ARCS: Accurate Rotation and Correspondence Search Supplementary Materials

Liangzu Peng
Johns Hopkins University
lpeng25@jhu.edu

Manolis C. Tsakiris
ShanghaiTech University
mtsakiris@shanghaitech.edu.cn

René Vidal
Johns Hopkins University
rvidal@jhu.edu

A. Proof of Proposition 4

We consider a stronger version of Proposition 4:

Proposition 1. *We have $\boldsymbol{w}^\top \boldsymbol{D}_i \boldsymbol{w} = \left\|\boldsymbol{y}_i - \boldsymbol{R}\boldsymbol{x}_i\right\|_2^2$, where $\boldsymbol{w}$ is a quaternion representation of $\boldsymbol{R}$ of (8), and $\boldsymbol{D}_i \in \mathbb{R}^{4\times 4}$ is a positive semi-definite matrix whose entries depend on $\boldsymbol{x}_i$, $\boldsymbol{y}_i$. So Problem (8) is equivalent to*

$$\min_{\boldsymbol{w}\in\mathbb{S}^3} h(\boldsymbol{w}), \quad h(\boldsymbol{w}) = \sum_{i=1}^{\ell} \sqrt{\boldsymbol{w}^\top \boldsymbol{D}_i \boldsymbol{w}}. \tag{1}$$

Moreover, $\boldsymbol{D}_i$ has eigenvalues $4, 4, 0, 0$ if $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ are normalized (that is $\left\|\boldsymbol{x}_i\right\|_2 = \left\|\boldsymbol{y}_i\right\|_2 = 1$).

We first recall some basics about unit quaternions, an algebraic construction invented by Hamilton in the 1840's, when the notion of *vector* does not exist; see the beautiful account of [1]. In our current notation, each element $\boldsymbol{w}$ of $\mathbb{S}^3$ is called a unit quaternion. The most crucial fact is that $\mathrm{SO}(3)$ is isomorphic to the 3-sphere $\mathbb{S}^3$ *up to sign*, that is $\mathrm{SO}(3) \equiv \mathbb{S}^3/\{\pm 1\}$. This implies a two-to-one correspondence between unit quaternions and 3D rotations. Algebraically, any $\boldsymbol{R} \in \mathrm{SO}(3)$ can be written as a 3×3 matrix

$$\begin{matrix} w_1^2+w_2^2-w_3^2-w_4^2 & 2(w_2w_3-w_1w_4) & 2(w_2w_4+w_1w_3) \\ 2(w_2w_3+w_1w_4) & w_1^2+w_3^2-w_2^2-w_4^2 & 2(w_3w_4-w_1w_2) \\ 2(w_2w_4-w_1w_3) & 2(w_3w_4+w_1w_2) & w_1^2+w_4^2-w_2^2-w_3^2 \end{matrix}$$

where $\boldsymbol{w} = [w_1, w_2, w_3, w_4]^\top \in \mathbb{S}^3$. We can now write the three entries of $\boldsymbol{R}\boldsymbol{x}_i$ as quadratic forms $\boldsymbol{w}^\top \boldsymbol{X}_{i,1}\boldsymbol{w}$, $\boldsymbol{w}^\top \boldsymbol{X}_{i,2}\boldsymbol{w}$, and $\boldsymbol{w}^\top \boldsymbol{X}_{i,3}\boldsymbol{w}$, respectively. Here $\boldsymbol{X}_{i,1}$, $\boldsymbol{X}_{i,2}$, and $\boldsymbol{X}_{i,3}$ are 4×4 symmetric matrices, defined as

$$\boldsymbol{X}_{i,1} = \begin{bmatrix} [\boldsymbol{x}_i]_1 & 0 & [\boldsymbol{x}_i]_3 & -[\boldsymbol{x}_i]_2 \\ 0 & [\boldsymbol{x}_i]_1 & [\boldsymbol{x}_i]_2 & [\boldsymbol{x}_i]_3 \\ [\boldsymbol{x}_i]_3 & [\boldsymbol{x}_i]_2 & -[\boldsymbol{x}_i]_1 & 0 \\ -[\boldsymbol{x}_i]_2 & [\boldsymbol{x}_i]_3 & 0 & -[\boldsymbol{x}_i]_1 \end{bmatrix} \tag{2}$$

$$\boldsymbol{X}_{i,2} = \begin{bmatrix} [\boldsymbol{x}_i]_2 & -[\boldsymbol{x}_i]_3 & 0 & [\boldsymbol{x}_i]_1 \\ -[\boldsymbol{x}_i]_3 & -[\boldsymbol{x}_i]_2 & [\boldsymbol{x}_i]_1 & 0 \\ 0 & [\boldsymbol{x}_i]_1 & [\boldsymbol{x}_i]_2 & [\boldsymbol{x}_i]_3 \\ [\boldsymbol{x}_i]_1 & 0 & [\boldsymbol{x}_i]_3 & -[\boldsymbol{x}_i]_2 \end{bmatrix} \tag{3}$$

$$\boldsymbol{X}_{i,3} = \begin{bmatrix} [\boldsymbol{x}_i]_3 & [\boldsymbol{x}_i]_2 & -[\boldsymbol{x}_i]_1 & 0 \\ [\boldsymbol{x}_i]_2 & -[\boldsymbol{x}_i]_3 & 0 & [\boldsymbol{x}_i]_1 \\ -[\boldsymbol{x}_i]_1 & 0 & -[\boldsymbol{x}_i]_3 & [\boldsymbol{x}_i]_2 \\ 0 & [\boldsymbol{x}_i]_1 & [\boldsymbol{x}_i]_2 & [\boldsymbol{x}_i]_3 \end{bmatrix} \tag{4}$$

Defining $\boldsymbol{C}_i := [\boldsymbol{y}_i]_1\boldsymbol{X}_{i,1} + [\boldsymbol{y}_i]_2\boldsymbol{X}_{i,2} + [\boldsymbol{y}_i]_3\boldsymbol{X}_{i,3}$, we get that $\boldsymbol{y}_i^\top \boldsymbol{R}\boldsymbol{x}_i = \boldsymbol{w}^\top \boldsymbol{C}_i \boldsymbol{w}$. And defining

$$\boldsymbol{D}_i = (\left\|\boldsymbol{y}_i\right\|_2^2 + \left\|\boldsymbol{x}_i\right\|_2^2)\boldsymbol{I}_4 - 2\boldsymbol{C}_i \tag{5}$$

with $\boldsymbol{I}_4$ the 4×4 identity matrix, we obtain the equality

$$\begin{aligned} \left\|\boldsymbol{y}_i - \boldsymbol{R}\boldsymbol{x}_i\right\|_2^2 &= \left\|\boldsymbol{y}_i\right\|_2^2 + \left\|\boldsymbol{x}_i\right\|_2^2 - 2\boldsymbol{y}_i^\top \boldsymbol{R}\boldsymbol{x}_i & (6) \\ &= \boldsymbol{w}^\top \boldsymbol{D}_i \boldsymbol{w}. & (7) \end{aligned}$$

Since $\boldsymbol{D}_i$ is symmetric and $\boldsymbol{w}^\top \boldsymbol{D}_i \boldsymbol{w} \geq 0$ for any $\boldsymbol{w} \in \mathbb{S}^3$, we know that $\boldsymbol{D}_i \in \mathbb{R}^{4\times 4}$ is positive semi-definite.

Suppose $\left\|\boldsymbol{y}_i\right\|_2 = \left\|\boldsymbol{x}_i\right\|_2 = 1$. Then there is at least two different 3D rotations $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ satisfying $\boldsymbol{y}_i = \boldsymbol{R}_1\boldsymbol{x}_i = \boldsymbol{R}_2\boldsymbol{x}_i$. Thus, with the factorization $\boldsymbol{D}_i = \boldsymbol{Z}_i\boldsymbol{Z}_i^\top$, there are at least two quaternions $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ with $\boldsymbol{w}_1 \neq \pm\boldsymbol{w}_2$ satisfying that $\boldsymbol{Z}_i^\top \boldsymbol{w}_1 = \boldsymbol{Z}_i^\top \boldsymbol{w}_2 = 0$. So $\mathrm{rank}(\boldsymbol{D}_i) = \mathrm{rank}(\boldsymbol{Z}_i) \leq 2$. Recalling $\boldsymbol{D}_i = 2\boldsymbol{I}_4 - 2\boldsymbol{C}_i$, we see that 1 is an eigenvalue of $\boldsymbol{C}_i$ that has multiplicity at least 2. Similarly, we can derive that $\left\|\boldsymbol{y}_i + \boldsymbol{R}\boldsymbol{x}_i\right\|_2^2 = \boldsymbol{w}^\top \boldsymbol{D}_i' \boldsymbol{w}$ where $\boldsymbol{D}_i' = (\left\|\boldsymbol{y}_i\right\|_2^2 + \left\|\boldsymbol{x}_i\right\|_2^2)\boldsymbol{I}_4 + 2\boldsymbol{C}_i = 2\boldsymbol{I}_4 + 2\boldsymbol{C}_i$ is positive semi-definite of rank at most 2. That is, -1 is an eigenvalue of $\boldsymbol{C}_i$ of multiplicity at least 2. Concluding, $\boldsymbol{C}_i$ has eigenvalues $1, 1, -1, -1$ and $\boldsymbol{D}_i$ has eigenvalues $4, 4, 0, 0$.

B. Proposition 5: Proof and Interpretation

Here we provide a proof (Appendix B.1) and probabilistic interpretation (Appendix B.2) for Proposition 5. In this section, we use the notation $\boldsymbol{D}_i = \boldsymbol{Z}_i \boldsymbol{Z}_i^\top$ from Appendix A where we decomposed every positive semidefinite matrix $\boldsymbol{D}_i$ into the product of its root $\boldsymbol{Z}_i$. Since we could always normalize the point sets $\boldsymbol{y}_i$ and $\boldsymbol{x}_i$, and then normalize $\boldsymbol{D}_i$, we assume without loss of generality that $\boldsymbol{D}_i$ has eigenvalues $1, 1, 0, 0$ (cf. Proposition 1). In this situation, we can now specify that $\boldsymbol{Z}_i$ is a matrix of size 4×2 and it has orthonormal columns, i.e., $\boldsymbol{Z}_i^\top \boldsymbol{Z}_i = \boldsymbol{I}_2$. Also, we see that the objective function (9) of interest can be rewritten as

$$\min_{\boldsymbol{w} \in \mathbb{S}^3} h(\boldsymbol{w}), \quad h(\boldsymbol{w}) = \sum_{i=1}^{\ell} \left\| \boldsymbol{Z}_i^\top \boldsymbol{w} \right\|_2. \tag{8}$$

Note that, if $\boldsymbol{Z}_i$ had a single column, then (8) is exactly the problem of *dual principal component pursuit* (DPCP) [25]. On the other hand, one could think of (8) as a *group* version of DPCP, as $\left\| \boldsymbol{Z}_i^\top \boldsymbol{w} \right\|_2$ here promotes group sparsity. A similar group version of DPCP was considered by [11] in the context of homography estimation. In [11], the authors provided conditions under which any global minimizer of (8) coincides with the ground-truth normal vector, or, in our context, the ground-truth unit quaternion $\pm \boldsymbol{w}^*$. Thus, our contribution here, if viewed from the angle of group-DPCP, is to show that, there is actually an efficient algorithm that exactly reaches the guaranteed ground-truth normal. We present our contribution next.

B.1. Proof of Proposition 5

The proof follows from Proposition 4 of [17] with some simplification for specializing arbitrary Stiefel manifolds to $\mathbb{S}^3$, and with some modification to tighten a constant factor (from 2 to $\sqrt{2}$). We also note that $\eta_{\min}$ and $\eta_{\max}$ are motivated from their corresponding definitions.

Write $\boldsymbol{w} := c_0 \boldsymbol{w}_0 + c^* \boldsymbol{w}^*$ with $c_0^2 + (c^*)^2 = 1$ and $\boldsymbol{w}_0 \in \mathcal{S}^*$. Without loss of generality assume $c^* \geq 0$. Then

$$\begin{aligned} \operatorname{dist}(\boldsymbol{w}, \pm \boldsymbol{w}^*) &= \min \left\{ \sqrt{2 + 2c^*}, \sqrt{2 - 2c^*} \right\} \\ &= \sqrt{2 - 2c^*} \\ &\leq \sqrt{2 - 2(c^*)^2} = \sqrt{2} c_0. \end{aligned} \tag{9}$$

If $i \in \mathcal{I}^*$ then by Proposition 4 we have

$$\left\| \boldsymbol{Z}_i^\top \boldsymbol{w}^* \right\|_2 = \sqrt{\boldsymbol{w}^\top \boldsymbol{D}_i \boldsymbol{w}} = \left\| \boldsymbol{y}_i - \boldsymbol{R}^* \boldsymbol{x}_i \right\|_2 = 0. \tag{10}$$

Hence the difference $h(\boldsymbol{w}) - h(\boldsymbol{w}^*)$ is equal to

$$c_0 \sum_{i \in \mathcal{I}^*} \left\| \boldsymbol{Z}_i^\top \boldsymbol{w}_0 \right\|_2 + \sum_{i \in [\ell] \setminus \mathcal{I}^*} \left(\left\| \boldsymbol{Z}_i^\top \boldsymbol{w} \right\|_2 - \left\| \boldsymbol{Z}_i^\top \boldsymbol{w}^* \right\|_2 \right).$$

By (9) and the definition of $\eta_{\min}$ (12), we know that

$$c_0 \sum_{i \in \mathcal{I}^*} \left\| \boldsymbol{Z}_i^\top \boldsymbol{w}_0 \right\|_2 \geq \frac{k^* \eta_{\min} \operatorname{dist}(\boldsymbol{w}, \pm \boldsymbol{w}^*)}{\sqrt{2}}. \tag{11}$$

By triangle inequality the second summation in the above the difference $h(\boldsymbol{w}) - h(\boldsymbol{w}^*)$ is smaller than or equal to $\sum_{i \in [\ell] \setminus \mathcal{I}^*} \left\| \boldsymbol{Z}_i^\top (\boldsymbol{w} - \boldsymbol{w}^*) \right\|_2$, but this bound satisfies

$$\sum_{i \in [\ell] \setminus \mathcal{I}^*} \left\| \boldsymbol{Z}_i^\top (\boldsymbol{w} - \boldsymbol{w}^*) \right\|_2 \leq (\ell - k^*) \eta_{\max} \operatorname{dist}(\boldsymbol{w}, \pm \boldsymbol{w}^*),$$

where we used $\operatorname{dist}(\boldsymbol{w}, \pm \boldsymbol{w}^*) = \sqrt{2 - 2c^*} = \left\| \boldsymbol{w} - \boldsymbol{w}^* \right\|_2$ (9) and the definition of (13). We finished the proof.

B.2. Probabilistic Interpretation of Proposition 5

B.2.1 Technical Assumptions

We assume there is no noise for two reasons. First, analysis for noisy data is more challenging and requires a full different chapter to penetrate. Second, analysis in the noiseless case typically serves as a starting point for and sheds enough light on analysis for noise. For example, see the trajectory of the development from the noiseless case [25] to the noisy case [12] in the context of DPCP.

Next, we discuss probabilistic assumptions on inliers. For an inlier index $i \in \mathcal{I}^*$, each column of $\boldsymbol{Z}_i$ lies in the ground-truth hyperplane $\mathcal{S}^* \subset \mathbb{R}^4$ that is perpendicular to the ground-truth unit quaternion $\pm \boldsymbol{w}^*$, and the two columns of $\boldsymbol{Z}_i$ span a subspace $\mathcal{S}_i$ of dimension 2 that is contained in $\mathcal{S}^*$. Note that any $\boldsymbol{Z}_i' \in \mathbb{R}^{4 \times 2}$ whose columns are in $\mathcal{S}_i \cap \mathbb{S}^3$ are equivalent to $\boldsymbol{Z}_i$ in the sense that $\left\| \boldsymbol{Z}_i^\top \boldsymbol{w}^* \right\|_2 = \left\| (\boldsymbol{Z}_i')^\top \boldsymbol{w}^* \right\|_2 = 0$. To impose randomness assumptions on $\boldsymbol{Z}_i$, one could simply replace $\boldsymbol{Z}_i$ by a 4×2 random matrix whose columns are independently sampled uniformly at random from the intersection $\mathcal{S}_i \cap \mathbb{S}^3$. In fact, we need a slightly stronger assumption:

Assumption 1 (*randomness on inliers*)**.** For each $i \in \mathcal{I}^*$, every column of $\boldsymbol{Z}_i$ is independently sampled uniformly at random from the intersection $\mathcal{S}^* \cap \mathbb{S}^3$.

This assumption destroys some good property of $\boldsymbol{Z}_i$: it might not be orthonormal in general. However, it is orthonormal in expectation, i.e., it satisfies $\mathbb{E}[\boldsymbol{Z}_i^\top \boldsymbol{Z}_i] = \boldsymbol{I}_2$. This will suffice for our later analysis.

On the other hand, Assumption 1 simplifies matters by a lot. This can be appreciated in comparison with a "common" approach, where one makes assumptions on the "source data", which are point pairs $(\boldsymbol{y}_i, \boldsymbol{x}_i)$'s in our case. Let us first recall the "data flow" from $(\boldsymbol{y}_i, \boldsymbol{x}_i)$ to $\boldsymbol{Z}_i$:

$$(\boldsymbol{y}_i, \boldsymbol{x}_i) \xmapsto{\text{Proposition 1}} \boldsymbol{D}_i \xmapsto{\text{factorizing } \boldsymbol{D}_i} \boldsymbol{Z}_i \tag{12}$$

In view of the above flow (or graphical model), one intuitively (not very rigorously) feels that, if $(\boldsymbol{y}_i, \boldsymbol{x}_i)$'s are independent, then $\boldsymbol{Z}_i$'s are independent; the latter is implied by Assumption 1. On the other hand, it seems hard to know the distribution of $\boldsymbol{Z}_i$'s, even if the distribution of $(\boldsymbol{y}_i, \boldsymbol{x}_i)$'s is given or assumed. It is via Assumption 1 that this challenge is circumvented and that our theorems are developed.

Finally, we need randomness on outliers:

Assumption 2 (*randomness on outliers*)**.** Each column of any outlier $\boldsymbol{Z}_j$, where $j \in [\ell] \backslash \mathcal{I}^*$, is independently sampled uniformly at random from $\mathbb{S}^3$.

Since an outlier $\boldsymbol{Z}_j$ could be distributed arbitrarily, this assumption is the most natural, if not the most challenging, as the outliers try their best to mimic the distribution of inliers. Assumptions 1 and 2 (together with the noiseless assumption) are all we need for the next section.

B.2.2 Probabilistic Interpretation

Recall that the quantities $\eta_{\min}, \eta_{\max}$ of interest are equal to

$$\eta_{\min} = \frac{1}{k^*} \min_{\boldsymbol{w} \in \mathcal{S}^* \cap \mathbb{S}^3} \sum_{i \in \mathcal{I}^*} \left\| \boldsymbol{Z}_i^\top \boldsymbol{w} \right\|_2, \quad \text{and} \tag{13}$$

$$\eta_{\max} = \frac{1}{\ell - k^*} \max_{\boldsymbol{w} \in \mathbb{S}^3} \sum_{j \in [\ell] \backslash \mathcal{I}^*} \left\| \boldsymbol{Z}_j^\top \boldsymbol{w} \right\|_2. \tag{14}$$

The following proposition gives probabilistic upper and lower bounds for $\eta_{\max}$ and $\eta_{\min}$ respectively.

Proposition 2. *Under the assumptions of §B.2.1, we have*

(i) With probability at least $1 - \exp(-\zeta^2/2)$ *it holds that*

$$\eta_{\max} \leq \frac{1}{\sqrt{2}} + \frac{(4+\zeta)}{\sqrt{\ell - k^*}}. \tag{15}$$

(ii) With probability at least $1 - \exp(-\zeta^2/2)$ *it holds that*

$$\eta_{\min} \geq \sqrt{\frac{2}{3}} - \frac{(4+\zeta)}{\sqrt{k^*}} \tag{16}$$

To prove Proposition 2 (cf. Appendix B.2.3), we combine the proof strategies of [17] and [31], where both sets of the authors found inspirations from [16]. We can now see that the condition of Proposition 5, $\alpha^* := k^* \eta_{\min}/\sqrt{2} - (\ell - k^*)\eta_{\max} > 0$, holds with high probability as long as

$$\sqrt{\frac{2}{3}} k^* - (4+\zeta)\sqrt{k^*} \geq \frac{1}{\sqrt{2}}(\ell - k^*) + (4+\zeta)\sqrt{\ell - k^*}.$$

Ignoring lower-order terms we get the condition

$$k^* \gtrsim \frac{\sqrt{3}}{2}(\ell - k^*) \Leftrightarrow \frac{k^*}{\ell} \gtrsim \frac{\sqrt{3}}{\sqrt{3}+2}, \tag{17}$$

which holds true whenever there are sufficiently many inliers. This condition ensures the α^*-sharpness, from which local linear convergence to $\pm \boldsymbol{w}^*$ from a good enough initialization with proper stepsize ensues.

B.2.3 Details: Proof of Proposition 2

We need the following simple result, with its proof omitted.

Lemma 1. *If* $\boldsymbol{z} = [z_1, z_2, z_3, z_4]^\top$ *sampled uniformly at random from* $\mathbb{S}^3$*, we have for any* $\boldsymbol{w} \in \mathbb{S}^3$ *that*

$$\mathbb{E}\left[(\boldsymbol{z}^\top \boldsymbol{w})^2\right] = \frac{1}{4}. \tag{18}$$

On the other hand, if $\hat{\boldsymbol{z}}$ *is sampled uniformly at random from* $\mathbb{S}^3 \cap \mathcal{S}$ *where* $\mathcal{S}$ *is a linear subspace of* $\mathbb{R}^4$ *of dimension* 3*, then we have for every* $\hat{\boldsymbol{w}} \in \mathbb{S}^3 \cap \mathcal{S}$ *that*

$$\mathbb{E}\left[(\hat{\boldsymbol{z}}^\top \hat{\boldsymbol{w}})^2\right] = \frac{1}{3}. \tag{19}$$

Upper Bounding $\eta_{\max}$ **(i).** We first prove (i) of Proposition 2. Consider matrix $\boldsymbol{Z} \in \mathbb{R}^{4 \times 2}$ whose columns are sampled independently and uniformly at random from the 3-sphere $\mathbb{S}^3$. We will give upper bounds respectively for

$$(\ell - k^*) \max_{\boldsymbol{w} \in \mathbb{S}^3} \mathbb{E}\left[\left\| \boldsymbol{Z}^\top \boldsymbol{w} \right\|_2 \right] \text{ and} \tag{20}$$

$$\max_{\boldsymbol{w} \in \mathbb{S}^3} \sum_{j \in [\ell] \backslash \mathcal{I}^*} \left(\left\| \boldsymbol{Z}_j^\top \boldsymbol{w} \right\|_2 - \mathbb{E}\left[\left\| \boldsymbol{Z}^\top \boldsymbol{w} \right\|_2 \right] \right), \tag{21}$$

while summing the two bounds gives an upper bound for $(\ell - k^*)\eta_{\max}$. For (20), Jensen's inequality gives

$$\max_{\boldsymbol{w} \in \mathbb{S}^3} \mathbb{E}\left[\left\| \boldsymbol{Z}^\top \boldsymbol{w} \right\|_2 \right] \leq \max_{\boldsymbol{w} \in \mathbb{S}^3} \sqrt{\mathbb{E}\left[\left\| \boldsymbol{Z}^\top \boldsymbol{w} \right\|_2^2 \right]} \tag{22}$$

$$= \max_{\boldsymbol{w} \in \mathbb{S}^3} \sqrt{2 \cdot \frac{1}{4}} = \frac{1}{\sqrt{2}}. \tag{23}$$

To obtain (23) we used (18) and the linearity of the expectation. The second term (21) is harder to handle, and we first consider its expectation $\mathbb{E}[(21)]$. We know from a standard symmetrization argument (cf. [15], Lemma 11.4 of [7]) that, since $\boldsymbol{Z}_j$'s are independent (Assumption 2), the expectation $\mathbb{E}[(21)]$ has the following bound:

$$\mathbb{E}[(21)] \leq 2\, \mathbb{E}\left[\max_{\boldsymbol{w} \in \mathbb{S}^3} \sum_{j \in [\ell] \backslash \mathcal{I}^*} r_j \left\| \boldsymbol{Z}_j^\top \boldsymbol{w} \right\|_2 \right], \tag{24}$$

where r_j's are independent Radeamacher random variables which take values $1, -1$ with probabilities $1/2$ each and independent of $\boldsymbol{Z}_j$'s. We also know from the *vector contraction inequality* (cf. Corollary 1 of [19]) that the right-hand

side of (24), and thus $\mathbb{E}[(21)]$, is has the following bound:

$$\begin{aligned}\mathbb{E}[(21)] &\leq 2\sqrt{2}\,\mathbb{E}\Big[\max_{\boldsymbol{w}\in\mathbb{S}^3}\sum_{j\in[\ell]\setminus\mathcal{I}^*}\Big(r_{j1}\boldsymbol{Z}_{j1}^\top\boldsymbol{w}+r_{j2}\boldsymbol{Z}_{j2}^\top\boldsymbol{w}\Big)\Big]\\ &= 2\sqrt{2}\,\mathbb{E}\Big[\Big\|\sum_{j\in[\ell]\setminus\mathcal{I}^*}\big(r_{j1}\boldsymbol{Z}_{j1}+r_{j2}\boldsymbol{Z}_{j2}\big)\Big\|_2\Big] \quad (25)\end{aligned}$$

where $\boldsymbol{Z}_{j1}$'s and $\boldsymbol{Z}_{j2}$'s are the first and second columns of $\boldsymbol{Z}_j$ respectively, while r_{j1}'s and r_{j2}'s are independent Radeamacher random variables that are also independent of entries of $\boldsymbol{Z}_j$'s. Applying Jensen's inequality to (25) we get

$$\begin{aligned}\mathbb{E}[(21)] &\leq 2\sqrt{2}\sqrt{\mathbb{E}\Big[\Big\|\sum_{j\in[\ell]\setminus\mathcal{I}^*}\big(r_{j1}\boldsymbol{Z}_{j1}+r_{j2}\boldsymbol{Z}_{j2}\big)\Big\|_2^2\Big]}\\ &= 2\sqrt{2}\sqrt{\mathbb{E}\Big[\sum_{j\in[\ell]\setminus\mathcal{I}^*}\big(r_{j1}^2\boldsymbol{Z}_{j1}^\top\boldsymbol{Z}_{j1}+r_{j2}^2\boldsymbol{Z}_{j2}^\top\boldsymbol{Z}_{j2}\big)\Big]}\\ &= 4\sqrt{\ell-k^*}\end{aligned}$$

To summarize, we have $\mathbb{E}[(21)] \leq 4\sqrt{\ell-k^*}$. Treat now (21) as a function of $\boldsymbol{Z}_j$'s. It is straightforward to verify that this function has *bounded difference* 2 (cf. [20]). Since $\boldsymbol{Z}_j$'s are independent (Assumption 2), Mcdiarmid's Lemma [20] or the *bounded difference inequality* is applicable, from which we obtain the following probability bound:

$$\mathbb{P}\Big((21)\geq\mathbb{E}[(21)]+\zeta_0\Big)\leq\exp\Big(-\frac{\zeta_0^2}{2(\ell-k^*)}\Big). \quad (26)$$

With $\mathbb{E}[(21)] \leq 4\sqrt{\ell-k^*}$ and $\zeta := \zeta_0/\sqrt{(\ell-k^*)}$, we get

$$\mathbb{P}\Big((21)\leq(4+\zeta)\sqrt{\ell-k^*}\Big)\geq 1-\exp\Big(-\frac{\zeta^2}{2}\Big). \quad (27)$$

Combining this with (23) finishes proving (i).

Lower Bounding $\eta_{\min}$ (ii). Let $\boldsymbol{U}\in\mathbb{R}^{4\times 3}$ have orthonormal columns and have $\mathcal{S}^*$ as its column space, then there is a unique $\boldsymbol{v}\in\mathbb{S}^2$ so that $\boldsymbol{U}\boldsymbol{v}=\boldsymbol{w}$ for any $\boldsymbol{w}\in\mathbb{S}^3$. Also, since for any $i\in\mathcal{I}^*$ every column of $\boldsymbol{Z}_i$ is in $\mathcal{S}^*$, there is a unique $\boldsymbol{A}_i\in\mathbb{R}^{3\times 2}$ with orthonormal columns satisfying $\boldsymbol{Z}_i=\boldsymbol{U}\boldsymbol{A}_i$. Moreover, by rotation invariance we know that each column of $\boldsymbol{A}_i$ is uniformly distributed on $\mathbb{S}^2$. As a result, we get $\boldsymbol{Z}_i^\top\boldsymbol{w}=\boldsymbol{A}_i^\top\boldsymbol{v},\forall i\in\mathcal{I}^*$, and $\eta_{\min}$ is equal to

$$\eta_{\min}=\frac{1}{k^*}\min_{\boldsymbol{v}\in\mathbb{S}^2}\sum_{i\in\mathcal{I}^*}\big\|\boldsymbol{A}_i^\top\boldsymbol{v}\big\|_2. \quad (28)$$

Now, lower bounding $\eta_{\min}$ can be done in a similar way to upper bounding $\eta_{\max}$; thus we only give a proof sketch next. Similarly to (20) and (21), to bound $\eta_{\min}$ we will find lower bounds respectively for the two terms

$$k^*\min_{\boldsymbol{v}\in\mathbb{S}^2}\mathbb{E}\Big[\big\|\boldsymbol{A}^\top\boldsymbol{v}\big\|_2\Big] \text{ and} \quad (29)$$

$$\min_{\boldsymbol{v}\in\mathbb{S}^2}\sum_{i\in\mathcal{I}^*}\Big(\big\|\boldsymbol{A}_i^\top\boldsymbol{v}\big\|_2-\mathbb{E}\Big[\big\|\boldsymbol{A}^\top\boldsymbol{v}\big\|_2\Big]\Big), \quad (30)$$

where $\boldsymbol{A}$ is an i.i.d. copy of $\boldsymbol{A}_i$. Similarly to (23), the first term here is bounded using (19) and Jensen's inequality:

$$\min_{\boldsymbol{v}\in\mathbb{S}^2}\mathbb{E}\Big[\big\|\boldsymbol{A}^\top\boldsymbol{v}\big\|_2\Big]\leq\sqrt{\frac{2}{3}} \quad (31)$$

Using the symmetric argument, the vector contraction inequality, and Jensen's inequality, the expectation $\mathbb{E}[(30)]$ of the second term (30) is bounded below by $-4\sqrt{k^*}$. Similarly, invoking Mcdiarmid's Lemma gives that

$$\mathbb{P}\Big((30)\leq\mathbb{E}[(30)]-\zeta_0\Big)\leq\exp\Big(-\frac{\zeta_0^2}{2k^*}\Big) \quad (32)$$

$$\Rightarrow\mathbb{P}\Big((30)\geq-(4+\zeta)\sqrt{k^*}\Big)\geq 1-\exp\Big(-\frac{\zeta^2}{2}\Big) \quad (33)$$

where ζ_0 is any positive constant and we set $\zeta := \zeta_0/\sqrt{k^*}$. Combining (31) with the above bound finishes the proof.

C. Proof of Proposition 1

Since $\boldsymbol{b}^*$ is the rotation axis of $\boldsymbol{R}^*$, we have $(\boldsymbol{R}^*)^\top\boldsymbol{b}^*=\boldsymbol{b}^*$. Recall $\boldsymbol{v}_i=\boldsymbol{y}_i-\boldsymbol{x}_i$ for every $i\in\mathcal{I}$. If $i\in\mathcal{I}^*$ then

$$\boldsymbol{v}_i^\top\boldsymbol{b}^*=(\boldsymbol{y}_i-\boldsymbol{x}_i)^\top\boldsymbol{b}^*=(\boldsymbol{y}_i-\boldsymbol{R}^*\boldsymbol{x}_i)^\top\boldsymbol{b}^*, \quad (34)$$

and further more if $(\boldsymbol{y}_i,\boldsymbol{x}_i)$ is an inlier pair we get that

$$|\boldsymbol{v}_i^\top\boldsymbol{b}^*|=\boldsymbol{\epsilon}_i^\top\boldsymbol{b}. \quad (35)$$

Clearly $\boldsymbol{\epsilon}_i^\top\boldsymbol{b}$ is a Gaussian random variable with zero mean and variance σ^2. The rest of the proof follows from a standard probability calculation.

D. Interval Stabbing

Here we provide proofs for Propositions 2 and 3. Along the way we will need multiple temporary variables to illustrate the idea; we use $a_{i,j}$'s to denote those variables. Here, i denotes the i-th point pair, and j denotes the order in which $a_{i,j}$ appears for the first time. In §4.2.1 we reviewed interval stabbing for closed intervals $\mathcal{J}_i$ of the form $[a,b]$. One should note and verify that this can be easily extended to the case where $\mathcal{J}_i$ is a finite (disjoint) union of closed intervals.

D.1. Proof of Proposition 2

Recall $\boldsymbol{b}=[\sin(\theta)\cos(\phi),\ \sin(\theta)\sin(\phi),\ \cos(\theta)]^\top$ with $\theta\in[0,\pi]$, $\phi\in[0,\pi]$. Denote by $a_{i,1}:=[\boldsymbol{v}_i]_1\cos(\phi)+[\boldsymbol{v}_i]_2\sin(\phi)$, then $|\boldsymbol{v}_i^\top\boldsymbol{b}|\leq c$ is equivalent to

$$|a_{i,1}\sin(\theta)+[\boldsymbol{v}_i]_3\cos(\theta)|\leq c. \quad (36)$$

Without loss of generality we can assume that $a_{i,1} \geq 0$. So there is a unique $a_{i,2} \in [0, \pi]$ which satisfies

$$\cos(a_{i,2}) = \frac{[\boldsymbol{v}_i]_3}{\sqrt{[\boldsymbol{v}_i]_3^2 + a_{i,1}^2}}, \quad \sin(a_{i,2}) = \frac{a_{i,1}}{\sqrt{[\boldsymbol{v}_i]_3^2 + a_{i,1}^2}}.$$

Hence (36) is equivalent to

$$|\cos(\theta - a_{i,2})| \leq c_i, \quad c_i := \min\left\{1, \frac{c}{\sqrt{[\boldsymbol{v}_i]_3^2 + a_{i,1}^2}}\right\}$$

Since the trigonometric function $\arccos : [0, \pi] \to [-1, 1]$ is decreasing and $|\theta - a_{i,2}| \leq \pi$, the above is equivalent to

$$a_{i,3} := \arccos(-c_i) \geq |\theta - a_{i,2}| \geq \arccos(c_i) =: a_{i,4}.$$

Define $a_{i,5} = a_{i,2} - a_{i,3}$, $a_{i,6} = a_{i,2} - a_{i,4}$, $a_{i,7} = a_{i,2} + a_{i,4}$, and $a_{i,8} = a_{i,2} + a_{i,3}$. Then $|\boldsymbol{v}_i^\top \boldsymbol{b}| \leq c$ is the same as

$$\theta \in \big([a_{i,5}, a_{i,6}] \cup [a_{i,7}, a_{i,8}]\big) \cap [0, \pi] \tag{37}$$

To summarize, given $\phi \in [0, \pi]$, the i-th constraint of (5) requires θ to lie in the union of some disjoint intervals defined in (37). So maximizing (6) amounts to finding a maximal set of intervals of the form (37) that overlap a point θ, and can be solved by interval stabbing in $O(\ell \log \ell)$ time.

D.2. Proof of Proposition 3

Assume that the rotation axis $\boldsymbol{b}$ of the 3D rotation

$$\boldsymbol{R} = \boldsymbol{b}\boldsymbol{b}^\top + [\boldsymbol{b}]_\times \sin(\omega) + (\boldsymbol{I}_3 - \boldsymbol{b}\boldsymbol{b}^\top)\cos(\omega) \tag{38}$$

is given, and we now solve (7). Let $a_{i,9} = \boldsymbol{y}_i^\top \boldsymbol{b}\boldsymbol{b}^\top \boldsymbol{x}_i$, $a_{i,10} = \boldsymbol{y}_i^\top [\boldsymbol{b}]_\times \boldsymbol{x}_i$ and $a_{i,11} = \boldsymbol{y}_i^\top (\boldsymbol{I}_3 - \boldsymbol{b}\boldsymbol{b}^\top)\boldsymbol{x}_i$. Then

$$\boldsymbol{y}_i^\top \boldsymbol{R}\boldsymbol{x}_i = a_{i,9} + a_{i,10}\sin(\omega) + a_{i,11}\cos(\omega). \tag{39}$$

Hence the constraint of (4) can be written as

$$\|\boldsymbol{y}_i\|_2^2 + \|\boldsymbol{x}_i\|_2^2 - c^2 \leq 2\boldsymbol{y}_i^\top \boldsymbol{R}\boldsymbol{x}_i \tag{40}$$

$$\Leftrightarrow \; a_{i,10}\sin(\omega) + a_{i,11}\cos(\omega) \geq a_{i,12} \tag{41}$$

where we defined $a_{i,12} = (\|\boldsymbol{y}_i\|_2^2 + \|\boldsymbol{x}_i\|_2^2 - c^2)/2 - a_{i,9}$. There is a unique angle $a_{i,13} \in [0, 2\pi)$ satisfying

$$\cos(a_{i,13}) = \frac{a_{i,11}}{\sqrt{a_{i,10}^2 + a_{i,11}^2}}, \; \sin(a_{i,13}) = \frac{a_{i,10}}{\sqrt{a_{i,10}^2 + a_{i,11}^2}}$$

Thus, the constraint $\|\boldsymbol{y}_i - \boldsymbol{R}\boldsymbol{x}_i\|_2 \leq c$ of (4) is the same as

$$\cos(\omega - a_{i,13}) \geq \max\left\{\frac{a_{i,12}}{\sqrt{a_{i,10}^2 + a_{i,11}^2}}, -1\right\} =: a_{i,14}.$$

Without loss of generality assume $a_{i,14} \leq 1$, for otherwise we could simply ignore this constraint. Define $a_{i,15} = \arccos(a_{i,14}) \in [0, \pi]$. Since $|\omega - a_{i,13}| \in [0, 2\pi]$, we consider two cases, namely $|\omega - a_{i,13}| \leq \pi$ and $|\omega - a_{i,13}| > \pi$. In the former case, since $\arccos$ is a decreasing function, the above constraint is equivalent to $|\omega - a_{i,13}| \leq a_{i,15}$. In the later case the above constraint is equivalent to

$$\cos(2\pi - |\omega - a_{i,13}|) \geq a_{i,14} \Leftrightarrow 2\pi - |\omega - a_{i,13}| \leq a_{i,15}.$$

Thus, the constraint $\|\boldsymbol{y}_i - \boldsymbol{R}\boldsymbol{x}_i\|_2 \leq c$ of (4) requires $\omega \in [0, 2\pi]$ to lie in the union of the following intervals.

$$[a_{i,13} - a_{i,15},\; a_{i,13} + a_{i,15}] \cap [0, 2\pi] \tag{42}$$

$$[a_{i,13} - a_{i,15} + 2\pi,\; 2\pi] \tag{43}$$

$$[0,\; a_{i,13} + a_{i,15} - 2\pi]. \tag{44}$$

In the above, the invalid interval where the right endpoint is smaller than its left endpoint, if any, should be discarded. To conclude, (7) can be solved via interval stabbing.

E. More Experiments

In this section we present more experiments. Besides rotation errors, we will also use another metric for evaluation, that is *success rate*. Given two point clouds as input, an algorithm *succeeds* if it outputs a rotation that has error smaller than a certain threshold; by default the threshold is set to 10 degree (as in [28]) but we will also vary it when appropriate. The success rate is the number of success divided by the total number of experiments that were run. This metric was referred to as *recall* in other related papers (cf. [9]).

Note that, like GORE [8] and QUASAR [27], ARCS+$_{\text{OR}}$ can be applied to image stitching, because sometimes the translation is negligible and thus the scene can be justified by a homography $\boldsymbol{H} \in \mathbb{R}^{3\times3}$ that involves a pure 3D rotation $\boldsymbol{R}$, i.e., $\boldsymbol{H} = \boldsymbol{K}\boldsymbol{R}\boldsymbol{K}^{-1}$ (cf. [24]); here $\boldsymbol{K} \in \mathbb{R}^{3\times3}$ is a matrix of intrinsic camera parameters given by the dataset. However, we noticed that the recent approaches MAGSAC++ [2–4] and VSAC [14] achieved surprising performance and run in fewer than 10 milliseconds for image stitching, hence we would recommend them for this task.

E.1. Robustness on Gaussian Point Sets

In previous synthetic experiments on robust rotation search (Table 4), we generated data by ensuring that each point pair $\boldsymbol{y}_i$ and $\boldsymbol{x}_i$ has nearly the same norm. This is for fair comparison of the methods, and it might not be true in practice. Here we show that, without this norm constraint, ARCS+$_{\text{OR}}$ can tolerate even more outliers. In the experiment here we generated point sets $\{\boldsymbol{y}_i, \boldsymbol{x}_i\}_{i=1}^{\ell}$ as in Table 4 except without the norm constraint. Then, we first perform a simple step, that removes all point pairs $(\boldsymbol{y}_i, \boldsymbol{x}_i)$ which satisfy $\big|\|\boldsymbol{y}_i\|_2 - \|\boldsymbol{x}_i\|_2\big| > c$, and then feed the remaining

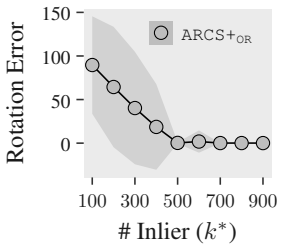


Figure 1. Performance of ARCS+OR on Gaussian point sets with $\ell = 10^7$ point pairs, $\sigma = 0.01$, 100 trials. ARCS+OR works well until there are fewer than $500/10^7 = 0.005\%$ inliers.

points to ARCS+OR. We reported the results in Figure 1, where we observed that ARCS+OR worked well until there are fewer than $500/10^7 = 0.005\%$ inliers.

E.2. Sensitivity to The Ground-Truth Rotation

In Figure 2 we presented the sensitivity of ARCS+OR to the ground-truth rotation $\boldsymbol{R}^*$. Figure 2a depicted that, with the ground-truth rotation angle ω^* changing, the mean estimation error of ARCS+O varied from 0.5 to 1, while the standard derivation ranged from 0 to 0.5. One the other hand, ARCS+R refined the estimate from ARCS+O, so that their combination ARCS+OR had much smaller mean error and standard derivation, nearly imperceivable from Figure 2a. In Figure 2b we kept ω^* fixed and presented how the errors of ARCS+OR vary with θ^* and ϕ^*, the two angles for the ground-truth rotation axis $\boldsymbol{b}^*$; we fixed one of them when varying the other. We observed that ARCS+OR is immune to the change of ϕ^*, as it consistently gave about 0.02 errors and 0.01 standard derivation. This is expected as ARCS+O selects from multiple ϕ_j's a best one based on consensus maximization. On the other hand, varying ϕ^* does make an impact on the performance of ARCS+OR; the standard deviation reached its peak, around 0.04, when $\theta^* = \pi/4$. Theoretically justifying the phenomenon presented here can be an interesting future work.

E.3. Phase Transition

In Figure 3, we showed the performances of algorithms for different inlier ratios k^*/ℓ and different number ℓ of points; whiter means smaller errors and errors larger than 1 were truncated to 1. The major point we would like to clarify here is that, whether or not an algorithm can tolerate say 99% outliers might depend on the total number of points (cf. Figures 3a and 3d), so sentences such as "our algorithm can tolerate 99% outliers" might be inaccurate, even though such description has been widely used in recent papers. Indeed, no algorithm can tolerate 99% outliers if $\ell = 100$. Also, as mentioned in §4.3, one theorem of [6] has shed light on this phenomenon. The other important observation here is that, GNC-TLS achieved higher accuracy than

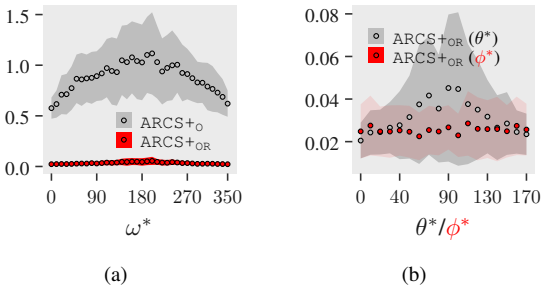


Figure 2. Average rotation errors (in degrees) and standard deviations with respect to the ground-truth rotation angle ω^* and axis $\boldsymbol{b}^* = [\sin(\theta^*)\cos(\phi^*),\ \sin(\theta^*)\sin(\phi^*),\ \cos(\theta^*)]^\top$. Experiments run with 100 trials, $\ell = 10^5$, $k^* = 1000$, $\sigma = 0.01$.

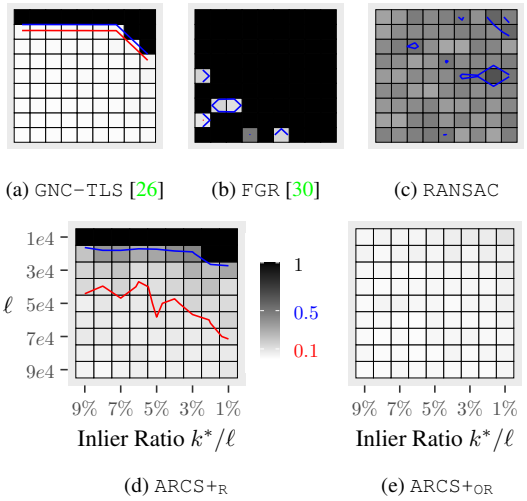


Figure 3. Average rotation errors in degrees of different robust rotation search approaches on medium-scale synthetic 3D point sets of sizes varying from 10^4 to 9×10^4 with inlier ratios ranging from 1% to 9%. Experiments run with 50 trials, $\sigma = 0.01$ fixed.

ARCS+R, although they exhibited nearly the same breaking down points. One reason is that GNC-TLS takes advantage of the inlier threshold c as extra information. This empirically suggests that combing ARCS+O and GNC-TLS might further boost the performance for robust rotation search.

E.4. Robustness to Noise

Figure 4 showed that ARCS+N is sensitive to noise: In particular, for $c = 5.54\sigma$ fixed, the number ℓ of output point pairs grows proportionally as a linear function of σ. A similar phenomenon can be found in [28] and its follow-up works: Higher noise leads to denser graphs, and thus to intractable maximal clique problems (recall Section 1).

However, ARCS+O and ARCS+OR behave reasonably well as noise varies. This was shown in Figure 5, where we observed that, for $k^*/\ell = 100/1000$, ARCS+OR are com-

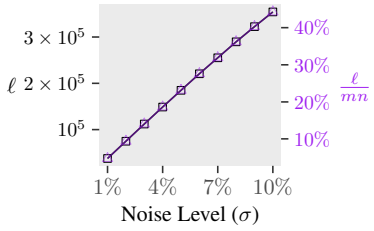
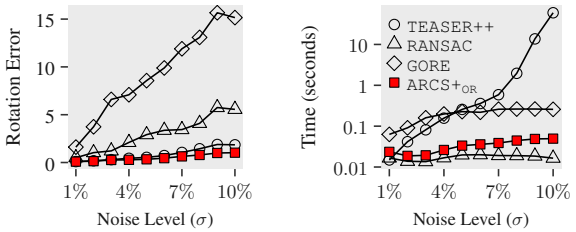


Figure 4. Sensitivity of ARCS+$_{\text{N}}$ to noise: ℓ increases linearly as σ grows. 100 trials, $m = 1000, n = 800, k^* = 200, c = 5.54\sigma$.



(a) $k^*/\ell = 100/1000$ (b) $k^*/\ell = 100/1000$

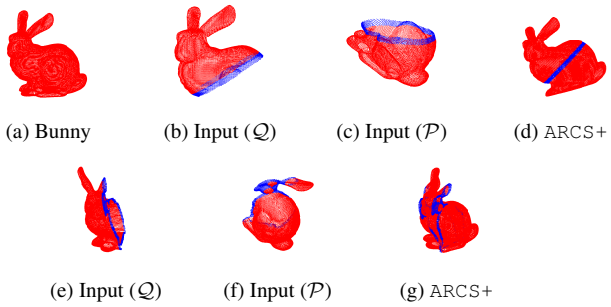
Figure 5. Robustness of various methods to noise. 20 trials.

petitive to TEASER++ in terms of accuracy (Figure 5a) and to RANSAC in terms of speed (Figure 5b); 10% inliers are enough for RANSAC to be fast. Also note that the running time of TEASER++ increases exponentially as noise grows, and that GORE would achieve higher accuracy if some local refinement methods were applied.

E.5. Procrustes's Experiments on Stanford Bunny

Here we use ARCS+ for simultaneous search of rotation & correspondences on a popular benchmark, the Stanford Bunny dataset [10].[1] Bunny has 35947 points with every coordinate of the points located in $[-1, 1]$ (Figure 6a). We randomly cut it into two parts, $\mathcal{Q}$ and $\mathcal{P}$, of sizes m and n respectively and of different overlapping ratios $k^*/\max\{m, n\} = k^*/m$ (Figures 6b-6c or 6e-6f). For simplicity we set $n = \lceil 35947/2 \rceil = 17974$ and, $m = \lfloor 35947/2 \rfloor + k^*$, so the exact values of m and k^* can be calculated as per a given overlapping ratio k^*/m. We then randomly rotated $\mathcal{P}$ and added 1% random Gaussian noise to it. The goal is to align $\mathcal{P}$ and $\mathcal{Q}$. ARCS+ can be applied directly to this task (Figures 6d or 6g). For comparison, we gave GORE and TEASER++ the correspondences established by FPFH. For all methods we set $c = 5 \times 10^{-5}$. Figure 7 showed the results for different overlapping ratios, from which we made a few observations: ARCS+ achieved higher success rates in all experiments, while the performance of FPFH, and thus of TEASER++ and GORE, improved as the overlapping ratios increased. We did not put

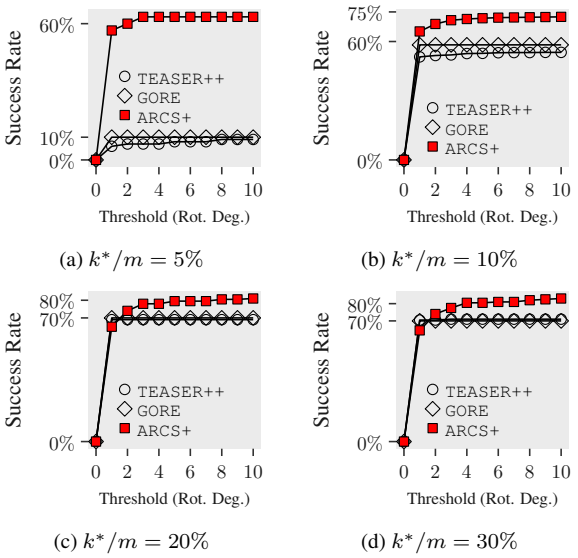
[1] In view of our opening quote, Bunny here is a victim of Procrustes.



(a) Bunny (b) Input ($\mathcal{Q}$) (c) Input ($\mathcal{P}$) (d) ARCS+

(e) Input ($\mathcal{Q}$) (f) Input ($\mathcal{P}$) (g) ARCS+

Figure 6. Bunny (6a) was cut through its body into two parts, $\mathcal{Q}$ (6b) and $\mathcal{P}$ (6c), with $k^*/m = 1997/19970 = 10\%$ overlapping points in blue. $\mathcal{P}$ was randomly rotated and corrupted by 1% random noise. ARCS+ successfully aligned $\mathcal{Q}$ and $\mathcal{P}$ (6d). For a different cut through the ear of Bunny (6e-6f), ARCS+ failed (6g).



(a) $k^*/m = 5\%$ (b) $k^*/m = 10\%$

(c) $k^*/m = 20\%$ (d) $k^*/m = 30\%$

Figure 7. Success Rates of the methods on Stanford Bunny with different overlapping ratios k^*/m. 1000 trials. In each trial, Bunny was randomly cut into two parts $\mathcal{Q}$ and $\mathcal{P}$, and $\mathcal{P}$ was then rotated randomly and corrupted by 1% random Gaussian noise.

RANSAC into comparison here, because FPFH often gave few to none inlier pairs for small k^*/m and so RANSAC used much longer time to reach a confidence of 0.99.

F. Handling The Translation Case

Utilizing ideas that have been known in prior works, it is easy to extend our algorithms to the situation where there is an extra unknown translation. As we did for Problems 1 and 2, we first define the two problems that we will discuss:

Problem 1 (*simultaneous pose and correspondences*)**.** Let the two point sets $\mathcal{Q}$ and $\mathcal{P}$ of Problem 1 instead satisfy

$$\boldsymbol{q}_i = \boldsymbol{R}^* \boldsymbol{p}_j + \boldsymbol{t}^* + \boldsymbol{o}_{i,j} + \boldsymbol{\epsilon}_{i,j}, \tag{45}$$

where $\boldsymbol{t}^* \in \mathbb{R}^3$ is an extra unknown translation vector. The task is to simultaneously estimate the rotation $\boldsymbol{R}^*$, translation $\boldsymbol{t}^*$, and correspondences $\mathcal{C}^*$ from $\mathcal{Q}$ and $\mathcal{P}$.

Problem 2 (*robust registration*)**.** Let the ℓ pairs of 3D points $\{(\boldsymbol{y}_i, \boldsymbol{x}_i)\}_{i=1}^{\ell}$ of Problem 2 instead satisfy

$$\boldsymbol{y}_i = \boldsymbol{R}^* \boldsymbol{x}_i + \boldsymbol{t}^* + \boldsymbol{o}_i + \boldsymbol{\epsilon}_i. \tag{46}$$

The task is to find $\boldsymbol{R}^*$, $\boldsymbol{t}^*$, and correspondences $\mathcal{I}^*$.

We will discuss more about Problem 1 in our future work; here we focus on the its special case, Problem 2. Specifically, we next extend our ARCS+$_{\text{OR}}$ algorithm to handle Problem 2 (Appendix F.1), and present its performance on the 3DMatch dataset [29] (Appendix F.2).

F.1. Extension for Robust Registration

Here we present an extension of our ARCS+$_{\text{OR}}$ algorithm for solving Problem 2. In this extension, we essentially combine ARCS+$_{\text{OR}}$ with *known* techniques. Thus, the presentation here serves more as an useful demonstration of concepts, and less as an entirely novel insight into, or the most efficient method for, solving Problem 2. Nevertheless, we will show in Appendix F.2 that our extension does enjoy state-of-the-art performance on the 3DMatch dataset [29].

We first review three crucial ingredients that are useful for solving Problem 2: *translation elimination (*TE), rotation elimination (RE), and *outlier removal*.

Translation Elimination (TE). For each $i, j \in [\ell]$, $i > j$, define $\mathbf{y}_{ij} = \boldsymbol{y}_i - \boldsymbol{y}_j$ and $\mathbf{x}_{ij} = \boldsymbol{x}_i - \boldsymbol{x}_j$, then

$$\mathbf{y}_{ij} = \boldsymbol{R}^* \mathbf{x}_{ij} + (\boldsymbol{o}_i - \boldsymbol{o}_j) + (\boldsymbol{\epsilon}_i - \boldsymbol{\epsilon}_j). \tag{47}$$

Here $(\mathbf{y}_{ij}, \mathbf{x}_{ij})$ is referred to in the literature as *translation invariant measurements*, as (47) no longer involves translation. As a consequence, robust rotation search might be performed over $\{(\mathbf{y}_{ij}, \mathbf{x}_{ij})\}_{i>j}$, yielding an estimate of rotation and correspondences. After this, the translation can be easily computed. A disadvantage here is that computing all $(\mathbf{y}_{ij}, \mathbf{x}_{ij})$'s needs $O(\ell^2)$ time; also note though that this computation can be implemented in parallel and thus can be efficient for medium-size datasets (*e.g.*, $\ell \leq 3 \times 10^4$).

Rotation Elimination (RE). Every inlier $(\boldsymbol{y}_i, \boldsymbol{x}_i)$ satisfying (46) with $\boldsymbol{o}_i = 0$ also necessarily satisfies

$$\left\|\boldsymbol{x}_i + \boldsymbol{\epsilon}_i\right\|_2 = \left\|\boldsymbol{y}_i - \boldsymbol{t}^*\right\|_2 \Leftrightarrow \left\|\boldsymbol{x}_i\right\|_2 \approx \left\|\boldsymbol{y}_i - \boldsymbol{t}^*\right\|_2. \tag{48}$$

If there were no outliers, estimating t^* from relation (48) is the problem of *source localization* that appears in signal processing applications [5]. Estimating translation from (48) in the presence of outliers is more challenging. A possible algorithm is combining the least-squares solvers of [5] with an iterative reweighting strategy, but this does not have global optimality guarantee. The other approach, which we employ, is to estimate t^* via branch & bound, solving the following optimization problem:

$$\begin{aligned} \max_{\mathcal{I} \subset [\ell], \boldsymbol{t} \in \mathbb{R}^3} \quad & |\mathcal{I}| \\ \text{s.t.} \quad & \left| \|\boldsymbol{y}_i - \boldsymbol{t}\|_2 - \|\boldsymbol{x}_i\|_2 \right| \leq c, \ \ \forall i \in \mathcal{I} \end{aligned} \tag{49}$$

If directly applying branch & bound to (49), one would branch over $\mathbb{R}^3$ (cf. [18]). On the other hand, our development in §4.2 implies that branching over $\mathbb{R}^2$, where the first two coordinates of $\boldsymbol{t}$ lie, suffices, as the third coordinate can be determined by interval stabbing. In short, we solve (49) via branching over the two-dimensional space $\mathbb{R}^2$ if needed. As a matter of fact, branch & bound runs much faster even if the parameter space has smaller dimension.

Remark 1 (TE versus RE). Translation elimination (TE) yields $O(\ell^2)$ measurements, leads to the problem of robust rotation search, and is also used in the 2D-3D *perspective-three-point* problem (see, *e.g.*, [22]); many recent papers on 3D-3D registration used TE (see, *e.g.*, [28] and its follow-up works). RE yields $O(\ell)$ measurements, leads to a less familiar problem, and receives fewer attention; [18] is the only paper, which we know, that uses RE (for Problem 1).

Outlier Removal. Even though rotation or translation can be estimated independently of each other (using TE or RE respectively), they might not be able to handle the case of extreme outlier rates. In particular, if using TE then the inlier ratio decreases from k^*/ℓ to $O\big((k^*/\ell)^2\big)$. This is why an outlier removal procedure is needed prior to estimation. For this, create (in mind) a graph $\mathcal{G}$ with ℓ vertices representing the ℓ point pairs $\{(\boldsymbol{y}_i, \boldsymbol{x}_i)\}_{i=1}^{\ell}$. Moreover, create an edge between two vertices i and j, if $|\mathbf{y}_{ij} - \mathbf{x}_{ij}| \leq 2c$, where $\mathbf{y}_{ij}$ and $\mathbf{x}_{ij}$ are defined in (47). Then, find a maximum clique of $\mathcal{G}$, and remove all point pairs whose corresponding vertices are not contained in the maximum clique. See [21,23,28] for more transparent discussion on this idea.

For implementation, we use the code of [21] to create $\mathcal{G}$ and compute a maximum clique of it.

Algorithms. Having reviewed the three ingredients, we are ready to extend ARCS+$_{\text{OR}}$ for Problem 2. We have two extensions, (ARCS++$_{\text{OR}}$)$^{\text{TE}}$ and (ARCS++$_{\text{OR}}$)$^{\text{RE}}$, summarized in Table 1. Both of them have the same first step, outlier removal via finding a maximum clique from the constructed graph. Their next steps proceed by working with point pairs that survive from outlier removal. Step 2 of (ARCS++$_{\text{OR}}$)$^{\text{TE}}$ is to eliminate the translation (TE), and step 3 is to estimate the rotation via ARCS+$_{\text{OR}}$ from the point pairs $\{(\mathbf{y}_{ij}, \mathbf{x}_{ij})\}_{i>j}$ (47). Step 4 of (ARCS++$_{\text{OR}}$)$^{\text{TE}}$ would estimate the translation from the remaining point pairs, with an estimated rotation given by ARCS+$_{\text{OR}}$. But we leave step 4 unspecified, as translation estimation in this situation is straightforward. On the other hand, step 2 of (ARCS++$_{\text{OR}}$)$^{\text{RE}}$ is to eliminate the rotation (RE), step 3 is

Table 1. Two Extensions of ARCS++$_{\text{OR}}$ for Problem 2.

	(ARCS++$_{\text{OR}}$)$^{\text{TE}}$	(ARCS++$_{\text{OR}}$)$^{\text{RE}}$
Step 1	Outlier Removal	
Step 2	TE	RE
Step 3	ARCS+$_{\text{OR}}$ (47)	Branch & Bound (49)
Step 4	—	ARCS+$_{\text{OR}}$
Step 5	Local Refinement (optional)	

to compute a translation $\hat{\boldsymbol{t}}$ by solving (49), and step 4 is to estimate the rotation via ARCS+$_{\text{OR}}$, operating on point pairs $(\boldsymbol{y}_i - \hat{\boldsymbol{t}}, \boldsymbol{x}_i)$'s. Finally, one might use an extra step 5, to refine the solution, *e.g.*, by singular value decomposition.

F.2. Experiments on 3DMatch

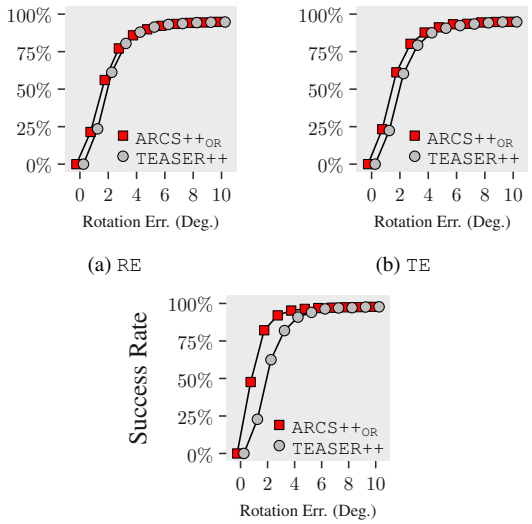
Data. The 3DMatch dataset [29] contains more than 1000 point clouds for testing, representing 8 different scenes (such as kitchen, hotel, etc.), while the number of point clouds for each scene ranges from 77 to 506. Each point cloud has more than 10^5 points, yet in [29] there are 5000 keypoints for each cloud. We used the pretrained model[2] of the 3DSmoothNet [13] to extract descriptors from these key points, and matched them using the Matlab function pcmatchfeatures, with its parameter MatchThreshold set to the maximum 1. It remains to solve Problem 2 using these hypothetical correspondences.

Metrics. We report success rates of the methods. Success rates were defined in the beginning of Appendix E. The default threshold 10 on rotation degrees is the one that was used in TEASER++ [28]. We do not report errors in terms of translation for two reasons: i) rotation search is the main theme of the paper, ii) if the rotation is estimated accurately, then so will be the translation (see, *e.g.*, algorithms of [28]).

Methods. We apply (ARCS++$_{\text{OR}}$)$^{\text{TE}}$ and (ARCS++$_{\text{OR}}$)$^{\text{RE}}$ to restore the rotation and translation from these correspondences. We use singular value decomposition as an extra step 5 for (ARCS++$_{\text{OR}}$)$^{\text{RE}}$ to refine the solution and account for inaccuracy of translation estimation via branch & bound (49). For reference, we also apply (ARCS++$_{\text{OR}}$)*, which uses the ground-truth translation $\boldsymbol{t}^*$ and point pairs $(\boldsymbol{y}_i - \boldsymbol{t}^*, \boldsymbol{x}_i)$'s to estimate a rotation via ARCS++$_{\text{OR}}$.

We compare our algorithms with TEASER++ [28]. Similarly, we use three versions of TEASER++. The first version is (TEASER++)$^{\text{TE}}$. This is the standard TEASER++, and the difference between (TEASER++)$^{\text{TE}}$ and (ARCS++$_{\text{OR}}$)$^{\text{TE}}$ is that, (TEASER++)$^{\text{TE}}$ estimates the rotation by GNC-TLS, not ARCS++$_{\text{OR}}$. The second version is (TEASER++)$^{\text{TE}}$, where we treat TEASER++ as a robust rotation search method and let it play the role of ARCS++$_{\text{OR}}$ in (ARCS++$_{\text{OR}}$)$^{\text{RE}}$. The third version is (TEASER++)*,

[2] https://github.com/zgojcic/3DSmoothNet



(a) RE (b) TE

(c) Ground-Truth Translation

Figure 8. Success rates of TEASER++ and ARCS++$_{\text{OR}}$ on the 3D Match dataset, using either estimated translation (Fig. 8a) or TIMs (Fig. 8b) or ground-truth translation (Fig. 8c).

where we assume the ground-truth translation $\boldsymbol{t}^*$ is given and run TEASER++ on $(\boldsymbol{y}_i - \boldsymbol{t}^*, \boldsymbol{x}_i)$'s. Finally, we did not compare other methods here, as TEASER++ currently has the best performance (to the best of our knowledge) on the 3DMatch dataset, see [28] for comparison with optimization-based methods, and also read from [9] the success rates (recall) of other deep learning methods.

Results. Following [28], we set $c = 0.05$. We presented results in Table 2 and Figure 8. In Table 2 we observed that ARCS++$_{\text{OR}}$ and TEASER++ have very close performance, although ARCS++$_{\text{OR}}$ has slight advantage (*e.g.*, in 12 cases in bold ARCS++$_{\text{OR}}$ has higher success rates). In terms of running times, ARCS++$_{\text{OR}}$ is slower than TEASER++. One reason is that we used an industrial-strength implementation[3] of TEASER++, while ARCS++$_{\text{OR}}$ was implemented in plain Matlab. This suggests our current idea of extending ARCS+$_{\text{OR}}$ into the translation case might be sub-optimal, and will motivate us to design even faster algorithms for that purpose, which though will require serious innovations. Finally, in Figure 8, we reported the success rates averaged over all testing scenes of 3DMatch and with the threshold (rotation degree) varying from 0 to 100. This delivers the same message that our direct extension of ARCS+$_{\text{OR}}$ maintains a state-of-the-art performance for solving Problem 2.

References

[1] Simon L Altmann. Hamilton, Rodrigues, and the quaternion scandal. *Mathematics Magazine*, 62(5):291–308, 1989. 1

[3] https://github.com/MIT-SPARK/TEASER-plusplus

Table 2. Success rates of methods run on the scene pairs of the 3DMatch dataset [29] for which the ground-truth transformations are provided (rotation error smaller than 10 degree means a success [28]; see also the first paragraph of Appendix E).

Scene Type	Kitchen	Home 1	Home 2	Hotel 1	Hotel 2	Hotel 3	Study Room	MIT Lab	Overall
# Scene Pairs	506	156	208	226	104	54	292	77	1623
$(\texttt{TEASER++})^{\texttt{TE}}$	98.4%	**92.9%**	89.9%	98.2%	92.3%	94.4%	**93.2%**	88.3%	94.82%
$(\texttt{TEASER++})^{\texttt{RE}}$	**99.0%**	**92.3%**	89.4%	98.7%	91.3%	94.4%	92.5%	88.3%	94.76%
$(\texttt{TEASER++})^{*}$	**99.0%**	**98.1%**	94.7%	98.7%	**99.0%**	98.1%	97.0%	94.8%	97.72%
$(\texttt{ARCS++}_{\texttt{OR}})^{\texttt{TE}}$	**98.6%**	92.3%	**90.4%**	**98.7%**	**93.3%**	94.4%	92.5%	88.3%	**94.89%**
$(\texttt{ARCS++}_{\texttt{OR}})^{\texttt{RE}}$	98.4%	91.7%	**89.9%**	**99.1%**	**94.2%**	94.4%	92.5%	88.3%	**94.82%**
$(\texttt{ARCS++}_{\texttt{OR}})^{*}$	98.4%	97.4%	**95.7%**	98.7%	98.1%	**100%**	97.3%	**96.1%**	97.72%

[2] Daniel Barath, Jiri Matas, and Jana Noskova. Magsac: marginalizing sample consensus. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10197–10205, 2019. 5

[3] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. Magsac++, a fast, reliable and accurate robust estimator. In *IEEE/CVF conference on computer vision and pattern recognition*, pages 1304–1312, 2020. 5

[4] Daniel Barath, Jana Noskova, and Jiri Matas. Marginalizing sample consensus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 5

[5] Amir Beck, Petre Stoica, and Jian Li. Exact and approximate solutions of source localization problems. *IEEE Transactions on signal processing*, 56(5):1770–1778, 2008. 8

[6] Cindy Orozco Bohorquez, Yuehaw Khoo, and Lexing Ying. Maximizing robustness of point-set registration by leveraging non-convexity. Technical report, arXiv:2004.08772v3 [math.OC], 2020. 6

[7] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013. 3

[8] Álvaro Parra Bustos and Tat-Jun Chin. Guaranteed outlier removal for rotation search. In *IEEE International Conference on Computer Vision*, pages 2165–2173, 2015. 5

[9] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 5, 9

[10] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Annual Conference on Computer Graphics and Interactive Techniques*, pages 303–312, 1996. 7

[11] Tianjiao Ding, Yunchen Yang, Zhihui Zhu, Daniel P Robinson, René Vidal, Laurent Kneip, and Manolis C Tsakiris. Robust homography estimation via dual principal component pursuit. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6080–6089, 2020. 2

[12] Tianyu Ding, Zhihui Zhu, Tianjiao Ding, Yunchen Yang, René Vidal, Manolis C. Tsakiris, and Daniel Robinson. Noisy dual principal component pursuit. In *International Conference on Machine Learning*, pages 1617–1625, 2019. 2

[13] Zan Gojcic, Caifa Zhou, Jan D. Wegner, and Andreas Wieser. The perfect match: 3D point cloud matching with smoothed densities. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5540–5549, 2019. 9

[14] Maksym Ivashechkin, Daniel Barath, and Jiří Matas. Vsac: Efficient and accurate estimator for h and f. In *IEEE/CVF International Conference on Computer Vision*, pages 15243–15252, 2021. 5

[15] Sham Kakade. Symmetrization and Rademacher averages. Technical report, Lecture 11 of Stat 928: Statistical Learning Theory, 2011. 3

[16] Gilad Lerman, Michael B. McCoy, Joel A. Tropp, and Teng Zhang. Robust computation of linear models by convex relaxation. *Foundations of Computational Mathematics*, 15(2):363–410, 2015. 3

[17] Xiao Li, Shixiang Chen, Zengde Deng, Qing Qu, Zhihui Zhu, and Anthony Man-Cho So. Weakly convex optimization over Stiefel manifold using Riemannian subgradient-type methods. *SIAM Journal on Optimization*, 31(3):1605–1634, 2021. 2, 3

[18] Yinlong Liu, Chen Wang, Zhijian Song, and Manning Wang. Efficient global point cloud registration by matching rotation invariant features through translation search. In *European Conference on Computer Vision*, 2018. 8

[19] Andreas Maurer. A vector-contraction inequality for Rademacher complexities. In *International Conference on Algorithmic Learning Theory*, pages 3–17. Springer, 2016. 3

[20] Colin McDiarmid et al. On the method of bounded differences. *Surveys in Combinatorics*, 141(1):148–188, 1989. 4

[21] Álvaro Parra, Tat-Jun Chin, Frank Neumann, Tobias Friedrich, and Maximilian Katzmann. A practical maximum clique algorithm for matching with pairwise constraints. Technical report, arXiv:1902.01534v2 [cs.CV], 2020. 8

[22] Mikael Persson and Klas Nordberg. Lambda twist: An accurate fast robust perspective three point (p3p) solver. In *European conference on computer vision*, pages 318–332, 2018. 8

[23] Jingnan Shi, Heng Yang, and Luca Carlone. ROBIN: a graph-theoretic approach to reject outliers in robust estimation using invariants. Technical report, arXiv:2011.03659v2 [cs.CV], 2021. 8

[24] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer Science & Business Media, 2010. 5

[25] Manolis C. Tsakiris and René Vidal. Dual principal component pursuit. *Journal of Machine Learning Research*, 19(18):1–50, 2018. 2

[26] Heng Yang, Pasquale Antonante, Vasileios Tzoumas, and Luca Carlone. Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection. *IEEE Robotics and Automation Letters*, 5(2):1127–1134, 2020. 6

[27] Heng Yang and Luca Carlone. A quaternion-based certifiably optimal solution to the Wahba problem with outliers. In *IEEE International Conference on Computer Vision*, pages 1665–1674, 2019. 5

[28] Heng Yang, Jingnan Shi, and Luca Carlone. TEASER: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 37(2):314–333, 2021. 5, 6, 8, 9, 10

[29] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and T Funkhouser. 3DMatch: Learning the matching of local 3D geometry in range scans. In *IEEE Conference on Computer Vision and Pattern Recognition*, page 4, 2017. 8, 9, 10

[30] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *European Conference on Computer Vision*, pages 766–782, 2016. 6

[31] Zhihui Zhu, Yifan Wang, Daniel Robinson, Daniel Naiman, René Vidal, and Manolis C. Tsakiris. Dual principal component pursuit: Improved analysis and efficient algorithms. In *Advances in Neural Information Processing Systems*, 2018. 3