given scale and rotation a single point would suffice to compute a translation (in other words, the distance among points does not play a role there).

The interested reader can find tighter (but more expensive to

compute and less intelligible) bounds, under similar assumptions as Theorem 17, in Appendix V.

X. TEASER++: A FAST C++ IMPLEMENTATION

In order to showcase the real-time capabilities of TEASER in real robotics, vision, and graphics applications, we have developed a fast C++ implementation of TEASER, named TEASER++. TEASER++ has been released as an open-source library and can be found at <https://github.com/MIT-SPARK/TEASER-plusplus>. This section describes the algorithmic and implementation choices we made in TEASER++.

TEASER++ follows the same decoupled approach described in Section VI. The only exception is that it circumvents the need to solve a large-scale SDP and uses the GNC approach described in [97] for the TLS rotation estimation problem. As described in Section VIII-C, we essentially run GNC and certify *a posteriori* that it retrieved the globally optimal solution using Theorem 14. This approach is very effective in practice since –as shown in the experimental section– the scale estimation (Section VI-A) and the max clique inlier selection (Section VI-D) are already able to remove a large number of outliers, hence GNC only needs to solve a more manageable problem with less outliers (a setup in which it has been shown to be very effective [97]). Therefore, by combining a fast heuristic (GNC) with the certification result in Theorem 14, TEASER++ is the first fast and certifiable registration algorithm in that (i) it enjoys the performance guarantees of Section IX, and (ii) it runs in milliseconds in practical problem instances.

TEASER++ uses Eigen3 for fast linear algebra operations. To further optimize for performance, we have implemented a large portion of TEASER++ using shared-memory parallelism. Using OpenMP [102], an industry-standard interface for scalable parallel programming, we have parallelized Algorithm 2, as well as the computation of the TIMs and TRIMs. In addition, we use a fast exact parallel maximum clique finder algorithm [103] to find the maximum clique for outlier pruning.

To facilitate rapid prototyping and visualization, in addition to the C++ library, we provide Python and MATLAB bindings. Furthermore, we provide a ROS [104] wrapper to enable easy integration and deployment in real-time robotics applications.

XI. EXPERIMENTS AND APPLICATIONS

The goal of this section is to (i) test the performance of each module presented in this paper, including scale, rotation, translation estimation, the MCIS pruning, and the optimality certification Algorithm 3 (Section XI-A), (ii) show that TEASER and TEASER++ outperform the state of the art on standard benchmarks (Section XI-B), (iii) show that TEASER++ also solves Simultaneous Pose and Correspondence (SPC) problems and dominates existing methods, such as ICP (Section XI-C), (iv) show an application of TEASER for object localization in an RGB-D robotics dataset (Section XI-D), and (v) show an application of TEASER++ to difficult scan-matching problems with deep-learned correspondences (Section XI-E).

Implementation Details. TEASER++ is implemented as discussed in Section X. TEASER is implemented in MATLAB, uses CVX [105] to solve the convex relaxation (24), and uses

⁷In other words: the inliers belong to the largest set S such that for any $i, j \in S$, $|\delta_{ij} - s| \leq \alpha_{ij}$, $\|b_{ij} - sR a_{ij}\| \leq \delta_{ij}$, and $\|b_i - sR a_i - t\| \leq \beta_i$, $\forall i = 1, 2, 3$ for any transformation (s, R, t) .

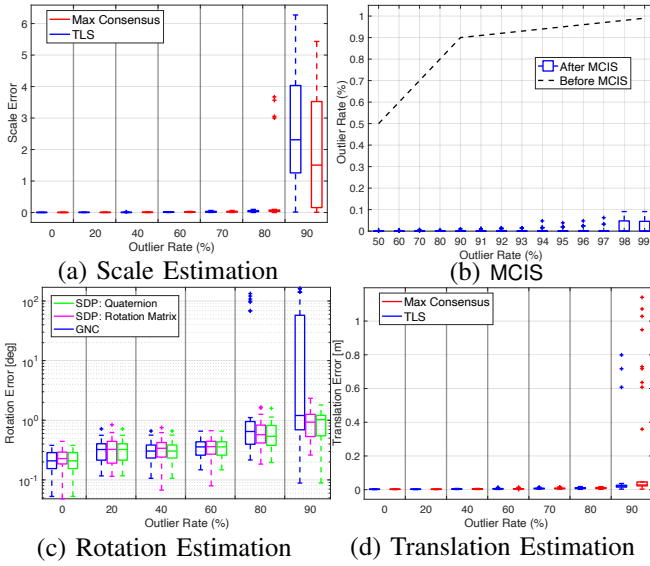


Fig. 4. Results for scale, rotation, translation estimation, and impact of maximal clique inlier selection (MCIS) for increasing outlier rates.

the algorithm in [94] to find the maximum clique in the pruned TIM graph (see Theorem 6). In all tests we set $\bar{c} = 1$. All tests are run on a laptop with an i7-8850H CPU and 32GB of RAM.

A. Testing TEASER’s and TEASER++’s Modules

Testing Setup. We use the Bunny point cloud from the Stanford 3D Scanning Repository [88] and resize it to be within the $[0, 1]^3$ cube. The Bunny is first downsampled to $N = 40$ points (unless specified otherwise), and then a random transformation $(s, \mathbf{R}, \mathbf{t})$ (with $1 \leq s \leq 5$, $\|\mathbf{t}\| \leq 1$, and $\mathbf{R} \in \text{SO}(3)$) is applied according to eq. (5). To generate the bounded noise ϵ_i , we sample $\epsilon_i \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$, until the resulting vector satisfies $\|\epsilon_i\| \leq \beta_i$. We set $\sigma = 0.01$ and $\beta_i = 5.54\sigma$ such that $\mathbb{P}(\|\epsilon_i\|^2 > \beta_i^2) \leq 10^{-6}$ (cf. Remark 20 in Appendix B). To generate outliers, we replace a fraction of the \mathbf{b}_i ’s with vectors uniformly sampled inside the sphere of radius 5. We test increasing outlier rates from 0% (all inliers) to 90% of the measurements; we test up to 99% outliers when relevant (e.g., we omit testing above 90% if failures are already observed at 90%). All statistics are computed over 40 Monte Carlo runs unless mentioned otherwise.

Scale Estimation. Given two point clouds \mathcal{A} and \mathcal{B} , we first create $N(N-1)/2$ TIMs corresponding to a complete graph and then use Algorithm 2 to solve for the scale. We compute both *Consensus Maximization* [106] and TLS estimates of the scale. Fig. 4(a) shows box plots of the scale error with increasing outlier ratios. The scale error is computed as $|\hat{s} - s^\circ|$, where \hat{s} is the scale estimate and s° is the ground-truth. We observe the TLS solver is robust against 80% outliers, while Consensus Maximization failed three times in that regime.

Maximal Clique Inlier Selection (MCIS). We downsample Bunny to $N = 1000$ and fix the scale to $s = 1$ when applying the random transformation. We first prune the outlier TIMs/TRIMs (edges) that are not consistent with the scale $s = 1$, while keeping all the points (vertices), to obtain the graph \mathcal{G}' . Then we compute the *maximum clique* in \mathcal{G}' and remove all edges and vertices outside the maximum clique, obtaining

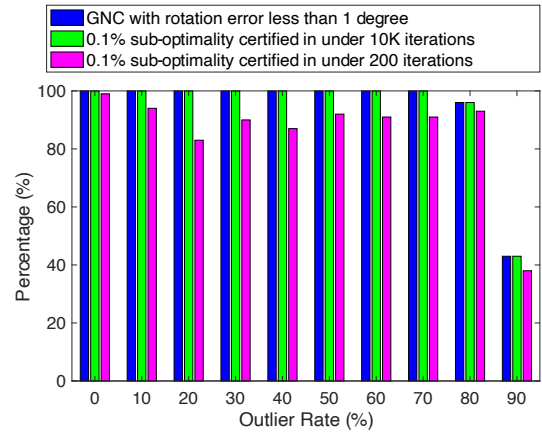


Fig. 5. GNC and optimality certification for robust rotation estimation for increasing outlier rates. We test Algorithm 3 with 200 and 10,000 iterations.

a pruned graph \mathcal{G}'' . Fig. 4(b) shows the outlier ratio in \mathcal{G}' (label: “Before MCIS”) and \mathcal{G}'' (label: “After MCIS”). The MCIS effectively reduces the amount of outliers to below 10%, facilitating rotation and translation estimation, which, in isolation, can already tolerate more than 90% outliers ($< 90\%$ when using GNC) as shown below.

Rotation Estimation. We simulate TIMs by applying a random rotation \mathbf{R} to the Bunny, and fixing $s = 1$ and $\mathbf{t} = \mathbf{0}$ (we set the number of TIMs to $K = 40$). We compare three approaches to solve the TLS rotation estimation problem (13): (i) the quaternion-based SDP relaxation in eq. (24) and [33], (ii) the rotation-matrix-based SDP relaxation we proposed in [32], and (iii) the GNC heuristic in [97]. For each approach, we evaluate the rotation error as $|\arccos((\text{tr}(\hat{\mathbf{R}}^T \mathbf{R}^\circ) - 1)/2)|$, which is the geodesic distance between the rotation estimate $\hat{\mathbf{R}}$ (produced by each approach) and the ground-truth \mathbf{R}° [107].

Fig. 4(c) reports the rotation error for increasing outlier rates. The two relaxations ensure similar performance, while the quaternion-based relaxation proposed in this paper is slightly more accurate at high outlier rates. Results in Appendix Q show that the quaternion-based relaxation is always tighter than the relaxation in [32], which often translates into better estimates. The quaternion-based relaxation also has the advantage of enabling the fast certification technique of Section VIII-C. The GNC heuristic performs well below 80% outliers and then starts failing. The reader can find more experiments using our quaternion-based relaxation in [33], where we also demonstrate its robustness against over 95% outliers and discuss applications to panorama stitching.

While the quaternion-based relaxation dominates in terms of accuracy and robustness, it requires solving a large SDP. Therefore, in TEASER++, we opted to use the fast GNC heuristic instead; this choice is motivated by the observation that MCIS is already able to remove most of the outliers (Fig. 4(b)) hence within TEASER++ the rotation estimation only requires solving a problem with less than 10% outliers.

Translation Solver. We apply a random translation \mathbf{t} to the Bunny, and fix $s = 1$ and $\mathbf{R} = \mathbf{I}_3$. Fig. 4(d) shows that component-wise translation estimation using both Consensus Maximization and TLS are robust against 80% outliers. The translation error is defined as $\|\hat{\mathbf{t}} - \mathbf{t}^\circ\|$, the 2-norm of the difference between the estimate $\hat{\mathbf{t}}$ and the ground-truth \mathbf{t}° . As

mentioned above, most outliers are typically removed before translation estimation, hence enabling TEASER and TEASER++ to be robust to extreme outlier rates (more in Section XI-B).

Optimality Certification for Rotation Estimation. We now test the effectiveness of Algorithm 3 in certifying optimality of a rotation estimate. We consider the same setup we used for testing rotation estimation but with $K = 100$ TIMs. We use the GNC scheme in [97] to solve problem (13) and then use Algorithm 3 to certify the solution of GNC. Fig. 5 shows the performance of GNC and the certification algorithm under increasing outlier rates, where 100 Monte Carlo runs are performed at each outlier rate. The blue bars show the percentage of runs for which GNC produced a solution with less than 1 degree rotation error with respect to the ground truth; the green bars show the percentage of runs for which Algorithm 3 produced a relative sub-optimality bound lower than 0.1% in less than 10,000 iterations of alternating projections; the magenta bars show the percentage of runs for which Algorithm 3 produced a relative sub-optimality bound lower than 0.1% in under 200 iterations of alternating projections. Fig. 5 demonstrates that: (i) the GNC scheme typically produces accurate solutions to the TLS rotation search problem with $< 80\%$ outliers (a result that confirms the errors we observed in Fig. 4(c)); (ii) the optimality certification algorithm 3 can certify *all* the correct GNC solutions with enough iterations (green bars); (iii) even with only 200 iterations, the certification algorithm achieves success rate higher than 80%. In other words, we can have a fast certification if we are willing to sacrifice *completeness* (i.e., we possibly fail to certify an optimal solution as optimal). On average, the certification algorithm takes 80 iterations to obtain $< 0.1\%$ sub-optimality bound, where each iteration of alternating projections takes 60ms in MATLAB.

B. Benchmarking on Standard Datasets

Testing Setup. We benchmark TEASER and TEASER++ against two state-of-the-art robust registration techniques: *Fast Global Registration* (FGR) [52] and *Guaranteed Outlier Removal* (GORE) [21]. In addition, we test two RANSAC variants: a fast version where we terminate RANSAC after a maximum of 1,000 iterations (RANSAC1K) and a slow version where we terminate RANSAC after 60 seconds (RANSAC1min). We use four datasets, Bunny, Armadillo, Dragon, and Buddha, from the Stanford 3D Scanning Repository [88] and downsample them to $N = 100$ points. The tests below follow the same protocol of Section XI-A. Here we focus on the results on the Bunny dataset and we postpone the (qualitatively similar) results obtained on the other three datasets to Appendix R. Appendix R also provides an example of the performance of TEASER on registration problems with high noise ($\sigma = 0.1$), which further validates our findings.

Known Scale. We first evaluate the compared techniques with known scale $s = 1$. Fig. 6(a) shows the rotation error, translation error, and timing for increasing outlier rates on the Bunny dataset. From the rotation and translation errors, we note that TEASER, TEASER++, GORE, and RANSAC1min are robust against up to 90% outliers, although TEASER and

TEASER++ tend to produce more accurate estimates than GORE, and RANSAC1min typically requires over 10^5 iterations for convergence at 90% outlier rate. FGR can only resist 70% outliers and RANSAC1K starts breaking at 90% outliers. TEASER++’s performance is on par with TEASER for all outlier rates, which is expected from the observations in Section X. The timing subplot at the bottom of Fig. 6(a) shows that TEASER is impractical for real-time robotics applications. On the other hand, TEASER++ is one of the fastest techniques across the spectrum, and is able to solve problems with large number of outliers in less than 10ms on a standard laptop.

Extreme Outlier Rates. We further benchmark the performance of TEASER and TEASER++ under extreme outlier rates from 95% to 99% with known scale and $N = 1000$ correspondences on the Bunny. We replace RANSAC1K with RANSAC10K, since RANSAC1K already performs poorly at 90% outliers. Fig. 6(b) shows the boxplots of the rotation errors, translation errors, and timing. TEASER, TEASER++, and GORE are robust against up to 99% outliers, while RANSAC1min with 60s timeout can resist 98% outliers with about 10^6 iterations. RANSAC10K and FGR perform poorly under extreme outlier rates. While GORE, TEASER and TEASER++ are both robust against 99% outliers, TEASER, and TEASER++ produce lower estimation errors, with TEASER++ being one order of magnitude faster than GORE (bottom subfigure).

Unknown Scale. GORE is unable to solve for the scale, hence we only benchmark TEASER and TEASER++ against FGR,⁸ RANSAC1K, and RANSAC1min. Fig. 6(c) plots scale, rotation, translation error and timing for increasing outliers on the Bunny dataset. All the compared techniques perform well when the outlier ratio is below 60%. FGR has the lowest breakdown point and fails at 70%. RANSAC1K, TEASER, and TEASER++ only fail at 90% outlier ratio when the scale is unknown. Although RANSAC1min with 60s timeout outperforms other methods at 90% outliers, it typically requires more than 10^5 iterations to converge, which is not practical for real-time applications. TEASER++ consistently runs in less than 30ms.

C. Simultaneous Pose and Correspondences (SPC)

Here we provide a proof-of-concept of how to use TEASER++ in the case where we do not have correspondences. As mentioned in Section II, ICP and variants are the go-to solutions in these cases, but these techniques (i) require an initial guess (which is not available in many robotics applications), and (ii) their performance is very sensitive to the quality of the initial guess. In the following, we show that TEASER++ can provide a more reliable alternative to ICP.

Testing Setup. We obtain the source point cloud \mathcal{A} by downsampling the Bunny dataset to 100 points. Then, we create the point cloud \mathcal{B} by applying a random rotation and translation to \mathcal{A} . Finally, we remove a percentage of the points in \mathcal{B} to simulate *partial overlap* between \mathcal{A} and \mathcal{B} . For instance, when the overlap is 80%, we discard 20% (randomly chosen) points from \mathcal{B} . Note that the point clouds can have different sizes. We compare TEASER++ against ICP. We use the MATLAB [108]

⁸Although the original algorithm in [52] did not solve for the scale, we extend it by using Horn’s method to compute the scale at each iteration

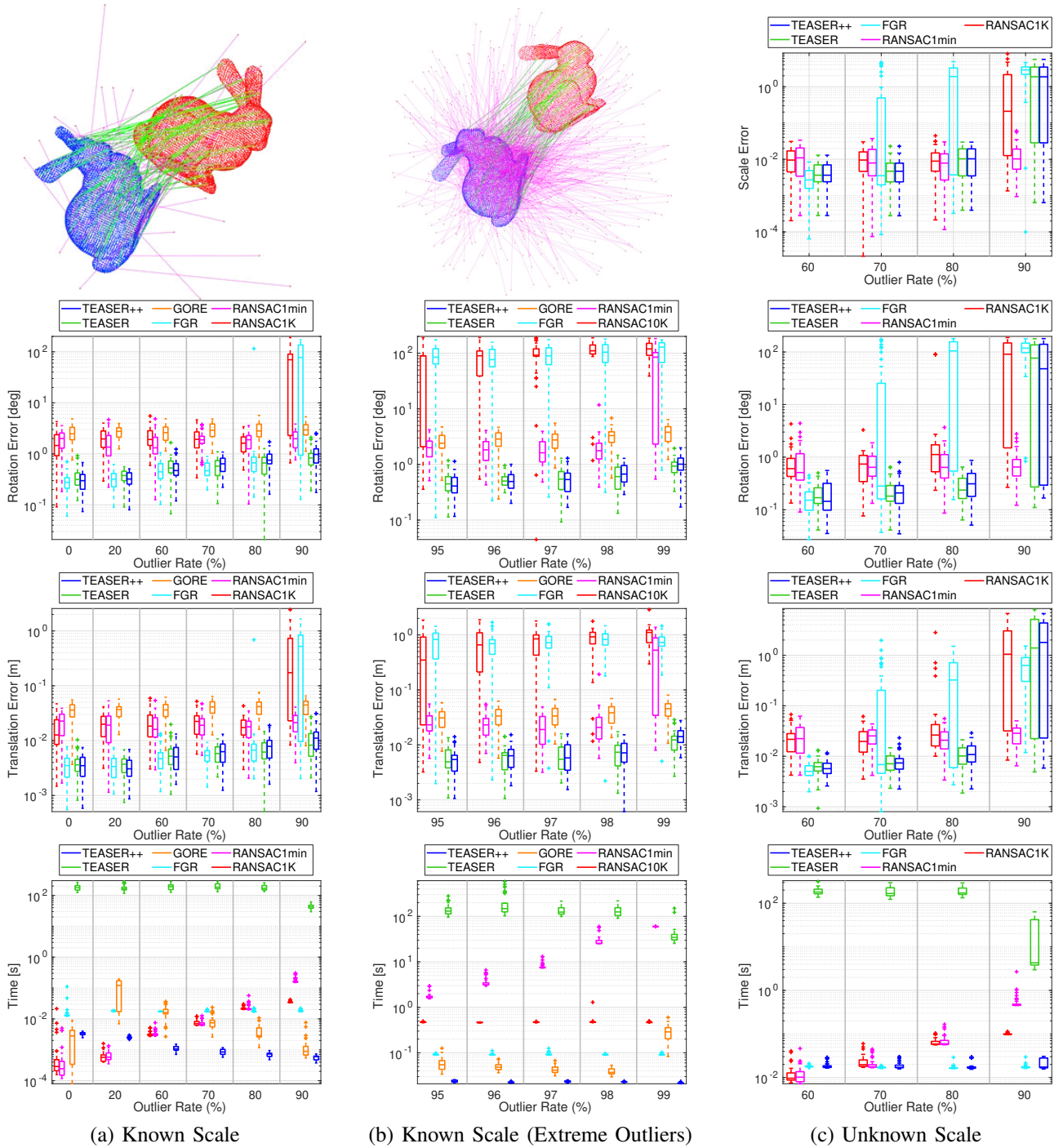


Fig. 6. Benchmark results. (a) Boxplots of rotation errors, translation errors, and timing for the six compared methods on the Bunny dataset with known scale (the top figure shows a registration example with 50% outlier correspondences). (b) Same as before, but for outlier rates between 95% and 99% (the top figure shows an example with 95% outlier correspondences). (c) Boxplots of scale, rotation, translation errors and timing for the five registration methods that support scale estimation on the Bunny dataset with unknown scale.

implementation of ICP with default parameters and the identity transformation as the initial guess. For TEASER++, we generate *all possible correspondences*: in other words, for each point in \mathcal{A} we add all points in \mathcal{B} as a potential correspondence (for a total of $|\mathcal{A}| \cdot |\mathcal{B}|$ correspondences). We then feed the correspondences to TEASER++ that computes a transformation without the need for an initial guess. Clearly, most of the correspondences fed to TEASER are outliers, but we rely on TEASER++ to find the small set of inliers.

Fig. 7(a)-(b) show the rotation and translation errors for

different levels of overlap between \mathcal{A} and \mathcal{B} . ICP fails to compute the correct transformation in practically all instances, since the initial guess is not in the basin of convergence of the optimal solution. TEASER++ *computes a correct solution across the spectrum without the need for an initial guess*. TEASER++ only starts failing when the overlap drops below 10%. The price we pay for this enhanced robustness is an increase in runtime. We feed $|\mathcal{A}| \cdot |\mathcal{B}| \approx 10^4$ correspondences to TEASER++, which increases the runtime compared to the correspondence-based setup in which the number of correspondences scales

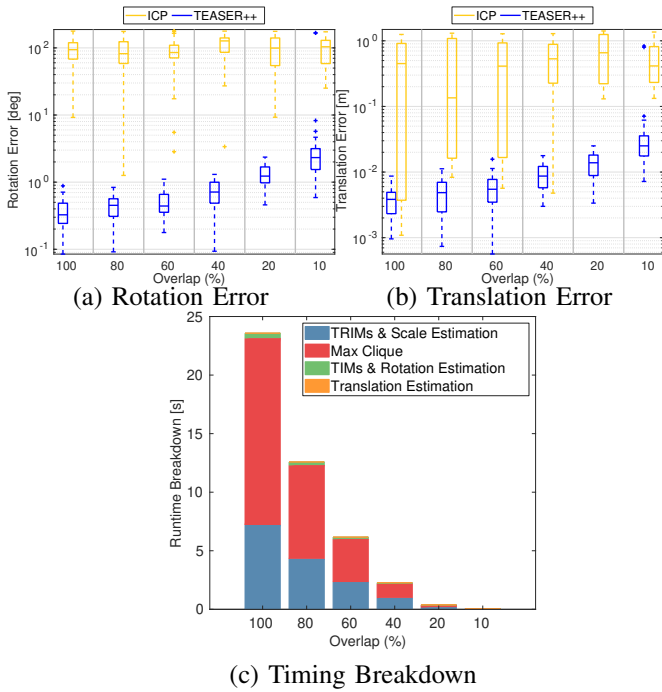


Fig. 7. (a)-(b) Rotation and translation errors for TEASER++ and ICP in a correspondence-free (SPC) problem. (c) Timing breakdown for TEASER++.

linearly (rather than quadratically) with the point cloud size. Fig. 7(c) reports the timing breakdown for the different modules in TEASER++. From the figure, we observe that (i) for small overlap, TEASER++ is not far from real-time, and (ii) the timing is dominated by the maximum clique computation and scale estimation, where the latter includes the computation of the translation and rotation invariant measurements (TRIMs).

D. Application 1: Object Pose Estimation and Localization

Testing Setup. We use the large-scale point cloud datasets from [30] to test TEASER in *object pose estimation and localization* applications. We first use the ground-truth object labels to extract the *cereal box/cap* out of the scene and treat it as the object, then apply a random transformation to the scene, to get an object-scene pair. To register the object-scene pair, we first use FPFH feature descriptors [1] to establish putative correspondences. Given correspondences from FPFH, TEASER is used to find the relative pose between the object and scene. We downsample the object and scene using the same ratio (about 0.1) to make the object have 2,000 points.

Results. Fig. 8 shows the noisy FPFH correspondences, the inlier correspondences obtained by TEASER, and successful localization and pose estimation of the *cereal box*. Another example is given in Fig. 1(g)-(h). Qualitative results for eight scenes are given in Appendix S. The inlier correspondence ratios for *cereal box* are all below 10% and typically below 5%. TEASER is able to compute a highly accurate estimate of the pose using a handful of inliers. Table I shows the mean and standard deviation (SD) of the rotation and translation errors, the number of FPFH correspondences, and the inlier ratio estimated by TEASER on the eight scenes.

E. Application 2: Scan Matching

Testing Setup. TEASER++ can also be used in robotics

applications that need robust scan matching, such as 3D reconstruction and loop closure detection in SLAM [109]. We evaluate TEASER++’s performance in such scenarios using the 3DMatch dataset [31], which consists of RGB-D scans from 62 real-world indoor scenes. The dataset is divided into 54 scenes for training, and 8 scenes for testing. The authors provide 5000 randomly sampled keypoints for each scan. On average, there are 205 pairs of scans per scene (maximum: 519 in the Kitchen scene, minimum: 54 in the Hotel 3 scene).

We use 3DSmoothNet [2], a state-of-the-art neural network, to compute local descriptors for each 3D keypoint, and generate correspondences using nearest-neighbor descriptor matching.⁹ We then feed the correspondences to TEASER++ and RANSAC (using an implementation in Open3D [110]) and compare their performance in terms of percentage of successfully matched scans and runtime. Two scans are successfully matched when the transformation computed by a technique has (i) rotation error smaller than 10° , and (ii) translation error less than 30 cm. We report RANSAC’s results with maximum number of iterations equal to 1,000 (RANSAC1K) and 10,000 (RANSAC10K). We also compare the percentage of successful registrations out of the cases where TEASER++ certified the rotation estimation as optimal, a setup we refer to as TEASER++ (CERT). The latter essentially evaluates the quality of the poses TEASER++ certified as optimal. We use $\beta_i = \beta = 5$ cm for TEASER++ in all tests.

Results. Table II shows the percentage of successfully matched scans and the average timing for the four compared techniques. TEASER dominates both RANSAC variants. RANSAC1K has a success rate up to 8% worst than TEASER++. RANSAC10K is an optimized C++ implementation and, while running slower than TEASER++, it cannot reach the same accuracy. These results further highlight that TEASER++ can be safely used as a faster and more robust replacement for RANSAC in SLAM pipelines. This conclusion is further reinforced by the last row in Table II, where we show the success rate for the poses certified as optimal by TEASER++. The success rate strictly dominates both RANSAC variants and TEASER++, since TEASER++ (CERT) is able to identify and reject unreliable registration results. This is a useful feature when scan matching is used for loop closure detection in SLAM, since bad registration results lead to incorrect loop closures and can compromise the quality of the resulting map (see [51], [97]). We omitted the timing result for TEASER++ (CERT) since our certification routine (Algorithm 3) is currently implemented in MATLAB (the certification time of our non-optimized MATLAB code is around 1 minute).

Why is TEASER++ not able to solve 100% of the tests in Table II? This should not come as a surprise from the statements in Theorem 15-17. *The estimation contracts discussed in Section IX require a minimum number of inliers:* TEASER++ can mine a small number of inliers among a large number of outliers, but it cannot solve problems where no inliers are given or where the inliers are not enough to identify a unique registration! For the experiments in this section, it is not uncommon to have no or fewer than 3 inliers in a scene

⁹3DSmoothNet descriptors have been shown to outperform FPFH in [2].



Fig. 8. Successful object pose estimation by TEASER on a real RGB-D dataset. Blue lines are the original FPFH [1] correspondences with outliers, green lines are the inlier correspondences computed by TEASER, and the final registered object is highlighted in red.

	Mean	SD
Rotation error [rad]	0.066	0.043
Translation error [m]	0.069	0.053
# of FPFH correspondences	525	161
FPFH inlier ratio [%]	6.53	4.59

TABLE I
REGISTRATION RESULTS ON EIGHT SCENES OF THE RGB-D DATASET [30].

Scenes	Kitchen (%)	Home 1 (%)	Home 2 (%)	Hotel 1 (%)	Hotel 2 (%)	Hotel 3 (%)	Study Room (%)	MIT Lab (%)	Avg. Runtime [ms]
RANSAC-1K	90.9	91.0	73.1	88.1	80.8	87.0	79.1	81.8	9.7
RANSAC-10K	96.4	92.3	73.1	92.0	84.6	90.7	82.2	81.8	96.5
TEASER++	97.7	92.3	82.7	96.9	88.5	94.4	88.7	84.4	85.1
TEASER++ (CERT)	99.2	97.5	90.0	98.8	94.9	97.7	94.8	93.9	-

TABLE II
PERCENTAGE OF CORRECT REGISTRATION RESULTS USING TEASER++, TEASER++ CERTIFIED, AND RANSAC ON THE 3DMatch DATASET.

due to the quality of the keypoint descriptors. Moreover, it is not uncommon to have symmetries in the scene, which make the registration non-unique. To intuitively highlight this issue, Fig. 9 shows the rotation errors of TEASER++, with different markers for certified (blue dots) and non-certified (red crosses) solutions. The figure shows that (i) TEASER++ (CERT) is able to reject a large number of incorrect estimates (red crosses with large errors), and (ii) some of the incorrect but certified solutions exhibit errors around 90° and 180° which correspond to symmetries of the scene. A visual example of the phenomenon is shown in Fig. 10. These results highlight the need for better keypoint detectors, since even state-of-the-art deep-learning-based methods struggle to produce acceptable outlier rates in real problems.

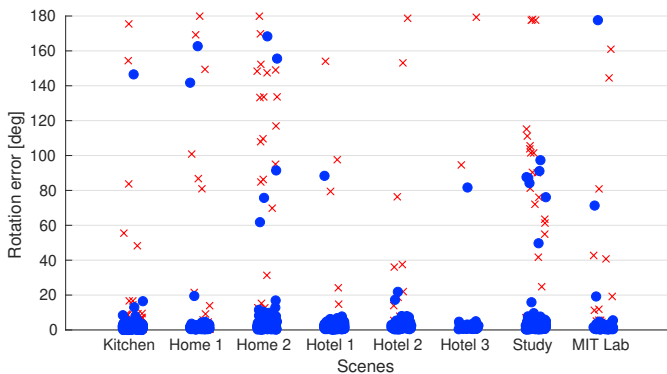


Fig. 9. Rotation errors for each scene (data points correspond to pairs of scans in the scene), with certified TEASER++ solutions (blue dots) vs. non-certified (red crosses).

XII. CONCLUSION

We propose the first fast and certifiable algorithm for correspondence-based registration with extreme outlier rates. We leverage insights from estimation theory (e.g., the unknown-but-bounded noise setup), geometry (e.g., invariant measurements), graph theory (e.g., maximum clique for inlier selection), and optimization (e.g., tight SDP relaxations). These insights lead to the design of two certifiable registration

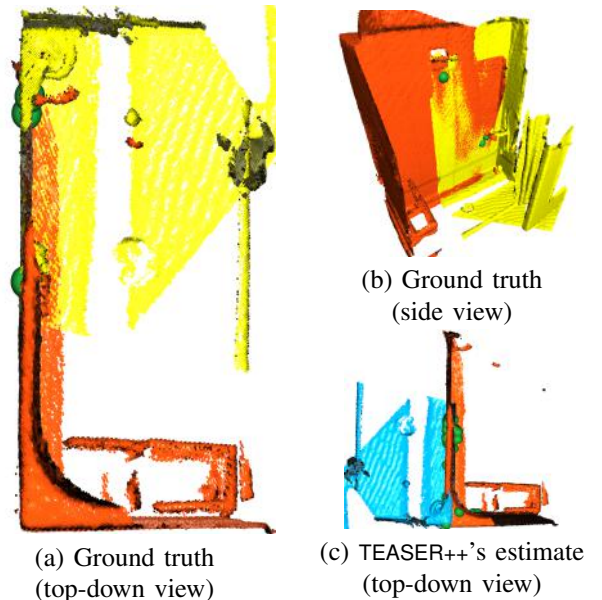


Fig. 10. Example from the Hotel 3 scene of the 3DMatch dataset [31]. The scene pictures a hotel shower where symmetries in the keypoint distribution cause TEASER++ to fail. (a)-(b) Ground-truth alignment (top view and side view). The ground-truth alignment admits 3 inlier correspondences (in green). (c) TEASER++'s estimate of the alignment. TEASER++ is able to find an estimate with 57 inliers. The symmetries of the scene allow for multiple potential registrations, and the ground truth inliers are not part of the maximum clique of the TIM graph. In this case, the max clique selected by TEASER++ contains an incorrect solution that differs from the ground truth by 180° .

algorithms. TEASER is accurate and robust but requires solving a large SDP. TEASER++ has similar performance in practice but circumvents the need to solve an SPD and can run in milliseconds. For both algorithms, we provide theoretical bounds on the estimation errors, which are the first of their kind for robust registration problems. Moreover, we test their performance on standard benchmarks, object detection datasets, and the 3DMatch scan matching dataset and show that (i) both algorithms dominate the state of the art (e.g., RANSAC, branch-&-bound, heuristics) and are robust to more than 99% outliers, (ii) TEASER++ can run in milliseconds and

it is currently the fastest robust registration algorithm at high outlier rates, (iii) TEASER++ is so robust it can also solve problems without correspondences (e.g., hypothesizing all-to-all correspondences) where it largely outperforms ICP. We release a fast open-source C++ implementation of TEASER++.

While not central to the technical contribution, this paper also establishes the foundations of *certifiable perception*, as discussed in Appendix A. This research area offers many research opportunities and the resulting progress has the potential to boost trustworthiness and reliability in safety-critical application of robotics and computer vision. Future work includes developing certifiable algorithms for other spatial perception problems (ideally leading to certifiable approaches for all the applications discussed in [97] and more). Another avenue for future research is the use of the certifiably robust algorithms presented in this paper for self-supervision of deep learning methods for keypoint detection and matching.

REFERENCES

- [1] R. Rusu, N. Blodow, and M. Beetz, “Fast point feature histograms (fpfh) for 3d registration,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. Citeseer, 2009, pp. 3212–3217.
- [2] Z. Gojcic, C. Zhou, J. D. Wegner, and A. Wieser, “The perfect match: 3d point cloud matching with smoothed densities,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5545–5554.
- [3] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, “Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments,” *Intl. J. of Robotics Research*, vol. 31, no. 5, pp. 647–663, 2012.
- [4] G. Blais and M. D. Levine, “Registering multiview range data to create 3d computer objects,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 8, pp. 820–824, 1995.
- [5] S. Choi, Q. Y. Zhou, and V. Koltun, “Robust reconstruction of indoor scenes,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5556–5565.
- [6] J. Zhang and S. Singh, “Visual-lidar odometry and mapping: Low-drift, robust, and fast,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 2174–2181.
- [7] B. Drost, M. Ulrich, N. Navab, and S. Ilic, “Model globally, match locally: Efficient and robust 3D object recognition,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 998–1005.
- [8] J. M. Wong, V. Kee, T. Le, S. Wagner, G. L. Mariottini, A. Schneider, L. Hamilton, R. Chipalkatty, M. Hebert, D. M. S. Johnson *et al.*, “Segicp: Integrated deep semantic segmentation and pose estimation,” in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 5784–5789.
- [9] A. Zeng, K. T. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez, and J. Xiao, “Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1386–1383.
- [10] P. Marion, P. R. Florence, L. Manuelli, and R. Tedrake, “Label fusion: A pipeline for generating ground truth labels for real rgbd data of cluttered scenes,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.
- [11] J. Bazin, Y. Seo, R. Hartley, and M. Pollefeys, “Globally optimal inlier set maximization with unknown rotation and focal length,” in *European Conf. on Computer Vision (ECCV)*, 2014, pp. 803–817.
- [12] M. A. Audette, F. P. Ferrie, and T. M. Peters, “An algorithmic overview of surface registration techniques for medical imaging,” *Med. Image Anal.*, vol. 4, no. 3, pp. 201–217, 2000.
- [13] G. K. L. Tam, Z. Q. Cheng, Y. K. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X. F. Sun, and P. L. Rosin, “Registration of 3d point clouds and meshes: a survey from rigid to nonrigid,” *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 7, pp. 1199–1217, 2013.
- [14] B. K. P. Horn, “Closed-form solution of absolute orientation using unit quaternions,” *J. Opt. Soc. Amer.*, vol. 4, no. 4, pp. 629–642, Apr 1987.
- [15] K. Arun, T. Huang, and S. Blostein, “Least-squares fitting of two 3-D point sets,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 9, no. 5, pp. 698–700, sept. 1987.
- [16] J. Briaes and J. Gonzalez-Jimenez, “Convex Global 3D Registration with Lagrangian Duality,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [17] F. Tombari, S. Salti, and L. D. Stefano, “Performance evaluation of 3d keypoint detectors,” *Intl. J. of Computer Vision*, vol. 102, no. 1-3, pp. 198–220, 2013.
- [18] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, “Aligning point cloud views using persistent feature histograms,” in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2008, pp. 3384–3391.
- [19] P. J. Besl and N. D. McKay, “A method for registration of 3-D shapes,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, no. 2, 1992.
- [20] M. Fischler and R. Bolles, “Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography,” *Commun. ACM*, vol. 24, pp. 381–395, 1981.
- [21] Á. Parra Bustos and T. J. Chin, “Guaranteed outlier removal for point cloud registration with correspondences,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 40, no. 12, pp. 2868–2882, 2018.
- [22] J. Yang, H. Li, D. Campbell, and Y. Jia, “Go-ICP: A globally optimal solution to 3D ICP point-set registration,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 38, no. 11, pp. 2241–2254, Nov. 2016.
- [23] A. Makadia, A. Patterson, and K. Daniilidis, “Fully automatic registration of 3d point clouds,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2006, pp. 1297–1304.
- [24] Y. Liu, C. Wang, Z. Song, and M. Wang, “Efficient global point cloud registration by matching rotation invariant features through translation search,” in *European Conf. on Computer Vision (ECCV)*, September 2018.
- [25] M. Milanese, *Estimation and Prediction in the Presence of Unknown but Bounded Uncertainty: A Survey*. Boston, MA: Springer US, 1989, pp. 3–24.
- [26] D. Bertsekas, “Control of uncertain systems with a set-membership description of the uncertainty,” Ph.D. dissertation, Massachusetts Institute of Technology, 1971.
- [27] R. Hartley and F. Kahl, “Global optimization through rotation space search,” *Intl. J. of Computer Vision*, vol. 82, no. 1, pp. 64–79, 2009.
- [28] G. Wahba, “A least squares estimate of satellite attitude,” *SIAM review*, vol. 7, no. 3, pp. 409–409, 1965.
- [29] I. Diakonikolas, G. Kamath, D. Kane, J. Li, A. Moitra, and A. Stewart, “Robust estimators in high dimensions without the computational intractability,” in *IEEE 57th Annual Symposium on Foundations of Computer Science*. IEEE, 2016, pp. 655–664.
- [30] K. Lai, L. Bo, X. Ren, and D. Fox, “A large-scale hierarchical multi-view RGB-D object dataset,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 1817–1824.
- [31] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, “3dmatch: Learning the matching of local 3d geometry in range scans,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, no. 2, 2017, p. 4.
- [32] H. Yang and L. Carlone, “A polynomial-time solution for robust registration with extreme outlier rates,” in *Robotics: Science and Systems (RSS)*, 2019, (pdf), (video), (media), (media).
- [33] —, “A quaternion-based certifiably optimal solution to the Wahba problem with outliers,” in *Intl. Conf. on Computer Vision (ICCV)*, 2019, (Oral Presentation, accept rate: 4%), Arxiv version: 1905.12536, (pdf).
- [34] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan, “3D object recognition in cluttered scenes with local surface features: a survey,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 36, no. 11, pp. 2270–2287, 2014.
- [35] H. Koppula, A. Anand, T. Joachims, and A. Saxena, “Semantic labeling of 3d point clouds for indoor scenes,” in *Advances in Neural Information Processing Systems (NIPS)*, 2011.
- [36] C. Olsson, F. Kahl, and M. Oskarsson, “Branch-and-bound methods for euclidean registration problems,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 31, no. 5, pp. 783–794, 2009.
- [37] F. L. Markley and J. L. Crassidis, *Fundamentals of spacecraft attitude determination and control*. Springer, 2014, vol. 33.
- [38] B. Horn, H. Hilden, and S. Negahdaripour, “Closed-form solution of absolute orientation using orthonormal matrices,” *J. Opt. Soc. Amer.*, vol. 5, no. 7, pp. 1127–1135, 1988.
- [39] F. L. Markley, “Attitude determination using vector observations and the singular value decomposition,” *The Journal of the Astronautical Sciences*, vol. 36, no. 3, pp. 245–258, 1988.
- [40] J. Forbes and A. de Ruiter, “Linear-matrix-inequality-based solution to Wahba’s problem,” *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 1, pp. 147–151, 2015.

- [41] J. Saunderson, P. Parrilo, and A. Willsky, "Semidefinite descriptions of the convex hull of rotation matrices," *SIAM J. OPTIM.*, vol. 25, no. 3, pp. 1314–1343, 2015.
- [42] J. Gower and G. Dijksterhuis, "Procrustes problems," *Procrustes Problems, Oxford Statistical Science Series*, vol. 30, 01 2005.
- [43] P. Schonemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, pp. 1–10, 1966.
- [44] Y. Cheng and J. L. Crassidis, "A total least-squares estimate for attitude determination," in *AIAA Scitech 2019 Forum*, 2019, p. 1176.
- [45] S. Ahmed, E. C. Kerrigan, and I. M. Jaimoukha, "A semidefinite relaxation-based algorithm for robust attitude estimation," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 3942–3952, 2012.
- [46] C. S. Chen, Y. P. Hung, and J. B. Cheng, "RANSAC-based DARCES: A new approach to fast automatic registration of partially overlapping range images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, no. 11, pp. 1229–1234, 1999.
- [47] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [48] P. Meer, D. Mintz, A. Rosenfeld, and D. Y. Kim, "Robust regression methods for computer vision: A review," *Intl. J. of Computer Vision*, vol. 6, no. 1, pp. 59–70, Apr 1991.
- [49] K. M. Tavish and T. D. Barfoot, "At all costs: A comparison of robust cost functions for camera correspondence outliers," in *Computer and Robot Vision (CRV), 2015 12th Conference on*. IEEE, 2015, pp. 62–69.
- [50] M. J. Black and A. Rangarajan, "On the unification of line processes, outlier rejection, and robust statistics with applications in early vision," *Intl. J. of Computer Vision*, vol. 19, no. 1, pp. 57–91, 1996.
- [51] P. Lajoie, S. Hu, G. Beltrame, and L. Carlone, "Modeling perceptual aliasing in SLAM via discrete-continuous graphical models," *IEEE Robotics and Automation Letters (RA-L)*, 2019, extended ArXiv version: (pdf), Supplemental Material: (pdf).
- [52] Q. Zhou, J. Park, and V. Koltun, "Fast global registration," in *European Conf. on Computer Vision (ECCV)*. Springer, 2016, pp. 766–782.
- [53] A. Parra Bustos, T.-J. Chin, F. Neumann, T. Friedrich, and M. Katzmann, "A practical maximum clique algorithm for matching with pairwise constraints," *arXiv preprint arXiv:1902.01534*, 2019.
- [54] T. J. Chin and D. Suter, "The maximum consensus problem: recent algorithmic advances," *Synthesis Lectures on Computer Vision*, vol. 7, no. 2, pp. 1–194, 2017.
- [55] F. Wen, R. Ying, Z. Gong, and P. Liu, "Efficient algorithms for maximum consensus robust fitting," *IEEE Trans. Robotics*, 2019.
- [56] Z. Cai, T.-J. Chin, and V. Koltun, "Consensus maximization tree search revisited," in *Intl. Conf. on Computer Vision (ICCV)*, 2019, pp. 1637–1645.
- [57] H. M. Le, T.-J. Chin, A. Eriksson, T.-T. Do, and D. Suter, "Deterministic approximate methods for maximum consensus robust fitting," *IEEE Trans. Pattern Anal. Machine Intell.*, 2019.
- [58] D. Campbell, L. Petersson, L. Kneip, and H. Li, "Globally-optimal inlier set maximisation for simultaneous camera pose and feature correspondence," in *Intl. Conf. on Computer Vision (ICCV)*, 2017, pp. 1–10.
- [59] J. C. Bazin, Y. Seo, and M. Pollefeys, "Globally optimal consensus set maximization through rotation search," in *Asian Conference on Computer Vision*. Springer, 2012, pp. 539–551.
- [60] C. Olsson, O. Enqvist, and F. Kahl, "A polynomial-time bound for matching and registration with outliers," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2008, pp. 1–8.
- [61] O. Enqvist, E. Ask, F. Kahl, and K. Åström, "Robust fitting for multiple view geometry," in *European Conf. on Computer Vision (ECCV)*. Springer, 2012, pp. 738–751.
- [62] S. Granger and X. Pennec, "Multi-scale EM-ICP: A fast and robust approach for surface registration," in *European Conf. on Computer Vision (ECCV)*, 2002.
- [63] R. Sandhu, S. Dambreville, and A. Tannenbaum, "Point set registration via particle filtering and stochastic dynamics," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 32, no. 8, pp. 1459–1473, 2010.
- [64] C. Zhang, S. Du, J. Liu, and J. Xue, "Robust 3d point set registration using iterative closest point algorithm with bounded rotation angle," *Signal Processing*, vol. 120, pp. 777–788, 2016.
- [65] L. Maier-Hein, A. M. Franz, T. R. dos Santos, M. Schmidt, M. Fangerau, H. P. Meinzer, and J. M. Fitzpatrick, "Convergent iterative closest-point algorithm to accomodate anisotropic and inhomogenous localization error," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 34, no. 8, pp. 1520–1532, 2012.
- [66] D. Chetverikov, D. Stepanov, and P. Krsek, "Robust euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm," *Image and Vision Computing*, vol. 23, no. 3, pp. 299–309, 2005.
- [67] S. Kaneko, T. Kondo, and A. Miyamoto, "Robust matching of 3D contours using iterative closest point algorithm improved by M-estimation," *Pattern Recognition*, vol. 36, no. 9, pp. 2041–2047, 2003.
- [68] S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," in *ACM Symp. Geom. Process.* Eurographics Association, 2013, pp. 113–123.
- [69] B. Jian and B. C. Vemuri, "A robust algorithm for point set registration using mixture of gaussians," in *Intl. Conf. on Computer Vision (ICCV)*, vol. 2. IEEE, 2005, pp. 1246–1251.
- [70] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 32, no. 12, pp. 2262–2275, 2010.
- [71] B. Jian and B. C. Vemuri, "Robust point set registration using gaussian mixture models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 33, no. 8, pp. 1633–1645, 2011.
- [72] D. Campbell and L. Petersson, "An adaptive data representation for robust point-set registration and merging," in *Intl. Conf. on Computer Vision (ICCV)*, 2015, pp. 4292–4300.
- [73] W. Clark, M. Ghaffari, and A. Bloch, "Nonparametric continuous sensor registration," *arXiv preprint arXiv:2001.04286*, 2020.
- [74] T. M. Breuel, "Implementation techniques for geometric branch-and-bound matching methods," *Comput. Vis. Image Underst.*, vol. 90, no. 3, pp. 258–294, 2003.
- [75] D. Campbell and L. Petersson, "Gogma: Globally-optimal gaussian mixture alignment," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5685–5694.
- [76] Á. Parra Bustos, T. J. Chin, and D. Suter, "Fast rotation search with stereographic projections for 3d registration," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 3930–3937.
- [77] X. Li, Y. Liu, Y. Wang, C. Wang, M. Wang, and Z. Song, "Fast and globally optimal rigid registration of 3d point sets by transformation decomposition," *arXiv preprint arXiv:1812.11307*, 2019.
- [78] G. Izatt, H. Dai, and R. Tedrake, "Globally optimal object pose estimation in point clouds with mixed-integer programming," in *Proc. of the Intl. Symp. of Robotics Research (ISRR)*, 2017.
- [79] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3D classification and segmentation," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 652–660.
- [80] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 5, p. 146, 2019.
- [81] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, "Pointnetlk: Robust & efficient point cloud registration using PointNet," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7163–7172.
- [82] Y. Wang and J. M. Solomon, "Deep Closest Point: Learning Representations for Point Cloud Registration," in *Intl. Conf. on Computer Vision (ICCV)*, 2019.
- [83] —, "PRNet: Self-Supervised Learning for Partial-to-Partial Registration," in *Advances in Neural Information Processing Systems (NIPS)*, 2019, pp. 8812–8824.
- [84] A. Avetisyan, M. Dahner, A. Dai, M. Savva, A. X. Chang, and M. Nießner, "Scan2CAD: Learning CAD model alignment in RGB-D scans," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2614–2623.
- [85] A. Avetisyan, A. Dai, and M. Nießner, "End-to-End CAD Model Retrieval and 9DoF Alignment in 3D Scans," in *Intl. Conf. on Computer Vision (ICCV)*, 2019.
- [86] T.-Y. Liu and H. Jiang, "Minimizing sum of truncated convex functions and its applications," *Journal of Computational and Graphical Statistics*, vol. 28, no. 1, pp. 1–10, 2019.
- [87] S. Barratt, G. Angeris, and S. Boyd, "Minimizing a sum of clipped convex functions," *arXiv preprint arXiv:1910.12342*, 2019.
- [88] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *SIGGRAPH*, 1996, pp. 303–312.
- [89] S. Agarwal, V. Shree, and S. Chakraborty, "RFM-SLAM: Exploiting relative feature measurements to separate orientation and position estimation in slam," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 6307–6314.
- [90] L. Carlone, R. Aragues, J. Castellanos, and B. Bona, "A fast and accurate approximation for planar pose graph optimization," *Intl. J. of Robotics Research*, vol. 33, no. 7, pp. 965–987, 2014, (pdf) (ppt) (code) (video) (datasets: (web)).

- [91] C. Bron and J. Kerbosch, "Algorithm 457: finding all cliques of an undirected graph," *Communications of the ACM*, vol. 16, no. 9, pp. 575–577, 1973.
- [92] B. Pattabiraman, M. M. A. Patwary, A. H. Gebremedhin, W. K. Liao, and A. Choudhary, "Fast algorithms for the maximum clique problem on massive graphs with applications to overlapping community detection," *Internet Mathematics*, vol. 11, no. 4–5, pp. 421–448, 2015.
- [93] Q. Wu and J. Hao, "A review on algorithms for maximum clique problems," *European Journal of Operational Research*, vol. 242, no. 3, pp. 693–709, 2015.
- [94] D. Eppstein, M. Löffler, and D. Strash, "Listing all maximal cliques in sparse graphs in near-optimal time," in *International Symposium on Algorithms and Computation*. Springer, 2010, pp. 403–414.
- [95] D. Scaramuzza, "1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints," *Intl. J. of Computer Vision*, pp. 1–12, 2011.
- [96] M. ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 8.1.*, 2017. [Online]. Available: <http://docs.mosek.com/8.1/toolbox/index.html>
- [97] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone, "Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection," *IEEE Robotics and Automation Letters (RA-L)*, 2020, (pdf).
- [98] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.
- [99] D. Henrion and J. Malick, "Projection methods in conic optimization," in *Handbook on Semidefinite, Conic and Polynomial Optimization*. Springer, 2012, pp. 565–600.
- [100] H. H. Bauschke and J. M. Borwein, "On projection algorithms for solving convex feasibility problems," *SIAM review*, vol. 38, no. 3, pp. 367–426, 1996.
- [101] Y. Nesterov, "A method for solving the convex programming problem with convergence rate $o(1/k^2)$," *Soviet Mathematics Doklady*, vol. 27, pp. 372–376, 1983.
- [102] OpenMP Architecture Review Board, "OpenMP application program interface version 3.0," May 2008. [Online]. Available: <http://www.openmp.org/mp-documents/spec30.pdf>
- [103] R. A. Rossi, D. F. Gleich, and A. H. Gebremedhin, "Parallel maximum clique algorithms with applications to network analysis," *SIAM Journal on Scientific Computing*, vol. 37, no. 5, pp. C589–C616, 2015.
- [104] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.
- [105] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming," [Online]. Available: <http://cvxr.com/cvx>
- [106] P. Speciale, D. P. Paudel, M. R. Oswald, T. Kroeger, L. V. Gool, and M. Pollefeys, "Consensus maximization with linear matrix inequality constraints," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 5048–5056.
- [107] R. Hartley, J. Trumpf, Y. Dai, and H. Li, "Rotation averaging," *IJCV*, vol. 103, no. 3, pp. 267–305, 2013.
- [108] MATLAB, version 9.6.0 (R2019a). Natick, Massachusetts: The MathWorks Inc., 2019.
- [109] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016, arxiv preprint: 1606.05830, (pdf).
- [110] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.
- [111] V. Tzoumas, P. Antonante, and L. Carlone, "Outlier-robust spatial perception: Hardness, general-purpose algorithms, and guarantees," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019, extended arxiv version: 1903.11683, (pdf).
- [112] T.-J. Chin, Z. Cai, and F. Neumann, "Robust fitting in computer vision: Easy or hard?" in *European Conf. on Computer Vision (ECCV)*, 2018.
- [113] A. Bandeira, "A note on probably certifiably correct algorithms," *arXiv:1509.00824*, 2015.
- [114] L. Carlone and F. Dellaert, "Duality-based verification techniques for 2D SLAM," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2015, pp. 4589–4596, (pdf) (code).
- [115] D. Rosen, L. Carlone, A. Bandeira, and J. Leonard, "SE-Sync: a certifiably correct algorithm for synchronization over the Special Euclidean group," *Intl. J. of Robotics Research*, 2018, accepted, arxiv preprint: 1611.00128, (pdf).
- [116] Q. M. Rahman, N. Sünderhauf, and F. Dayoub, "Did you miss the sign? A false negative alarm system for traffic sign detectors," *CoRR*, vol. abs/1903.06391, 2019. [Online]. Available: <http://arxiv.org/abs/1903.06391>
- [117] S. Seshia and D. Sadigh, "Towards verified artificial intelligence," *ArXiv*, vol. abs/1606.08514, 2016.
- [118] A. Desai, T. Dreossi, and S. Seshia, "Combining model checking and runtime verification for safe robotics," in *International Conference on Runtime Verification*. Springer, 2017, pp. 172–189.
- [119] F. Chung, *Spectral Graph Theory*. American Mathematical Soc., CBMS Regional Conference Series in Mathematics, No. 92, 1996.
- [120] W. Cheney and A. A. Goldstein, "Proximity maps for convex sets," *Proceedings of the American Mathematical Society*, vol. 10, no. 3, pp. 448–450, 1959.
- [121] N. J. Higham, "Computing a nearest symmetric positive semidefinite matrix," *Linear algebra and its applications*, vol. 103, pp. 103–118, 1988.



Heng Yang is a PhD candidate in the Department of Mechanical Engineering and the Laboratory for Information & Decision Systems (LIDS) at the Massachusetts Institute of Technology, where he is working with Prof. Luca Carlone at the SPARK Lab. He has obtained a B.S. degree in Mechanical Engineering (with honors) from the Tsinghua University, Beijing, China, in 2015; and an S.M. degree in Mechanical Engineering from the Massachusetts Institute of Technology in 2017. His research interests include convex optimization, semidefinite and sums-of-squares relaxation, robust estimation and machine learning, applied to robotic perception and computer vision. His work includes developing robust and certifiable algorithms for geometric understanding.



Jingnan Shi is a Master student in the Department of Aeronautics and Astronautics at the Massachusetts Institute of Technology. He is currently a member of the SPARK lab, led by Professor Luca Carlone. He has obtained a B.S. degree in Engineering from Harvey Mudd College in 2019. His research interests include robust perception and estimation, with applications specific to various robotic systems.



Luca Carlone is the *Charles Stark Draper* Assistant Professor in the Department of Aeronautics and Astronautics at the Massachusetts Institute of Technology, and a Principal Investigator in the Laboratory for Information & Decision Systems (LIDS). He has obtained a B.S. degree in mechatronics from the Polytechnic University of Turin, Italy, in 2006; an S.M. degree in mechatronics from the Polytechnic University of Turin, Italy, in 2008; an S.M. degree in automation engineering from the Polytechnic University of Milan, Italy, in 2008; and a Ph.D. degree in robotics also the Polytechnic University of Turin in 2012. He joined LIDS as a postdoctoral associate (2015) and later as a Research Scientist (2016), after spending two years as a postdoctoral fellow at the Georgia Institute of Technology (2013–2015). His research interests include nonlinear estimation, numerical and distributed optimization, and probabilistic inference, applied to sensing, perception, and decision-making in single and multi-robot systems. His work includes seminal results on certifiably correct algorithms for localization and mapping, as well as approaches for visual-inertial navigation and distributed mapping. He is a recipient of the 2017 Transactions on Robotics King-Sun Fu Memorial Best Paper Award, the Best Paper award at WAFR 2016, the Best Student Paper award at the 2018 Symposium on VLSI Circuits, and was best paper finalist at RSS 2015.

APPENDICES

A. Manifesto of Certifiable Perception

This section provides a broader context for the algorithms developed in this paper, and motivates our efforts towards perception algorithms with formal performance guarantees.

What is a certifiable algorithm? Outlier-robust estimation is a hard combinatorial problem, where one has to separate inliers from outliers while computing an estimate for the variables of interest. Many algorithms, such as RANSAC and ICP, are heuristic solvers for such a combinatorial problem [54]. These algorithms are not certifiable (formalized later) in the sense that the estimate they return can be arbitrarily far from the optimal solution of the combinatorial problem, and there is no easy way to check how suboptimal an estimate is.

Then a natural question is: *can we directly compute an optimal solution for outlier-robust estimation or develop algorithms that always compute near-optimal solutions?* Unfortunately, in general the answer is no. A recent set of papers [111], [112] has shown that a broad family of robust estimation problems (including Consensus Maximization [54] and the Truncated Least Squares formulation considered in this paper) are *inapproximable* in polynomial-time: there exist worst-case instances in which no polynomial-time algorithm can compute a near-optimal solution (see [112] for a more formal treatment and [111] for even more pessimistic inapproximability results ruling out our quasi-polynomial-time algorithms).

This fundamental intractability calls for a paradigm shift: since it is not possible to solve every robust estimation problem in polynomial time, we claim that a useful goal is to design algorithms that perform well in typical instances and are able to certify the correctness of the resulting estimates, but at the same time can detect worst-case instances and declare “failure” on those rather than blindly returning an invalid estimate.

We formalize this notion in the definition below.

Definition 19 (Certifiable Algorithms). *Given an optimization problem $\mathbb{P}(\mathbb{D})$ that depends on input data \mathbb{D} , we say that an algorithm \mathbb{A} is certifiable if, after solving $\mathbb{P}(\mathbb{D})$, algorithm \mathbb{A} either provides a certificate for the quality of its solution (e.g., a proof of optimality, a finite bound on its sub-optimality, or a finite bound on the distance of the estimate from the optimal solution), or declares failure otherwise.*

The notion of certifiable algorithms is inspired by (and is indeed a particularization of) the notion of *Probably Certifiably Correct (PCC) Algorithm* introduced by Bandeira in [113]. We observe that a certifiable algorithm is *sound* (the algorithm does not certify incorrect solutions), but is not necessarily *complete* (the algorithm may declare failure in a problem that can be solved in polynomial time by a different algorithm). In this paper, we provide the first certifiable algorithm for registration (and for rotation search, one of the subproblems we encounter) and what we believe are the first certifiable algorithms for outlier-robust estimation in robotics and vision. We sometimes refer to the algorithms in Definition 19 as *certifiably robust* to stress that in outlier-robust estimation the notion of optimality has implications on the robustness to outliers of the resulting estimate. However, the definition also

applies to hard outlier-free problems such as [114], [115].

When applied to estimation problems in robotics or vision, we note that an optimal solution is not necessarily a “good” solution (*i.e.*, close to the ground truth for the quantity we want to estimate). For instance, if the data \mathbb{D} we feed to the algorithm is completely random and uncorrelated with our unknown, solving $\mathbb{P}(\mathbb{D})$ would not bring us any closer to knowing the value of the unknown. In this sense, any meaningful performance guarantee will need to take assumptions on the generative model of the data. We refer to the corresponding theoretical results as “estimation contracts”, which, informally, say that as long as the data is informative about the ground truth, a certifiable algorithm can produce an estimate close to the ground truth. In other words, while a certifiable algorithm is one that can assess if it completed the assigned task to optimality, an estimation contract ensures that optimality is useful towards estimating a variable of interest.

Why certifiable algorithms? Besides being an intellectual pursuit towards the design of better algorithms for robot perception, we believe that the current adoption of algorithms that can fail without notice is a major cause of brittleness in modern robotics applications, ranging from self-driving cars to autonomous drones. This is true across spatial perception applications, including object detection [116] and Simultaneous Localization and Mapping [109]. A certifiable algorithm can detect failures before they cascade to other modules in the autonomy stack. Moreover, we hope that the development of estimation contracts can pave the way towards formal verification and monitoring of complex perception systems involving multiple modules and algorithms, establishing connections with parallel efforts on safe autonomy and decision-making, *e.g.*, [117], [118].

B. Choice of β_i and \bar{c}

The parameters β_i and \bar{c} are straightforward to set in (7). In the following we discuss how to set them depending on the assumptions we make on the *inlier* noise.

Remark 20 (Probabilistic inlier noise). *If we assume the inliers follow the generative model (5) with $\epsilon_i \sim \mathcal{N}(\mathbf{0}_3, \sigma_i^2 \mathbf{I}_3)$ and $\mathbf{o}_i = \mathbf{0}$, it holds:*

$$\frac{1}{\sigma_i^2} \|\mathbf{b}_i - \mathbf{R}\mathbf{a}_i\|^2 = \frac{1}{\sigma_i^2} \|\epsilon_i\|^2 \sim \chi^2(3), \quad (30)$$

where $\chi^2(3)$ is the Chi-squared distribution with three degrees of freedom. Therefore, with desired probability p , the weighted error $\frac{1}{\sigma_i^2} \|\epsilon_i\|^2$ for the inliers satisfies:

$$\mathbb{P}\left(\frac{\|\epsilon_i\|^2}{\sigma_i^2} \leq \gamma^2\right) = p, \quad (31)$$

where γ^2 is the quantile of the χ^2 distribution with three degrees of freedom and lower tail probability equal to p (e.g., $\gamma = 3$ for $p = 0.97$). Therefore, one can simply set the noise bound β_i in Problem (7) to be $\beta_i = \sigma_i$, set the $\bar{c} = \gamma$. As mentioned in Section IV, from optimization standpoint, this is equivalent to setting $\beta_i = \gamma\sigma_i$ and setting $\bar{c} = 1$. The parameter γ monotonically increases with p ; therefore, setting p close to 1 makes the formulation (7) more prone to accept

measurements with large residuals, while a small p makes (7) more selective.

Remark 21 (Set membership inlier noise). If we assume the inliers follow the generative model (5) with unknown-but-bounded noise $\|\epsilon_i\| \leq \beta_i$, where β_i is a given noise bound,¹⁰ it is easy to see that the inliers satisfy:

$$\begin{aligned} \|\epsilon_i\| \leq \beta_i &\iff \|\mathbf{b}_i - \mathbf{R}\mathbf{a}_i\|^2 \leq \beta_i^2 \\ &\iff \frac{1}{\beta_i^2} \|\mathbf{b}_i - \mathbf{R}\mathbf{a}_i\|^2 \leq 1 \end{aligned} \quad (32)$$

Therefore, we directly plug β_i into (7) and choose $\bar{c} = 1$.

C. Truncated Least Squares vs. Consensus Maximization

Truncated Least Squares (and Algorithm 2) are related to Consensus Maximization (MC), a popular approach for outlier rejection in vision [106], [24]. Consensus Maximization looks for an estimate that maximizes the number of inliers (or, equivalently, minimizes the number of outliers):

$$\min_{\substack{\mathcal{O} \subseteq \mathcal{M} \\ \mathbf{x} \in \mathcal{X}}} |\mathcal{O}|, \text{ s.t. } \|\mathbf{r}_i(\mathbf{x})\|^2 \leq \bar{c}^2 \quad \forall i \in \mathcal{M} \setminus \mathcal{O} \quad (33)$$

where \mathbf{x} is the variable we want to estimate (possibly belonging to some domain \mathcal{X}), \mathcal{M} is the available set of measurements, $\mathbf{r}_i(\cdot)$ is a given residual error function, \bar{c} is the maximum admissible error for an inlier, and $|\cdot|$ denotes the cardinality of a set. Problem (33) looks for the smallest set of outliers (\mathcal{O}) such that all the other measurements (i.e., the inliers $\mathcal{M} \setminus \mathcal{O}$) have residual error below \bar{c} for some \mathbf{x} . While Consensus Maximization is intractable in general, by following the same lines of Theorem 7, it is easy to show it can be solved in polynomial time in the scalar case.

MC and TLS do not return the same solution in general, since TLS may prefer discarding measurements that induce a large bias in the estimate, as shown by the simple example below.

Example 22 (TLS \neq MC). Consider a simple scalar estimation problem, where we are given three measurements, and compare the two formulations:

$$\min_s \sum_{k \in \mathcal{M}} \min((s - s_k)^2, \bar{c}^2) \quad (\text{TLS}) \quad (34)$$

$$\min_{\mathcal{O} \subseteq \mathcal{M}, s} |\mathcal{O}| \text{ s.t. } (s - s_k)^2 \leq \bar{c}^2 \quad \forall k \in \mathcal{M} \setminus \mathcal{O} \quad (\text{MC}) \quad (35)$$

where $\mathcal{M} = \{1, 2, 3\}$. Assume $s_1 = s_2 = 0$, $s_3 = 3$, and $\bar{c} = 2$. Then, it is possible to see that $s_{\text{MC}} = 1.5$ attains a maximum consensus set including all measurements $\{1, 2, 3\}$, while the TLS estimate is $\hat{s} = 0$ which attains an optimal cost equal to 2, and has consensus set $\mathcal{I}(\hat{s}) = \{1, 2\}$.

Note that Consensus Maximization is not necessarily preferable over TLS: indeed (i) a human might have arguably preferred to flag $s_3 = 3$ as an outlier in the toy example above, and more importantly (ii) in the experiments in Section XI-A we observe that often Consensus Maximization has a lower breakdown point than TLS (intuitively, TLS is more sensitive to the distribution of the inliers, since it penalizes their residuals).

While MC and TLS do not choose the same set of inliers in general, in cases where there is a large set of inliers, we expect TLS and MC to produce similar solutions, as shown below.

Lemma 23 (Necessary Conditions for TLS \equiv MC). Assume the maximum consensus set returned by Consensus Maximization (MC) has size $N_{\text{in}}^{\text{MC}}$ and the sum of the squared residual errors for the inliers is $r_{\text{in}}^{\text{MC}}$. If the second largest consensus set has size smaller than $N_{\text{in}}^{\text{MC}} - r_{\text{in}}^{\text{MC}}/\bar{c}^2$, then TLS also selects the maximum consensus set as inliers.

Proof: The lemma establishes a general condition under which the set computed by (33) matches the set of inliers found by the TLS formulation:

$$\min_{\mathbf{x} \in \mathcal{X}} \sum_{i \in \mathcal{M}} \min(\|\mathbf{r}_i(\mathbf{x})\|^2, \bar{c}^2) \quad (36)$$

For the TLS formulation, the inliers are the measurements that—at the optimal solution—have residual smaller than \bar{c} . We denote as $f_{\text{TLS}}(\mathcal{I})$ the value of the TLS cost in (36) for a given choice of the consensus set \mathcal{I} .

Denote the size of the measurement set as N , i.e., $N \doteq |\mathcal{M}|$. If \mathcal{O}_{MC} is the optimal solution of (33), we define $N_{\text{out}}^{\text{MC}} \doteq |\mathcal{O}_{\text{MC}}|$. Lemma 23 denotes the number of inliers found by Consensus Maximization as $N_{\text{in}}^{\text{MC}} \doteq |\mathcal{M} \setminus \mathcal{O}| = N - N_{\text{out}}^{\text{MC}}$. Moreover, Lemma 23 denotes the sum of the squared residual errors for the inliers as:

$$r_{\text{in}}^{\text{MC}} \doteq \sum_{i \in \mathcal{M} \setminus \mathcal{O}_{\text{MC}}} \|\mathbf{r}_i(\mathbf{x})\|^2 \quad (37)$$

Then the lemma claims that if the size of the second largest consensus set is smaller than $N_{\text{in}}^{\text{MC}} - r_{\text{in}}^{\text{MC}}/\bar{c}^2$, TLS returns the maximum consensus set as inliers.

Assume by contradiction that the maximum consensus set \mathcal{I}_{MC} leads to a suboptimal TLS solution. Then, there exists another TLS solution with consensus set \mathcal{I}' such that:

$$f_{\text{TLS}}(\mathcal{I}') < N_{\text{out}}^{\text{MC}} \bar{c}^2 + r_{\text{in}}^{\text{MC}} \quad (38)$$

which follows from the assumption that the solution corresponding to \mathcal{I}_{MC} (attaining cost $N_{\text{out}}^{\text{MC}} \bar{c}^2 + r_{\text{in}}^{\text{MC}}$) is suboptimal.

Now, if we call $N'_{\text{in}} = |\mathcal{I}'|$ and define $N'_{\text{out}} = N - N'_{\text{in}}$, the assumption that any consensus set other than \mathcal{I}_{MC} has size smaller than $N_{\text{in}}^{\text{MC}} - r_{\text{in}}^{\text{MC}}/\bar{c}^2$ implies:

$$\begin{aligned} N'_{\text{in}} &< N_{\text{in}}^{\text{MC}} - r_{\text{in}}^{\text{MC}}/\bar{c}^2 &\iff \\ N - N'_{\text{out}} &< N - N_{\text{out}}^{\text{MC}} - r_{\text{in}}^{\text{MC}}/\bar{c}^2 &\iff \\ N'_{\text{out}} &> N_{\text{out}}^{\text{MC}} + r_{\text{in}}^{\text{MC}}/\bar{c}^2 \end{aligned} \quad (39)$$

Using (39) and recalling that the residual error for the inliers is nonnegative (it is a sum of squares):

$$f_{\text{TLS}}(\mathcal{I}') \geq N'_{\text{out}} \bar{c}^2 > (N_{\text{out}}^{\text{MC}} + r_{\text{in}}^{\text{MC}}/\bar{c}^2) \bar{c}^2 = N_{\text{out}}^{\text{MC}} \bar{c}^2 + r_{\text{in}}^{\text{MC}} \quad (40)$$

We conclude the lemma by observing that (38) and (40) cannot be simultaneously satisfied, leading to contradiction. ■

¹⁰This is the typical setup assumed in *set membership estimation* [25].

Measurements	Points	TIMs	TRIMs
Symbol	$\mathbf{a}_i, \mathbf{b}_i$	$\bar{\mathbf{a}}_{ij}, \bar{\mathbf{b}}_{ij}$	s_{ij}
Definition	-	$\begin{cases} \bar{\mathbf{a}}_{ij} = \mathbf{a}_j - \mathbf{a}_i \\ \bar{\mathbf{b}}_{ij} = \mathbf{b}_j - \mathbf{b}_i \end{cases}$	$s_{ij} = \frac{\ \bar{\mathbf{b}}_{ij}\ }{\ \bar{\mathbf{a}}_{ij}\ }$
Generative model	$\mathbf{b}_i = s\mathbf{R}\mathbf{a}_i + \mathbf{t} + \mathbf{o}_i + \boldsymbol{\epsilon}_i$	$\mathbf{b}_{ij} = s\mathbf{R}\bar{\mathbf{a}}_{ij} + \mathbf{o}_{ij} + \boldsymbol{\epsilon}_{ij}$	$s_{ij} = s + o_{ij}^s + \epsilon_{ij}^s$
Noise bounds	$\ \boldsymbol{\epsilon}_i\ \leq \beta_i$	$\ \boldsymbol{\epsilon}_{ij}\ \leq \delta_{ij} \doteq \beta_i + \beta_j$	$ \epsilon_{ij}^s \leq \alpha_{ij} \doteq \delta_{ij} / \ \bar{\mathbf{a}}_{ij}\ $
Dependent transformations	$(s, \mathbf{R}, \mathbf{t})$	(s, \mathbf{R})	s
Number	N	$K \leq \frac{N(N-1)}{2}$	K

TABLE III
SUMMARY OF INVARIANT MEASUREMENTS.

D. Proof of Theorem 4: Translation Invariant Measurements

Using the vector notation $\mathbf{a} \in \mathbb{R}^{3N}$ and $\mathbf{b} \in \mathbb{R}^{3N}$ already introduced in the statement of the theorem, we can write the generative model (5) compactly as:

$$\mathbf{b} = s(\mathbf{I}_N \otimes \mathbf{R})\mathbf{a} + (\mathbf{1}_N \otimes \mathbf{t}) + \mathbf{o} + \boldsymbol{\epsilon} \quad (41)$$

where $\mathbf{o} \doteq [\mathbf{o}_1^\top \dots \mathbf{o}_N^\top]^\top$, $\boldsymbol{\epsilon} \doteq [\boldsymbol{\epsilon}_1^\top \dots \boldsymbol{\epsilon}_N^\top]^\top$, and $\mathbf{1}_N$ is a column vector of ones of size N . Denote $K = |\mathcal{E}|$ as the cardinality of \mathcal{E} , such that $\mathbf{A} \in \mathbb{R}^{K \times N}$ is the incidence matrix of the graph with edges \mathcal{E} . Let us now multiply both members by $(\mathbf{A} \otimes \mathbf{I}_3)$:

$$\bar{\mathbf{b}} = (\mathbf{A} \otimes \mathbf{I}_3)[s(\mathbf{I}_N \otimes \mathbf{R})\mathbf{a} + (\mathbf{1}_N \otimes \mathbf{t}) + (\mathbf{o} + \boldsymbol{\epsilon})] \quad (42)$$

Using the property of the Kronecker product we simplify:

$$\begin{aligned} (i) \quad & (\mathbf{A} \otimes \mathbf{I}_3)(\mathbf{I}_N \otimes \mathbf{R})\mathbf{a} = (\mathbf{A} \otimes \mathbf{R})\mathbf{a} \\ & = (\mathbf{I}_K \otimes \mathbf{R})(\mathbf{A} \otimes \mathbf{I}_3)\mathbf{a} \\ & = (\mathbf{I}_K \otimes \mathbf{R})\bar{\mathbf{a}} \\ (ii) \quad & (\mathbf{A} \otimes \mathbf{I}_3)(\mathbf{1}_N \otimes \mathbf{t}) = (\mathbf{A}\mathbf{1}_N \otimes \mathbf{t}) = \mathbf{0} \end{aligned} \quad (43)$$

where we used the fact that $\mathbf{1}_N$ is in the Null space of the incidence matrix \mathbf{A} [119]. Using (i) and (ii), eq. (42) becomes:

$$\bar{\mathbf{b}} = s(\mathbf{I}_K \otimes \mathbf{R})\bar{\mathbf{a}} + (\mathbf{A} \otimes \mathbf{I}_3)(\mathbf{o} + \boldsymbol{\epsilon}) \quad (44)$$

which is invariant to the translation \mathbf{t} , concluding the proof.

E. Summary of Invariant Measurements

Table III provides a summary of the invariant measurements introduced in Section V.

F. Proof of Theorem 6: Max Clique Inlier Selection

Consider a graph $\mathcal{G}'(\mathcal{V}, \mathcal{E}')$ whose edges were selected as inliers during scale estimation, by pruning the complete graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. An edge (i, j) (and the corresponding TIM) is an inlier if both i and j are correct correspondences (see discussion before Theorem 4). Therefore, \mathcal{G}' contains edges connecting all points for which we have inlier correspondences. Therefore, these points are vertices of a clique in the graph \mathcal{G}' and the edges (or equivalently the TIMs) connecting those points form a clique in \mathcal{G}' . We conclude the proof by observing that the clique formed by the inliers has to belong to at least one maximal clique of \mathcal{G}' .

G. Proof of Theorem 7: Optimal Scalar TLS Estimation

Let us first prove that there are at most $2K - 1$ different non-empty consensus sets. We attach a confidence interval $[s_k - \alpha_k \bar{c}, s_k + \alpha_k \bar{c}]$ to each measurement s_k , $\forall k \in \{1, \dots, K\}$. For a given scalar $s \in \mathbb{R}$, a measurement k is in the consensus set of s if $s \in [s_k - \alpha_k \bar{c}, s_k + \alpha_k \bar{c}]$ (satisfies $\frac{\|s - s_k\|^2}{\alpha_k^2} \leq \bar{c}^2$), see Fig. 3(a). Therefore, the only points on the real line where the consensus set may change are the boundaries (shown in red in Fig. 3(a)) of the intervals $[s_k - \alpha_k \bar{c}, s_k + \alpha_k \bar{c}]$, $k \in \{1, \dots, K\}$. Since there are at most $2K - 1$ such intervals, there are at most $2K - 1$ non-empty consensus sets (Fig. 3(b)), concluding the first part of the proof. The second part follows from the fact that the consensus set of \hat{s} is necessarily one of the $2K - 1$ possible consensus sets, and problem (12) simply computes the least squares estimate of the measurements for every possible consensus set (at most $2K - 1$) and chooses the estimate that induces the lowest cost as the optimal estimate.

H. Proof of Proposition 9: Binary Cloning

Here we prove the equivalence between the mixed-integer program (18) and the optimization in (19) involving $N + 1$ quaternions. To do so, we note that since $\theta_k \in \{+1, -1\}$ and $\frac{1+\theta_k}{2} \in \{0, 1\}$, we can safely move $\frac{1+\theta_k}{2}$ inside the squared norm (because $0 = 0^2, 1 = 1^2$) in each summand of the cost function (18):

$$\begin{aligned} & \sum_{k=1}^K \frac{1+\theta_k}{2} \frac{\|\hat{\mathbf{b}}_k - \mathbf{q} \circ \hat{\mathbf{a}}_k \circ \mathbf{q}^{-1}\|^2}{\delta_k^2} + \frac{1-\theta_k}{2} \bar{c}^2 \\ & = \sum_{k=1}^K \frac{\|\hat{\mathbf{b}}_k - \mathbf{q} \circ \hat{\mathbf{a}}_k \circ \mathbf{q}^{-1} + \theta_k \hat{\mathbf{b}}_k - \mathbf{q} \circ \hat{\mathbf{a}}_k \circ (\theta_k \mathbf{q}^{-1})\|^2}{4\delta_k^2} + \frac{1-\theta_k}{2} \bar{c}^2 \end{aligned} \quad (45)$$

Now we introduce N new unit quaternions $\mathbf{q}_k = \theta_k \mathbf{q}$, $k = 1, \dots, N$ by multiplying \mathbf{q} by the N binary variables $\theta_k \in \{+1, -1\}$, a re-parametrization we called *binary cloning*. One can easily verify that $\mathbf{q}^\top \mathbf{q}_k = \theta_k (\mathbf{q}^\top \mathbf{q}) = \theta_k$. Hence, by substituting $\theta_k = \mathbf{q}^\top \mathbf{q}_k$ into (45), we can rewrite the mixed-integer program (18) as:

$$\begin{aligned} & \min_{\substack{\mathbf{q} \in \mathcal{U}^3 \\ \mathbf{q}_k \in \{\pm \mathbf{q}\}}} \sum_{k=1}^K \frac{\|\hat{\mathbf{b}}_k - \mathbf{q} \circ \hat{\mathbf{a}}_k \circ \mathbf{q}^{-1} + \mathbf{q}^\top \mathbf{q}_k \hat{\mathbf{b}}_k - \mathbf{q} \circ \hat{\mathbf{a}}_k \circ \mathbf{q}_k^{-1}\|^2}{4\delta_k^2} \\ & \quad + \frac{1 - \mathbf{q}^\top \mathbf{q}_k}{2} \bar{c}^2, \end{aligned} \quad (46)$$

which is the same as the optimization in (19).

I. Proof of Proposition 10: Binary Cloning as a QCQP

Here we show that the optimization involving $K + 1$ quaternions in (19) can be reformulated as the Quadratically-Constrained Quadratic Program (QCQP) in (20). Towards this

goal, we prove that the objective function and the constraints in the QCQP are a re-parametrization of the ones in (19).

Equivalence of the objective functions. We start by developing the squared 2-norm term in (19) (we scale \bar{a}_k by $\hat{s}\bar{a}_k$, i.e., $\bar{a}_k \doteq \hat{s}\bar{a}_k$ for notation simplicity):

$$\begin{aligned}
& \|\hat{b}_k - q \circ \hat{a}_k \circ q^{-1} + q^\top q_k \hat{b}_k - q \circ \hat{a}_k \circ q_k^{-1}\|^2 \\
& (\|q^\top q_k \hat{b}_k\|^2 = \|\hat{b}_k\|^2 = \|\bar{b}_k\|^2, \hat{b}_k^\top (q^\top q_k) \hat{b}_k = q^\top q_k \|\bar{b}_k\|^2) \\
& (\|q \circ \hat{a}_k \circ q^{-1}\|^2 = \|\bar{R}\bar{a}_k\|^2 = \|\bar{a}_k\|^2) \\
& (\|q \circ \hat{a}_k \circ q_k^{-1}\|^2 = \|\theta_k \bar{R}\bar{a}_k\|^2 = \|\bar{a}_k\|^2) \\
& ((q \circ \hat{a}_k \circ q^{-1})^\top (q \circ \hat{a}_k \circ q_k^{-1}) = (\bar{R}\bar{a}_k)^\top (\theta_k \bar{R}\bar{a}_k) = q^\top q_k \|\bar{a}_k\|^2) \\
& = 2\|\bar{b}_k\|^2 + 2\|\bar{a}_k\|^2 + 2q^\top q_k \|\bar{b}_k\|^2 + 2q^\top q_k \|\bar{a}_k\|^2 \\
& \quad - 2\hat{b}_k^\top (q \circ \hat{a}_k \circ q^{-1}) - 2\hat{b}_k^\top (q \circ \hat{a}_k \circ q_k^{-1}) \\
& - 2q^\top q_k \hat{b}_k^\top (q \circ \hat{a}_k \circ q^{-1}) - 2q^\top q_k \hat{b}_k^\top (q \circ \hat{a}_k \circ q_k^{-1}) \quad (47) \\
& (q^\top q_k \hat{b}_k^\top (q \circ \hat{a}_k \circ q_k^{-1}) = (\theta_k)^2 \hat{b}_k^\top (q \circ \hat{a}_k \circ q^{-1}) = \hat{b}_k^\top (q \circ \hat{a}_k \circ q^{-1})) \\
& (\hat{b}_k^\top (q \circ \hat{a}_k \circ q_k^{-1}) = q^\top q_k \hat{b}_k^\top (q \circ \hat{a}_k \circ q^{-1})) \\
& = 2\|\bar{b}_k\|^2 + 2\|\bar{a}_k\|^2 + 2q^\top q_k \|\bar{b}_k\|^2 + 2q^\top q_k \|\bar{a}_k\|^2 \\
& \quad - 4\hat{b}_k^\top (q \circ \hat{a}_k \circ q^{-1}) - 4q^\top q_k \hat{b}_k^\top (q \circ \hat{a}_k \circ q^{-1}) \quad (48)
\end{aligned}$$

where we have used multiple times the binary cloning equalities $q_k = \theta_k q$, $\theta_k = q^\top q_k$, the equivalence between applying rotation to a homogeneous vector \hat{a}_k using quaternion product and using rotation matrix in eq. (4) from the main document, as well as the fact that vector 2-norm is invariant to rotation and homogenization (with zero padding).

Before moving to the next step, we make the following observation by combining eq. (2) and eq. (3):

$$\Omega_1(q^{-1}) = \Omega_1^\top(q), \quad \Omega_2(q^{-1}) = \Omega_2^\top(q) \quad (49)$$

which states the linear operators $\Omega_1(\cdot)$ and $\Omega_2(\cdot)$ of q and its inverse q^{-1} are related by a simple transpose operation. In the next step, we use the equivalence between quaternion product and linear operators in $\Omega_1(q)$ and $\Omega_2(q)$ as defined in eq. (1)-(2) to simplify $\hat{b}_k^\top (q \circ \hat{a}_k \circ q^{-1})$ in eq. (48):

$$\begin{aligned}
& \hat{b}_k^\top (q \circ \hat{a}_k \circ q^{-1}) \\
& (q \circ \hat{a}_k = \Omega_1(q) \hat{a}_k, \Omega_1(q) \hat{a}_k \circ q^{-1} = \Omega_2(q^{-1}) \Omega_1(q) \hat{a}_k = \Omega_2^\top(q) \Omega_1(q) \hat{a}_k) \\
& = \hat{b}_k^\top (\Omega_2^\top(q) \Omega_1(q) \hat{a}_k) \quad (50) \\
& (\Omega_2(q) \hat{b}_k = \hat{b}_k \circ q = \Omega_1(\hat{b}_k) q, \Omega_1(q) \hat{a}_k = q \circ \hat{a}_k = \Omega_2(\hat{a}_k) q) \\
& = q^\top \Omega_1^\top(\hat{b}_k) \Omega_2(\hat{a}_k) q. \quad (51)
\end{aligned}$$

Now we can insert eq. (51) back to eq. (48) and write:

$$\begin{aligned}
& \|\hat{b}_k - q \circ \hat{a}_k \circ q^{-1} + q^\top q_k \hat{b}_k - q \circ \hat{a}_k \circ q_k^{-1}\|^2 \\
& = 2\|\bar{b}_k\|^2 + 2\|\bar{a}_k\|^2 + 2q^\top q_k \|\bar{b}_k\|^2 + 2q^\top q_k \|\bar{a}_k\|^2 \\
& \quad - 4\hat{b}_k^\top (q \circ \hat{a}_k \circ q^{-1}) - 4q^\top q_k \hat{b}_k^\top (q \circ \hat{a}_k \circ q^{-1}) \quad (52) \\
& = 2\|\bar{b}_k\|^2 + 2\|\bar{a}_k\|^2 + 2q^\top q_k \|\bar{b}_k\|^2 + 2q^\top q_k \|\bar{a}_k\|^2 \\
& \quad - 4q^\top \Omega_1^\top(\hat{b}_k) \Omega_2(\hat{a}_k) q - 4q^\top q_k q^\top \Omega_1^\top(\hat{b}_k) \Omega_2(\hat{a}_k) q \quad (53)
\end{aligned}$$

$$\begin{aligned}
& (q^\top q_k q^\top \Omega_1^\top(\hat{b}_k) \Omega_2(\hat{a}_k) q = \theta_k q^\top \Omega_1^\top(\hat{b}_k) \Omega_2(\hat{a}_k) q = q^\top \Omega_1^\top(\hat{b}_k) \Omega_2(\hat{a}_k) q_k) \\
& = 2\|\bar{b}_k\|^2 + 2\|\bar{a}_k\|^2 + 2q^\top q_k \|\bar{b}_k\|^2 + 2q^\top q_k \|\bar{a}_k\|^2 \\
& \quad - 4q^\top \Omega_1^\top(\hat{b}_k) \Omega_2(\hat{a}_k) q - 4q^\top \Omega_1^\top(\hat{b}_k) \Omega_2(\hat{a}_k) q_k \quad (54) \\
& \quad (-\Omega_1^\top(\hat{b}_k) = \Omega_1(\hat{b}_k)) \\
& = 2\|\bar{b}_k\|^2 + 2\|\bar{a}_k\|^2 + 2q^\top q_k \|\bar{b}_k\|^2 + 2q^\top q_k \|\bar{a}_k\|^2 \\
& \quad + 4q^\top \Omega_1(\hat{b}_k) \Omega_2(\hat{a}_k) q + 4q^\top \Omega_1(\hat{b}_k) \Omega_2(\hat{a}_k) q_k, \quad (55)
\end{aligned}$$

which is quadratic in q and q_k . Substituting eq. (55) back to (19), we can write the cost function as:

$$\begin{aligned}
& \sum_{k=1}^K \frac{\|\hat{b}_k - q \circ \hat{a}_k \circ q^{-1} + q^\top q_k \hat{b}_k - q \circ \hat{a}_k \circ q_k^{-1}\|^2}{4\delta_k^2} + \frac{1 - q^\top q_k}{2} \bar{c}^2 \\
& = \sum_{k=1}^K q_k^\top \left(\underbrace{\frac{(\|\bar{b}_k\|^2 + \|\bar{a}_k\|^2) \mathbf{I}_4 + 2\Omega_1(\hat{b}_k) \Omega_2(\hat{a}_k)}{2\delta_k^2} + \frac{\bar{c}^2}{2} \mathbf{I}_4}_{:=Q_{kk}} \right) q_k \\
& \quad + 2q^\top \left(\underbrace{\frac{(\|\bar{b}_k\|^2 + \|\bar{a}_k\|^2) \mathbf{I}_4 + 2\Omega_1(\hat{b}_k) \Omega_2(\hat{a}_k)}{4\delta_k^2} - \frac{\bar{c}^2}{4} \mathbf{I}_4}_{:=Q_{0k}} \right) q_k, \quad (56)
\end{aligned}$$

where we have used two facts: (i) $q^\top A q = \theta_k^2 q^\top A q = q_k^\top A q_k$ for any matrix $A \in \mathbb{R}^{4 \times 4}$, (kk) $c = c q^\top q = q^\top (c \mathbf{I}_4) q$ for any real constant c , which allowed writing the quadratic forms of q and constant terms in the cost as quadratic forms of q_k . Since we have not changed the decision variables q and $\{q_k\}_{k=1}^K$, the optimization in (19) is therefore equivalent to the following optimization:

$$\min_{\substack{q \in S^3 \\ q_k \in \{\pm q\}}} \sum_{k=1}^K q_k^\top Q_{kk} q_k + 2q^\top Q_{0k} q_k \quad (57)$$

where Q_{kk} and Q_{0k} are the known 4×4 data matrices as defined in eq. (56).

Now it remains to prove that the above optimization (57) is equivalent to the QCQP in (20). Recall that x is the column vector stacking all the $K+1$ quaternions, i.e., $x = [q^\top q_1^\top \dots q_N^\top]^\top \in \mathbb{R}^{4(K+1)}$. Let us introduce symmetric matrices $Q_k \in \mathbb{R}^{4(K+1) \times 4(K+1)}$, $k = 1, \dots, K$ and let the 4×4 sub-block of Q_k corresponding to sub-vector u and v , be denoted as $[Q_k]_{uv}$; each Q_k is defined as:

$$[Q_k]_{uv} = \begin{cases} Q_{kk} & \text{if } u = q_k \text{ and } v = q_k \\ Q_{0k} & \text{if } u = q \text{ and } v = q_k \\ & \text{or } u = q_k \text{ and } v = q \\ 0_{4 \times 4} & \text{otherwise} \end{cases} \quad (58)$$

i.e., Q_k has the diagonal 4×4 sub-block corresponding to (q_k, q_k) be Q_{kk} , has the two off-diagonal 4×4 sub-blocks corresponding to (q, q_k) and (q_k, q) be Q_{0k} , and has all the other 4×4 sub-blocks be zero. Then we can write the cost function in eq. (57) compactly using x and Q_k :

$$\sum_{k=1}^K q_k^\top Q_{kk} q_k + 2q^\top Q_{0k} q_k = \sum_{k=1}^K x^\top Q_k x \quad (59)$$

Therefore, by denoting $Q = \sum_{k=1}^K Q_k$, we proved that the objective functions in (19) and the QCQP (20) are the same.

Equivalence of the constraints. We are only left to prove that (19) and (20) have the same feasible set, i.e., the following two sets of constraints are equivalent:

$$\begin{cases} q \in \mathcal{U}^3 \\ q_k \in \{\pm q\}, \\ k = 1, \dots, K \end{cases} \Leftrightarrow \begin{cases} x_q^\top x_q = 1 \\ x_{q_k} x_{q_k}^\top = x_q x_q^\top, \\ k = 1, \dots, K \end{cases} \quad (60)$$

We first prove the (\Rightarrow) direction. Since $\mathbf{q} \in \mathcal{U}^3$, it is obvious that $\mathbf{x}_q^\top \mathbf{x}_q = \mathbf{q}^\top \mathbf{q} = 1$. In addition, since $\mathbf{q}_k \in \{+\mathbf{q}, -\mathbf{q}\}$, it follows that $\mathbf{x}_{q_k} \mathbf{x}_{q_k}^\top = \mathbf{q}_k \mathbf{q}_k^\top = \mathbf{q} \mathbf{q}^\top = \mathbf{x}_q \mathbf{x}_q^\top$. Then we prove the reverse direction (\Leftarrow). Since $\mathbf{x}_q^\top \mathbf{x}_q = \mathbf{q}^\top \mathbf{q}$, so $\mathbf{x}_q^\top \mathbf{x}_q = 1$ implies $\mathbf{q}^\top \mathbf{q} = 1$ and therefore $\mathbf{q} \in \mathcal{U}^3$. On the other hand, $\mathbf{x}_{q_k} \mathbf{x}_{q_k}^\top = \mathbf{x}_q \mathbf{x}_q^\top$ means $\mathbf{q}_k \mathbf{q}_k^\top = \mathbf{q} \mathbf{q}^\top$. If we write $\mathbf{q}_k = [q_{k1}, q_{k2}, q_{k3}, q_{k4}]^\top$ and $\mathbf{q} = [q_1, q_2, q_3, q_4]$, then the following matrix equality holds:

$$\begin{bmatrix} q_{k1}^2 & q_{k1}q_{k2} & q_{k1}q_{k3} & q_{k1}q_{k4} \\ \star & q_{k2}^2 & q_{k2}q_{k3} & q_{k2}q_{k4} \\ \star & \star & q_{k3}^2 & q_{k3}q_{k4} \\ \star & \star & \star & q_{k4}^2 \end{bmatrix} = \begin{bmatrix} q_1^2 & q_1q_2 & q_1q_3 & q_1q_4 \\ \star & q_2^2 & q_2q_3 & q_2q_4 \\ \star & \star & q_3^2 & q_3q_4 \\ \star & \star & \star & q_4^2 \end{bmatrix} \quad (61)$$

First, from the diagonal equalities, we can get $q_{ki} = \theta_i q_i, \theta_i \in \{+1, -1\}, i = 1, 2, 3, 4$. Then we look at the off-diagonal equality: $q_{ki}q_{kj} = q_iq_j, i \neq j$, since $q_{ki} = \theta_i q_i$ and $q_{kj} = \theta_j q_j$, we have $q_{ki}q_{kj} = \theta_i \theta_j q_i q_j$, from which we obtain $\theta_i \theta_j = 1, \forall i \neq j$. This implies that all the binary values $\{\theta_i\}_{i=1}^4$ have the same sign, and therefore they are equal to each other. As a result, $\mathbf{q}_k = \theta_k \mathbf{q} = \{+\mathbf{q}, -\mathbf{q}\}$, showing the two sets of constraints in eq. (60) are indeed equivalent. Therefore, the QCQP in eq. (20) is equivalent to the optimization in (57), and the original optimization in (19) that involves $K+1$ quaternions, concluding the proof.

J. Proof of Proposition 11: Matrix Formulation of Binary Cloning

Here we show that the non-convex QCQP written in terms of the vector \mathbf{x} in Proposition 10 (and eq. (20)) is equivalent to the non-convex problem written using the matrix \mathbf{Z} in Proposition 11 (and eq. (23)). We do so by showing that the objective function and the constraints in (23) are a reparametrization of the ones in (20).

Equivalence of the objective function. Since $\mathbf{Z} = \mathbf{x} \mathbf{x}^\top$ and using the cyclic property of the trace, we rewrite the objective in (20) as:

$$\mathbf{x}^\top \mathbf{Q} \mathbf{x} = \text{tr}(\mathbf{Q} \mathbf{x} \mathbf{x}^\top) = \text{tr}(\mathbf{Q} \mathbf{Z}) \quad (62)$$

showing the equivalence of the objectives in (20) and (23).

Equivalence of the constraints. It is trivial to see that $\mathbf{x}_q^\top \mathbf{x}_q = \text{tr}(\mathbf{x}_q \mathbf{x}_q^\top) = 1$ is equivalent to $\text{tr}([\mathbf{Z}]_{qq}) = 1$ by using the cyclic property of the trace operator and inspecting the structure of \mathbf{Z} . In addition, $\mathbf{x}_{q_k} \mathbf{x}_{q_k}^\top = \mathbf{x}_q \mathbf{x}_q^\top$ also directly maps to $[\mathbf{Z}]_{q_k q_k} = [\mathbf{Z}]_{qq}$ for all $i = 1, \dots, K$. Lastly, requiring $\mathbf{Z} \succeq 0$ and $\text{rank}(\mathbf{Z}) = 1$ is equivalent to restricting \mathbf{Z} to the form $\mathbf{Z} = \mathbf{x} \mathbf{x}^\top$ for some vector $\mathbf{x} \in \mathbb{R}^{4(K+1)}$. Therefore, the constraint sets of eq. (20) and (23) are also equivalent, concluding the proof.

K. Proof of Theorem 13: SDP Relaxation with Redundant Constraints and Global Optimality Guarantee

We first restate the QCQP (20) and explicitly add the redundant constraints:

Lemma 24 (QCQP with Redundant Constraints). *The QCQP (20) is equivalent to the following QCQP with redundant constraints:*

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^{4(K+1)}} \quad & \mathbf{x}^\top \mathbf{Q} \mathbf{x} \\ \text{s.t.} \quad & \mathbf{x}_q^\top \mathbf{x}_q = 1, \\ & \mathbf{x}_{q_k} \mathbf{x}_{q_k}^\top = \mathbf{x}_q \mathbf{x}_q^\top, \forall k = 1, \dots, K, \\ & \mathbf{x}_{q_i} \mathbf{x}_{q_j}^\top = (\mathbf{x}_{q_i} \mathbf{x}_{q_j}^\top)^\top, \forall 0 \leq i < j \leq K, \end{aligned} \quad (\text{P})$$

where $\mathbf{q}_0 \doteq \mathbf{q}$ for notation simplicity.

The redundancy of the last set of constraints can be easily understood from:

$$\mathbf{x}_{q_i} \mathbf{x}_{q_j}^\top = (\theta_i \mathbf{q})(\theta_j \mathbf{q}^\top) = (\theta_j \mathbf{q})(\theta_i \mathbf{q}^\top) = (\mathbf{x}_{q_i} \mathbf{x}_{q_j}^\top)^\top. \quad (63)$$

Since the original QCQP (20) is equivalent to the rank-constrained SDP (23), problem (P) is also equivalent to (23).

Now let us prove Theorem 13. While the SDP relaxation of (P) can be simply obtained by adding the redundant constraints and dropping the rank constraint in (23). Here we take a longer path, using Lagrangian duality, which is useful towards getting the guarantees stated in the theorem.

Lemma 25 (Dual of QCQP with Redundant Constraints). *The Lagrangian dual problem of the QCQP with redundant constraints (P) is the following convex semidefinite program:*

$$\begin{aligned} \max_{\mu, \mathbf{\Lambda}, \mathbf{W}} \quad & \mu \\ \text{s.t.} \quad & (\mathbf{Q} - \mu \mathbf{J} + \mathbf{\Lambda} + \mathbf{W}) \succeq 0, \end{aligned} \quad (\text{D})$$

where $\mathbf{J} \in \mathcal{S}^{4(K+1)}$ is an all-zero matrix except the top-left diagonal 4×4 block $[\mathbf{J}]_{00} = \mathbf{I}_4$; $\mathbf{\Lambda} \in \mathcal{S}^{4(K+1)}$ is a block-diagonal matrix $\mathbf{\Lambda} = \text{blkdiag}(\mathbf{\Lambda}_{00}, \mathbf{\Lambda}_{11}, \dots, \mathbf{\Lambda}_{KK})$, where each $\mathbf{\Lambda}_{kk}$ is a 4×4 symmetric matrix and the sum $\sum_{k=0}^K \mathbf{\Lambda}_{kk} = \mathbf{0}$; and $\mathbf{W} \in \mathcal{S}^{4(K+1)}$ satisfies $[\mathbf{W}]_{ij} = \mathbf{0}$ for $i = j$ and $[\mathbf{W}]_{ij} \in \mathfrak{S}^4$ (i.e., $[\mathbf{W}]_{ij}$ is a skew symmetric matrix) for $i \neq j$.

The dual problem (D) is derived from the redundant QCQP (P) by following standard Lagrangian duality [98]. For any \mathbf{x} that is feasible for (P), one can verify that:

$$\mathbf{x}^\top \mathbf{J} \mathbf{x} = 1, \quad \mathbf{x}^\top \mathbf{\Lambda} \mathbf{x} = 0, \quad \mathbf{x}^\top \mathbf{W} \mathbf{x} = 0. \quad (64)$$

Lemma 26 (Dual of the Dual). *The dual SDP of the dual problem (D) is the following SDP:*

$$\begin{aligned} \min_{\mathbf{Z} \succeq 0} \quad & \text{tr}(\mathbf{Q} \mathbf{Z}) \\ \text{s.t.} \quad & \text{tr}([\mathbf{Z}]_{00}) = 1 \\ & [\mathbf{Z}]_{kk} = [\mathbf{Z}]_{00}, \forall k = 1, \dots, K \\ & [\mathbf{Z}]_{ij} = [\mathbf{Z}]_{ij}^\top, \forall 0 \leq i < j \leq K, \end{aligned} \quad (\text{DD})$$

which is the same as the SDP relaxation with redundant constraints (24).

Now we are ready to prove Theorem 13. First of all, we have the following weak duality:

$$f_D^* = f_{DD}^* \leq f_P^*, \quad (65)$$

where $f_D^* = f_{DD}^*$ (strong duality of the SDP pair) is due to the fact that (DD) has a strictly feasible solution, by having $[\mathbf{Z}]_{00} = [\mathbf{Z}]_{11} = \dots = [\mathbf{Z}]_{KK} = \frac{1}{4}\mathbf{I}_4$ and $[\mathbf{Z}]_{ij} = \mathbf{0}$ for $i \neq j$ (in this case $\mathbf{Z} = \frac{1}{4}\mathbf{I}_{4(K+1)} \succ \mathbf{0}$), and $f_{DD}^* \leq f_P^*$ is due to the fact that (DD) is a convex relaxation for (P).

If the optimal solution of (DD) satisfies $\text{rank}(\mathbf{Z}^*) = 1$, then this means that \mathbf{Z}^* is also feasible for the rank-constrained SDP (23). Therefore, the minimum f_{DD}^* can indeed be attained by a point in the feasible set of problem (23). Because the rank-constrained problem (23) is equivalent to (P), $f_{DD}^* = f_P^*$ must hold and the rank-1 decomposition of \mathbf{Z}^* , \mathbf{x}^* must be a global minimizer of problem (P).

To prove the uniqueness of $\pm \mathbf{x}^*$ as global minimizers of (P), denote $\mu^* = (\mathbf{x}^*)^\top \mathbf{Q}(\mathbf{x}^*) = \text{tr}(\mathbf{Q}\mathbf{Z}^*)$ as the global minimum of (P), (D) and (DD), and $\mathbf{\Lambda}^*$, \mathbf{W}^* as the corresponding dual variables. Let $\mathbf{M}^* = \mathbf{Q} - \mu^*\mathbf{J} + \mathbf{\Lambda}^* + \mathbf{W}^*$, we have $\text{rank}(\mathbf{M}^*) = 4(K+1) - 1$ and its kernel is $\ker(\mathbf{M}^*) = \{\mathbf{x} : \mathbf{x} = a\mathbf{x}^*\}$ from complementary slackness of the pair of SDP duals. Denote the feasible set of (P) as $\Omega(\mathbf{P})$, then we have $\ker(\mathbf{M}^*) \cap \Omega(\mathbf{P}) = \pm \mathbf{x}^*$. Therefore, for any $\mathbf{x} \in \Omega(\mathbf{P}) \setminus \{\pm \mathbf{x}^*\}$, we have:

$$\mathbf{x}^\top \mathbf{M}^* \mathbf{x} > 0 \quad (66)$$

$$\Leftrightarrow \mathbf{x}^\top \mathbf{Q} \mathbf{x} - \mu^* + \mathbf{x}^\top \mathbf{\Lambda}^* \mathbf{x} + \mathbf{x}^\top \mathbf{W}^* \mathbf{x} > 0 \quad (67)$$

$$\Leftrightarrow \mathbf{x}^\top \mathbf{Q} \mathbf{x} > \mu^*, \quad (68)$$

where $\mathbf{x}^\top \mathbf{\Lambda}^* \mathbf{x} = \mathbf{x}^\top \mathbf{W}^* \mathbf{x} = 0$ holds due to (64).

L. Proof of Theorem 14: Optimality Certification

We prove Theorem 14 in three steps. First, we show that that proving global optimality of an estimate for the rotation estimation problem (16) relates to the existence of a matrix at the intersection of the positive semidefinite (PSD) cone and an affine subspace (Appendix L1). Second, we show that we can compute a sub-optimality gap (this is the bound η in the theorem) for an estimate using any matrix in the affine subspace (Appendix L2). Finally, we show that under the conditions of Theorem 14 such bound converges to zero when the provided estimate is globally optimal (Appendix L3).

1) *Matrix Certificate for Global Optimality*: Here we derive necessary and sufficient conditions for global optimality of the TLS rotation estimation problem (16).

Theorem 27 (Sufficient Condition for Global Optimality). *Given a feasible solution $(\hat{\mathbf{q}}, \hat{\theta}_1, \dots, \hat{\theta}_K)$ of the TLS rotation estimation problem (16), denote $\hat{\mathbf{x}} = [\hat{\mathbf{q}}^\top, \hat{\theta}_1 \hat{\mathbf{q}}^\top, \dots, \hat{\theta}_K \hat{\mathbf{q}}^\top]^\top$, and $\hat{\mu} = \hat{\mathbf{x}}^\top \mathbf{Q} \hat{\mathbf{x}}$, as the corresponding solution and cost of the QCQP (P). Then $\hat{\mu}$ (resp. $(\hat{\mathbf{q}}, \hat{\theta}_1, \dots, \hat{\theta}_K)$) is the global minimum (resp. minimizer) of problem (16) if there exists a matrix \mathbf{M} that satisfies the following:*

$$\mathbf{M} \in \mathcal{L}, \quad (69)$$

$$\mathbf{M} \succeq \mathbf{0}, \quad (70)$$

where \mathcal{L} is the following affine subspace:

$$\mathcal{L} \doteq \left\{ \mathbf{M} \in \mathcal{S}^{4(K+1)} \mid \begin{matrix} \mathbf{M} \hat{\mathbf{x}} = \mathbf{0}, \\ \mathbf{M} - \mathbf{Q} + \hat{\mu} \mathbf{J} \in \mathcal{H}. \end{matrix} \right\} \quad (71)$$

and \mathcal{H} is defined as:

$$\mathcal{H} \doteq \left\{ \mathbf{\Delta} \in \mathcal{S}^{4(N+1)} \mid \begin{matrix} \sum_{k=1}^K [\mathbf{\Delta}]_{kk} = \mathbf{0}, \\ [\mathbf{\Delta}]_{ij} \in \mathfrak{S}^4, \forall 0 \leq i < j \leq K. \end{matrix} \right\}, \quad (72)$$

The matrix $\mathbf{J} \in \mathcal{S}^{4(K+1)}$ is an all-zero matrix except the top-left 4×4 block $[\mathbf{J}]_{00} = \mathbf{I}_4$ (the same defined in Appendix K).

Proof: Because $\mathbf{M} - \mathbf{Q} + \hat{\mu} \mathbf{J} \in \mathcal{H}$, and from the definition of \mathcal{H} , we can write \mathbf{M} as:

$$\mathbf{M} = \mathbf{Q} - \hat{\mu} \mathbf{J} + \mathbf{\Lambda} + \mathbf{W}, \quad (73)$$

where $\mathbf{\Lambda}, \mathbf{W} \in \mathcal{S}^{4(K+1)}$ are the dual variables defined in Lemma 25. From $\mathbf{M} \succeq \mathbf{0}$, we have the following inequality for any \mathbf{x} that is primal feasible for (P):

$$\mathbf{x}^\top \mathbf{M} \mathbf{x} \geq 0 \Leftrightarrow \mathbf{x}^\top (\mathbf{Q} - \hat{\mu} \mathbf{J} + \mathbf{\Lambda} + \mathbf{W}) \mathbf{x} \geq 0 \quad (74)$$

$$\Leftrightarrow \mathbf{x}^\top \mathbf{Q} \mathbf{x} \geq \hat{\mu} + \mathbf{x}^\top \mathbf{\Lambda} \mathbf{x} + \mathbf{x}^\top \mathbf{W} \mathbf{x} = \hat{\mu}, \quad (75)$$

which states that $\hat{\mu}$ (resp. $\hat{\mathbf{x}}$) is the global minimum (resp. minimizer) of (P).

In addition, denote $\hat{\mathbf{Z}} = \hat{\mathbf{x}} \hat{\mathbf{x}}^\top$, then $\hat{\mathbf{Z}}$ and $(\hat{\mu}, \mathbf{\Lambda}, \mathbf{W})$ are a pair of primal-dual optimal solutions for (DD) and (D), which leads to the following SDP complementary slackness condition:

$$(\mathbf{Q} - \hat{\mu} \mathbf{J} + \mathbf{\Lambda} + \mathbf{W}) \hat{\mathbf{Z}} = \mathbf{0} \quad (76)$$

$$\Leftrightarrow (\mathbf{Q} - \hat{\mu} \mathbf{J} + \mathbf{\Lambda} + \mathbf{W}) \hat{\mathbf{x}} = \mathbf{0}. \quad (77)$$

Therefore, $\mathbf{M} \hat{\mathbf{x}} = \mathbf{0}$ must hold in eq. (69). ■

Corollary 28 (Sufficient and Necessary Condition for Global Optimality). *If strong duality is achieved, i.e., $f_D^* = f_{DD}^* = f_P^*$ is attained, then the existence of a matrix \mathbf{M} that satisfies eq. (69) and (70) is also a necessary condition for $\hat{\mu}$ (resp. $(\hat{\mathbf{q}}, \hat{\theta}_1, \dots, \hat{\theta}_K)$) to be the global minimum (resp. minimizer) of problem (16).*

Proof: If strong duality is achieved, then there must exist dual variables $\mathbf{\Lambda}$ and \mathbf{W} that satisfy $\mathbf{Q} - \hat{\mu} \mathbf{J} + \mathbf{\Lambda} + \mathbf{W} \succeq \mathbf{0}$. The rest follows from the proof of Theorem 27. ■

2) *Sub-optimality Bound*: Theorem 27 suggests that finding a matrix \mathbf{M} that lies at the intersection of the PSD cone \mathcal{S}_+ and the affine subspace \mathcal{L} gives a *certificate* for $\hat{\mathbf{x}}$ to be globally optimal for (P). Corollary 29 below states that even if $\hat{\mathbf{x}}$ is not a globally optimal solution, finding any $\mathbf{M} \in \mathcal{L}$ (not necessarily PSD) still gives a valid sub-optimality bound using the minimum eigenvalue of \mathbf{M} .¹¹

Corollary 29 (Relative Sub-optimality Bound). *Given a feasible solution $(\hat{\mathbf{q}}, \hat{\theta}_1, \dots, \hat{\theta}_K)$ of the TLS rotation estimation problem (16), denote $\hat{\mathbf{x}} = [\hat{\mathbf{q}}^\top, \hat{\theta}_1 \hat{\mathbf{q}}^\top, \dots, \hat{\theta}_K \hat{\mathbf{q}}^\top]^\top$, and $\hat{\mu} = \hat{\mathbf{x}}^\top \mathbf{Q} \hat{\mathbf{x}}$, as the corresponding solution and cost of the QCQP (P). Define the affine subspace \mathcal{L} as in (71). For any $\mathbf{M} \in \mathcal{L}$, let $\lambda_1(\mathbf{M})$ be its minimum eigenvalue. Then the relative sub-optimality gap of $\hat{\mathbf{x}}$ is bounded by:*

$$\frac{\hat{\mu} - \mu^*}{\hat{\mu}} \leq \eta \doteq \frac{|\lambda_1(\mathbf{M})|(K+1)}{\hat{\mu}}, \quad (78)$$

¹¹ Note that the expression of \mathcal{L} depends on the estimate $\hat{\mathbf{x}}$ as per (71).

where μ^* is the true global minimum of (P).

Proof: For any \mathbf{x} that is primal feasible for (P), we have $\|\mathbf{x}\|^2 = \sum_{k=1}^K \|\mathbf{q}_k\|^2 = K + 1$. Therefore, for any \mathbf{x} in the feasible set of (P) and $\mathbf{M} \in \mathcal{L}$, the following holds:

$$\mathbf{x}^\top \mathbf{M} \mathbf{x} \geq \lambda_1(\mathbf{M}) \|\mathbf{x}\|^2 = \lambda_1(\mathbf{M})(K + 1) \quad (79)$$

$$\stackrel{(i)}{\Leftrightarrow} \mathbf{x}^\top (\mathbf{Q} - \hat{\mu} \mathbf{J} + \mathbf{\Lambda} + \mathbf{W}) \mathbf{x} \geq \lambda_1(\mathbf{M})(K + 1) \quad (80)$$

$$\stackrel{(ii)}{\Leftrightarrow} \mathbf{x}^\top \mathbf{Q} \mathbf{x} \geq \hat{\mu} + \lambda_1(\mathbf{M})(K + 1). \quad (81)$$

where (i) follows from the fact that any $\mathbf{M} \in \mathcal{L}$ can be written as $\mathbf{Q} - \hat{\mu} \mathbf{J} + \mathbf{\Lambda} + \mathbf{W}$ (see eq. (73)), and (ii) follows from (64). Since μ^* is also achieved in the feasible set, eq. (81) also holds for μ^* :

$$\mu^* \geq \hat{\mu} + \lambda_1(\mathbf{M})(K + 1) \quad (82)$$

$$\Leftrightarrow \frac{\hat{\mu} - \mu^*}{\hat{\mu}} \leq \frac{|\lambda_1(\mathbf{M})|(K + 1)}{\hat{\mu}}, \quad (83)$$

where in eq. (83) we have used $\lambda_1(\mathbf{M}) \leq 0$ since \mathbf{M} has a zero eigenvalue with eigenvector $\hat{\mathbf{x}}$ ($\mathbf{M}\hat{\mathbf{x}} = \mathbf{0}$ in eq. (69)). ■

3) *Alternating Projections and Algorithm 3:* Theorems 27-28 relate global optimality to the existence of a matrix $\mathbf{M} \in \mathcal{S}_+ \cap \mathcal{L}$, while Theorem 29 obtains a suboptimality bound from any matrix $\mathbf{M} \in \mathcal{L}$. The next Theorem states that finding a matrix $\mathbf{M} \in \mathcal{S}_+ \cap \mathcal{L}$ is equivalent to finding a matrix $\bar{\mathbf{M}} \in \mathcal{S}_+ \cap \bar{\mathcal{L}}$, where $\bar{\mathcal{L}}$ is a “rotated” version of \mathcal{L} (this will enable faster projections to this subspace).

Theorem 30 (Rotated Affine Subspace). *Given a feasible solution $(\hat{q}, \hat{\theta}_1, \dots, \hat{\theta}_K)$ of the TLS rotation estimation problem (16), define the block-diagonal matrix $\hat{\Omega}_q = \mathbf{I}_{K+1} \otimes \Omega_1(\hat{q})$. Then a matrix $\mathbf{M} \in \mathcal{L} \cap \mathcal{S}_+$ if and only if $\bar{\mathbf{M}} \doteq \hat{\Omega}_q^\top \mathbf{M} \hat{\Omega}_q \in \bar{\mathcal{L}} \cap \mathcal{S}_+$ with $\bar{\mathcal{L}}$ defined as:*

$$\bar{\mathcal{L}} \doteq \left\{ \bar{\mathbf{M}} \in \mathcal{S}^{4(K+1)} \mid \begin{array}{l} \bar{\mathbf{M}} \bar{\mathbf{x}} = \mathbf{0}, \\ \bar{\mathbf{M}} - \bar{\mathbf{Q}} + \hat{\mu} \mathbf{J} \in \mathcal{H}. \end{array} \right\}, \quad (84)$$

where $\bar{\mathbf{x}} = [1, \hat{\theta}_1, \dots, \hat{\theta}_K]^\top \otimes \mathbf{e}$ (with $\mathbf{e} = [0, 0, 0, 1]^\top$) and $\bar{\mathbf{Q}} = \hat{\Omega}_q^\top \mathbf{Q} \hat{\Omega}_q$. Moreover, $\bar{\mathbf{M}}$ and \mathbf{M} produce the same relative sub-optimality bound in Theorem 29.

Proof: Using the expression of $\Omega_1(q)$ in (2), one can verify that $\Omega_1(q)^\top q = \mathbf{e}$, $\hat{\Omega}_q^\top \hat{\mathbf{x}} = \bar{\mathbf{x}}$ and $\hat{\Omega}_q^\top \hat{\Omega}_q = \hat{\Omega}_q \hat{\Omega}_q^\top = \mathbf{I}_{4(K+1)}$. Therefore, we have the following equivalence:

$$\begin{aligned} \bar{\mathbf{M}} \bar{\mathbf{x}} = \mathbf{0} &\Leftrightarrow \hat{\Omega}_q^\top \mathbf{M} \hat{\Omega}_q \hat{\Omega}_q^\top \hat{\mathbf{x}} = \mathbf{0} \\ &\Leftrightarrow \hat{\Omega}_q \mathbf{M} \hat{\mathbf{x}} = \mathbf{0} \Leftrightarrow \mathbf{M} \hat{\mathbf{x}} = \mathbf{0}. \end{aligned} \quad (85)$$

In addition, it is easy to verify that $\Delta \in \mathcal{H}$ if and only if $\bar{\Delta} \doteq \hat{\Omega}_q^\top \Delta \hat{\Omega}_q \in \mathcal{H}$, and $\hat{\Omega}_q^\top \mathbf{J} \hat{\Omega}_q = \mathbf{J}$. Therefore,

$$\mathbf{M} - \mathbf{Q} - \hat{\mu} \mathbf{J} \in \mathcal{H} \Leftrightarrow \bar{\mathbf{M}} - \bar{\mathbf{Q}} - \hat{\mu} \mathbf{J} \in \mathcal{H}, \quad (86)$$

and we conclude that $\mathbf{M} \in \mathcal{L} \Leftrightarrow \bar{\mathbf{M}} \in \bar{\mathcal{L}}$. Finally, $\bar{\mathbf{M}} = \hat{\Omega}_q^\top \mathbf{M} \hat{\Omega}_q = \hat{\Omega}_q^{-1} \mathbf{M} \hat{\Omega}_q$ is a similarity transform on \mathbf{M} and shares the same eigenvalues with \mathbf{M} , which trivially leads to $\mathbf{M} \in \mathcal{S}_+ \Leftrightarrow \bar{\mathbf{M}} \in \mathcal{S}_+$. In addition, since \mathbf{M} and $\bar{\mathbf{M}}$ share the same eigenvalues, we have $\lambda_1(\bar{\mathbf{M}}) = \lambda_1(\mathbf{M})$, which means that $\bar{\mathbf{M}}$ and \mathbf{M} produces the same relative sub-optimality bound in Corollary 29. ■

Therefore, to certify optimality of an estimate, we have to search for a matrix $\bar{\mathbf{M}} \in \bar{\mathcal{L}} \cap \mathcal{S}_+$, i.e., at the intersection between two convex sets. The following Lemma, due to Cheney and Goldstein [120], states that a simple *alternating projections* method converges to a point at the intersection of two closed convex sets.

Lemma 31 (Alternating Projections to Convex Sets). *Let $\mathcal{X}_1, \mathcal{X}_2$ be two closed convex sets in the Hilbert space \mathcal{X} , and $\Pi_{\mathcal{X}_1}, \Pi_{\mathcal{X}_2}$ be their corresponding projection maps, then for any $\mathbf{x} \in \mathcal{X}$,*

$$\lim_{n \rightarrow \infty} (\Pi_{\mathcal{X}_1} \Pi_{\mathcal{X}_2})^n \mathbf{x} \in \mathcal{X}_1 \cap \mathcal{X}_2, \quad (87)$$

provided that $\mathcal{X}_1 \cap \mathcal{X}_2$ is non-empty.

Therefore, the alternating projections (line 26 and line 28) in Algorithm 3 guarantee to converge to a matrix $\bar{\mathbf{M}} \in \bar{\mathcal{L}} \cap \mathcal{S}_+$ when $\bar{\mathcal{L}} \cap \mathcal{S}_+ \neq \emptyset$, and the sub-optimality bound η asymptotically converges to zero. Moreover, Algorithm 3 produces a sub-optimality bound even if it fails to certify global optimality, thanks to Corollary 29.

M. Closed-form Projections

Although Lemma 31 is theoretically sound, in practice computing the projection onto an arbitrary convex set could be computationally expensive¹². However, we will show that both projections, $\Pi_{\bar{\mathcal{L}}}$ and $\Pi_{\mathcal{S}_+}$ in Algorithm 3 can be computed efficiently in closed form.

The projection onto the PSD cone is presented in the following Lemma due to Higham [121].

Lemma 32 (Projection onto \mathcal{S}_+^n). *Given any matrix $\mathbf{M} \in \mathcal{S}^n$, let $\mathbf{M} = \mathbf{U} \text{diag}(\lambda_1, \dots, \lambda_n) \mathbf{U}^\top$ be its spectral decomposition, then the projection of \mathbf{M} onto the PSD cone \mathcal{S}_+^n is:*

$$\Pi_{\mathcal{S}_+^n}(\mathbf{M}) = \mathbf{U} \text{diag}(\max(0, \lambda_1), \dots, \max(0, \lambda_n)) \mathbf{U}^\top. \quad (88)$$

Now let us focus on the affine projection onto $\bar{\mathcal{L}}$. The expression of the projection $\Pi_{\bar{\mathcal{L}}}$ is given in Proposition 33 below. Before stating the result, let us introduce some notation. For two 4×4 matrices $\mathbf{X} \in \mathcal{S}^4$ and $\mathbf{Y} \in \mathcal{S}^4$, we partition \mathbf{X} and \mathbf{Y} as:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}^m \in \mathcal{S}^3 & \mathbf{X}^v \in \mathbb{R}^3 \\ (\mathbf{X}^v)^\top & \mathbf{X}^s \in \mathbb{R} \end{bmatrix}, \mathbf{Y} = \begin{bmatrix} \mathbf{Y}^m \in \mathcal{S}^3 & \mathbf{Y}^v \in \mathbb{R}^3 \\ -(\mathbf{Y}^v)^\top & \mathbf{Y}^s = 0 \end{bmatrix}, \quad (89)$$

where the superscripts m, v, s denotes the top-left 3×3 matrix part, the top-right 3×1 vector part and the bottom-right scalar part, respectively. For a matrix $\mathbf{M} \in \mathcal{S}^{4(K+1)}$, we partition it into $(K+1)^2$ blocks of size 4×4 , and we create a set of $L \doteq \frac{K(K+1)}{2}$ ordered indices (left to right, top to bottom) that enumerate the upper-triangular off-diagonal blocks:

$$\mathcal{Z} = \{(0, 1), \dots, (0, K), (1, 2), \dots, (1, K), \dots, (K-1, K)\}, \quad (90)$$

where each $[\mathbf{M}]_{\mathcal{Z}(l)}, l = 1, \dots, L$, takes a 4×4 off-diagonal block from \mathbf{M} . Then, we define a linear map $\mathcal{C} : \mathcal{S}^{4(K+1)} \rightarrow$

¹²Even for an affine subspace, $\mathcal{A} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} = \mathbf{b}\}$, the naive projection of any \mathbf{x} onto \mathcal{A} is: $\Pi_{\mathcal{A}}(\mathbf{x}) = \mathbf{x} - \mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top)^{-1}(\mathbf{A}\mathbf{x} - \mathbf{b})$, which could be expensive for large n due to the computation of the inverse $(\mathbf{A}\mathbf{A}^\top)^{-1}$.

$\mathbb{R}^{L \times 3}$ that assembles the top-right 3×1 vector part of each upper-triangular off-diagonal block into a matrix:

$$\mathcal{C}(\mathbf{M}) = \left[[\mathbf{M}]_{\mathcal{Z}(1)}^v, \dots, [\mathbf{M}]_{\mathcal{Z}(L)}^v \right]^\top. \quad (91)$$

Given a solution $\hat{\mathbf{x}} = [\hat{\mathbf{q}}^\top, \hat{\theta}_1 \hat{\mathbf{q}}^\top, \dots, \hat{\theta}_K \hat{\mathbf{q}}^\top]^\top$ to problem (P), denote $\hat{\mathbf{R}}$ to be the unique rotation matrix corresponding to $\hat{\mathbf{q}}$, and define the following residual vectors:

$$\hat{\xi}_k = \hat{\mathbf{R}}^\top (\bar{\mathbf{b}}_k - \hat{\mathbf{R}} \bar{\mathbf{a}}_k), k = 1, \dots, K, \quad (92)$$

for each pair of TIMs $(\bar{\mathbf{a}}_k, \bar{\mathbf{b}}_k)$ (scale $\bar{\mathbf{a}}_k$ by: $\bar{\mathbf{a}}_k \leftarrow \hat{s} \bar{\mathbf{a}}_k$).

Proposition 33 (Projection onto $\bar{\mathcal{L}}$). *Given any matrix $\mathbf{M} \in \mathcal{S}^{4(K+1)}$, its projection onto $\bar{\mathcal{L}}$, denoted $\bar{\mathbf{M}} \doteq \Pi_{\bar{\mathcal{L}}}(\mathbf{M})$, is:*

$$\bar{\mathbf{M}} = \Pi_{\bar{\mathcal{H}}}(\mathbf{H}) + \bar{\mathbf{Q}} - \hat{\mu} \mathbf{J}, \quad (93)$$

where $\mathbf{H} \doteq \mathbf{M} - \bar{\mathbf{Q}} + \hat{\mu} \mathbf{J}$ and $\bar{\mathbf{H}} \doteq \Pi_{\bar{\mathcal{H}}}(\mathbf{H})$ can be computed as follows:

1) *Project the matrix part of each diagonal block:*

$$[\bar{\mathbf{H}}]_{kk}^m = [\mathbf{H}]_{kk}^m - \frac{\sum_{k=0}^K [\mathbf{H}]_{kk}^m}{K+1}, \forall k = 0, \dots, K. \quad (94)$$

2) *Project the scalar part of each diagonal block:*

$$[\bar{\mathbf{H}}]_{kk}^s = \begin{cases} -(\frac{1}{4}\hat{\theta}_k + \frac{1}{2})\|\hat{\xi}_k\|^2 + (\frac{1}{4}\hat{\theta}_k - \frac{1}{2})\bar{c}^2 & k \neq 0 \\ \sum_{k=1}^K (\frac{1}{4}\hat{\theta}_k + \frac{1}{2})\|\hat{\xi}_k\|^2 - (\frac{1}{4}\hat{\theta}_k - \frac{1}{2})\bar{c}^2 & k = 0. \end{cases} \quad (95)$$

3) *Project the matrix part of each off-diagonal block:*

$$[\bar{\mathbf{H}}]_{\mathcal{Z}(l)}^m = \frac{[\mathbf{H}]_{\mathcal{Z}(l)}^m - ([\mathbf{H}]_{\mathcal{Z}(l)}^m)^\top}{2}, \forall l = 1, \dots, L. \quad (96)$$

4) *Project the scalar part of each off-diagonal block:*

$$[\bar{\mathbf{H}}]_{\mathcal{Z}(l)}^s = 0, \forall l = 1, \dots, L. \quad (97)$$

5) *Project the vector part of each off-diagonal block:*

$$\mathcal{C}(\bar{\mathbf{H}}) = \mathbf{P} \cdot \mathcal{F}(\mathbf{H}), \quad (98)$$

where $\mathbf{P} \in \mathcal{S}^L$ and $\mathcal{F}(\mathbf{H}) \in \mathbb{R}^{L \times 3}$ are defined in Appendix T.

6) *Project the vector part of each diagonal block using the results of step 5:*

$$\begin{aligned} [\bar{\mathbf{H}}]_{kk}^v &= -\sum_{i=0, i \neq k}^K \hat{\theta}_k \hat{\theta}_i [\bar{\mathbf{H}}]_{ki}^v + \phi_k, \\ \phi_k &= \begin{cases} -\left(\frac{1}{2}\hat{\theta}_k + 1\right) [\hat{\xi}_k]_\times \bar{\mathbf{a}}_k & k = 1, \dots, K \\ \sum_{k=1}^K \left(\frac{1}{2}\hat{\theta}_k + 1\right) [\hat{\xi}_k]_\times \bar{\mathbf{a}}_k & k = 0 \end{cases} \end{aligned} \quad (99)$$

$$\phi_k = \begin{cases} -\left(\frac{1}{2}\hat{\theta}_k + 1\right) [\hat{\xi}_k]_\times \bar{\mathbf{a}}_k & k = 1, \dots, K \\ \sum_{k=1}^K \left(\frac{1}{2}\hat{\theta}_k + 1\right) [\hat{\xi}_k]_\times \bar{\mathbf{a}}_k & k = 0 \end{cases} \quad (100)$$

A complete proof of Proposition 33 is algebraically involved and is given in Appendix T.

N. Proof of Theorem 15: Estimation Contract with Noiseless Inliers and Random Outliers

We prove Theorem 15 in two steps. First we show that under assumptions (iii)-(iv) of the theorem, the inliers found by TEASER match the maximum consensus solution. Then, we prove that under assumptions (i)-(ii) the maximum consensus solution is unique and recovers the ground truth, from which the claims of the theorem follow.

1) *TLS \equiv MC:* Let the number of inliers in the maximum consensus be $N_{\text{in}}^{\text{MC}}$, since the inliers are noise-free, the sum of the squared residual errors in the maximum consensus set, $r_{\text{in}}^{\text{MC}}$, must be zero. By Lemma 23, if TLS selects a different consensus set than the maximum consensus set, then the consensus set that TLS selects must have size $N_{\text{in}}^{\text{TLS}}$ greater than $N_{\text{in}}^{\text{MC}} - r_{\text{in}}^{\text{MC}}/\bar{c}^2$, i.e., $N_{\text{in}}^{\text{TLS}} > N_{\text{in}}^{\text{MC}} - r_{\text{in}}^{\text{MC}}/\bar{c}^2 = N_{\text{in}}^{\text{MC}}$, contradicting the fact that $N_{\text{in}}^{\text{MC}}$ is the size of the maximum consensus set. Therefore, $\text{TLS} \equiv \text{MC}$ for each subproblem.

2) *Exact recovery of maximum consensus:* We now prove that maximum consensus recovers the ground truth under the assumptions of the theorem. Let us start with the scale subproblem. The inliers are noiseless and distinct (assumption (i) in the theorem), so each pair of inliers will produce a TRIM $s_k = s^\circ$. Since we have at least 3 inliers, there are at least 3 such TRIMs. Since the outliers are in *generic position* (assumption (ii) in the theorem), the event that more than 3 TRIMs formed by the *outliers* will have the same scale estimate \bar{s} that is different from s° happens with probability zero. Therefore, the maximum consensus solution coincides with the ground-truth scale. The same logic can be repeated for the rotation and translation subproblems, leading to identical conclusions under the assumption that the rotation subproblem finds a certifiably optimal TLS estimate (condition (iv)).

3) *Exact recovery of TEASER:* Since under the assumptions of the theorem, TEASER matches the maximum consensus solution in each subproblem, and maximum consensus recovers the ground truth in each subproblem with probability 1, TEASER almost surely recovers the ground-truth transformation, i.e., $\hat{s} = s^\circ, \hat{\mathbf{R}} = \mathbf{R}^\circ, \hat{\mathbf{t}} = \mathbf{t}^\circ$, proving the theorem.

O. Proof of Theorem 16: Estimation Contract with Noiseless Inliers and Adversarial Outliers

As for Theorem 15, we prove Theorem 16 in two steps.

1) *TLS \equiv MC:* The proof of this part is identical to the corresponding proof in Theorem 15.

2) *Exact recovery of maximum consensus:* Different than the case with random outliers, we need condition (i), a stronger assumption on the number of inliers to guarantee recovery of the ground truth. Let the ground-truth consensus set be \mathcal{C}° whose size is at least N_{in} (the outliers can accidentally become inliers) in condition (i). Suppose an adversary wants to trick MC to return a solution $(\bar{s}, \bar{\mathbf{R}}, \bar{\mathbf{t}})$ that is different from the ground truth, s/he must have a consensus set, $\bar{\mathcal{C}}$, of size N_{in} that is no smaller than N_{in} , in which the measurements are consistent w.r.t. $(\bar{s}, \bar{\mathbf{R}}, \bar{\mathbf{t}})$. Now because the total number of measurement is $N_{\text{in}} + N_{\text{out}}$, and both \mathcal{C}° and $\bar{\mathcal{C}}$ have size at least N_{in} , this means \mathcal{C}° and $\bar{\mathcal{C}}$ must share at least $2N_{\text{in}} - (N_{\text{in}} + N_{\text{out}}) = N_{\text{in}} - N_{\text{out}}$ common correspondences. From condition (i), we know that $N_{\text{in}} - N_{\text{out}} \geq 3$. However, if \mathcal{C}° and $\bar{\mathcal{C}}$ share at least 3 common noise-free correspondences, then $(\bar{s}, \bar{\mathbf{R}}, \bar{\mathbf{t}}) \equiv (s^\circ, \mathbf{R}^\circ, \mathbf{t}^\circ)$ must be true. Therefore, we conclude that there is no chance for the adversary to trick MC, and MC must recover the ground-truth transformation when condition (i) holds.

3) *Exact recovery of TEASER:* Condition (iv) guarantees that TEASER returned a consensus set that satisfies condition

(i). However, according to Section O2 only the ground truth can produce an estimate that satisfies condition (i), hence TEASER's solution must match the ground truth, proving the theorem. Note that recovering the ground truth does not guarantee that all outliers are rejected, since in the adversarial regime outliers can be indistinguishable from inliers (*i.e.*, an outlier i can satisfy $\mathbf{b}_i = s^\circ \mathbf{R}^\circ \mathbf{a}_i + \mathbf{t}^\circ$), a fact that, however, does not have negative repercussions on the accuracy of TEASER.

P. Proof of Theorem 17: Estimation Contract with Noisy Inliers and Adversarial Outliers

We prove Theorem 17 in two steps. First we show that under assumptions (ii) and (iii) of the theorem, the inliers produced by TEASER match the maximum consensus set and contain the set of true inliers. Then, we prove that under assumption (i) and (iv) the maximum consensus solution is also close to the ground truth, in the sense of inequalities (27)-(29).

1) *TLS \equiv MC and inliers are preserved*: We prove that under assumptions (ii) and (iii), the inliers produced by TEASER match the maximum consensus set and contain the set of true inliers. In each subproblem, Lemma 23 and assumption (iii) guarantee that TEASER computes a maximum consensus set. Assumption (iii) implies the maximum consensus set contains the inliers (and hence TEASER preserves the inliers).

2) *Noisy recovery of maximum consensus*: We now prove that maximum consensus produces a solution "close" to the ground truth under the assumptions of the theorem. Let us start with the scale subproblem. Assumption (ii) in the theorem ensures that the maximum consensus set contains all inliers in each subproblem. Let us now derive the bounds on the resulting estimate, starting from the scale estimation.

Scale error bound. Since the scale estimation selects all the inliers, plus potentially some outliers, it holds for some of the selected measurements i, j :

$$|s^\circ - \hat{s}| = |s^\circ - \hat{s} + \epsilon_k^s - \epsilon_k^s| \quad (101)$$

$$\stackrel{(a)}{\leq} |s^\circ + \epsilon_k^s - \hat{s}| + |\epsilon_k^s| \quad (102)$$

$$\stackrel{(b)}{\leq} |s_k - \hat{s}| + \alpha_{ij} \stackrel{(c)}{\leq} 2\alpha_{ij} \quad (103)$$

where in (a) we used the triangle inequality, in (b) we noticed that for inliers i, j , it holds $|\epsilon_k^s| \leq \alpha_{ij}$ and $s^\circ + \epsilon_k^s = s_k$, and in (c) we noticed that for s_k to be considered an inlier by TLS, $|s_k - \hat{s}| \leq \alpha_{ij}$. Since the inequality has to hold for all the true inliers, but we do not know which ones of the measurements selected by TLS are true inliers, we substitute the bound with:

$$|s^\circ - \hat{s}| \leq 2 \max_{ij} \alpha_{ij} \quad (104)$$

which proves the first bound in (27).

Rotation error bound. Let us now move to rotation estimation. Since the rotation subproblem also selects all the inliers, plus potentially some outliers, it holds for some of the selected measurements i, j :

$$\|s^\circ \mathbf{R}^\circ \bar{\mathbf{a}}_{ij} - \hat{\mathbf{R}} \bar{\mathbf{a}}_{ij}\| = \|s^\circ \mathbf{R}^\circ \bar{\mathbf{a}}_{ij} - \hat{\mathbf{R}} \bar{\mathbf{a}}_{ij} + \epsilon_{ij} - \epsilon_{ij}\| \quad (105)$$

$$\stackrel{(a)}{=} \|\bar{\mathbf{b}}_{ij} - \hat{\mathbf{R}} \bar{\mathbf{a}}_{ij} - \epsilon_{ij}\| \quad (106)$$

$$\stackrel{(b)}{\leq} \|\bar{\mathbf{b}}_{ij} - \hat{\mathbf{R}} \bar{\mathbf{a}}_{ij}\| + \|\epsilon_{ij}\| \stackrel{(c)}{\leq} 2\delta_{ij} \quad (107)$$

where in (a) we noticed that for inliers i, j , $\bar{\mathbf{b}}_{ij} = s^\circ \mathbf{R}^\circ \bar{\mathbf{a}}_{ij} + \epsilon_{ij}$, in (b) we used the triangle inequality, and in (c) we noticed that it holds $\|\epsilon_{ij}\| \leq \delta_{ij}$ and that for (i, j) to be considered an inlier by TLS, $\|\bar{\mathbf{b}}_{ij} - \hat{\mathbf{R}} \bar{\mathbf{a}}_{ij}\| \leq \delta_{ij}$.

Squaring the inequality and defining $\mathbf{u}_{ij} \doteq \frac{\bar{\mathbf{a}}_{ij}}{\|\bar{\mathbf{a}}_{ij}\|}$:

$$\|s^\circ \mathbf{R}^\circ \bar{\mathbf{a}}_{ij} - \hat{\mathbf{R}} \bar{\mathbf{a}}_{ij}\|^2 \leq 4\delta_{ij}^2 \iff \quad (108)$$

$$\|s^\circ \mathbf{R}^\circ \mathbf{u}_{ij} - \hat{\mathbf{R}} \mathbf{u}_{ij}\|^2 \leq 4 \frac{\delta_{ij}^2}{\|\bar{\mathbf{a}}_{ij}\|^2} = 4\alpha_{ij}^2 \quad (109)$$

Since we have at least 4 inliers i, j, h, k , whose invariant measurements include $\bar{\mathbf{a}}_{ij}, \bar{\mathbf{a}}_{ih}, \bar{\mathbf{a}}_{ik}$, and each invariant measurement satisfies (109), we can sum (member-wise) the 3 corresponding inequalities and obtain:

$$\|s^\circ \mathbf{R}^\circ \mathbf{u}_{ij} - \hat{\mathbf{R}} \mathbf{u}_{ij}\|^2 + \|s^\circ \mathbf{R}^\circ \mathbf{u}_{ih} - \hat{\mathbf{R}} \mathbf{u}_{ih}\|^2 + \quad (110)$$

$$\|s^\circ \mathbf{R}^\circ \mathbf{u}_{ik} - \hat{\mathbf{R}} \mathbf{u}_{ik}\|^2 \leq 12\alpha_{ij}^2 \quad (111)$$

$$\iff \|(s^\circ \mathbf{R}^\circ - \hat{\mathbf{R}}) \mathbf{U}_{ijhk}\|_F^2 \leq 12\alpha_{ij}^2 \quad (112)$$

where $\mathbf{U}_{ijhk} = [\mathbf{u}_{ij} \ \mathbf{u}_{ih} \ \mathbf{u}_{ik}]$ and matches the definition in the statement of the theorem. Now we note that for any two matrices \mathbf{A} and \mathbf{B} it holds $\|\mathbf{A}\mathbf{B}\|_F^2 \geq \sigma_{\min}(\mathbf{B})^2 \|\mathbf{A}\|_F^2$, which applied to (112) becomes:

$$\|(s^\circ \mathbf{R}^\circ - \hat{\mathbf{R}}) \mathbf{U}_{ijhk}\|_F^2 \geq \quad (113)$$

$$\sigma_{\min}(\mathbf{U}_{ijhk})^2 \|s^\circ \mathbf{R}^\circ - \hat{\mathbf{R}}\|_F^2 \quad (114)$$

Chaining the inequality (114) back into (112):

$$\sigma_{\min}(\mathbf{U}_{ijhk})^2 \|s^\circ \mathbf{R}^\circ - \hat{\mathbf{R}}\|_F^2 \leq 12\alpha_{ij}^2 \iff \quad (115)$$

$$\|s^\circ \mathbf{R}^\circ - \hat{\mathbf{R}}\|_F \leq 2\sqrt{3} \frac{\alpha_{ij}}{\sigma_{\min}(\mathbf{U}_{ijhk})} \quad (116)$$

Since we do not know what are the true inliers, we have to take the worst case over all possible i, j, h, k in (116), yielding:

$$\|s^\circ \mathbf{R}^\circ - \hat{\mathbf{R}}\|_F \leq 2\sqrt{3} \frac{\max_{ij} \alpha_{ij}}{\min_{ijhk} \sigma_{\min}(\mathbf{U}_{ijhk})} \quad (117)$$

We finally observe that $\sigma_{\min}(\mathbf{U}_{ijhk})$ is different from zero as long as the vectors $\mathbf{u}_{ij}, \mathbf{u}_{ih}, \mathbf{u}_{ik}$ are linearly independent (*i.e.*, do not lie in the same plane), which is the case when the four inliers are not coplanar. This proves the second bound in (28).

Translation error bound. Since the translation subproblem also selects all the inliers, plus potentially some outliers, it holds for each inlier point i :

$$\begin{aligned} \mathbf{b}_i &= s^\circ \mathbf{R}^\circ \mathbf{a}_i + \mathbf{t}^\circ + \epsilon_i \quad \text{and} \\ \mathbf{b}_i &= \hat{\mathbf{R}} \mathbf{a}_i + \hat{\mathbf{t}} + \phi_i \quad \text{with } \|\phi_i\| \leq \sqrt{3}\beta_i \end{aligned} \quad (118)$$

where the first follows from the definition of inlier, and the second from the maximum error of a measurement considered inlier by TLS (the factor $\sqrt{3}$ comes from the fact that translation is estimated component-wise, which means $\|\phi_i\|^2 = [\phi_i]_1^2 + [\phi_i]_2^2 + [\phi_i]_3^2 \leq 3\beta_i^2$). Combining the equalities (118):

$$\begin{aligned} \hat{\mathbf{R}} \mathbf{a}_i + \hat{\mathbf{t}} + \phi_i &= s^\circ \mathbf{R}^\circ \mathbf{a}_i + \mathbf{t}^\circ + \epsilon_i \\ \iff \mathbf{t}^\circ - \hat{\mathbf{t}} &= (\hat{\mathbf{R}} - s^\circ \mathbf{R}^\circ) \mathbf{a}_i + (\phi_i - \epsilon_i) \end{aligned} \quad (119)$$

Since we have at least 4 inliers i, j, h, k , each satisfying the equality (119), we subtract (member-wise) the first equality

(including inlier i) from the sum of the last 3 equalities (including points j, h, k):

$$2(\mathbf{t}^\circ - \hat{\mathbf{t}}) = \sum_{l \in \{j, h, k\}} (\hat{\mathbf{s}}\hat{\mathbf{R}} - s^\circ \mathbf{R}^\circ)(\mathbf{a}_l - \mathbf{a}_i) \quad (120)$$

$$+ \sum_{l \in \{j, h, k\}} (\phi_l - \epsilon_l - \phi_i + \epsilon_i) \quad (121)$$

Taking the norm of both members and using the triangle inequality:

$$2\|\mathbf{t}^\circ - \hat{\mathbf{t}}\| \leq \sum_{l \in \{j, h, k\}} \|(\hat{\mathbf{s}}\hat{\mathbf{R}} - s^\circ \mathbf{R}^\circ)(\mathbf{a}_l - \mathbf{a}_i)\| \quad (122)$$

$$+ \sum_{l \in \{j, h, k\}} (\|\phi_l\| + \|\epsilon_l\| + \|\phi_i\| + \|\epsilon_i\|) \quad (123)$$

Noting that $\|(\hat{\mathbf{s}}\hat{\mathbf{R}} - s^\circ \mathbf{R}^\circ)(\mathbf{a}_l - \mathbf{a}_i)\| = \|(\hat{\mathbf{s}}\hat{\mathbf{R}} - s^\circ \mathbf{R}^\circ)\bar{\mathbf{a}}_{il}\|$ must satisfy the inequality (107), and the last 4 terms in (123) have norm bounded by β and $\sqrt{3}\beta$:

$$2\|\mathbf{t}^\circ - \hat{\mathbf{t}}\| \leq \sum_{l \in \{j, h, k\}} 2\delta_{ij} + \sum_{l \in \{j, h, k\}} (2 + 2\sqrt{3})\beta \quad (124)$$

Since we assume all the points to have the same noise bound β , it holds $\delta_{ij} = 2\beta$ and the right-hand-side of (124) becomes $(18 + 6\sqrt{3})\beta$, from which it follows:

$$2\|\mathbf{t}^\circ - \hat{\mathbf{t}}\| \leq (18 + 6\sqrt{3})\beta \iff \|\mathbf{t}^\circ - \hat{\mathbf{t}}\| \leq (9 + 3\sqrt{3})\beta \quad (125)$$

which proves the last bound in (29).

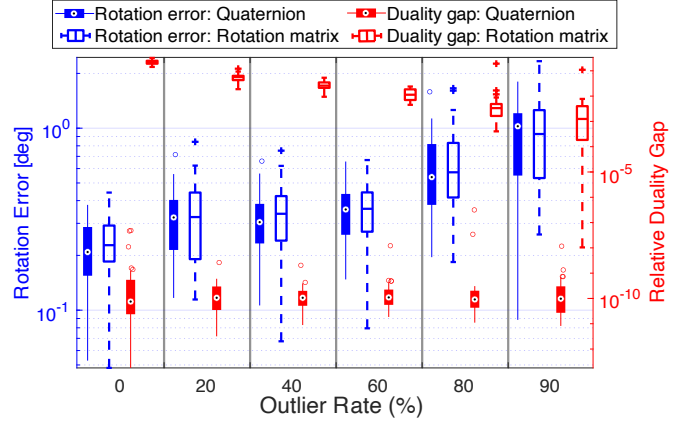
Q. SDP Relaxations: Quaternion vs. Rotation Matrix

In Section XI, we compare the rotation estimation error between the quaternion-based SDP relaxation (eq. (24), [33]) and the rotation-matrix-based SDP relaxation ([32]) on the Bunny dataset. In this section, we also report the *relative duality gap* for the two relaxations on the Bunny dataset. Fig. 11(a) shows the relative duality gap (boxes in red, scale on the right-hand y-axis) and the rotation error (boxes in blue, scale on the left-hand y-axis). The figure shows that the quaternion-based relaxation is always *tighter* than the rotation-matrix-based relaxation, although both relaxations achieve similar performance in terms of rotation estimation errors.

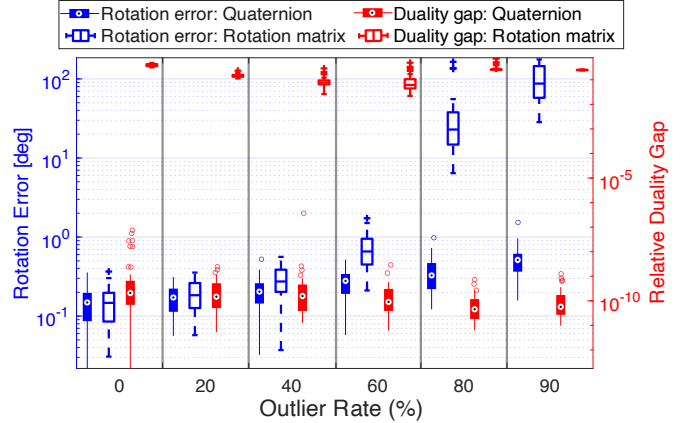
To further stress-test the two techniques, we compare their performance in a more randomized test setting [33]: at each Monte Carlo run, we first randomly sample $K = 40$ unit vectors from the 3D unit sphere, then apply a random rotation and isotropic Gaussian noise (standard deviation $\sigma = 0.01$) to the K unit vectors. Fig. 11(b) shows the results under this setting. We can see that the quaternion-based relaxation remains tight, while the rotation-matrix-based relaxation has a much larger relative duality gap. In addition, a tighter relaxation translates to more accurate rotation estimation: the quaternion-based relaxation dominates the rotation-matrix relaxation and obtains much more accurate rotation estimates, especially at $> 80\%$ outlier rates.

R. Stanford Dataset: Extra Results

In all the following experiments, $\bar{c}^2 = 1$ for scale, rotation and translation estimation.



(a) Results on the Bunny dataset.



(b) Results on random simulated unit vectors.

Fig. 11. Comparison of rotation errors (left-axis, blue) and relative duality gap (right-axis, red) between the quaternion-based SDP relaxation (filled boxplots) and the rotation-matrix-based SDP relaxation (empty boxplots) on (a) the Bunny dataset and (b) random simulated unit vectors.

1) *Benchmark on Synthetic Datasets:* As stated in Section XI-B in the main document, we have benchmarked TEASER against four state-of-the-art methods in point cloud registration (FGR [52], GORE [21], RANSAC1K, RANSAC) at increasing level of outliers, using four datasets from the Stanford 3D Scanning Repository [88]: Bunny, Armadillo, Dragon and Buddha. Results for the Bunny are showed in the main document and here we show the results for the other three datasets in Fig. 12. The figure confirms that TEASER dominates the other techniques is insensitive to extreme outlier rates.

2) *TEASER on High Noise:* In the main document, the standard deviation σ of the isotropic Gaussian noise is set to be 0.01, with the point cloud scaled inside a unit cube $[0, 1]^3$. In this section, we give a visual illustration of how corrupted the point cloud is after adding noise of such magnitude. In addition, we increase the noise standard deviation σ to be 0.1, visualize the corresponding corrupted point cloud, and show that TEASER can still recover the ground-truth transformation with reasonable accuracy. Fig. 14 illustrates a *clean* (noise-free) Bunny model (Fig. 14(a)) scaled inside the unit cube and its variants by adding different levels of isotropic Gaussian noise and outliers. From Fig. 14(b)(c), we can see $\sigma = 0.01$ (noise used in the main document) is a reasonable noise standard deviation for real-world applications, while $\sigma = 0.1$ destroys the geometric structure of the Bunny and it is beyond

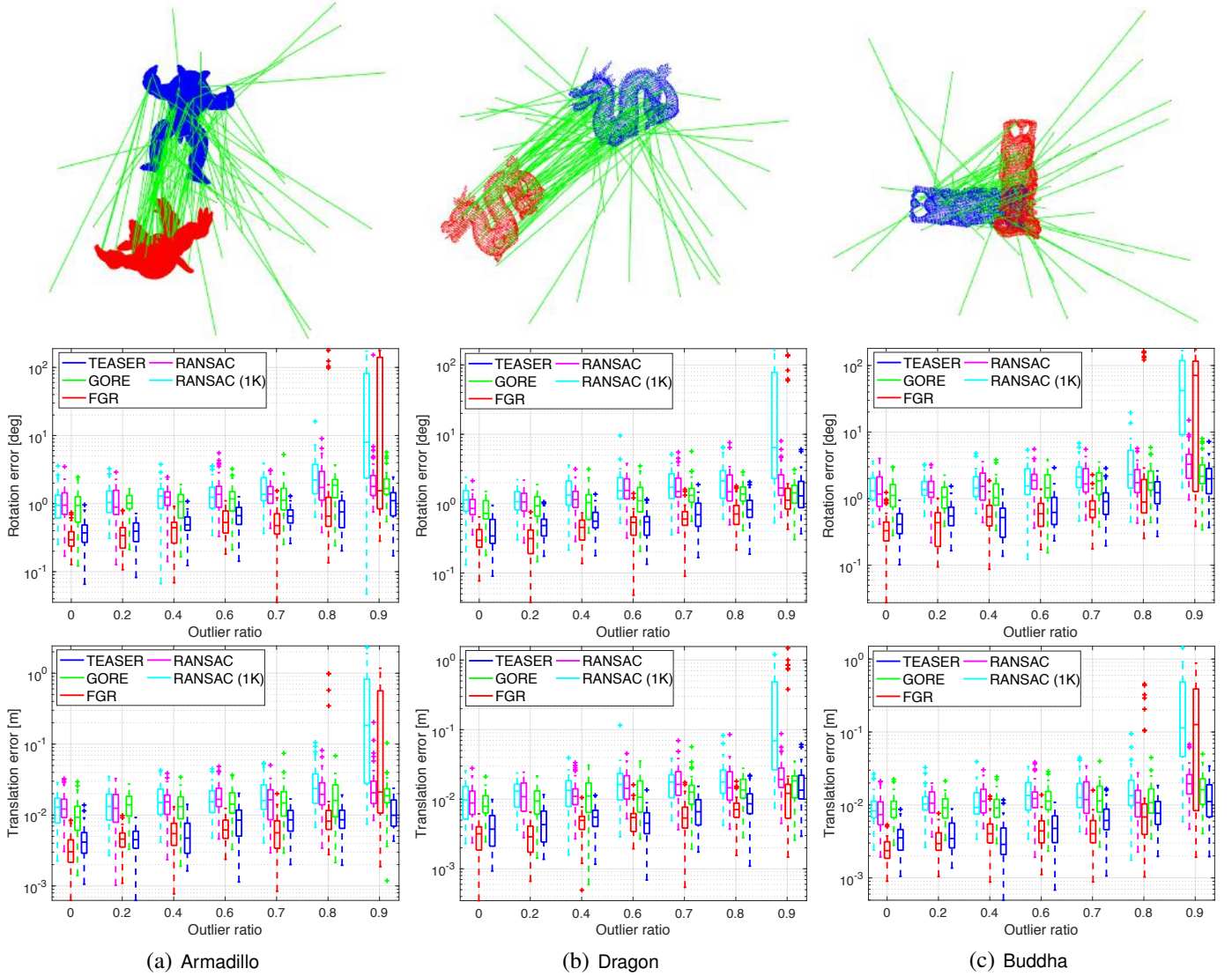


Fig. 12. Results for the Armadillo, Dragon and Buddha datasets at increasing levels of outliers. First row: example of putative correspondences with 50% outliers. Blue points are the model point cloud, red points are the transformed scene model. Second row: rotation error produced by five methods including TEASER. Third row: translation error produced by the five methods.

the noise typically encountered in robotics and computer vision applications. Fig. 14(d) shows the Bunny with high noise ($\sigma = 0.1$) and 50% outliers. Under this setup, we run TEASER to register the two point clouds and recover the relative transformation. TEASER can still return rotation and translation estimates that are close to the ground-truth transformation. However, due to the severe noise corruption, even a successful registration fails to yield a visually convincing registration result to human perception. For example, Fig. 13 shows a representative TEASER registration result with $\sigma = 0.1$ noise corruption and 50% outlier rate, where accurate transformation estimates are obtained (rotation error is 3.42° and translation error is 0.098m), but the Bunny in the cluttered scene is hardly visible, even when super-imposed to the clean model.

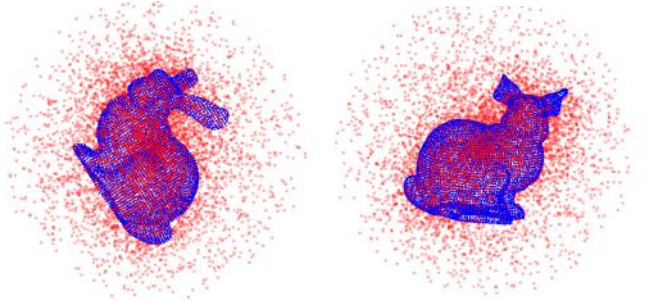


Fig. 13. A single representative TEASER registration result with $\sigma = 0.1$ noise corruption and 50% outliers, viewed from two distinctive perspectives. Clean Bunny model showed in blue and cluttered scene showed in red. Although the rotation error compared to ground-truth is 3.42° and the translation error compared to ground-truth is 0.098m , it is challenging for a human to confirm the correctness of the registration result.

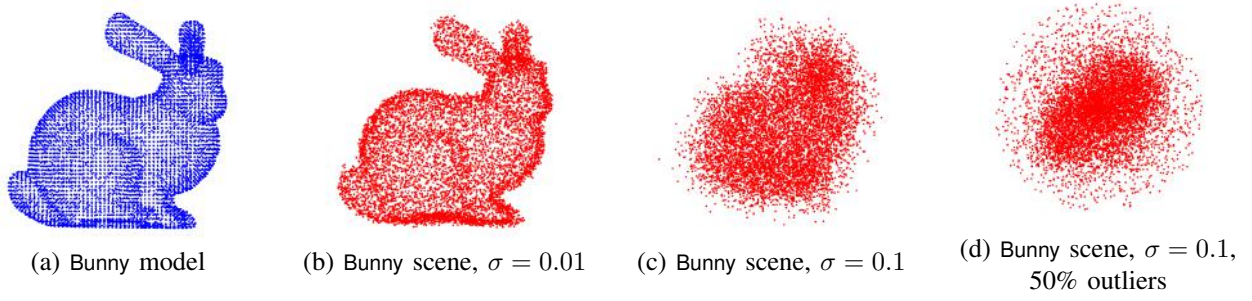
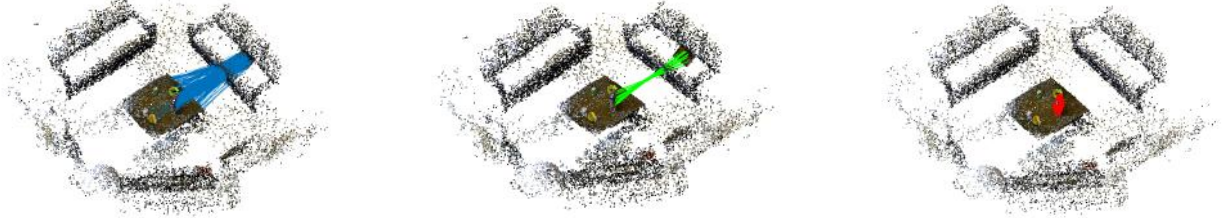


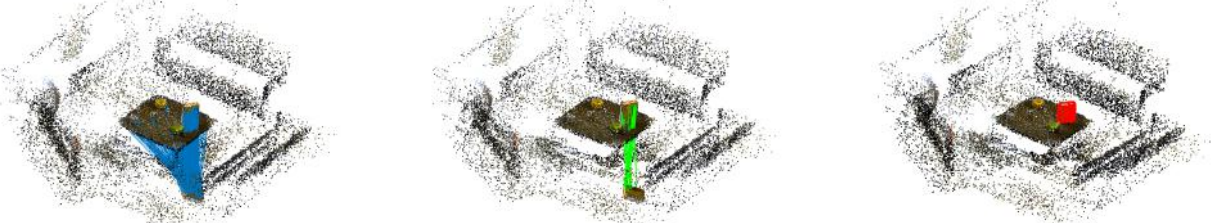
Fig. 14. Bunny point cloud scaled inside unit cube $[0, 1]^3$ and corrupted by different levels of noise and outliers, all viewed from the same perspective angle. (a) Clean Bunny model point cloud, scaled inside unit cube $[0, 1]^3$. (b) Bunny scene, generated from (a) by adding isotropic Gaussian noise with standard deviation $\sigma = 0.01$. (c) Bunny scene, generated from (a) by adding isotropic Gaussian noise with $\sigma = 0.1$. (d) Bunny scene, generated from (a) by adding isotropic Gaussian noise with $\sigma = 0.1$ and 50% random outliers.

S. Object Pose Estimation: Extra Results

Fig. 15 shows the registration results for the 8 selected scenes from the University of Washington RGB-D datasets [30]. The inlier correspondence ratios for *cereal box* are all below 10% and typically below 5%. TEASER is able to compute a highly-accurate estimate of the relative pose using a handful of inliers. Due to the distinctive shape of *cap*, FPFH produces a higher number of inliers and TEASER is able to register the object-scene pair without problems.



scene-2, # of FPFH correspondences: 550, Inlier ratio: 4.55%, Rotation error: 0.120, Translation error: 0.052.



scene-4, # of FPFH correspondences: 636, Inlier ratio: 4.56%, Rotation error: 0.042, Translation error: 0.051.



scene-5, # of FPFH correspondences: 685, Inlier ratio: 2.63%, Rotation error: 0.146, Translation error: 0.176.



scene-7, # of FPFH correspondences: 416, Inlier ratio: 3.13%, Rotation error: 0.058, Translation error: 0.097.

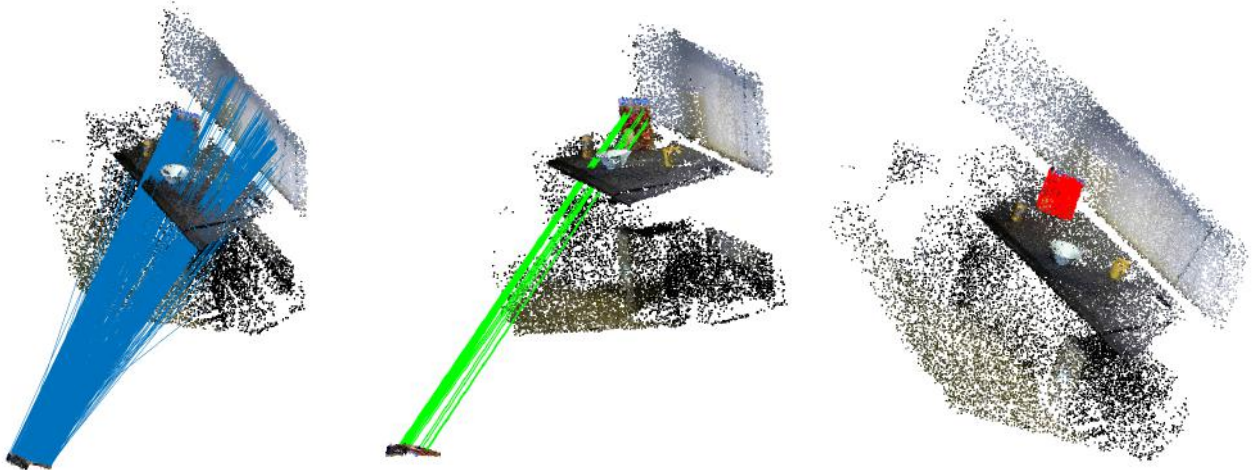
Fig. 15. Object pose estimation on the large-scale RGB-D dataset [30]. First column: FPFH correspondences, second column: inlier correspondences after TEASER, third column: registration result with the registered object highlighted in red. Scene number and related registration information are listed below each scene.



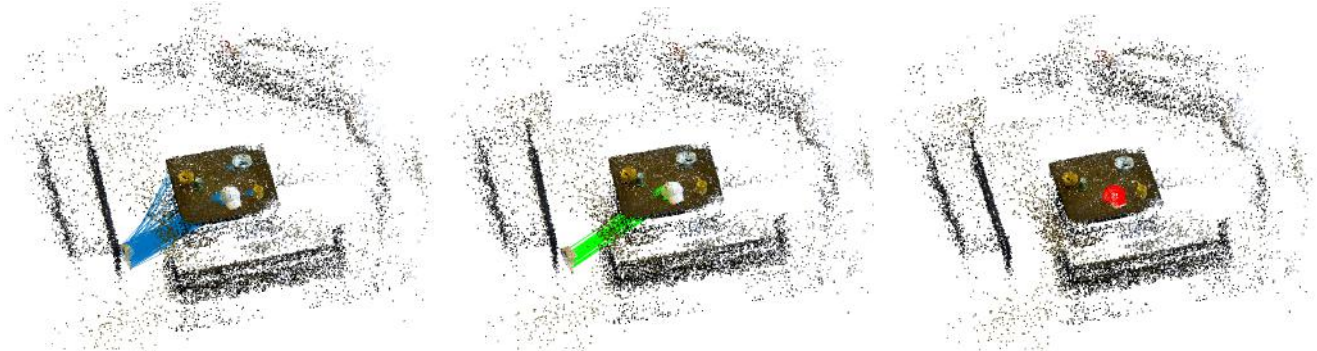
scene-9, # of FPFH correspondences: 651, Inlier ratio: 8.29%, Rotation error: 0.036, Translation error: 0.011.



scene-11, # of FPFH correspondences: 445, Inlier ratio: 6.97%, Rotation error: 0.028, Translation error: 0.016.



scene-13, # of FPFH correspondences: 612, Inlier ratio: 5.23%, Rotation error: 0.036, Translation error: 0.064.



scene-1, # of FPFH correspondences: 207, Inlier ratio: 16.91%, Rotation error: 0.066, Translation error: 0.090.

Fig. 15. Object pose estimation on the large-scale RGB-D dataset [30] (cont.).

T. Proof of Proposition 33: Projection onto $\bar{\mathcal{L}}$

Proof: We first analyze the constraints defined by the affine subspace $\bar{\mathcal{L}}$ (84). For any matrix $\bar{M} \in \bar{\mathcal{L}}$, \bar{M} can be written as $\bar{M} = \bar{Q} - \hat{\mu}J + \Delta$, with $\Delta \in \mathcal{H}$ and \bar{M} satisfies $\bar{M}\bar{x} = 0$, we write this in matrix form:

$$\begin{bmatrix} -\hat{\mu}I_4 - \sum_{k=1}^K [\Delta]_{kk} & [\bar{Q}]_{01} + [\Delta]_{01} & \cdots & [\bar{Q}]_{0k} + [\Delta]_{0k} & \cdots & [\bar{Q}]_{0K} + [\Delta]_{0K} \\ [\bar{Q}]_{01} - [\Delta]_{01} & [\bar{Q}]_{11} + [\Delta]_{11} & \cdots & [\Delta]_{1k} & \cdots & [\Delta]_{1K} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ [\bar{Q}]_{0k} - [\Delta]_{0k} & -[\Delta]_{1k} & \cdots & [\bar{Q}]_{kk} + [\Delta]_{kk} & \cdots & [\Delta]_{kK} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ [\bar{Q}]_{0K} - [\Delta]_{0K} & -[\Delta]_{1K} & \cdots & -[\Delta]_{kK} & \cdots & [\bar{Q}]_{KK} + [\Delta]_{KK} \end{bmatrix} \begin{bmatrix} e \\ \hat{\theta}_1 e \\ \vdots \\ \hat{\theta}_k e \\ \vdots \\ \hat{\theta}_K e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad (126)$$

where the expressions for $[\bar{Q}]_{0k}$ and $[\bar{Q}]_{kk}$ are given by the following:

$$\begin{cases} [\bar{Q}]_{0k} = \begin{bmatrix} -[\bar{a}_k]_\times^2 + \frac{1}{4}(\|\hat{\xi}_k\|^2 - \bar{c}^2)I_3 + \frac{1}{2}\bar{a}_k^\top \hat{\xi}_k I_3 - \frac{1}{2}[\hat{\xi}_k]_\times [\bar{a}_k]_\times - \frac{1}{2}\hat{\xi}_k \bar{a}_k^\top & \frac{1}{2}[\hat{\xi}_k]_\times \bar{a}_k \\ \frac{1}{2}([\hat{\xi}_k]_\times \bar{a}_k)^\top & \frac{1}{4}(\|\hat{\xi}_k\|^2 - \bar{c}^2) \end{bmatrix} \\ [\bar{Q}]_{kk} = \begin{bmatrix} -2[\bar{a}_k]_\times^2 + \frac{1}{2}(\|\hat{\xi}_k\|^2 + \bar{c}^2)I_3 + \bar{a}_k^\top \hat{\xi}_k I_3 - [\hat{\xi}_k]_\times [\bar{a}_k]_\times - \hat{\xi}_k \bar{a}_k^\top & [\hat{\xi}_k]_\times \bar{a}_k \\ ([\hat{\xi}_k]_\times \bar{a}_k)^\top & \frac{1}{2}(\|\hat{\xi}_k\|^2 + \bar{c}^2) \end{bmatrix} \end{cases}, \forall k = 1, \dots, K. \quad (127)$$

Denote $\hat{\theta}_0 = +1$, we write the k -th ($k = 1, \dots, K$) block-row equality of (126):

$$\left(\hat{\theta}_0 [\bar{Q}]_{0k} + \hat{\theta}_k [\bar{Q}]_{kk} + \hat{\theta}_k [\Delta]_{kk} + \sum_{i=0, i \neq k}^K \hat{\theta}_i [\Delta]_{ki} \right) e = 0 \quad (128)$$

$$\Leftrightarrow \begin{bmatrix} \star & \hat{\theta}_0 [\bar{Q}]_{0k}^v + \hat{\theta}_k [\bar{Q}]_{kk}^v + \hat{\theta}_k [\Delta]_{kk}^v + \sum_{i=0, i \neq k}^K \hat{\theta}_i [\Delta]_{ki}^v \\ \star & \hat{\theta}_0 [\bar{Q}]_{0k}^s + \hat{\theta}_k [\bar{Q}]_{kk}^s + \hat{\theta}_k [\Delta]_{kk}^s \end{bmatrix} e = 0 \quad (129)$$

$$\Leftrightarrow \begin{cases} [\Delta]_{kk}^v = -\hat{\theta}_0 \hat{\theta}_k [\bar{Q}]_{0k}^v - [\bar{Q}]_{kk}^v - \sum_{i=0, i \neq k}^K \hat{\theta}_i \hat{\theta}_k [\Delta]_{ki}^v = -(\frac{1}{2}\hat{\theta}_0 \hat{\theta}_k + 1)[\hat{\xi}_k]_\times \bar{a}_k - \sum_{i=0, i \neq k}^K \hat{\theta}_i \hat{\theta}_k [\Delta]_{ki}^v \\ [\Delta]_{kk}^s = -\hat{\theta}_0 \hat{\theta}_k [\bar{Q}]_{0k}^s - [\bar{Q}]_{kk}^s = -(\frac{1}{4}\hat{\theta}_0 \hat{\theta}_k + \frac{1}{2})\|\hat{\xi}_k\|^2 + (\frac{1}{4}\hat{\theta}_0 \hat{\theta}_k - \frac{1}{2})\bar{c}^2 \end{cases}, \quad (130)$$

and from eq. (130), we can get the a sum of all $[\Delta]_{kk}^v$ to be:

$$\begin{aligned} \sum_{k=1}^K [\Delta]_{kk}^v &= \sum_{k=1}^K -\left(\frac{1}{2}\hat{\theta}_0 \hat{\theta}_k + 1\right) [\hat{\xi}_k]_\times \bar{a}_k - \sum_{k=1}^K \sum_{i=0, i \neq k}^K \hat{\theta}_i \hat{\theta}_k [\Delta]_{ki}^v \\ &= \sum_{k=1}^K -\left(\frac{1}{2}\hat{\theta}_0 \hat{\theta}_k + 1\right) [\hat{\xi}_k]_\times \bar{a}_k + \sum_{k=1}^K \hat{\theta}_0 \hat{\theta}_k [\Delta]_{0k}^v, \end{aligned} \quad (131)$$

and the sum of all $[\Delta]_{kk}^s$ to be:

$$\sum_{k=1}^K [\Delta]_{kk}^s = \sum_{k=1}^K -\left(\frac{1}{4}\hat{\theta}_0 \hat{\theta}_k + \frac{1}{2}\right) \|\hat{\xi}_k\|^2 + \left(\frac{1}{4}\hat{\theta}_0 \hat{\theta}_k - \frac{1}{2}\right) \bar{c}^2. \quad (132)$$

Now we write the 0-th block row equality of (126):

$$\left(-\hat{\theta}_0 \hat{\mu} I_4 - \sum_{k=1}^K \hat{\theta}_0 [\Delta]_{kk} + \sum_{k=1}^K \hat{\theta}_k [\bar{Q}]_{0k} + \sum_{k=1}^K \hat{\theta}_k [\Delta]_{0k} \right) e = 0 \quad (133)$$

$$\Leftrightarrow \begin{bmatrix} \star & -\sum_{k=1}^K \hat{\theta}_0 [\Delta]_{kk}^v + \sum_{k=1}^K \hat{\theta}_k [\bar{Q}]_{0k}^v + \sum_{k=1}^K \hat{\theta}_k [\Delta]_{0k}^v \\ \star & -\hat{\theta}_0 \hat{\mu} - \sum_{k=1}^K \hat{\theta}_0 [\Delta]_{kk}^s + \sum_{k=1}^K \hat{\theta}_k [\bar{Q}]_{0k}^s \end{bmatrix} e = 0 \quad (134)$$

$$\Leftrightarrow \begin{cases} \sum_{k=1}^K [\Delta]_{kk}^v = \sum_{k=1}^K \hat{\theta}_0 \hat{\theta}_k [\bar{Q}]_{0k}^v + \sum_{k=1}^K \hat{\theta}_0 \hat{\theta}_k [\Delta]_{0k}^v \\ \sum_{k=1}^K [\Delta]_{kk}^s = -\hat{\mu} + \sum_{k=1}^K \hat{\theta}_0 \hat{\theta}_k [\bar{Q}]_{0k}^s \end{cases} \quad (135)$$

$$\Leftrightarrow \begin{cases} \sum_{k=1}^K [\Delta]_{kk}^v = \sum_{k=1}^K \frac{1}{2}\hat{\theta}_0 \hat{\theta}_k [\hat{\xi}_k]_\times \bar{a}_k + \sum_{k=1}^K \hat{\theta}_0 \hat{\theta}_k [\Delta]_{0k}^v \\ \sum_{k=1}^K [\Delta]_{kk}^s = -\hat{\mu} + \sum_{k=1}^K \frac{1}{4}\hat{\theta}_0 \hat{\theta}_k (\|\hat{\xi}_k\|^2 - \bar{c}^2) \end{cases}, \quad (136)$$

and compare eq. (136) with eq. (131):

$$\sum_{k=1}^K -\left(\frac{1}{2}\hat{\theta}_0\hat{\theta}_k + 1\right) [\hat{\xi}_k]_{\times} \bar{\mathbf{a}}_k + \sum_{k=1}^K \hat{\theta}_0\hat{\theta}_k [\Delta]_{0k}^v = \sum_{k=1}^K \frac{1}{2}\hat{\theta}_0\hat{\theta}_k [\hat{\xi}_k]_{\times} \bar{\mathbf{a}}_k + \sum_{k=1}^K \hat{\theta}_0\hat{\theta}_k [\Delta]_{0k}^v \quad (137)$$

$$\Leftrightarrow \sum_{k=1}^K -\left(\frac{1}{2}\hat{\theta}_0\hat{\theta}_k + 1\right) [\hat{\xi}_k]_{\times} \bar{\mathbf{a}}_k = \sum_{k=1}^K \frac{1}{2}\hat{\theta}_0\hat{\theta}_k [\hat{\xi}_k]_{\times} \bar{\mathbf{a}}_k \quad (138)$$

$$\Leftrightarrow \sum_{\hat{\theta}_k=+1} -\frac{3}{2}[\hat{\xi}_k]_{\times} \bar{\mathbf{a}}_k + \sum_{\hat{\theta}_k=-1} -\frac{1}{2}[\hat{\xi}_k]_{\times} \bar{\mathbf{a}}_k = \sum_{\hat{\theta}_k=+1} \frac{1}{2}[\hat{\xi}_k]_{\times} \bar{\mathbf{a}}_k + \sum_{\hat{\theta}_k=-1} -\frac{1}{2}[\hat{\xi}_k]_{\times} \bar{\mathbf{a}}_k \quad (139)$$

$$\Leftrightarrow \sum_{\hat{\theta}_k=+1} [\hat{\xi}_k]_{\times} \bar{\mathbf{a}}_k = \mathbf{0}, \quad (140)$$

which states that for inliers $\hat{\theta}_k = +1$, the sum $\sum_{\hat{\theta}_k=+1} [\hat{\xi}_k]_{\times} \bar{\mathbf{a}}_k = \mathbf{0}$ must hold¹³. We then compare eq. (136) with eq. (132):

$$\sum_{k=1}^K -\left(\frac{1}{4}\hat{\theta}_0\hat{\theta}_k + \frac{1}{2}\right) \|\hat{\xi}_k\|^2 + \left(\frac{1}{4}\hat{\theta}_0\hat{\theta}_k - \frac{1}{2}\right) \bar{c}^2 = -\hat{\mu} + \sum_{k=1}^K \frac{1}{4}\hat{\theta}_0\hat{\theta}_k (\|\hat{\xi}_k\|^2 - \bar{c}^2) \quad (141)$$

$$\Leftrightarrow \sum_{k=1}^K -\left(\frac{1}{4}\hat{\theta}_k + \frac{1}{2}\right) \|\hat{\xi}_k\|^2 + \left(\frac{1}{4}\hat{\theta}_k - \frac{1}{2}\right) \bar{c}^2 = -\left(\frac{1}{2}(\hat{\theta}_k + 1)\|\hat{\xi}_k\|^2 + \frac{1}{2}(1 - \hat{\theta}_k)\bar{c}^2\right) + \sum_{k=1}^K \frac{1}{4}\hat{\theta}_k (\|\hat{\xi}_k\|^2 - \bar{c}^2) \quad (142)$$

$$\Leftrightarrow \sum_{k=1}^K -\left(\frac{1}{4}\hat{\theta}_k + \frac{1}{2}\right) \|\hat{\xi}_k\|^2 + \left(\frac{1}{4}\hat{\theta}_k - \frac{1}{2}\right) \bar{c}^2 = \sum_{k=1}^K -\left(\frac{1}{4}\hat{\theta}_k + \frac{1}{2}\right) \|\hat{\xi}_k\|^2 + \left(\frac{1}{4}\hat{\theta}_k - \frac{1}{2}\right) \bar{c}^2, \quad (143)$$

which holds true without any condition. Therefore, we conclude that the equality constraint of the 0-th block row adds no extra constraints on the matrix Δ . This leads to the following Theorem.

Theorem 34 (Equivalent Projection to $\bar{\mathcal{L}}$). *The projection of a matrix M onto $\bar{\mathcal{L}}$ is equivalent to:*

$$\Pi_{\bar{\mathcal{L}}}(M) = \Pi_{\bar{\mathcal{H}}}(M - \bar{Q} + \hat{\mu}J) + \bar{Q} - \hat{\mu}J, \quad (144)$$

where $\bar{\mathcal{H}}$ is defined by:

$$\bar{\mathcal{H}} \doteq \{\Delta : \Delta \in \mathcal{H}, \Delta \text{ satisfies eq. (130)}\} \quad (145)$$

Next we prove that the projection steps in Proposition 33 define the correct projection $\Pi_{\bar{\mathcal{H}}}$. To this end, we use \bar{H} to denote the projection of a matrix $H \in \mathcal{S}^{4(K+1)}$ onto the set $\bar{\mathcal{H}}$. By the definition of projection, \bar{H} is the minimizer of the following optimization:

$$\bar{H} = \arg \min_{\Delta \in \bar{\mathcal{H}}} \|H - \Delta\|_F^2. \quad (146)$$

Denoting the objective function of eq. (146) as $f = \|H - \Delta\|_F^2$, we separate the Frobenius norm of $H - \Delta$ into a sum of block-wise Frobenius norms:

$$\begin{aligned} f &= \|H - \Delta\|_F^2 \quad (147) \\ &= \underbrace{\sum_{k=0}^K \|\Delta_{kk}^m - [H]_{kk}^m\|_F^2}_{f_{kk}^m: \text{diagonal block matrix part}} + \underbrace{\sum_{k=0}^K \|\Delta_{kk}^s - [H]_{kk}^s\|_F^2}_{f_{kk}^s: \text{diagonal block scalar part}} + 2 \underbrace{\sum_{l=1}^L \|\Delta_{\mathcal{Z}(l)}^m - [H]_{\mathcal{Z}(l)}^m\|_F^2}_{f_{\mathcal{Z}}^m: \text{off-diagonal block matrix part}} + 2 \underbrace{\sum_{l=1}^L \|\Delta_{\mathcal{Z}(l)}^s - [H]_{\mathcal{Z}(l)}^s\|_F^2}_{f_{\mathcal{Z}}^s: \text{off-diagonal block scalar part}} \\ &\quad + 2 \underbrace{\left(\sum_{k=0}^K \|\Delta_{kk}^v - [H]_{kk}^v\|_F^2 + \sum_{l=1}^L \|\Delta_{\mathcal{W}(l)}^v - [H]_{\mathcal{W}(l)}^v\|_F^2 + \sum_{l=1}^L \|\Delta_{\mathcal{Z}(l)}^v - [H]_{\mathcal{Z}(l)}^v\|_F^2 \right)}_{f^v: \text{diagonal and off-diagonal vector part}}, \quad (148) \end{aligned}$$

where \mathcal{W} is the following set of L ordered indices that enumerate the lower-triangular blocks of any matrix $\Delta \in \mathcal{S}^{4(K+1)}$:

$$\mathcal{W} = \{(1, 0), \dots, (K, 0), (2, 1), \dots, (K, 1), \dots, (K, K-1)\}, \quad (149)$$

and $\mathcal{W}(l)$ equals to the flipped copy of $\mathcal{Z}(l)$ (eq. (90)). Using the separated Frobenius norm (148), we can get the projection \bar{H} block by block in the following steps.

¹³This condition is always true because this is the first-order stationary condition for the TLS optimization considering only the least squares part applied on inliers.

- 1) **Diagonal block matrix part.** From the expression of f_{kk}^m and the constraint that $\sum_{k=0}^K [\Delta]_{kk}^m = \mathbf{0}$, we can get the diagonal block matrix part of $\bar{\mathbf{H}}$ to be:

$$[\bar{\mathbf{H}}]_{kk}^m = [\mathbf{H}]_{kk}^m - \frac{\sum_{k=0}^K [\mathbf{H}]_{kk}^m}{K+1}, \quad \forall k = 0, \dots, K, \quad (150)$$

which first computes the mean of all diagonal block matrix parts, $\sum_{k=0}^K [\mathbf{H}]_{kk}^m / (K+1)$, and then subtracts the mean from each individual diagonal block matrix part $[\mathbf{H}]_{kk}^m$.

- 2) **Diagonal block scalar part.** From the expression of f_{kk}^s and the constraint from eq. (130) (which fixes the scalar parts to be known constants), we can get the diagonal block scalar part of $\bar{\mathbf{H}}$ to be:

$$[\bar{\mathbf{H}}]_{kk}^s = \begin{cases} -(\frac{1}{4}\hat{\theta}_k + \frac{1}{2})\|\hat{\xi}_k\|^2 + (\frac{1}{4}\hat{\theta}_k - \frac{1}{2})\bar{c}^2 & k = 1, \dots, K, \\ \sum_{k=1}^K (\frac{1}{4}\hat{\theta}_k + \frac{1}{2})\|\hat{\xi}_k\|^2 - (\frac{1}{4}\hat{\theta}_k - \frac{1}{2})\bar{c}^2 & k = 0. \end{cases}, \quad (151)$$

which first computes the diagonal scalar parts for $k = 1, \dots, K$ and then assigns the negative sum of all diagonal scalar parts to the scalar part of the top-left diagonal block ($[\bar{\mathbf{H}}]_{00}^s$).

- 3) **Off-diagonal block matrix part.** From the expression of $f_{\mathcal{Z}}^m$ and the constraint that $[\Delta]_{\mathcal{Z}(l)}$ is skew-symmetric, we can obtain the off-diagonal block matrix part of $\bar{\mathbf{H}}$ to be:

$$[\bar{\mathbf{H}}]_{\mathcal{Z}(l)}^m = \frac{[\mathbf{H}]_{\mathcal{Z}(l)}^m - ([\mathbf{H}]_{\mathcal{Z}(l)}^m)^\top}{2}, \quad \forall l = 1, \dots, L, \quad (152)$$

which simply projects $[\mathbf{H}]_{\mathcal{Z}(l)}^m$ to its nearest skew-symmetric matrix. Note that it suffices to only project the upper-triangular off-diagonal blocks because both \mathbf{H} and $\bar{\mathbf{H}}$ are symmetric matrices.

- 4) **Off-diagonal block scalar part.** The off-diagonal blocks are skew-symmetric matrices and the scalar parts must be zeros. Therefore, the off-diagonal block scalar parts of $\bar{\mathbf{H}}$ are:

$$[\bar{\mathbf{H}}]_{\mathcal{Z}(l)}^s = 0, \quad \forall l = 1, \dots, L. \quad (153)$$

- 5) **Off-diagonal block vector part.** The projection of the vector parts of each block is non-trivial due to the coupling between the diagonal block vector parts $[\Delta]_{kk}^v$ and the off-diagonal block vector parts $[\Delta]_{\mathcal{Z}(l)}^v, [\Delta]_{\mathcal{W}(l)}^v$. We first write the diagonal block vector parts as functions of the off-diagonal block vector parts by recalling eq. (130) and (131):

$$\begin{cases} [\Delta]_{kk}^v = -\underbrace{(\frac{1}{2}\hat{\theta}_0\hat{\theta}_k + 1)[\hat{\xi}_k] \times \bar{\mathbf{a}}_k}_{:=\phi_k} - \sum_{i=0, i \neq k} \hat{\theta}_i \hat{\theta}_k [\Delta]_{ki}^v = -\sum_{i=0, i \neq k} \hat{\theta}_i \hat{\theta}_k [\Delta]_{ki}^v + \phi_k \\ [\Delta]_{00}^v = -\sum_{k=1}^K [\Delta]_{kk}^v = \underbrace{\sum_{k=1}^K (\frac{1}{2}\hat{\theta}_0\hat{\theta}_k + 1)[\hat{\xi}_k] \times \bar{\mathbf{a}}_k}_{:=\phi_0} - \sum_{k=1}^K \hat{\theta}_0 \hat{\theta}_k [\Delta]_{0k}^v = -\sum_{i=0, i \neq 0} \hat{\theta}_i \hat{\theta}_k [\Delta]_{0i}^v + \phi_0 \end{cases}, \quad (154)$$

which shows that we can write $[\Delta]_{kk}^v, k = 0, \dots, K$ in a unified notation:

$$[\Delta]_{kk}^v = -\sum_{i=0, i \neq k}^K \hat{\theta}_k \hat{\theta}_i [\Delta]_{ki}^v + \phi_k, \quad \forall k = 0, \dots, K. \quad (155)$$

Inserting eq. (155) into the expression of f^v in eq. (148), and using the fact that $[\Delta]_{\mathcal{W}(l)}^v = -[\Delta]_{\mathcal{Z}(l)}^v$, we obtain the following unconstrained optimization problem:

$$\min_{[\Delta]_{\mathcal{Z}(l)}^v} \sum_{l=1}^L \left\| [\Delta]_{\mathcal{Z}(l)}^v - [\mathbf{H}]_{\mathcal{Z}(l)}^v \right\|^2 + \left\| [\Delta]_{\mathcal{Z}(l)}^v + [\mathbf{H}]_{\mathcal{W}(l)}^v \right\|^2 + \sum_{k=0}^K \left\| \sum_{i=0, i \neq k}^K \hat{\theta}_k \hat{\theta}_i [\Delta]_{ki}^v - \phi_k + [\mathbf{H}]_{kk}^v \right\|^2. \quad (156)$$

To solve problem (156), we first derive the partial derivatives w.r.t. each variable $[\Delta]_{\mathcal{Z}(l)}^v$. For notation simplicity, we write $\mathcal{Z}(l) = (r_l, c_l)$, where $r_l < c_l$ represent the block-row and block-column indices. In addition, we use $u(r, c)$ to denote

the pointer that returns the index of (r, c) in the set \mathcal{Z} . Using this notation, the partial derivative of f^v w.r.t. $[\Delta]_{\mathcal{Z}(l)}^v$ is:

$$\frac{\partial f^v}{\partial [\Delta]_{\mathcal{Z}(l)}^v} = \frac{\partial \left(\frac{\|[\Delta]_{\mathcal{Z}(l)}^v - [\mathbf{H}]_{\mathcal{Z}(l)}^v\|^2 + \|[\Delta]_{\mathcal{Z}(l)}^v + [\mathbf{H}]_{\mathcal{W}(l)}^v\|^2 + \|\sum_{i=0, i \neq r_l}^K \hat{\theta}_{r_l} \hat{\theta}_i [\Delta]_{r_l, i}^v - \phi_{r_l} + [\mathbf{H}]_{r_l, r_l}^v\|^2 + \|\sum_{i=0, i \neq c_l}^K \hat{\theta}_{c_l} \hat{\theta}_i [\Delta]_{c_l, i}^v - \phi_{c_l} + [\mathbf{H}]_{c_l, c_l}^v\|^2}{\partial [\Delta]_{\mathcal{Z}(l)}^v} \right)}{\partial [\Delta]_{\mathcal{Z}(l)}^v} \quad (157)$$

$$= 2 \left(\hat{\theta}_{r_l} \hat{\theta}_{c_l} \left(\sum_{\substack{i=0 \\ i \neq r_l}}^K \hat{\theta}_{r_l} \hat{\theta}_i [\Delta]_{r_l, i}^v - \phi_{r_l} + [\mathbf{H}]_{r_l, r_l}^v \right) - \hat{\theta}_{c_l} \hat{\theta}_{r_l} \left(\sum_{\substack{i=0 \\ i \neq c_l}}^K \hat{\theta}_{c_l} \hat{\theta}_i [\Delta]_{c_l, i}^v - \phi_{c_l} + [\mathbf{H}]_{c_l, c_l}^v \right) \right) \quad (158)$$

$$= 2 \left(4[\Delta]_{\mathcal{Z}(l)}^v - [\mathbf{H}]_{\mathcal{Z}(l)}^v + [\mathbf{H}]_{\mathcal{W}(l)}^v + \sum_{i \neq r_l, c_l}^K \hat{\theta}_{r_l} \hat{\theta}_i [\Delta]_{r_l, i}^v - \sum_{i \neq r_l, c_l}^K \hat{\theta}_{r_l} \hat{\theta}_i [\Delta]_{c_l, i}^v + \hat{\theta}_{r_l} \hat{\theta}_{c_l} (\phi_{c_l} - \phi_{r_l} + [\mathbf{H}]_{r_l, r_l}^v - [\mathbf{H}]_{c_l, c_l}^v) \right). \quad (159)$$

Notice that not all of the $[\Delta]_{r_l, i}^v$ and $[\Delta]_{c_l, i}^v$ in eq. (159) belong to the upper-triangular part of Δ , therefore, we use the fact that $[\Delta]_{ij}^v = -[\Delta]_{ji}^v$ for any $i \neq j$ to convert all lower-triangular vectors to upper-triangular. In addition, we set the partial derivative to zero to obtain sufficient and necessary conditions to solve the convex unconstrained optimization (156):

$$4[\Delta]_{\mathcal{Z}(l)}^v + \sum_{i \neq r_l, c_l}^K \hat{\theta}_{c_l} \hat{\theta}_i [\Delta]_{r_l, i}^v - \sum_{i \neq r_l, c_l}^K \hat{\theta}_{r_l} \hat{\theta}_i [\Delta]_{c_l, i}^v = \underbrace{[\mathbf{H}]_{\mathcal{Z}(l)}^v - [\mathbf{H}]_{\mathcal{W}(l)}^v + \hat{\theta}_{r_l} \hat{\theta}_{c_l} (\phi_{r_l} - \phi_{c_l} - [\mathbf{H}]_{r_l, r_l}^v + [\mathbf{H}]_{c_l, c_l}^v)}_{:= \mathcal{F}_l(\mathbf{H})} \quad (160)$$

$$\Leftrightarrow 4[\Delta]_{\mathcal{Z}(l)}^v + \sum_{i < r_l} \hat{\theta}_{c_l} \hat{\theta}_i [\Delta]_{r_l, i}^v + \sum_{i > r_l, i \neq c_l} \hat{\theta}_{c_l} \hat{\theta}_i [\Delta]_{r_l, i}^v - \sum_{i < c_l, i \neq r_l} \hat{\theta}_{r_l} \hat{\theta}_i [\Delta]_{c_l, i}^v - \sum_{i > c_l} \hat{\theta}_{r_l} \hat{\theta}_i [\Delta]_{c_l, i}^v = \mathcal{F}_l(\mathbf{H}) \quad (161)$$

$$\Leftrightarrow 4[\Delta]_{\mathcal{Z}(l)}^v - \sum_{i < r_l} \hat{\theta}_{c_l} \hat{\theta}_i [\Delta]_{u(i, r_l)}^v + \sum_{i > r_l, i \neq c_l} \hat{\theta}_{c_l} \hat{\theta}_i [\Delta]_{u(r_l, i)}^v + \sum_{i < c_l, i \neq r_l} \hat{\theta}_{r_l} \hat{\theta}_i [\Delta]_{u(i, c_l)}^v - \sum_{i > c_l} \hat{\theta}_{r_l} \hat{\theta}_i [\Delta]_{u(c_l, i)}^v = \mathcal{F}_l(\mathbf{H}) \quad (162)$$

$$\Leftrightarrow \mathbf{A}_l^T \mathcal{C}(\Delta) = \mathcal{F}_l^T(\mathbf{H}), \quad (163)$$

where the linear map $\mathcal{C} : \mathcal{S}^{4(K+1)} \rightarrow \mathbb{R}^{L \times 3}$ is as defined in eq. (91), and $\mathcal{F}_l(\mathbf{H}) \in \mathbb{R}^3$ is defined as in eq. (160), and $\mathbf{A}_l \in \mathbb{R}^L$ is the following vector:

$$\mathbf{A}_l(u) = \begin{cases} 4 & \text{if } u = l \\ -\hat{\theta}_{c_l} \hat{\theta}_i & \text{if } u = u(i, r_l), \forall 0 \leq i < r_l \\ \hat{\theta}_{c_l} \hat{\theta}_i & \text{if } u = u(r_l, i), \forall r_l < i \leq K, i \neq c_l \\ \hat{\theta}_{r_l} \hat{\theta}_i & \text{if } u = u(i, c_l), \forall 0 \leq i < c_l, i \neq r_l \\ -\hat{\theta}_{r_l} \hat{\theta}_i & \text{if } u = u(c_l, i), \forall c_l < i \leq K \\ 0 & \text{otherwise} \end{cases}, \quad (164)$$

where $\mathbf{A}_l(u)$ denotes the u -th entry of vector \mathbf{A}_l . Because the partial derivative w.r.t. each variable $[\Delta]_{\mathcal{Z}(l)}^v, l = 1, \dots, L$, must be zero at the global minimizer, $\bar{\mathbf{H}}$, of problem (156), we have L linear equations of the form eq. (163), which allows us to write the following matrix form:

$$\underbrace{[\mathbf{A}_1, \dots, \mathbf{A}_l, \dots, \mathbf{A}_L]^T}_{\mathbf{A} \in \mathbb{R}^{L \times L}} \underbrace{\mathcal{C}(\bar{\mathbf{H}})}_{\in \mathbb{R}^{L \times 3}} = \underbrace{[\mathcal{F}_1(\mathbf{H}), \dots, \mathcal{F}_l(\mathbf{H}), \dots, \mathcal{F}_L(\mathbf{H})]^T}_{\mathcal{F}(\mathbf{H}) \in \mathbb{R}^{L \times 3}} \quad (165)$$

and we can compute $\mathcal{C}(\bar{\mathbf{H}})$ by:

$$\mathcal{C}(\bar{\mathbf{H}}) = \mathbf{A}^{-1} \mathcal{F}(\mathbf{H}). \quad (166)$$

Surprisingly, due to the structure of \mathbf{A} , its inverse can be computed in closed form.

Theorem 35 (Closed-form Matrix Inverse). *The inverse of matrix \mathbf{A} (164), denoted \mathbf{P} , can be computed in closed form, with each column of \mathbf{P} being:*

$$\mathbf{P}_l(u) = \begin{cases} p_1 & \text{if } u = l \\ \hat{\theta}_{c_l} \hat{\theta}_i p_2 & \text{if } u = u(i, r_l), \forall 0 \leq i < r_l \\ -\hat{\theta}_{c_l} \hat{\theta}_i p_2 & \text{if } u = u(r_l, i), \forall r_l < i \leq K, i \neq c_l \\ -\hat{\theta}_{r_l} \hat{\theta}_i p_2 & \text{if } u = u(i, c_l), \forall 0 \leq i < c_l, i \neq r_l \\ \hat{\theta}_{r_l} \hat{\theta}_i p_2 & \text{if } u = u(c_l, i), \forall c_l < i \leq K \\ 0 & \text{otherwise} \end{cases}, \quad (167)$$

where p_1 and p_2 are the following constants:

$$p_1 = \frac{K+1}{2K+6}, \quad p_2 = \frac{1}{2K+6}. \quad (168)$$

Proof: To show \mathbf{P} is the inverse of \mathbf{A} , we need to show $\mathbf{AP} = \mathbf{I}_L$. To do so, we first show that $\mathbf{A}_l^\top \mathbf{P}_l = 1$, which means the diagonal entries of \mathbf{AP} are all equal to 1. Note that from the definition of \mathbf{A}_l in (164) and \mathbf{P}_l in (167), one can see that there are $2(K-1)+1$ non-zero entries in \mathbf{A}_l and \mathbf{P}_l and $\mathbf{P}_l(u) = -p_2 \mathbf{A}_l(u)$ when $u \neq l$, and $\mathbf{P}_l(u) = p_1, \mathbf{A}_l(u) = 4$ when $u = l$, therefore, we have:

$$\mathbf{A}_l^\top \mathbf{P}_l = 4p_1 - 2(K-1)p_2 = \frac{4K+4}{2K+6} - \frac{2K-2}{2K+6} = \frac{2K+6}{2K+6} = 1. \quad (169)$$

We then show that $\mathbf{A}_l^\top \mathbf{P}_m = 0$ for any $l \neq m$. The non-zero indices of $\mathbf{A}_l, \mathbf{P}_m$ and their corresponding entries are:

u	$\mathbf{A}_l(u)$	u	$\mathbf{P}_m(u)$
$u(r_l, c_l)$	4	$u(r_m, c_m)$	p_1
$u(i, r_l), \forall 0 \leq i < r_l$	$-\hat{\theta}_{c_l} \hat{\theta}_i$	$u(i, r_m), \forall 0 \leq i < r_m$	$\hat{\theta}_{c_m} \hat{\theta}_i p_2$
$u(r_l, i), \forall r_l < i \leq K, i \neq c_l$	$\hat{\theta}_{c_l} \hat{\theta}_i$	$u(r_m, i), \forall r_m < i \leq K, i \neq c_m$	$-\hat{\theta}_{c_m} \hat{\theta}_i p_2$
$u(i, c_l), \forall 0 \leq i < c_l, i \neq r_l$	$\hat{\theta}_{r_l} \hat{\theta}_i$	$u(i, c_m), \forall 0 \leq i < c_m, i \neq r_m$	$-\hat{\theta}_{r_m} \hat{\theta}_i p_2$
$u(c_l, i), \forall c_l < i \leq K,$	$-\hat{\theta}_{r_l} \hat{\theta}_i$	$u(c_m, i), \forall c_m < i \leq K,$	$\hat{\theta}_{r_m} \hat{\theta}_i p_2$

(170)

where we have used $\mathcal{Z}(l) = (r_l, c_l)$ and $\mathcal{Z}(m) = (r_m, c_m)$. Now depending on the 12 different relations between (r_l, c_l) and (r_m, c_m) , \mathbf{A}_l and \mathbf{P}_m can have different ways of sharing common indices whose corresponding entries in both \mathbf{A}_l and \mathbf{P}_m are non-zero. We next show that in all these 12 different cases, $\mathbf{A}_l^\top \mathbf{P}_m = 0$ holds true.

a) $r_l < c_l < r_m < c_m$: \mathbf{A}_l and \mathbf{P}_m share 4 non-zero entries with the same indices:

u	$\mathbf{A}_l(u)$	$\mathbf{P}_m(u)$
$u(r_l, r_m)$	$\hat{\theta}_{c_l} \hat{\theta}_{r_m}$	$\hat{\theta}_{c_m} \hat{\theta}_{r_l} p_2$
$u(r_l, c_m)$	$\hat{\theta}_{c_l} \hat{\theta}_{c_m}$	$-\hat{\theta}_{r_m} \hat{\theta}_{r_l} p_2$
$u(c_l, r_m)$	$-\hat{\theta}_{r_l} \hat{\theta}_{r_m}$	$\hat{\theta}_{c_m} \hat{\theta}_{c_l} p_2$
$u(c_l, c_m)$	$-\hat{\theta}_{r_l} \hat{\theta}_{c_m}$	$-\hat{\theta}_{r_m} \hat{\theta}_{c_l} p_2$

(171)

Using the above table, we easily see that $\mathbf{A}_l^\top \mathbf{P}_m = 0$.

b) $r_l < c_l = r_m < c_m$: \mathbf{A}_l and \mathbf{P}_m share the following common indices whose corresponding entries are non-zero:

u	$\mathbf{A}_l(u)$	$\mathbf{P}_m(u)$
$u(r_l, c_l)$	4	$\hat{\theta}_{c_m} \hat{\theta}_{r_l} p_2$
$u(r_l, c_m)$	$\hat{\theta}_{c_l} \hat{\theta}_{c_m}$	$-\hat{\theta}_{c_l} \hat{\theta}_{r_l} p_2$
$u(i, c_l), \forall 0 \leq i < c_l, i \neq r_l$	$\hat{\theta}_{r_l} \hat{\theta}_i$	$\hat{\theta}_{c_m} \hat{\theta}_i p_2$
$u(c_l, i), \forall c_l < i \leq K, i \neq c_m$	$-\hat{\theta}_{r_l} \hat{\theta}_i$	$-\hat{\theta}_{c_m} \hat{\theta}_i p_2$
$u(c_l, c_m)$	$-\hat{\theta}_{r_l} \hat{\theta}_{c_m}$	p_1

(172)

Using the table above, we can calculate $\mathbf{A}_l^\top \mathbf{P}_m$:

$$\begin{aligned} \mathbf{A}_l^\top \mathbf{P}_m &= 4\hat{\theta}_{c_m} \hat{\theta}_{r_l} p_2 - \hat{\theta}_{r_l} \hat{\theta}_{c_m} p_2 + (c_l - 1)\hat{\theta}_{r_l} \hat{\theta}_{c_m} p_2 + (K - c_l - 1)\hat{\theta}_{r_l} \hat{\theta}_{c_m} p_2 - \hat{\theta}_{r_l} \hat{\theta}_{c_m} p_1 \\ &= ((K+1)p_2 - p_1)\hat{\theta}_{r_l} \hat{\theta}_{c_m} = 0, \end{aligned} \quad (173)$$

which ends up being 0 due to $p_1 = (K+1)p_2$ (cf. eq. (168)).

c) $r_l < r_m < c_l < c_m$: \mathbf{A}_l and \mathbf{P}_m share 4 common indices with non-zero entries:

u	$\mathbf{A}_l(u)$	$\mathbf{P}_m(u)$
$u(r_l, r_m)$	$\hat{\theta}_{c_l} \hat{\theta}_{r_m}$	$\hat{\theta}_{c_m} \hat{\theta}_{r_l} p_2$
$u(r_l, c_m)$	$\hat{\theta}_{c_l} \hat{\theta}_{c_m}$	$-\hat{\theta}_{r_m} \hat{\theta}_{r_l} p_2$
$u(r_m, c_l)$	$\hat{\theta}_{r_l} \hat{\theta}_{r_m}$	$-\hat{\theta}_{c_m} \hat{\theta}_{c_l} p_2$
$u(c_l, c_m)$	$-\hat{\theta}_{r_l} \hat{\theta}_{c_m}$	$-\hat{\theta}_{r_m} \hat{\theta}_{c_l} p_2$

(174)

from which we can compute that $\mathbf{A}_l^\top \mathbf{P}_m = 0$.

d) $r_l = r_m < c_l < c_m$: \mathbf{A}_l and \mathbf{P}_m share the following common indices with non-zero entries:

u	$\mathbf{A}_l(u)$	$\mathbf{P}_m(u)$
$u(r_l, c_l)$	4	$-\hat{\theta}_{c_m} \hat{\theta}_{c_l} p_2$
$u(i, r_l), \forall 0 \leq i < r_l$	$-\hat{\theta}_{c_l} \hat{\theta}_i$	$\hat{\theta}_{c_m} \hat{\theta}_i p_2$
$u(r_l, i), \forall r_l < i \leq K, i \neq c_l, c_m$	$\hat{\theta}_{c_l} \hat{\theta}_i$	$-\hat{\theta}_{c_m} \hat{\theta}_i p_2$
$u(r_l, c_m)$	$\hat{\theta}_{c_l} \hat{\theta}_{c_m}$	p_1
$u(c_l, c_m)$	$-\hat{\theta}_{r_l} \hat{\theta}_{c_m}$	$-\hat{\theta}_{r_k} \hat{\theta}_{c_l} p_2$

(175)

from which we can compute $\mathbf{A}_l^\top \mathbf{P}_m$:

$$\mathbf{A}_l^\top \mathbf{P}_m = (-4p_2 - r_l p_2 - (K - r_l - 2)p_2 + p_1 + p_2) \hat{\theta}_{c_l} \hat{\theta}_{c_m} = (p_1 - (K + 1)p_2) \hat{\theta}_{c_l} \hat{\theta}_{c_m} = 0 \quad (176)$$

e) $r_l < r_m < c_l = c_m$: \mathbf{A}_l and \mathbf{P}_m share the following common indices with non-zero entries:

u	$\mathbf{A}_l(u)$	$\mathbf{P}_m(u)$
$u(r_l, c_l)$	4	$-\hat{\theta}_{r_m} \hat{\theta}_{r_l} p_2$
$u(r_l, r_m)$	$\hat{\theta}_{c_l} \hat{\theta}_{r_m}$	$\hat{\theta}_{c_l} \hat{\theta}_{r_l} p_2$
$u(i, c_l), \forall 0 \leq i < c_l, i \neq r_l, r_m$	$\hat{\theta}_{r_l} \hat{\theta}_i$	$-\hat{\theta}_{r_m} \hat{\theta}_i p_2$
$u(r_m, c_l)$	$\hat{\theta}_{r_l} \hat{\theta}_{r_m}$	p_1
$u(c_l, i), \forall c_l < i \leq K$	$-\hat{\theta}_{r_l} \hat{\theta}_i$	$\hat{\theta}_{r_m} \hat{\theta}_i p_2$

(177)

from which we compute $\mathbf{A}_l^\top \mathbf{P}_m$:

$$\mathbf{A}_l^\top \mathbf{P}_m = (-4p_2 + p_2 - (c_l - 2)p_2 + p_1 - (K - c_l)p_2) \hat{\theta}_{r_l} \hat{\theta}_{r_m} = (p_1 - (K + 1)p_2) \hat{\theta}_{r_l} \hat{\theta}_{r_m} = 0. \quad (178)$$

f) $r_m < r_l < c_l < c_m$: \mathbf{A}_l and \mathbf{P}_m share 4 common indices with non-zero entries:

u	$\mathbf{A}_l(u)$	$\mathbf{P}_m(u)$
$u(r_m, r_l)$	$-\hat{\theta}_{c_l} \hat{\theta}_{r_m}$	$-\hat{\theta}_{c_m} \hat{\theta}_{c_m} \hat{\theta}_{r_l} p_2$
$u(r_l, c_m)$	$\hat{\theta}_{c_l} \hat{\theta}_{c_m}$	$-\hat{\theta}_{r_m} \hat{\theta}_{r_l} p_2$
$u(r_m, c_l)$	$\hat{\theta}_{r_l} \hat{\theta}_{r_m}$	$-\hat{\theta}_{c_m} \hat{\theta}_{c_l} p_2$
$U(c_l, c_m)$	$-\hat{\theta}_{r_l} \hat{\theta}_{c_m}$	$-\hat{\theta}_{r_m} \hat{\theta}_{c_l} p_2$

(179)

from which we easily see that $\mathbf{A}_l^\top \mathbf{P}_m = 0$.

g) $r_m < r_l < c_l = c_m$: same as $r_l < r_m < c_m = c_l$ by symmetry (switch m and l).

h) $r_m < r_l < c_m < c_l$: same as $r_l < r_m < c_l < c_m$ by symmetry.

i) $r_m < r_l = c_m < c_l$: same as $r_l < r_m = c_l < c_m$ by symmetry.

j) $r_m < c_m < r_l < c_l$: same as $r_l < c_l < r_m < c_m$ by symmetry.

k) $r_l < r_m < c_m < c_l$: same as $r_m < r_l < c_l < c_m$ by symmetry.

l) $r_m = r_l < c_m < c_l$: same as $r_l = r_m < c_l < c_m$ by symmetry.

As we show above, in all cases, $\mathbf{A}_l \mathbf{P}_m = 0$, when $l \neq m$. Therefore, we prove that $\mathbf{A} \mathbf{P} = \mathbf{I}_L$ and \mathbf{P} is the closed-form inverse of \mathbf{A} . ■

Therefore, using Theorem 35, we can first generate \mathbf{P} directly (using eq. (167)) and then apply eq. (166) to obtain the off-diagonal block vector parts of $\bar{\mathbf{H}}$.

6) **Diagonal block vector part.** After obtaining off-diagonal block vector parts, we can insert them back into eq. (155) to obtain the diagonal block vector parts:

$$[\bar{\mathbf{H}}]_{kk}^v = - \sum_{i=0, i \neq k}^K \hat{\theta}_k \hat{\theta}_i [\bar{\mathbf{H}}]_{ki}^v + \phi_k, \quad (180)$$

$$\phi_k = \begin{cases} - \left(\frac{1}{2} \hat{\theta}_k + 1 \right) [\hat{\boldsymbol{\xi}}_k]_{\times} \bar{\mathbf{a}}_k & k = 1, \dots, K \\ \sum_{k=1}^K \left(\frac{1}{2} \hat{\theta}_k + 1 \right) [\hat{\boldsymbol{\xi}}_k]_{\times} \bar{\mathbf{a}}_k & k = 0 \end{cases}. \quad (181)$$

■

U. Initial Guess for Algorithm 3

We describe the initial guess $\bar{\mathbf{M}}_0$ as:

$$\bar{\mathbf{M}}_0 = \bar{\mathbf{Q}} - \hat{\mu} \mathbf{J} + \boldsymbol{\Delta}_0, \quad (182)$$

where $\Delta_0 \in \bar{\mathcal{H}}$ is the following:

$$\text{Off-diagonal blocks:} \quad [\Delta]_{\mathcal{U}(l)} = \mathbf{0}, \forall l = 1, \dots, L \quad (183)$$

$$\text{Diagonal block scalar part:} \quad [\Delta]_{kk}^s = \begin{cases} -(\frac{1}{4}\hat{\theta}_k + \frac{1}{2})\|\hat{\xi}_k\|^2 + (\frac{1}{4}\hat{\theta}_k - \frac{1}{2})\bar{c}^2 & k = 1, \dots, K, \\ \sum_{k=1}^K (\frac{1}{4}\hat{\theta}_k + \frac{1}{2})\|\hat{\xi}_k\|^2 - (\frac{1}{4}\hat{\theta}_k - \frac{1}{2})\bar{c}^2 & k = 0. \end{cases} \quad (184)$$

$$\text{Diagonal block vector part:} \quad [\Delta]_{kk}^v = \begin{cases} -(\frac{1}{2}\hat{\theta}_k + 1) [\hat{\xi}_k]_{\times} \bar{\mathbf{a}}_k & k = 1, \dots, K, \\ \sum_{k=1}^K (\frac{1}{2}\hat{\theta}_k + 1) [\hat{\xi}_k]_{\times} \bar{\mathbf{a}}_k & k = 0 \end{cases} \quad (185)$$

$$\text{Diagonal block matrix part:} \quad [\Delta]_{kk}^m = \begin{cases} -[\bar{\mathbf{Q}}]_{0k} - (\frac{1}{4}\hat{\theta}_k + \frac{1}{4}) \|\hat{\xi}_k\|^2 \mathbf{I}_3 - \frac{1}{2}\bar{c}^2 \mathbf{I}_3 & k = 1, \dots, K \\ \sum_{k=1}^K ([\bar{\mathbf{Q}}]_{0k} + (\frac{1}{4}\hat{\theta}_k + \frac{1}{4}) \|\hat{\xi}_k\|^2 \mathbf{I}_3 + \frac{1}{2}\bar{c}^2 \mathbf{I}_3) & k = 0 \end{cases}. \quad (186)$$

One can verify that Δ_0 satisfies all the constraints defined by set $\bar{\mathcal{H}}$ (145).

V. Tighter Bounds for Theorem 17

In this section, we improve over the bounds presented in Theorem 17 and derive tighter but more complex bounds for scale, rotation, and translation estimation.

Theorem 36 (Estimation Contract with Noisy Inliers and Adversarial Outliers). Assume (i) the set of correspondences contains at least 3 distinct and non-collinear inliers, where for any inlier i , $\|\mathbf{b}_i - s^\circ \mathbf{R}^\circ \mathbf{a}_i - \mathbf{t}^\circ\| \leq \beta_i$, and $(s^\circ, \mathbf{R}^\circ, \mathbf{t}^\circ)$ denotes the ground truth transformation. Assume (ii) the inliers belong to the maximum consensus set in each subproblem¹⁴, and (iii) the second largest consensus set is “sufficiently smaller” than the maximum consensus set (as formalized in Lemma 23). If (iv) the rotation subproblem produces a valid certificate (as per Theorems 13-14), then the output $(\hat{s}, \hat{\mathbf{R}}, \hat{\mathbf{t}})$ of TEASER satisfies:

$$|\hat{s} - s^\circ| \leq \min\{\zeta_i^s, i \in \mathcal{I}\}, \quad (187)$$

$$s^\circ(1 - \cos(\theta_R)) \leq \frac{\sum_{i \in \mathcal{I}} (\zeta_i^R)^2 - (\hat{s} - s^\circ)^2 \|\mathbf{A}_{\mathcal{I}}\|_F^2}{2\hat{s}(\sigma_1^2(\mathbf{A}_{\mathcal{I}}) + \sigma_2^2(\mathbf{A}_{\mathcal{I}}))}, \quad (188)$$

$$|\hat{t}_l - t_l^\circ| \leq \min\{((\hat{s} - s^\circ)^2 + 2\hat{s}s^\circ(1 - \cos(\theta_R))) \|\mathbf{a}_i\|^2 + [\zeta_i^t]_l : i \in \mathcal{I}\}, \forall l = 1, 2, 3. \quad (189)$$

where \mathcal{I} is the set of ground-truth inliers with size N_I , i.e., ground-truth TRIMs in the scale bound, ground-truth TIMs in the rotation bound, and ground-truth correspondences in the translation bound; $\hat{t}_l, t_l^\circ, l = 1, 2, 3$, denotes the l -th entry of the translation vector; ζ_i^s, ζ_i^R and $[\zeta_i^t]_l$ are defined as follows:

$$\begin{cases} \zeta_i^s = |s_i - \hat{s}| + \alpha_i \\ \zeta_i^R = \frac{\|\mathbf{b}_i - \hat{s}\hat{\mathbf{R}}\bar{\mathbf{a}}_i\|}{\|\bar{\mathbf{a}}_i\|} + \alpha_i \\ [\zeta_i^t]_l = |[\mathbf{b}_i]_l - [\hat{s}\hat{\mathbf{R}}\bar{\mathbf{a}}_i]_l - \hat{t}_l| + \beta_i \end{cases}, \forall i \in \mathcal{I}, \forall l = 1, 2, 3; \quad (190)$$

θ_R is the angular (geodesic) distance between $\hat{\mathbf{R}}$ and \mathbf{R}° ; $\mathbf{A}_{\mathcal{I}} = \begin{bmatrix} \frac{\bar{\mathbf{a}}_{i_1}}{\|\bar{\mathbf{a}}_{i_1}\|}, \dots, \frac{\bar{\mathbf{a}}_{i_{N_I}}}{\|\bar{\mathbf{a}}_{i_{N_I}}\|} \end{bmatrix} \in \mathbb{R}^{3 \times N_I}$ is the matrix that assembles all normalized TIMs in the set \mathcal{I} , and $\sigma_1(\mathbf{A}_{\mathcal{I}}), \sigma_2(\mathbf{A}_{\mathcal{I}})$ denotes the smallest and second smallest singular values of matrix $\mathbf{A}_{\mathcal{I}}$. Note that since there are at least 3 non-collinear ground-truth inliers, $\text{rank}(\mathbf{A}_{\mathcal{I}}) \geq 2$ and $\sigma_1(\mathbf{A}_{\mathcal{I}}) + \sigma_2(\mathbf{A}_{\mathcal{I}}) > 0$, meaning the rotation bound is well defined. Since the bounds rely on the knowledge of the inliers \mathcal{I} , to compute the worst-case bounds, one has to enumerate over all possible subsets of $N_I = 3$ inliers to compute the largest bounds for scale, rotation and translation.

Proof: We will prove the bounds for scale, rotation, and translation separately.

Scale Error Bound. Let us denote with $\bar{\mathcal{C}}$ the consensus set associated with the optimal solution of the TLS scale estimation problem, with $|\bar{\mathcal{C}}| = \bar{N}_{\text{in}}$. To establish a connection with the ground-truth scale s° , we partition $\bar{\mathcal{C}}$ into two sets: \mathcal{I} that contains all the indices of the *true inliers*, and \mathcal{J} that contains all the indices of the *false inliers*, which are not generated from the true inlier model (TRIM) where classified as inliers by TLS and ended up in $\bar{\mathcal{C}}$. Let $N_I = |\mathcal{I}|$ and $N_J = |\mathcal{J}|$, and we have $N_I + N_J = \bar{N}_{\text{in}}$. For scale measurements that are in \mathcal{I} and \mathcal{J} , we have the following relations:

$$\begin{cases} s_i = s^\circ + \epsilon_i = \hat{s} + \phi_i & \forall i \in \mathcal{I} \\ s_j = \hat{s} + \phi_j & \forall j \in \mathcal{J} \end{cases}, \quad (191)$$

where ϕ_i 's and ϕ_j 's are *residuals* that can be computed from s_i, s_j , and \hat{s} , and ϵ_i 's are *unknown noise* terms. Note that we have $|\epsilon_i| \leq \alpha_i, |\phi_i| \leq \alpha_i, |\phi_j| \leq \alpha_j$. From the first equation in eq. (191), we have:

$$\hat{s} - s^\circ = \epsilon_i - \phi_i \Rightarrow |\hat{s} - s^\circ| \leq |\phi_i| + \alpha_i \doteq \zeta_i^s, \forall i \in \mathcal{I}, \quad (192)$$

$$\Rightarrow |\hat{s} - s^\circ| \leq \min\{\zeta_i^s, i \in \mathcal{I}\} \quad (193)$$

¹⁴In other words: the inliers belong to the largest set \mathcal{S} such that for any $i, j \in \mathcal{S}$, $|s_{ij} - s| \leq \alpha_{ij}$, $\|\bar{\mathbf{b}}_{ij} - s\mathbf{R}\bar{\mathbf{a}}_{ij}\| \leq \delta_{ij}$, and $|\mathbf{b}_i - s\mathbf{R}\mathbf{a}_i - \mathbf{t}|_l \leq \beta_i, \forall l = 1, 2, 3$ for any transformation $(s, \mathbf{R}, \mathbf{t})$. Note that for translation the assumption is applied component-wise.

where we have defined a new quantity $\zeta_i^s = |\phi_i| + \alpha_i$, that can be computed for every TRIM in the consensus set $\bar{\mathcal{C}}$. Eq. (192) states that the scale error is bounded by the minimum ζ_i in the set of true inliers. Therefore, if we know what the set of true inliers \mathcal{I} is, then we can simply take the minimum of the set $\{\zeta_i, i \in \mathcal{I}\}$. However, if we do not know the true inlier set, assuming $N_I \geq 3$, we get the worst-case bound:

$$|\hat{s} - s^\circ| \leq \zeta_{m_3}^s, \quad (194)$$

where ζ_{m_3} is the third-largest ζ_k^s in the set $\{\zeta_k^s = |s_k - \hat{s}| + \alpha_k : k \in \bar{\mathcal{C}}\}$.

Rotation Error Bound. Similar to the previous case, we use $\bar{\mathcal{C}}$ to denote the consensus set of TIM at the optimal TLS estimate. We also use \mathcal{I} to denote the set of *true inliers* and use \mathcal{J} to denote the set *false inliers*, i.e., outliers that TLS selected as inliers. We have the following equations for the true inliers and false inliers:

$$\begin{cases} \bar{b}_i = s^\circ \mathbf{R}^\circ \bar{a}_i + \epsilon_i = \hat{s} \hat{\mathbf{R}} \bar{a}_i + \phi_i & \forall i \in \mathcal{I} \\ \bar{b}_j = \hat{s} \hat{\mathbf{R}} \bar{a}_j + \phi_j & \forall j \in \mathcal{J} \end{cases}, \quad (195)$$

where ϕ_i, ϕ_j are the residuals that can be computed and ϵ_i is the unknown noise term. From the first equation of eq. (195), we have:

$$(\hat{s} \hat{\mathbf{R}} - s^\circ \mathbf{R}^\circ) \bar{a}_i = \epsilon_i - \phi_i \Leftrightarrow (\hat{s} \hat{\mathbf{R}} - s^\circ \mathbf{R}^\circ) \frac{\bar{a}_i}{\|\bar{a}_i\|} = \frac{\epsilon_i - \phi_i}{\|\bar{a}_i\|} \quad (196)$$

$$\Rightarrow \|(\hat{s} \hat{\mathbf{R}} - s^\circ \mathbf{R}^\circ) \mathbf{u}_i\| \leq \frac{\|\phi_i\|}{\|\bar{a}_i\|} + \alpha_i \doteq \zeta_i^R, \forall i \in \mathcal{I}, \quad (197)$$

where $\mathbf{u}_i \doteq \frac{\bar{a}_i}{\|\bar{a}_i\|}$ and we have defined a new quantity $\zeta_i^R \doteq \|\phi_i\|/\|\bar{a}_i\| + \alpha_i$ that can be computed from the TLS solution. Since eq. (197) holds for every TIM in \mathcal{I} , we can aggregate all inequalities:

$$\|(\hat{s} \hat{\mathbf{R}} - s^\circ \mathbf{R}^\circ) \mathbf{A}_{\mathcal{I}}\|_F^2 \leq \sum_{i \in \mathcal{I}} (\zeta_i^R)^2, \quad (198)$$

where $\mathbf{A}_{\mathcal{I}} = [\mathbf{u}_{i_1}, \dots, \mathbf{u}_{i_{N_I}}] \in \mathbb{R}^{3 \times N_I}$, $i_1, \dots, i_{N_I} \in \mathcal{I}$, assembles all normalized TIMs \mathbf{u}_i in the set \mathcal{I} . Now we want to get a lower bound on $\|(\hat{s} \hat{\mathbf{R}} - s^\circ \mathbf{R}^\circ) \mathbf{A}_{\mathcal{I}}\|_F^2$ so that we can upper-bound the rotation error.

Towards this goal, we first analyze the singular values of $(\hat{s} \hat{\mathbf{R}} - s^\circ \mathbf{R}^\circ)$. Let the three singular values of $(\hat{s} \hat{\mathbf{R}} - s^\circ \mathbf{R}^\circ)$ be $\sigma_l, l = 1, 2, 3$, then we know that $\sigma_l^2, l = 1, 2, 3$, are equal to the eigenvalues of the matrix $\mathbf{B} = (\hat{s} \hat{\mathbf{R}} - s^\circ \mathbf{R}^\circ)^\top (\hat{s} \hat{\mathbf{R}} - s^\circ \mathbf{R}^\circ)$. Expanding \mathbf{B} , we get:

$$\mathbf{B} = (\hat{s} \hat{\mathbf{R}}^\top - s^\circ (\mathbf{R}^\circ)^\top) (\hat{s} \hat{\mathbf{R}} - s^\circ \mathbf{R}^\circ) = (\hat{s}^2 + (s^\circ)^2) \mathbf{I}_3 - \hat{s} s^\circ (\hat{\mathbf{R}}^\top \mathbf{R}^\circ + (\mathbf{R}^\circ)^\top \hat{\mathbf{R}}) = (\hat{s}^2 + (s^\circ)^2) \mathbf{I}_3 - \hat{s} s^\circ (\mathbf{R}_s + \mathbf{R}_s^\top) \quad (199)$$

where we have denoted $\mathbf{R}_s = \hat{\mathbf{R}}^\top \mathbf{R}^\circ \in \text{SO}(3)$, which is the composition of two rotation matrices. Let the axis-angle representation of \mathbf{R}_s be (Φ_s, θ_R) , then the axis-angle representation for \mathbf{R}_s is simply $(\Phi_s, -\theta_R)$. According to the Rodrigues' rotation formula, we can write \mathbf{R}_s and \mathbf{R}_s^\top as:

$$\begin{cases} \mathbf{R}_s = \cos(\theta_R) \mathbf{I}_3 + \sin(\theta_R) [\Phi_s]_\times + (1 - \cos(\theta_R)) \Phi_s \Phi_s^\top \\ \mathbf{R}_s^\top = \cos(-\theta_R) \mathbf{I}_3 + \sin(-\theta_R) [\Phi_s]_\times + (1 - \cos(-\theta_R)) \Phi_s \Phi_s^\top = \cos(\theta_R) \mathbf{I}_3 - \sin(\theta_R) [\Phi_s]_\times + (1 - \cos(\theta_R)) \Phi_s \Phi_s^\top \end{cases} \quad (200)$$

Inserting eq. (200) back into eq. (199), we have:

$$\mathbf{B} = (\hat{s}^2 + (s^\circ)^2) \mathbf{I}_3 - \hat{s} s^\circ (2 \cos(\theta_R) \mathbf{I}_3 + 2(1 - \cos(\theta_R)) \Phi_s \Phi_s^\top), \quad (201)$$

from which we can see that the three eigenvectors of \mathbf{B} are Φ_s and Φ_1^\perp and Φ_2^\perp , where we pick Φ_1^\perp and Φ_2^\perp to be 2 unit orthogonal vectors in the plane that is orthogonal to Φ_s :

$$\begin{cases} \mathbf{B} \Phi_s = ((\hat{s}^2 + (s^\circ)^2) - 2\hat{s}s^\circ) \Phi_s = (\hat{s} - s^\circ)^2 \Phi_s \\ \mathbf{B} \Phi_l^\perp = (\hat{s}^2 + (s^\circ)^2 - 2\hat{s}s^\circ \cos(\theta_R)) \Phi_l^\perp, \forall l = 1, 2 \end{cases} \quad (202)$$

Therefore, we have $\sigma_1 = \sigma_2 = \hat{s}^2 + (s^\circ)^2 - 2\hat{s}s^\circ \cos(\theta_R) = (\hat{s} - s^\circ)^2 + 2\hat{s}s^\circ(1 - \cos(\theta_R))$ and $\sigma_3 = (\hat{s} - s^\circ)^2$. Let the SVD of $(\hat{s} \hat{\mathbf{R}} - s^\circ \mathbf{R}^\circ)$ be $(\hat{s} \hat{\mathbf{R}} - s^\circ \mathbf{R}^\circ) = \mathbf{U}_\Delta \mathbf{S}_\Delta \mathbf{V}_\Delta^\top$, we lower-bound $\|(\hat{s} \hat{\mathbf{R}} - s^\circ \mathbf{R}^\circ) \mathbf{A}_{\mathcal{I}}\|_F^2$:

$$\|(\hat{s} \hat{\mathbf{R}} - s^\circ \mathbf{R}^\circ) \mathbf{A}_{\mathcal{I}}\|_F^2 = \|\mathbf{U}_\Delta \mathbf{S}_\Delta \mathbf{V}_\Delta^\top \mathbf{A}_{\mathcal{I}}\|_F^2 = \|\mathbf{S}_\Delta \mathbf{V}_\Delta^\top \mathbf{A}_{\mathcal{I}}\|_F^2 = \left\| \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \begin{bmatrix} (\mathbf{V}_\Delta^\top \mathbf{A}_{\mathcal{I}})_1 \\ (\mathbf{V}_\Delta^\top \mathbf{A}_{\mathcal{I}})_2 \\ (\mathbf{V}_\Delta^\top \mathbf{A}_{\mathcal{I}})_3 \end{bmatrix} \right\|_F^2 \quad (203)$$

$$= (\hat{s} - s^\circ)^2 (\|(\mathbf{V}_\Delta^\top \mathbf{A}_{\mathcal{I}})_1\|^2 + \|(\mathbf{V}_\Delta^\top \mathbf{A}_{\mathcal{I}})_2\|^2 + \|(\mathbf{V}_\Delta^\top \mathbf{A}_{\mathcal{I}})_3\|^2) + 2\hat{s}s^\circ(1 - \cos(\theta_R)) (\|(\mathbf{V}_\Delta^\top \mathbf{A}_{\mathcal{I}})_1\|^2 + \|(\mathbf{V}_\Delta^\top \mathbf{A}_{\mathcal{I}})_2\|^2) \quad (204)$$

$$= (\hat{s} - s^\circ)^2 \|\mathbf{A}_{\mathcal{I}}\|_F^2 + 2\hat{s}s^\circ(1 - \cos(\theta_R)) (\|\mathbf{A}_{\mathcal{I}}\|_F^2 - \|(\mathbf{V}_\Delta^\top \mathbf{A}_{\mathcal{I}})_3\|^2) \quad (205)$$

$$\geq (\hat{s} - s^\circ)^2 \|\mathbf{A}_{\mathcal{I}}\|_F^2 + 2\hat{s}s^\circ(1 - \cos(\theta_R)) (\sigma_1^2(\mathbf{A}_{\mathcal{I}}) + \sigma_2^2(\mathbf{A}_{\mathcal{I}})), \quad (206)$$

where we have used $\|\mathbf{V}_\Delta \mathbf{A}_\mathcal{I}\|_F^2 = \|\mathbf{A}_\mathcal{I}\|_F^2 = \sigma_1^2(\mathbf{A}_\mathcal{I}) + \sigma_2^2(\mathbf{A}_\mathcal{I}) + \sigma_3^2(\mathbf{A}_\mathcal{I})$, with $\sigma_1(\mathbf{A}_\mathcal{I}) \leq \sigma_2(\mathbf{A}_\mathcal{I}) \leq \sigma_3(\mathbf{A}_\mathcal{I})$ being the three singular values of $\mathbf{A}_\mathcal{I}$ in ascending order. The last inequality in eq. (206) follows from:

$$\|(\mathbf{V}_\Delta^\top \mathbf{A}_\mathcal{I})_3\|^2 = \left\| \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{V}_\Delta^\top \mathbf{A}_\mathcal{I} \right\|_F^2 \leq \sigma_3^2(\mathbf{A}_\mathcal{I}) \left\| \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{V}_\Delta^\top \right\|_F^2 = \sigma_3^2(\mathbf{A}_\mathcal{I}). \quad (207)$$

Now we combine eq. (206) and (198) to bound $1 - \cos(\theta_R)$:

$$(\hat{s} - s^\circ)^2 \|\mathbf{A}_\mathcal{I}\|_F^2 + 2\hat{s}s^\circ(1 - \cos(\theta_R)) (\sigma_1^2(\mathbf{A}_\mathcal{I}) + \sigma_2^2(\mathbf{A}_\mathcal{I})) \leq \sum_{i \in \mathcal{I}} (\zeta_i^R)^2 \quad (208)$$

$$\Leftrightarrow s^\circ(1 - \cos(\theta_R)) \leq \frac{\sum_{i \in \mathcal{I}} (\zeta_i^R)^2 - (\hat{s} - s^\circ)^2 \|\mathbf{A}_\mathcal{I}\|_F^2}{2\hat{s}(\sigma_1^2(\mathbf{A}_\mathcal{I}) + \sigma_2^2(\mathbf{A}_\mathcal{I}))}, \quad (209)$$

where we note that (i) the existence of at least 3 non-collinear inliers guarantees that $\text{rank}(\mathbf{A}_\mathcal{I}) \geq 2$, which means $\sigma_1^2(\mathbf{A}_\mathcal{I}) + \sigma_2^2(\mathbf{A}_\mathcal{I}) > 0$ and the upper bound is well-defined; (ii) since (Φ_s, θ_R) is the axis-angle representation for $\hat{\mathbf{R}}^\top \mathbf{R}^\circ$, θ_R is the angular (geodesic) distance between $\hat{\mathbf{R}}$ and \mathbf{R}° ; (iii) the rotation bound is dependent on the scale bound $|\hat{s} - s^\circ|$ (193).

Again, when the true inlier set \mathcal{I} is known, one can directly compute the rotation bound using eq. (209); when the true inlier set \mathcal{I} is unknown, one has to enumerate over all possible subsets of 3 TIMs (formed by a triplet of (i, j, k) correspondences) in $\bar{\mathcal{C}}$ to compute the worst-case bound.

Translation Error Bound. Let $\mathbf{t}^\circ = [t_1^\circ, t_2^\circ, t_3^\circ]^\top$ and $\hat{\mathbf{t}} = [\hat{t}_1, \hat{t}_2, \hat{t}_3]^\top$ be the ground-truth and estimated translation. Let $\bar{\mathcal{C}}_l, l = 1, 2, 3$, be the set of measurements $(\mathbf{a}_k, \mathbf{b}_k)$ that pass the l -th component-wise adaptive voting, i.e., $|\mathbf{b}_k - \hat{\mathbf{R}}\mathbf{a}_k - \hat{\mathbf{t}}|_l \leq \beta_k, \forall k \in \bar{\mathcal{C}}_l, l = 1, 2, 3$. Again, we let \mathcal{I}_l and \mathcal{J}_l be the set of true and false inliers within $\bar{\mathcal{C}}_l, l = 1, 2, 3$. From condition (ii), we have that $\mathcal{I}_1 = \mathcal{I}_2 = \mathcal{I}_3 = \mathcal{I}$, which says that the true inliers pass all three component-wise adaptive voting. However, $\mathcal{J}_l, l = 1, 2, 3$, could potentially be different. Then we write the following relations:

$$\begin{cases} [\mathbf{b}_i]_l = [\hat{\mathbf{R}}\mathbf{a}_i]_l + \hat{t}_l + [\phi_i]_l = [s^\circ \mathbf{R}^\circ \mathbf{a}_i]_l + t_l^\circ + [\epsilon_i]_l & \forall i \in \mathcal{I}, l = 1, 2, 3 \\ [\mathbf{b}_j]_l = [\hat{\mathbf{R}}\mathbf{a}_j]_l + \hat{t}_l + [\phi_j]_l & \forall j \in \mathcal{J}_l, l = 1, 2, 3 \end{cases} \quad (210)$$

where ϵ_i is the unknown noise that satisfies $\|\epsilon_i\| \leq \beta_i$ and $[\phi_i]_l$ and $[\phi_j]_l$ are the residuals that can be computed from the TLS estimate and satisfy $|\phi_i|_l \leq \beta_i$ and $|\phi_j|_l \leq \beta_j$. From the first equation in (210), we have that:

$$\hat{t}_l - t_l^\circ = [s^\circ \mathbf{R}^\circ \mathbf{a}_i]_l - [\hat{\mathbf{R}}\mathbf{a}_i]_l + [\epsilon_i]_l - [\phi_i]_l \quad (211)$$

$$\Rightarrow |\hat{t}_l - t_l^\circ| \leq |[s^\circ \mathbf{R}^\circ \mathbf{a}_i]_l - [\hat{\mathbf{R}}\mathbf{a}_i]_l| + |\phi_i|_l + \beta_i = |[s^\circ \mathbf{R}^\circ \mathbf{a}_i]_l - [\hat{\mathbf{R}}\mathbf{a}_i]_l| + [\zeta_i^t]_l, \quad (212)$$

where we have defined a new quantity $[\zeta_i^t]_l = |\phi_i|_l + \beta_i$ that can be computed from the solution. Now we want to upper-bound $|[s^\circ \mathbf{R}^\circ \mathbf{a}_i]_l - [\hat{\mathbf{R}}\mathbf{a}_i]_l|$, using a similar SVD-based approach as in eq. (206) (but this time we upper-bound):

$$([s^\circ \mathbf{R}^\circ \mathbf{a}_i]_l - [\hat{\mathbf{R}}\mathbf{a}_i]_l)^2 \leq \|(\hat{\mathbf{R}} - s^\circ \mathbf{R}^\circ) \mathbf{a}_i\|^2 = \|\mathbf{U}_\Delta \mathbf{S}_\Delta \mathbf{V}_\Delta^\top \mathbf{a}_i\|^2 = \|\mathbf{S}_\Delta \mathbf{V}_\Delta^\top \mathbf{a}_i\|^2 \quad (213)$$

$$= (\hat{s} - s^\circ)^2 \|\mathbf{a}_i\|^2 + 2\hat{s}s^\circ(1 - \cos(\theta_R))((\mathbf{V}_\Delta^\top \mathbf{a}_i)_1^2 + (\mathbf{V}_\Delta^\top \mathbf{a}_i)_2^2) \leq ((\hat{s} - s^\circ)^2 + 2\hat{s}s^\circ(1 - \cos(\theta_R))) \|\mathbf{a}_i\|^2 \quad (214)$$

Inserting the inequality (214) back into eq. (212), we get the following translation bound for each $i \in \mathcal{I}$

$$|\hat{t}_l - t_l^\circ| \leq ((\hat{s} - s^\circ)^2 + 2\hat{s}s^\circ(1 - \cos(\theta_R))) \|\mathbf{a}_i\|^2 + [\zeta_i^t]_l, \forall i \in \mathcal{I}, \forall l = 1, 2, 3, \quad (215)$$

there the final translation bound is:

$$|\hat{t}_l - t_l^\circ| \leq \min \{ ((\hat{s} - s^\circ)^2 + 2\hat{s}s^\circ(1 - \cos(\theta_R))) \|\mathbf{a}_i\|^2 + [\zeta_i^t]_l : i \in \mathcal{I} \}, \forall l = 1, 2, 3. \quad (216)$$

We note that the translation bound depends on both the scale bound (193) and the rotation bound (209). When the true inlier set \mathcal{I} is known, one can directly compute the bound using eq. (215); when the true inlier set \mathcal{I} is unknown, one has to enumerate over all possible subsets of 3 inliers in $\bar{\mathcal{C}}$ to compute the worst-case translation bound. ■