This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

ARCS: Accurate Rotation and Correspondence Search

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

Abstract

This paper is about the old Wahba problem in its more general form, which we call "simultaneous rotation and correspondence search". In this generalization we need to find a rotation that best aligns two partially overlapping 3D point sets, of sizes m and n respectively with $m \geq n$. We first propose a solver, `ARCS`*, that i) assumes noiseless point sets in general position, ii) requires only 2 inliers, iii) uses $O(m \log m)$ time and $O(m)$ space, and iv) can successfully solve the problem even with,* e.g., *$m, n \approx 10^6$ in about 0.1 seconds. We next robustify* `ARCS` *to noise, for which we approximately solve consensus maximization problems using ideas from robust subspace learning and interval stabbing. Thirdly, we refine the approximately found consensus set by a Riemannian subgradient descent approach over the space of unit quaternions, which we show converges globally to an ε-stationary point in $O(\varepsilon^{-4})$ iterations, or locally to the ground-truth at a linear rate in the absence of noise. We combine these algorithms into* `ARCS+`*, to simultaneously search for rotations and correspondences. Experiments show that* `ARCS+` *achieves state-of-the-art performance on large-scale datasets with more than 10^6 points with a 10^4 time-speedup over alternative methods.* https://github.com/liangzu/ARCS

1. Introduction

> The villain Procrustes forced his victims to sleep on an iron bed; if they did not fit the bed he cut off or stretched their limbs to make them fit [27].
>
> Richard Everson

Modern sensors have brought the classic *Wahba* problem [75], or slightly differently the *Procrustes analysis* problem [31], into greater generality that has increasing importance to computer vision [34, 50], computer graphics [58], and robotics [12]. We formalize this generalization as follows.

Problem 1 (*simultaneous rotation and correspondence search*)**.** Consider point sets $\mathcal{Q} = \{\boldsymbol{q}_1, \dots, \boldsymbol{q}_m\} \subset \mathbb{R}^3$ and $\mathcal{P} = \{\boldsymbol{p}_1, \dots, \boldsymbol{p}_n\} \subset \mathbb{R}^3$ with $m \geq n$. Let $\mathcal{C}^*$ be a subset of $[m] \times [n] := \{1, \dots, m\} \times \{1, \dots, n\}$ of size k^*, called the *inlier correspondence* set, such that all pairs (i_1, j_1) and (i_2, j_2) of $\mathcal{C}^*$ satisfy $i_1 \neq i_2$ and $j_1 \neq j_2$. Assume that

$$\boldsymbol{q}_i = \boldsymbol{R}^* \boldsymbol{p}_j + \boldsymbol{\epsilon}_{i,j}, \quad \text{if } (i,j) \in \mathcal{C}^* \tag{1}$$

where $\boldsymbol{\epsilon}_{i,j} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_3)$ is noise, $\boldsymbol{R}^*$ is an unknown 3D rotation, and $(\boldsymbol{q}_i, \boldsymbol{p}_j)$ is called an *inlier*. If $(i,j) \notin \mathcal{C}^*$ then $(\boldsymbol{q}_i, \boldsymbol{p}_j)$ is arbitrary and is called an *outlier*. The goal of the simultaneous rotation and correspondence search problem is to simultaneously estimate the 3D rotation $\boldsymbol{R}^*$ and the *inlier correspondence set* $\mathcal{C}^*$ from point sets $\mathcal{Q}$ and $\mathcal{P}$.

We focus on Problem 1 for two reasons. First, it already encompasses several vision applications such as image stitching [16]. Second, the more general and more important *simultaneous pose and correspondence* problem, which involves an extra unknown translation in (1), reduces to Problem 1 by eliminating the translation parameters (at the cost of squaring the number of measurements) [80]. As surveyed in [38], whether accurate and fast algorithms exist for solving the pose and correspondence search is largely an open question. Therefore, solving the simpler Problem 1 efficiently is an important step for moving forward.

For Problem 1 or its variants, there is a vast literature of algorithms that are based on i) local optimization via *iterative closest points* (`ICP`) [13, 20, 66] or *graduated non-convexity* (`GNC`) [1, 76, 85] or others [23, 41, 59], ii) global optimization by branch & bound [18, 21, 50, 54, 55, 64, 70, 81], iii) outlier removal techniques [16, 62, 63, 69, 80], iv) semidefinite programming [39, 58, 71, 77, 79], v) RANSAC [28, 51, 52, 72], vi) deep learning [4, 9, 22, 37], and vii) spherical Fourier transform [12]. But all these methods, if able to accurately solve Problem 1 with the number k^* of inliers extremely small, take $\Omega(mn)$ time. Yet we have:

Theorem 1 (`ARCS`)**.** *Suppose there are at least two inliers, $k^* \geq 2$, and that the point sets $\mathcal{Q}$ and $\mathcal{P}$ of Problem 1 are noiseless "in general position". Then there is an algorithm that solves Problem 1 in $O(m \log m)$ time and $O(m)$ space.*

Remark 1 (*general position assumption*). In Theorem 1, by "*in general position*" we mean that i) for any outlier $(\boldsymbol{q}_i, \boldsymbol{p}_j)$, we have $\|\boldsymbol{q}_i\|_2 \neq \|\boldsymbol{p}_j\|_2$, ii) there exists some inlier pairs $(\boldsymbol{q}_{i_1}, \boldsymbol{p}_{j_1})$ and $(\boldsymbol{q}_{i_2}, \boldsymbol{p}_{j_2})$ such that $\boldsymbol{q}_{i_1}$ and $\boldsymbol{q}_{i_2}$ are not parallel. If point sets $\mathcal{Q}$ and $\mathcal{P}$ are randomly sampled from $\mathbb{R}^3$, these two conditions hold true with probability 1.

A numerical illustration of Theorem 1 is that our ARCS solver, to be described in §3, can handle the case where $m = 10^6, n = 8 \times 10^5$ and $k^* = 2$, in about 0.1 seconds (cf. Table 1).[1] However, like other correspondence-based minimal solvers for geometric vision [29,45–47,61], ARCS might be fragile to noise. That being said, it can be extended to the noisy case, leading to a three-step algorithm called ARCS+, which we summarize next.

The first step ARCS+_N of ARCS+ extends ARCS by *establishing correspondences under noise*. ARCS+_N outputs in $O(\ell + m \log m)$ time a candidate correspondence set $\overline{\mathcal{C}}$ of size ℓ that contains $\mathcal{C}^*$. Problem 1 then reduces to estimating $\boldsymbol{R}^*$ and $\mathcal{C}^*$ from $\mathcal{P}$, $\mathcal{Q}$, and hypothetical correspondences $\overline{\mathcal{C}}$, a simpler task of *robust rotation search* [16,63,77,85].

The second step ARCS+_O of ARCS+ is to *remove outliers* from the previous step 1. To do so we approximately maximize an appropriate consensus over $\text{SO}(3)$ (§4.2). Instead of mining inliers in $\text{SO}(3)$ [10,34,43,50,64], we show that the parameter space of consensus maximization can be reduced from $\text{SO}(3)$ to $\mathbb{S}^2$ and further to $[0, \pi]$ (see [16] for a different reduction). With this reduction, ARCS+_O removes outliers via repeatedly solving in $O(\ell \log \ell)$ time a computational geometry problem, *interval stabbing* [24] (§4.2.1). Note that ARCS+_O only repeats for $s \approx 90$ times to reach satisfactory accuracy. Therefore, conceptually, for $\ell \geq 10^6$, it is 10^4 times faster than the most related outlier removal method GORE [16], which uses $O(\ell^2 \log \ell)$ time (Table 4).

The third and final step ARCS+_R of our ARCS+ pipeline is to accurately estimate the rotation, using the consensus set from the second step (§4.3). In short, ARCS+_R is a *Riemannian subgradient descent* method. Our novelty here is to descend in the space $\mathbb{S}^3$ of unit quaternions, not $\text{SO}(3)$ [14]. This allows us to derive, based on [53], that ARCS+_R converges linearly though locally to the ground-truth unit quaternion, thus obtaining the first to our knowledge convergence rate guarantee for robust rotation search.

Numerical highlights are in order (§5). ARCS+_O is an outlier pruning procedure for robust rotation search that can handle extremely small inlier ratios $k^*/\ell = 3000/10^7 = 0.03\%$ in 5 minutes; $\text{ARCS+}_\text{O} + \text{ARCS+}_\text{R}$, or ARCS+_OR for short, accurately solves the robust rotation search problem with $k^*/\ell = 10^3/10^6$ in 23 seconds (see Table 4). $\text{ARCS+}_\text{N} + \text{ARCS+}_\text{OR}$, that is ARCS+, solves Problem 1 with $m = 10^4, n = 8000, k^* = 2000$ in 90 seconds (see Figure 2). To the best of our knowledge, all these challenging cases have not been considered in prior works. In fact, as we will review soon (§2), applying state-of-the-art methods to those cases either gives wrong estimates of rotations, or takes too much time ($\geq$ 8 hours), or exhausts the memory (Table 4).

[1] We run experiments on an Intel(R) i7-1165G7, 16GB laptop. In the paper we consider random instead of adversarial outliers.

2. Prior Art: Accuracy Versus Scalability

Early efforts on Problem 1 have encountered an *accuracy versus scalability* dilemma. The now classic ICP algorithm [13] estimates the rotation and correspondences in an alternating fashion, running in real time but requiring a high-quality and typically unavailable initialization to avoid local and usually poor minima; the same is true for its successors [20,23,41,59,66]. The GO-ICP method [81,82] of the branch & bound type enumerates initializations fed to ICP to reach a global minimum—in exponential time; the same running time bound is true for its successors [18,55,64].

The above ICP versus GO-ICP dilemma was somewhat alleviated by a two-step procedure: i) compute a candidate correspondence set $\hat{\mathcal{C}}$, via hand-crafted [67] or learned [30] feature descriptors, and ii) estimate the rotation from point sets indexed by $\hat{\mathcal{C}}$. But, as observed in [80], due to the quality of the feature descriptors, there could be fewer than 2 inliers remaining in $\hat{\mathcal{C}}$, from which the ground-truth rotation can never be determined. An alternative and more conservative idea is to use *all-to-all* correspondences $\hat{\mathcal{C}} := [m] \times [n]$, although now the inlier ratio becomes extremely small.

This justifies why researchers have recently focused on designing robust rotation search algorithms for extreme outlier rates, *e.g.*, $\geq$ 90 outliers out of 100. One such design is GORE [16], a guaranteed outlier removal algorithm of $O(\ell^2 \log \ell)$ time complexity that heavily exploits the geometry of $\text{SO}(3)$. The other one is the semidefinite relaxation QUASAR of [77], which involves sophisticated manipulation on unit quaternions; $\ell \approx 1000$ constitutes the current limit on the number of points this relaxation can handle. Yet another one is TEASER++ [80]; its robustness to outliers comes mainly from finding via parallel branch & bound [65] a maximum clique of the graph whose vertices represent point pairs and whose edges indicate whether two point pairs can simultaneously be inliers. This maximum clique formulation was also explored by [62] where it was solved via a different branch & bound algorithm. Since finding a maximum clique is in general NP-hard, their algorithms take exponential time in the worst case; in addition, TEASER++ was implemented to trade $O(\ell^2)$ space for speed. One should also note though that if noise is small then the graph is sparse so that the otherwise intractable branch & bound algorithm can be efficient. Since constructing such a graph entails checking $\binom{\ell}{2}$ point pairs, recent follow-up works [51, 56, 69, 71, 72] that use such a graph entail $O(\ell^2)$ time complexity. While all these methods are more accurate than scalable, the following two are on the other side. FGR [85] combines *graduated non-convexity*

(GNC) and alternating minimization, while GNC-TLS [76] combines *truncated least squares*, iteratively reweighted least-squares, and GNC. Both of them scale gracefully with ℓ, while being robust against up to $80/100 = 80\%$ outliers.

Is such accuracy versus scalability dilemma of an inherent nature of the problems here, or can we escape from it?

3. ARCS: Accuracy & Scalability

Basic Idea. Although perhaps not explicitly mentioned in the literature, it should be known that there is a simple algorithm that solves Problem 1 under the assumptions of Theorem 1. This algorithm first computes the ℓ_2 norm of each point in $\mathcal{Q}$ and $\mathcal{P}$ and the difference $d_{i,j} := \|\boldsymbol{q}_i\|_2 - \|\boldsymbol{p}_j\|_2$. Since $\mathcal{Q}$ and $\mathcal{P}$ are in general position (Remark 1), we have that $(\boldsymbol{q}_i, \boldsymbol{p}_j)$ is an inlier pair if and only if $d_{i,j} = 0$. Based on the $d_{i,j}$'s, extract all such inlier pairs. Since $k^* \geq 2$, and by the general position assumption (Remark 1), there exist two inlier pairs say $(\boldsymbol{q}_1, \boldsymbol{p}_1), (\boldsymbol{q}_2, \boldsymbol{p}_2)$ such that $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ are not parallel. As a result and as it has been well-known since the 1980's [3, 35, 36, 57], if not even earlier [68, 75], $\boldsymbol{R}^*$ can be determined from the two inlier pairs by SVD.

ARCS: Efficient Implementation. Not all the $d_{i,j}$'s should be computed in order to find the correspondence set $\mathcal{C}^*$, meaning that the otherwise $O(mn)$ time complexity can be reduced. Our ARCS Algorithm 1 seeks all point pairs $(\boldsymbol{q}_i, \boldsymbol{p}_j)$'s whose norms are close, *i.e.*, they satisfy $|d_{i,j}| \leq c$, for some sufficiently small $c \geq 0$. Here c is provided as an input of ARCS and set as 0 in the current context. It is clear that, under the general position assumption of Theorem 1, the set $\overline{\mathcal{C}}$ returned by ARCS is exactly the ground-truth correspondence set $\mathcal{C}^*$. It is also clear that ARCS takes $O(m \log m)$ time and $O(m)$ space (recall $m \geq n \geq |\mathcal{C}^*|$).

We proved Theorem 1. It is operating in the noiseless case that allows us to show that Problem 1 can be solved accurately and at large scale. Indeed, ARCS can handle more than 10^6 points with $k^* = 2$ in about 0.1 seconds, even though generating those points has taken more than 0.2 seconds, as shown in Table 1.[2] Note that in the setting of Table 1 we have only $k^* = 2$ overlapping points, a situation where all prior methods mentioned in §1 and §2, if directly applicable, in principle break down. One reason is that they are not designed to handle the noiseless case. The other reason is that the overlapping ratio k^*/m of Table 1 is the minimum possible. While the achievement in Table 1 is currently limited to the noiseless case, it forms a strong motivation that urges us to robustify ARCS to noise, while keeping as much of its accuracy and scalability as possible. Such robustification is the main theme of the next section.

[2]For experiments in Tables 1 and 2 we generate data as per Section 5.1.

Algorithm 1: ARCS

Input: $\mathcal{Q} = \{\boldsymbol{q}_i\}_{i=1}^m, \mathcal{P} = \{\boldsymbol{p}_j\}_{j=1}^n, c \geq 0$;
Sort $\mathcal{Q}$ so that (w.l.o.g.) $\|\boldsymbol{q}_1\|_2 \leq \cdots \leq \|\boldsymbol{q}_m\|_2$;
Sort $\mathcal{P}$ so that (w.l.o.g.) $\|\boldsymbol{p}_1\|_2 \leq \cdots \leq \|\boldsymbol{p}_n\|_2$;
$i = 1; j = 1; \overline{\mathcal{C}} = \varnothing$;
while $i \leq m$ *and* $j \leq n$ **do**
  $d_{i,j} \leftarrow \|\boldsymbol{q}_i\|_2 - \|\boldsymbol{p}_j\|_2$;
  if $d_{i,j} > c$ **then**
    $j \leftarrow j + 1$;
  end
  if $d_{i,j} < -c$ **then**
    $i \leftarrow i + 1$;
  end
  if $-c \leq d_{i,j} \leq c$ **then**
    $\overline{\mathcal{C}} \leftarrow \overline{\mathcal{C}} \cup (i,j)$; $(i,j) \leftarrow (i+1, j+1)$;
  end
end
return $\overline{\mathcal{C}}$;

Table 1. Time (msec) of generating noiseless Gaussian point sets (G) and solving Problem 1 by ARCS (100 trials, $k^* = 2$).

m	10^4	10^5	10^6
n	8×10^3	8×10^4	8×10^5
G	5.9	15.0	212.8
Brute Force	73.8	8304	8380441.5
ARCS	1.51	8.4	121.1

4. ARCS+: Robustifying ARCS to Noise

Here we consider Problem 1 with noise $\boldsymbol{\epsilon}_{i,j}$. We will illustrate our algorithmic ideas by assuming $\boldsymbol{\epsilon}_{i,j} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_3)$, although this is not necessary for actual implementation. As indicated in §1, ARCS+ has three steps. We introduce them respectively in the next three subsections.

4.1. Step 1: Finding Correspondences Under Noise

A simple probability fact is $\|\boldsymbol{q}_i - \boldsymbol{R}^* \boldsymbol{p}_j\|_2 \leq 5.54\sigma$ for any inlier $(\boldsymbol{q}_i, \boldsymbol{p}_j)$, so $|d_{i,j}| \leq 5.54\sigma$ with probability at least $1 - 10^{-6}$ (see, *e.g.*, [80]). To establish correspondences under noise, we need to modify[3] the while loop of Algorithm 1, such that, in $O(\ell + m \log m)$ time, it returns the set $\overline{\mathcal{C}}$ of all correspondences of size ℓ where each $(i,j) \in \overline{\mathcal{C}}$ satisfies $|d_{i,j}| \leq c$, with c now set to 5.54σ. Note that, to store the output correspondences, we need an extra $O(\ell)$ time, which can not be simply ignored as ℓ is in general larger than m in the presence of noise (Table 2). We call this modified version ARCS+$_\text{N}$. ARCS+$_\text{N}$ gives a set $\overline{\mathcal{C}}$ that

[3]The details of this modification can be found at: https://github.com/liangzu/ARCS/blob/main/ARCSplus_N.m

Table 2. The number ℓ of candidate correspondences produced by $\texttt{ARCS+}_{\texttt{N}}$ on synthetic noisy Gaussian point sets. A single trial.

m	1000	5000	10000
n	800	4000	8000
k^*	200	1000	2000
ℓ	36622	931208	3762888
$\ell/(mn)$	4.58%	4.66%	4.70%

contains all inlier correspondences $\mathcal{C}^*$ with probability at least $(1-10^{-6})^{k^*}$. This probability is larger than 99.9% if $k^* \leq 10^3$, or larger than 99% if $k^* \leq 10^4$.

Remark 2 (*feature matching versus all-to-all correspondences versus* $\texttt{ARCS+}_{\texttt{N}}$). Feature matching methods provide fewer than n hypothetical correspondences and thus speed up the subsequent computation, but they might give no inliers. Using all-to-all correspondences preserves all inliers, but a naive computation needs $O(mn)$ time and leads to a large-scale problem with extreme outlier rates. $\texttt{ARCS+}_{\texttt{N}}$ strikes a balance by delivering in $O(\ell + m\log m)$ time a candidate correspondence set $\overline{\mathcal{C}}$ of size ℓ containing all inliers with high probability and with $\ell \ll mn$.

For illustration, Table 2 reports the number ℓ of correspondences that $\texttt{ARCS+}_{\texttt{N}}$ typically yields. As shown, even though $\ell/(mn)$ is usually smaller than 5%, yet ℓ itself could be very large, and the inlier ratio k^*/ℓ is extremely small (*e.g.*, $\leq 0.05\%$). This is perhaps the best we could do for the current stage, because for now we only considered every point pair individually, while any pair $(\boldsymbol{q}_i, \boldsymbol{p}_i)$ is a potential inlier if it satisfies the necessary (but no longer sufficient) condition $|d_{i,j}| \leq c$. On the other hand, collectively analyzing the remaining point pairs allows to further remove outliers, and this is the major task of our next stage (§4.2).

4.2. Step 2: Outlier Removal

Let there be some correspondences given, by, *e.g.*, either $\texttt{ARCS+}_{\texttt{N}}$ or feature matching (cf. Remark 2). Then we arrive at an important special case of Problem 1, called *robust rotation search*. For convenience we formalize it below:

Problem 2. (*robust rotation search*) Consider ℓ pairs of 3D points $\{(\boldsymbol{y}_i, \boldsymbol{x}_i)\}_{i=1}^{\ell}$, with each pair satisfying

$$\boldsymbol{y}_i = \boldsymbol{R}^* \boldsymbol{x}_i + \boldsymbol{o}_i + \boldsymbol{\epsilon}_i. \tag{2}$$

Here $\boldsymbol{\epsilon}_i \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_3)$ is noise, $\boldsymbol{o}_i = \boldsymbol{0}$ if $i \in \mathcal{I}^*$ where $\mathcal{I}^* \subset [\ell]$ is of size k^*, and if $i \notin \mathcal{I}^*$ then $\boldsymbol{o}_i$ is nonzero and arbitrary. The task is to find $\boldsymbol{R}^*$ and $\mathcal{I}^*$.

The percentage of outliers in Problem 2 can be quite large (cf. Table 2), so our second step $\texttt{ARCS+}_{\texttt{O}}$ here is to remove outliers. In §4.2.1, we shortly review the interval stabbing problem, on which $\texttt{ARCS+}_{\texttt{O}}$ of §4.2.2 is based.

4.2.1 Preliminaries: Interval Stabbing

Consider a collection of subsets of $\mathbb{R}$, $\{\mathcal{J}_i\}_{i=1}^L$, where each $\mathcal{J}_i$ is an interval of the form $[a, b]$. In the interval stabbing problem, one needs to determine a point $\omega \in \mathbb{R}$ and a subset $\mathcal{I}$ of $\{\mathcal{J}_i\}_{i=1}^L$, so that $\mathcal{I}$ is a maximal subset whose intervals overlap at ω. Formally, we need to solve

$$\begin{aligned} \max_{\mathcal{I} \subset [L], \omega \in \mathbb{R}} \quad & |\mathcal{I}| \\ \text{s.t.} \quad & \omega \in \mathcal{J}_i, \ \ \forall i \in \mathcal{I} \end{aligned} \tag{3}$$

For this purpose, the following result is known.

Lemma 1 (*interval stabbing*)**.** *Problem* (3) *can be solved in* $O(L\log L)$ *time and* $O(L)$ *space.*

Actually, the interval stabbing problem can be solved using sophisticated data structures such as *interval tree* [24] or *interval skip list* [32]. On the other hand, it is a basic exercise to find an algorithm that solves Problem (3), which, though also in $O(L\log L)$ time, involves only a sorting operation and a for loop (details are omitted, see, *e.g.*, [17]). Finally, note that the use of interval stabbing for robust rotation search is not novel, and can be found in $\texttt{GORE}$ [16, 63]. However, as the reader might realize after §4.2.2, our use of interval stabbing is quite different from $\texttt{GORE}$.

4.2.2 The Outlier Removal Algorithm

We now consider the following consensus maximization:

$$\begin{aligned} \max_{\mathcal{I} \subset [\ell], \boldsymbol{R} \in \mathrm{SO}(3)} \quad & |\mathcal{I}| \\ \text{s.t.} \quad & \left\| \boldsymbol{y}_i - \boldsymbol{R}\boldsymbol{x}_i \right\|_2 \leq c, \ \ \forall i \in \mathcal{I}. \end{aligned} \tag{4}$$

It has been shown in [73] that for the very related *robust fitting* problem, such consensus maximization is in general NP-hard[4]. Thus it seems only prudent to switch our computational goal from solving (4) exactly to approximately.

From $\mathrm{SO}(3)$ **to** $\mathbb{S}^2$**.** Towards this goal, we first shift our attention to $\mathbb{S}^2$ where the rotation axis $\boldsymbol{b}^*$ of $\boldsymbol{R}^*$ lives. An interesting observation is that the axis $\boldsymbol{b}^*$ has the following interplay with data, independent of the rotation angle of $\boldsymbol{R}^*$.

Proposition 1. *Let* $\boldsymbol{v}_i := \boldsymbol{y}_i - \boldsymbol{x}_i$*. Recall* $\boldsymbol{\epsilon}_i \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_3)$*. If* $(\boldsymbol{y}_i, \boldsymbol{x}_i)$ *is an inlier pair, then* $\boldsymbol{v}_i^\top \boldsymbol{b}^* \sim \mathcal{N}(0, \sigma^2)$*, and so* $|\boldsymbol{v}_i^\top \boldsymbol{b}^*| \leq 4.9\sigma$ *with probability at least* $1 - 10^{-6}$*.*

Proposition 1 (cf. Appendix C) leads us to Problem (5):

$$\begin{aligned} \max_{\mathcal{I} \subset [\ell], \boldsymbol{b} \in \mathbb{S}^2} \quad & |\mathcal{I}| \\ \text{s.t.} \quad & |\boldsymbol{v}_i^\top \boldsymbol{b}| \leq \bar{c}, \ \ \forall i \in \mathcal{I} \\ & b_2 \geq 0. \end{aligned} \tag{5}$$

[4]Interestingly, consensus maximization over $\mathrm{SO}(2)$, *i.e.*, the 2D version of (4), can be solved in $O(\ell\log\ell)$ time; see [17].

In (5) the constraint on the second entry b_2 of $\boldsymbol{b}$ is to eliminate the symmetry, and Proposition 1 suggests to set $\bar{c} := 4.9\sigma$. Problem (5) is easier than (4) as it has fewer degrees of freedom; see also [16] where a different reduction to a 2 DoF (sub-)problem was derived for GORE.

Solving (5) is expected to yield an accurate estimate of $\boldsymbol{b}^*$, from which the rotation angle can later be estimated. Problem (5) reads: find a plane (defined by the normal $\boldsymbol{b}$) that approximately contains as much points $\boldsymbol{v}_i$'s as possible. This is an instance of the *robust subspace learning* problem [25, 26, 48, 74, 83, 86, 87], for which various scalable algorithms with strong theoretical guarantees have been developed in more tractable formulations (*e.g.*, ℓ_1 minimization) than consensus maximization. Most notably, the so-called *dual principal component pursuit* formulation [74] was proved in [87] to be able to tolerate $O\big((k^*)^2\big)$ outliers. Still, all these methods can not handle as many outliers as we currently have (cf. Table 2), even though they can often minimize their objective functions to global optimality.

From $\mathbb{S}^2$ to $[0,\pi]$. We can further "reduce" the degrees of freedom in (5) by 1, through the following lens. Certainly $\boldsymbol{b} \in \mathbb{S}^2$ in (5) is determined by two angles $\theta \in [0,\pi]$, $\phi \in [0,\pi]$. Now consider the following problem:

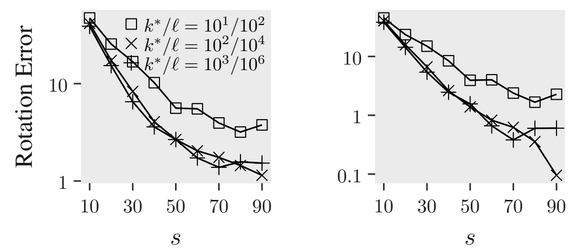
$$\begin{aligned} \max_{\mathcal{I}\subset[\ell],\theta\in[0,\pi]} \quad & |\mathcal{I}| \\ \text{s.t.} \quad & |\boldsymbol{v}_i^\top \boldsymbol{b}| \le \bar{c}, \;\; \forall i \in \mathcal{I} \\ & \boldsymbol{b} = [\sin(\theta)\cos(\phi),\ \sin(\theta)\sin(\phi),\ \cos(\theta)]^\top. \end{aligned} \tag{6}$$

Problem (6) is a simplified version of (5) with ϕ given. Clearly, to solve (5) it suffices to minimize the function $f : [0,\pi] \to \mathbb{R}$ which maps any $\phi_0 \in [0,\pi]$ to the objective value of (6) with $\phi = \phi_0$. Moreover, we have:

Proposition 2. *Problem (6) can be solved in $O(\ell \log \ell)$ time and $O(\ell)$ space via interval stabbing.*

Proposition 2 gives an $O(\ell \log \ell)$ time oracle to access the values of f. Since computing the objective value of (5) given θ, ϕ already needs $O(\ell)$ time, the extra cost of the logarithmic factor in Proposition 2 is nearly negligible. Since f has only one degree of freedom, its global minimizer can be found by *one-dimensional branch & bound* [42]. But this entails exponential time complexity in the worst case, a situation we wish to sidestep. Alternatively, the search space $[0,\pi]$ is now so small that the following algorithm ARCS+$_\text{O}$ turns out to be surprisingly efficient and robust: i) *sampling* from $[0,\pi]$, ii) *stabbing* in $\mathbb{S}^2$, and iii) *stabbing* in SO(3).

Sampling from $[0,\pi]$. Take s equally spaced points $\phi_j = (2j-1)\pi/(2s), \forall j \in [s]$, on $[0,\pi]$. The reader may find this choice of ϕ_j's similar to the *uniform grid approach* [60]; in the latter Nesterov commented that "the reason why it works here is related to the *dimension* of the problem".



(a) ARCS+$_\text{O}$ (b) ARCS+$_\text{OR}$

Figure 1. Rotation errors (in degrees) of steps 2 and 3 for robust rotation search methods with s varying (500 trials, $\sigma = 0.01$).

Stabbing in $\mathbb{S}^2$. For each $j \in [s]$, solve (6) with $\phi = \phi_j$ to get s candidate consensus set $\mathcal{I}_j$'s and s angles θ_j's. From each ϕ_j and θ_j we obtain a candidate rotation axis $\boldsymbol{b}_j$.

Stabbing in SO(3). Since now we have estimates of rotation axes, $\boldsymbol{b}_j$'s, there is one degree of freedom remaining, the rotation angle ω. For this we consider:

$$\begin{aligned} \max_{\mathcal{I}\subset[\ell],\omega\in[0,2\pi]} \quad & |\mathcal{I}| \\ \text{s.t.} \quad & \big\|\boldsymbol{y}_i - \boldsymbol{R}\boldsymbol{x}_i\big\|_2 \le c, \;\; \forall i \in \mathcal{I} \\ & \boldsymbol{R} = \boldsymbol{b}\boldsymbol{b}^\top + [\,\boldsymbol{b}\,]_\times \sin(\omega) + (\boldsymbol{I}_3 - \boldsymbol{b}\boldsymbol{b}^\top)\cos(\omega) \end{aligned} \tag{7}$$

Here $[\,\boldsymbol{b}\,]_\times \in \mathbb{R}^{3\times 3}$ denotes the matrix generating the cross product $\times$ by $\boldsymbol{b}$, that is $[\,\boldsymbol{b}\,]_\times \boldsymbol{a} = \boldsymbol{b} \times \boldsymbol{a}$ for all $\boldsymbol{a} \in \mathbb{R}^3$. Similarly to Proposition 2, we have the following result:

Proposition 3. *Problem (7) can be solved in $O(\ell \log \ell)$ time and $O(\ell)$ space via interval stabbing.*

After solving (7) with $\boldsymbol{b} = \boldsymbol{b}_j$ for each $j \in [s]$, we obtain s candidate consensus sets $\tilde{\mathcal{I}}_1, \dots, \tilde{\mathcal{I}}_s$, and we choose the one with maximal cardinality as an approximate solution to (4). Finally, notice that the time complexity $O(s\ell \log \ell)$ of ARCS+$_\text{O}$ depends on the hyper-parameter s. We set $s = 90$ as an invariant choice, as suggested by Figure 1.

This output consensus set $\tilde{\mathcal{I}}$ typically has very few outliers; see Table 3. Thus it will be used next in ARCS+$_\text{R}$, our final step for accurately estimating the rotation (§4.3).

Table 3. The output of ARCS+$_\text{O}$ with inputs from Table 2.

Input Inlier Ratio	$\frac{200}{36622}$	$\frac{1000}{931208}$	$\frac{2000}{3762888}$
Output Inlier Ratio	$\frac{199}{213}$	$\frac{993}{1314}$	$\frac{1951}{3184}$

4.3. Step 3: Rotation Estimation

The final step ARCS+$_\text{R}$ of ARCS+ is a refinement procedure that performs robust rotation search on the output correspondences $\tilde{\mathcal{I}}$ of ARCS+$_\text{O}$. Since $\tilde{\mathcal{I}}$ contains much fewer outlier correspondences than we previously had (cf. Table 2 and 3), in what follows we simplify the notations by focusing on the point set $\{(\boldsymbol{y}_i, \boldsymbol{x}_i)\}_{i\in[\ell]}$, which we assume has few outliers (say $\leq 50\%$). Then, a natural formulation is

$$\min_{\boldsymbol{R}\in\mathrm{SO}(3)} \sum_{i=1}^{\ell} \left\|\boldsymbol{y}_i - \boldsymbol{R}\boldsymbol{x}_i\right\|_2. \tag{8}$$

Problem (8) appears easier to solve than consensus maximization (4), as it has a convex objective function at least. Next we present the $\texttt{ARCS+}_{\texttt{R}}$ algorithm and its theory.

Algorithm. We start with the following equivalence.

Proposition 4. *We have* $\boldsymbol{w}^\top \boldsymbol{D}_i \boldsymbol{w} = \left\|\boldsymbol{y}_i - \boldsymbol{R}\boldsymbol{x}_i\right\|_2^2$, *where* $\boldsymbol{w} \in \mathbb{S}^3$ *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*

$$\min_{\boldsymbol{w}\in\mathbb{S}^3} h(\boldsymbol{w}), \quad h(\boldsymbol{w}) = \sum_{i=1}^{\ell} \sqrt{\boldsymbol{w}^\top \boldsymbol{D}_i \boldsymbol{w}}. \tag{9}$$

The exact relation between unit quaternions and rotations is reviewed in Appendix A, where Proposition 4 is proved and the expression of $\boldsymbol{D}_i$ is given. For what follows, it suffices to know that a unit quaternion is simply a unit vector of $\mathbb{R}^4$, and that the space of unit quaternions is $\mathbb{S}^3$.

Note that the objective h of (9) is convex, while both problems (8) and (9) are nonconvex (due to the constraint) and nonsmooth (due to the objective). Though (8) and (9) are equivalent, the advantage of (9) will manifest itself soon. Before that, we first introduce the $\texttt{ARCS+}_{\texttt{R}}$ algorithm for solving (9). $\texttt{ARCS+}_{\texttt{R}}$ falls into the general *Riemannian subgradient descent* framework (see, *e.g.*, [53]). It is initialized at some unit quaternion $\boldsymbol{w}^{(0)} \in \mathbb{S}^3$ and proceeds by

$$\boldsymbol{w}^{(t+1)} \leftarrow \mathrm{Proj}_{\mathbb{S}^3}\left(\boldsymbol{w}^{(t)} - \gamma^{(t)}\,\tilde{\nabla}_{\mathrm{s}}\, h(\boldsymbol{w}^{(t)})\right), \tag{10}$$

where $\mathrm{Proj}_{\mathbb{S}^3}(\cdot)$ projects a vector onto $\mathbb{S}^3$, $\gamma^{(t)}$ is some stepsize, $\tilde{\nabla}_{\mathrm{s}}\, h(\boldsymbol{w}^{(t)})$ is a *Riemannian subgradient*[5] of h at $\boldsymbol{w}^{(t)}$.

Theory. Now we are able to compare (8) and (9) from a theoretical perspective. As proved in [14], for any fixed outlier ratio and $k^* > 0$, *Riemannian subgradient descent* when applied to (8) with proper initialization converges to $\boldsymbol{R}^*$ in finite time, as long as i) ℓ is sufficiently large, ii) all points $\boldsymbol{y}_i$'s and $\boldsymbol{x}_i$'s are uniformly distributed on $\mathbb{S}^2$, iii) there is no noise. But in [14] no convergence rate is given. One main challenge of establishing convergence rates there is that projecting on SO(3) does not enjoy a certain kind of *nonexpansiveness* property, which is important for convergence analysis (cf. Lemma 1 of [53]). On the other hand, projection onto $\mathbb{S}^3$ of (9) does satisfy such property. As a result, we are able to provide convergence rate guarantees for $\texttt{ARCS+}_{\texttt{R}}$. For example, it follows directly from Theorem 2 of [53] that $\texttt{ARCS+}_{\texttt{R}}$ (10) converges to an ε-*stationary* point in $O(\varepsilon^{-4})$ iterations, even if initialized arbitrarily.

We next give conditions for $\texttt{ARCS+}_{\texttt{R}}$ to converge linearly to the ground-truth unit quaternion $\pm\boldsymbol{w}^*$ that represents $\boldsymbol{R}^*$. Let the distance between a unit quaternion $\boldsymbol{w}$ and $\pm\boldsymbol{w}^*$ be

$$\mathrm{dist}(\boldsymbol{w}, \pm\boldsymbol{w}^*) := \min\left\{\left\|\boldsymbol{w} - \boldsymbol{w}^*\right\|_2, \left\|\boldsymbol{w} + \boldsymbol{w}^*\right\|_2\right\}.$$

If $\mathrm{dist}(\boldsymbol{w}, \pm\boldsymbol{w}^*) < \rho$ with $\rho > 0$ then $\boldsymbol{w}$ is called ρ-*close* to $\pm\boldsymbol{w}^*$. We need the following notion of *sharpness*.

Definition 1 (*sharpness* [15, 44, 49, 53])**.** We say that $\pm\boldsymbol{w}^*$ is an α-*sharp minimum* of (9) if $\alpha > 0$ and if there exists a number $\rho_\alpha > 0$ such that any unit quaternion $\boldsymbol{w} \in \mathbb{S}^3$ that is ρ_α-*close* to $\pm\boldsymbol{w}^*$ satisfies the inequality

$$h(\boldsymbol{w}) - h(\boldsymbol{w}^*) \geq \alpha\,\mathrm{dist}(\boldsymbol{w}, \pm\boldsymbol{w}^*). \tag{11}$$

We provide a condition below for $\pm\boldsymbol{w}^*$ to be α^*-sharp:

Proposition 5. *If* $\alpha^* := k^*\eta_{\min}/\sqrt{2} - (\ell - k^*)\eta_{\max} > 0$ *and if* $\boldsymbol{\epsilon}_i = \mathbf{0}$ *in Problem* 2*, then Problem* (9) *admits* $\pm w^*$ *as an* α^**-sharp minimum. Here* $\eta_{\min}$, $\eta_{\max}$ *are respectively*

$$\eta_{\min} := \frac{1}{k^*} \min_{\boldsymbol{w}\in\mathcal{S}^*\cap\mathbb{S}^3} \sum_{i\in\mathcal{I}^*} \sqrt{\boldsymbol{w}^\top \boldsymbol{D}_i \boldsymbol{w}}, \text{ and} \tag{12}$$

$$\eta_{\max} := \frac{1}{\ell - k^*} \max_{\boldsymbol{w}\in\mathbb{S}^3} \sum_{i\in[\ell]\setminus\mathcal{I}^*} \sqrt{\boldsymbol{w}^\top \boldsymbol{D}_i \boldsymbol{w}}, \tag{13}$$

where $\mathcal{S}^*$ *is the hyperplane of* $\mathbb{R}^4$ *perpendicular to* $\pm\boldsymbol{w}^*$.

Proposition 5 is proved in Appendix B.1. The condition $\alpha^* > 0$ defines a relation between the number of inliers (k^*) and outliers $(\ell - k^*)$, and involves two quantities $\eta_{\min}$ and $\eta_{\max}$ whose values depend on how $\boldsymbol{D}_i$'s are distributed on the positive semi-definite cone. We offer probabilistic interpretations for $\eta_{\min}$ and $\eta_{\max}$ in Appendix B.2.

With Theorem 4 of [53] and Proposition 5 we have that $\texttt{ARCS+}_{\texttt{R}}$ (10), if initialized properly and with suitable stepsizes, converges linearly to the ground-truth unit quaternion $\pm\boldsymbol{w}^*$, as long as $\pm\boldsymbol{w}^*$ is α^*-sharp. A formal statement is:

Theorem 2. *Suppose* $\alpha^* := k^*\eta_{\min}/\sqrt{2} - (\ell - k^*)\eta_{\max} > 0$*. Let* L_h *be a Lipschitz constant of* h*. Run Riemannian subgradient descent* $\texttt{ARCS+}_{\texttt{R}}$ (10) *with initialization* $\boldsymbol{w}^{(0)}$ *satisfying* $\mathrm{dist}(\boldsymbol{w}^{(0)}, \pm\boldsymbol{w}^*) \leq \min\{\alpha^*/L_h, \rho_{\alpha^*}\}$ *and with geometrically diminishing stepsizes* $\gamma^{(t)} = \beta^t\gamma^{(0)}$*, where*

$$\gamma^{(0)} < \min\left\{\frac{2e_0(\alpha^* - L_h e_0)}{L_h^2}, \frac{e_0}{2(\alpha^* - L_h e_0)}\right\},$$

$$\beta^2 \in \left[1 + 2\Big(L_h - \frac{\alpha^*}{e_0}\Big)\gamma^{(0)} + \frac{L_h^2(\gamma^{(0)})^2}{e_0^2},\ 1\right),$$

$$e_0 = \min\left\{\max\left\{\mathrm{dist}(\boldsymbol{w}^{(0)}, \pm\boldsymbol{w}^*), \frac{\alpha^*}{2L_h}\right\}, \rho_{\alpha^*}\right\}.$$

[5]We follow [53] where a Riemannian subgradient $\tilde{\nabla}_{\mathrm{s}}\, h(\boldsymbol{w})$ at $\boldsymbol{w} \in \mathbb{S}^3$ is defined as the projection of some subgradient $\nabla_{\mathrm{s}}\, h(\boldsymbol{w})$ of h at $\boldsymbol{w}$ onto the tangent space of $\mathbb{S}^3$ at $\boldsymbol{w}$, *i.e.*, $\tilde{\nabla}_{\mathrm{s}}\, h(\boldsymbol{w}) := (\boldsymbol{I}_4 - \boldsymbol{w}\boldsymbol{w}^\top)\nabla_{\mathrm{s}}\, h(\boldsymbol{w})$. See [11] for how to compute a subgradient of some given function.

Table 4. Average errors in degrees | standard deviation | running times in seconds of various algorithms on synthetic data (20 trials).

Inlier Ratio $\frac{k^*}{\ell}$	$\frac{10^3}{10^5} = 1\%$	$\frac{10^3}{10^6} = 0.1\%$	$\frac{3\times10^3}{5\times10^6} = 0.06\%$	$\frac{3\times10^3}{10^7} = 0.03\%$	$\frac{10^3}{10^7} = 0.01\%$
`TEASER++` [80]	out-of-memory				
`RANSAC`	0.39 | 0.20 | 29.1	$\geq$ 8.4 hours			
`GORE` [16, 63]	3.43 | 2.10 | 1698	$\geq$ 12 hours			
`FGR` [85]	52.2 | 68.5 | 3.64	95.0 | 60.9 | 37.7	84.9 | 59.4 | 145	86.5 | 56.9 | 311	97.3 | 61.3 | 314
`GNC-TLS` [76]	3.86 | 9.51 | 0.13	63.4 | 50.5 | 2.26	49.9 | 31.1 | 15.9	90.2 | 45.6 | 40.1	120 | 34.3 | 36.3
`ARCS+`$_\text{R}$	9.92 | 13.1 | 0.12	65.2 | 48.9 | 0.96	55.6 | 38.3 | 5.58	88.4 | 36.2 | 12.6	98.2 | 36.0 | 12.2
`ARCS+`$_\text{O}$	0.86 | 0.29 | 1.71	0.99 | 0.37 | 23.2	0.91 | 0.30 | 125	0.98 | 0.42 | 287	55.6 | 60.9 | 281
`ARCS+`$_\text{OR}$	0.03 | 0.03 | 1.72	0.09 | 0.07 | 23.2	0.11 | 0.07 | 125	0.22 | 0.15 | 287	55.4 | 60.1 | 281

In the noiseless case ($\epsilon_i = 0$) we have each $\boldsymbol{w}^{(t)}$ satisfying

$$\operatorname{dist}(\boldsymbol{w}^{(t)}, \pm\boldsymbol{w}^*) \leq \beta^t e_0. \tag{14}$$

Remark 3 (*a posteriori optimality guarantees*). Theorem 2 endows `ARCS+`$_\text{R}$ (10) with convergence guarantee. On the other hand, *a posteriori* optimality guarantees can be obtained via semidefinite certification [5, 19, 78, 80].

5. Experiments

In this section we evaluate `ARCS+` via synthetic and real experiments for Problem 1, simultaneous rotation and correspondence search. We also evaluate its components, namely `ARCS+`$_\text{O}$ (§4.2) and `ARCS+`$_\text{R}$ (§4.3) for Problem 2, robust rotation search, as it is a task of independent interest. For both of the two problems we compare the following state-of-the-art methods (reviewed in §2): `FGR` [85], `GORE` [16], `RANSAC`, `GNC-TLS` [76], and `TEASER++` [80].

5.1. Experiments on Synthetic Point Clouds

Setup. We set $\sigma = 0.01$, $\bar{c} = c = 5.54\sigma$, $n = \lfloor 0.8m \rfloor$, and $s = 90$ unless otherwise specified. For all other methods we used default or otherwise appropriate parameters. We implemented `ARCS+` in MATLAB. No parallelization was explicitly used and no special care was taken for speed.

Robust Rotation Search. From $\mathcal{N}(0, \boldsymbol{I}_3)$ we randomly sampled point pairs $\{(\boldsymbol{y}_i, \boldsymbol{x}_i)\}_{i=1}^{\ell}$ with k^* inliers and noise $\boldsymbol{\epsilon}_i \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_3)$. Specifically, we generated the ground-truth rotation $\boldsymbol{R}^*$ from an axis randomly sampled from $\mathbb{S}^2$ and an angle from $[0, 2\pi]$, rotated k^* points randomly sampled from $\mathcal{N}(0, \boldsymbol{I}_3)$ by $\boldsymbol{R}^*$, and added noise to obtain k^* inlier pairs. Every outlier point $\boldsymbol{y}_j$ or $\boldsymbol{x}_j$ was randomly sampled from $\mathcal{N}(0, \boldsymbol{I}_3)$ with the constraint $-c \leq \|\boldsymbol{y}_j\|_2 - \|\boldsymbol{x}_j\|_2 \leq c$; otherwise $(\boldsymbol{y}_j, \boldsymbol{x}_j)$ might simply be detected and removed by computing $\|\boldsymbol{y}_j\|_2 - \|\boldsymbol{x}_j\|_2$.

We compared `ARCS+`$_\text{O}$ and `ARCS+`$_\text{R}$ and their combination `ARCS+`$_\text{OR}$ with prior works. The results are in Table 4. We first numerically illustrate the *accuracy versus scalability* dilemma in prior works (§2). On the one hand, we observed an extreme where *accuracy overcomes scalability*: `RANSAC` performed well with error 0.39 when $k^*/\ell = 10^3/10^5$, but its running time increased greatly with decreasing inlier ratio, from 29 seconds to more than 8.4 hours. The other extreme is where *scalability overcomes accuracy*: Both `GNC-TLS` and `FGR` failed in presence of such many outliers—as expected—even though their running time scales linearly with ℓ.

Table 4 also depicted the performance of our proposals `ARCS+`$_\text{O}$ and `ARCS+`$_\text{R}$. Our approximate consensus strategy `ARCS+`$_\text{O}$ reached a balance between accuracy and scalability. In terms of accuracy, it made errors smaller than 1 degree, as long as there are more than $3 \times 10^3/10^7 = 0.03\%$ inliers; this was further refined by Riemannian subgradient descent `ARCS+`$_\text{R}$, so that their combination `ARCS+`$_\text{OR}$ had even lower errors. In terms of scalability, we observed that `ARCS+`$_\text{OR}$ is uniformly faster than `FGR`, and is at least 1800 times faster than `GORE` for $k^*/\ell = 10^3/10^6 = 0.1\%$. But it had been harder to measure exactly how faster `ARCS+`$_\text{OR}$ is than `GORE` and `RANSAC` for even larger point sets. Finally, `ARCS+`$_\text{OR}$ failed at $k^*/\ell = 10^3/10^7 = 0.01\%$.

Simultaneous Rotation and Correspondence Search. We randomly sampled point sets $\mathcal{Q}$ and $\mathcal{P}$ from $\mathcal{N}(0, \boldsymbol{I}_3)$ with k^* inlier pairs and noise $\boldsymbol{\epsilon}_{i,j} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_3)$ (cf. Problem 1). Each outlier point was randomly and independently drawn also from $\mathcal{N}(0, \boldsymbol{I}_3)$. Figure 2 shows that `ARCS+` accurately estimated the rotations for $k^* \geq 2000$ (in 90 seconds), and broke down at $k^* = 1000$, a situation where there were $k^*/m = 10\%$ overlapping points. We did not compare methods like `TEASER++`, `GORE`, `RANSAC` here, because giving them correspondences from `ARCS+`$_\text{N}$ would result unsatisfactory running time or accuracy (recall Tables 2 and 4), while feature matching methods like `FPFH` do not perform well on random synthetic data.

5.2. Experiments on 3DMatch

The 3DMatch[6] dataset [84] contains more than 1000 point clouds for testing, representing 8 different scenes

[6]License info: https://3dmatch.cs.princeton.edu/

Table 5. Success rates of methods run on the scene pairs of the 3DMatch dataset [84] for which the ground-truth rotation and translation are provided (rotation error smaller than 10 degree means a success [80]; see also the first paragraph of Appendix E).

Scene Type # Scene Pairs	Kitchen 506	Home 1 156	Home 2 208	Hotel 1 226	Hotel 2 104	Hotel 3 54	Study Room 292	MIT Lab 77	Overall 1623
TEASER++	**99.0%**	**98.1%**	94.7%	98.7%	**99.0%**	98.1%	97.0%	94.8%	97.72%
$\texttt{ARCS++}_{\texttt{OR}}$	98.4%	97.4%	**95.7%**	98.7%	98.1%	**100%**	**97.3%**	**96.1%**	97.72%

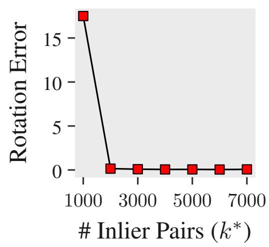


Figure 2. Rotation errors of ARCS+ on synthetic Gaussian point clouds. 20 trials, $m = 10^4$, $n = \lfloor 0.8m \rfloor$, $\sigma = 0.01$.

(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 [84] there are 5000 keypoints for each cloud. We used the pretrained model[7] of the 3DSmoothNet [30] to extract descriptors from these key points, and matched them using the Matlab function pcmatchfeatures, with its parameter MatchThreshold set to the maximum 1. We assume that the ground-truth translation $\boldsymbol{t}^*$ is given, and run TEASER++ and $\texttt{ARCS+}_{\texttt{OR}}$ on $(\boldsymbol{y}_i - \boldsymbol{t}^*, \boldsymbol{x}_i)$'s; the performance is comparable. We did not compare other methods here, as TEASER++ currently has the best performance on 3DMatch (to the best of our knowledge); see [80] for comparison with optimization-based methods, and see [22] for the success rates (recall) of deep learning methods.

See the supplementary materials for more experiments.

6. Discussion and Future Work

Despite of the progress that we made for robust rotation search and simultaneous rotation and correspondence search on large-scale point clouds, our ARCS+ pipeline has a few limitations, and we discuss them next.

For small datasets (*e.g.*, $\ell \leq 500$), as in homography fitting [16], other methods, *e.g.*, MAGSAC++ [6–8], VSAC [40], TEASER++ [80], and GORE [16] might be considered with higher priority; they come with efficient C++ implementations. For more points, *e.g.*, $\ell \geq 10^4$, but with higher inlier rates than in Table 4 (*e.g.*, $\geq 15\%$), GNC-TLS [76] and RANSAC are our recommendations for what to use.

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

Modern point clouds have more than 10^5 points, and are naturally correspondences-less (cf. [17]). ARCS operates at that scale in the absence of noise (Table 1), while ARCS+ can handle $m, n \approx 10000$ (Figure 2) and $\texttt{ARCS+}_{\texttt{OR}}$ can handle $\ell \approx 10^7$ correspondences (Table 4); all these are limited to the rotation-only case. To find rotation (and translation) from such point sets "in the wild", it seems inevitable to downsample them. An interesting future work is to theoretically quantify the tradeoff between downsampling factors and the final registration performance. Another tradeoff to quantify, as implied by Remark 2, is this: Can we design a correspondence matching algorithm that better balances the number of remaining points and the number of remaining inliers? In particular, such matching should take specific pose into consideration (cf. ARCS); many methods did not.

Like TEASER++, GORE, GNC-TLS, RANSAC, our algorithm relies on an inlier threshold c. While how to set this hyper-parameter suitably is known for Gaussian noise with given variance, *in practice the distance threshold is usually chosen empirically*, as Hartley & Zisserman wrote [33]. While mis-specification of c could fail the registration, certain heuristics have been developed to alleviate the sensitivity to such mis-specification; see [2, 6–8]. Finally, our experience is to set c based on the scale of the point clouds.

Our outlier removal component $\texttt{ARCS+}_{\texttt{O}}$ presented good performance (Table 3), yet with no optimality guarantees. Note that, with $s = 90$ we have $|\phi_j - \phi^*| \leq 1$ for some ϕ_j, while Figure 1a shows that $\texttt{ARCS+}_{\texttt{O}}$ gave roughly 1 degree error at $s = 90$. Theoretically justifying this is left as future work. Without guarantees, registration could fail, which might lead to undesired consequences in safety-critical applications. On the other hand, we believe that ARCS+ is a good demonstration of trading optimality guarantees for accuracy and scalability; enforcing all of the three properties amounts to requiring solving NP hard problems efficiently at large scale! In fact, since any solutions might get certified for optimality (Remark 3), bold algorithmic design ideas can be taken towards improving accuracy and scalability, while relying on other tools for optimality certification.

Acknowledgments. The first author was supported by the MINDS PhD fellowship at Johns Hopkins University. This work was supported by NSF Grants 1704458 and 1934979, and by the Northrop Grumman Mission Systems Research in Applications for Learning Machines (REALM) initiative.

References

[1] Pasquale Antonante, Vasileios Tzoumas, Heng Yang, and Luca Carlone. Outlier-robust estimation: Hardness, minimally-tuned algorithms, and applications. Technical report, arXiv:2007.15109v2 [cs.CV], 2020. 1

[2] Pasquale Antonante, Vasileios Tzoumas, Heng Yang, and Luca Carlone. Outlier-robust estimation: Hardness, minimally tuned algorithms, and applications. *IEEE Transactions on Robotics*, 2021. 8

[3] K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-squares fitting of two 3D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (5):698–700, 1987. 3

[4] Xuyang Bai, Zixin Luo, Lei Zhou, Hongkai Chen, Lei Li, Zeyu Hu, Hongbo Fu, and Chiew-Lan Tai. Pointdsc: Robust point cloud registration using deep spatial consistency. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 15859–15869, 2021. 1

[5] Afonso S Bandeira. A note on probably certifiably correct algorithms. *Comptes Rendus Mathematique*, 354(3):329–333, 2016. 7

[6] 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. 8

[7] 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. 8

[8] Daniel Barath, Jana Noskova, and Jiri Matas. Marginalizing sample consensus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 8

[9] Dominik Bauer, Timothy Patten, and Markus Vincze. Reagent: Point cloud registration using imitation and reinforcement learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 14586–14594, 2021. 1

[10] Jean-Charles Bazin, Yongduek Seo, and Marc Pollefeys. Globally optimal consensus set maximization through rotation search. In *Asian Conference on Computer Vision*, pages 539–551, 2012. 2

[11] Amir Beck. *First-Order Methods in Optimization*. Society for Industrial and Applied Mathematics, 2017. 6

[12] Lukas Bernreiter, Lionel Ott, Juan Nieto, Roland Siegwart, and Cesar Cadena. PHASER: A robust and correspondence-free global pointcloud registration. *IEEE Robotics and Automation Letters*, 6(2):855–862, 2021. 1

[13] PJ Besl and Neil D McKay. A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. 1, 2

[14] 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. 2, 6

[15] James V Burke and Michael C Ferris. Weak sharp minima in mathematical programming. *SIAM Journal on Control and Optimization*, 31(5):1340–1359, 1993. 6

[16] Álvaro Parra Bustos and Tat-Jun Chin. Guaranteed outlier removal for rotation search. In *IEEE International Conference on Computer Vision*, pages 2165–2173, 2015. 1, 2, 4, 5, 7, 8

[17] Zhipeng Cai, Tat-Jun Chin, Alvaro Parra Bustos, and Konrad Schindler. Practical optimal registration of terrestrial lidar scan pairs. *ISPRS Journal of Photogrammetry and Remote Sensing*, 147:118–131, 2019. 4, 8

[18] Dylan Campbell and Lars Petersson. GOGMA: Globally-optimal gaussian mixture alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1, 2

[19] Luca Carlone, Giuseppe C. Calafiore, Carlo Tommolillo, and Frank Dellaert. Planar pose graph optimization: Duality, optimal solutions, and verification. *IEEE Transactions on Robotics*, 32(3):545–565, 2016. 7

[20] Dmitry Chetverikov, Dmitry Svirko, Dmitry Stepanov, and Pavel Krsek. The trimmed iterative closest point algorithm. In *Object recognition supported by user interaction for service robots*, volume 3, pages 545–548, 2002. 1, 2

[21] Tat-Jun Chin, Yang Heng Kee, Anders Eriksson, and Frank Neumann. Guaranteed outlier removal with mixed integer linear programs. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5858–5866, 2016. 1

[22] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 8

[23] Haili Chui and Anand Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2-3):114–141, 2003. 1, 2

[24] Mark De Berg, Marc Van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry*. Springer, 1997. 2, 4

[25] 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. 5

[26] Tianyu Ding, Zhihui Zhu, René Vidal, and Daniel P Robinson. Dual principal component pursuit for robust subspace learning: Theory and algorithms for a holistic approach. In *International Conference on Machine Learning*, pages 2739–2748, 2021. 5

[27] Richard Everson. Orthogonal, but not orthonormal, Procrustes problems. *Advances in Computational Mathematics*, 3(4), 1998. 1

[28] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 1

[29] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):930–943, 2003. 2

[30] 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. 2, 8

[31] John C Gower, Garmt B Dijksterhuis, et al. *Procrustes Problems*, volume 30. Oxford University Press on Demand, 2004. 1

[32] Eric N. Hanson. The interval skip list: A data structure for finding all intervals that overlap a point. In *Algorithms and Data Structures*, pages 153–164, 1991. 4

[33] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. 8

[34] Richard I Hartley and Fredrik Kahl. Global optimization through rotation space search. *International Journal of Computer Vision*, 82(1):64–79, 2009. 1, 2

[35] Berthold KP Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987. 3

[36] Berthold KP Horn, Hugh M Hilden, and Shahriar Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A*, 5(7):1127–1135, 1988. 3

[37] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. PREDATOR: Registration of 3D point clouds with low overlap. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4267–4276, 2021. 1

[38] Xiaoshui Huang, Guofeng Mei, Jian Zhang, and Rana Abbas. A comprehensive survey on point cloud registration. Technical report, arXiv:2103.02690v2 [cs.CV], 2021. 1

[39] Jose Pedro Iglesias, Carl Olsson, and Fredrik Kahl. Global optimality for point set registration using semidefinite programming. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1

[40] 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. 8

[41] Bing Jian and Baba C. Vemuri. Robust point set registration using gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1633–1645, 2011. 1, 2

[42] Yanmei Jiao, Yue Wang, Bo Fu, Qimeng Tan, Lei Chen, Minhang Wang, Shoudong Huang, and Rong Xiong. Globally optimal consensus maximization for robust visual inertial localization in point and line map. In *International Conference on Intelligent Robots and Systems*, pages 4631–4638, 2020. 5

[43] Kyungdon Joo, Hongdong Li, Tae-Hyun Oh, and In So Kweon. Robust and efficient estimation of relative pose for cameras on selfie sticks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2

[44] Mohammad Mahdi Karkhaneei and Nezam Mahdavi-Amiri. Nonconvex weak sharp minima on Riemannian manifolds. *Journal of Optimization Theory and Applications*, 183(1):85–104, 2019. 6

[45] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Automatic generator of minimal problem solvers. In *European Conference on Computer Vision*, pages 302–315, 2008. 2

[46] Viktor Larsson, Kalle Astrom, and Magnus Oskarsson. Efficient solvers for minimal problems by syzygy-based reduction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2383–2392, 2017. 2

[47] Viktor Larsson, Magnus Oskarsson, Kalle Astrom, Alge Wallis, Zuzana Kukelova, and Tomas Pajdla. Beyond grobner bases: Basis selection for minimal solvers. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[48] Gilad Lerman and Tyler Maunu. An overview of robust subspace recovery. *Proceedings of the IEEE*, 106(8):1380–1410, 2018. 5

[49] Chong Li, Boris S. Mordukhovich, Jinhua Wang, and Jen-Chih Yao. Weak sharp minima on Riemannian manifolds. *SIAM Journal on Optimization*, 21(4):1523–1560, 2011. 6

[50] Hongdong Li and Richard Hartley. The 3D-3D registration problem revisited. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007. 1, 2

[51] Jiayuan Li, Qingwu Hu, and Mingyao Ai. GESAC: Robust graph enhanced sample consensus for point cloud registration. *ISPRS Journal of Photogrammetry and Remote Sensing*, 167:363–374, 2020. 1, 2

[52] Jiayuan Li, Qingwu Hu, and Mingyao Ai. Point cloud registration based on one-point RANSAC and scale-annealing biweight estimation. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–14, 2021. 1

[53] 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, 6

[54] Wei Lian, Lei Zhang, and Ming-Hsuan Yang. An efficient globally optimal algorithm for asymmetric point matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1281–1293, 2017. 1

[55] 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. 1, 2

[56] Parker C. Lusk, Kaveh Fathian, and Jonathan P. How. CLIPPER: A graph-theoretic framework for robust data association. Technical report, arXiv:2011.10202v2 [cs.RO], 2021. 2

[57] F Landis Markley. Attitude determination using vector observations and the singular value decomposition. *Journal of the Astronautical Sciences*, 36(3):245–258, 1988. 3

[58] Haggai Maron, Nadav Dym, Itay Kezurer, Shahar Kovalsky, and Yaron Lipman. Point registration via efficient convex relaxation. *ACM Transactions on Graphics*, 35(4), 2016. 1

[59] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, 2010. 1, 2

[60] Yurii Nesterov. *Lectures on Convex Optimization*. Springer, 2018. 5

[61] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770, 2004. 2

[62] 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. 1, 2

[63] Álvaro Parra Bustos and Tat-Jun Chin. Guaranteed outlier removal for point cloud registration with correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2868–2882, 2018. 1, 2, 4, 7

[64] Álvaro Parra Bustos, Tat-Jun Chin, Anders Eriksson, Hongdong Li, and David Suter. Fast rotation search with stereographic projections for 3D registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2227–2240, 2016. 1, 2

[65] Ryan A. Rossi, David F. Gleich, and Assefaw H. Gebremedhin. Parallel maximum clique algorithms with applications to network analysis. *SIAM Journal on Scientific Computing*, 37(5):C589–C616, 2015. 2

[66] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the ICP algorithm. In *International Conference on 3D Digital Imaging and Modeling*, pages 145–152. IEEE, 2001. 1, 2

[67] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (FPFH) for 3D registration. In *IEEE International Conference on Robotics and Automation*, pages 3212–3217, 2009. 2

[68] Peter H Schönemann. A generalized solution of the orthogonal Procrustes problem. *Psychometrika*, 31(1):1–10, 1966. 3

[69] 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. 1, 2

[70] Julian Straub, Trevor Campbell, Jonathan P How, and John W Fisher. Efficient global point cloud alignment using bayesian nonparametric mixtures. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2941–2950, 2017. 1

[71] Lei Sun. IRON: Invariant-based highly robust point cloud registration. Technical report, arXiv:2103.04357v2 [cs.CV], 2021. 1, 2

[72] Lei Sun. RANSIC: Fast and highly robust estimation for rotation search and point cloud registration using invariant compatibility. Technical report, arXiv:2104.09133v3 [cs.CV], 2021. 1, 2

[73] Chin Tat-Jun, Cai Zhipeng, and Frank Neumann. Robust fitting in computer vision: Easy or hard? *International Journal of Computer Vision*, 128(3):575–587, 2020. 4

[74] Manolis C. Tsakiris and René Vidal. Dual principal component pursuit. *Journal of Machine Learning Research*, 19(18):1–50, 2018. 5

[75] Grace Wahba. A least squares estimate of satellite attitude. *SIAM Review*, 7(3):409, 1965. 1, 3

[76] 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. 1, 3, 7, 8

[77] 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. 1, 2

[78] Heng Yang and Luca Carlone. One ring to rule them all: Certifiably robust geometric perception with outliers. In *Advances in Neural Information Processing Systems*, 2020. 7

[79] Heng Yang, Ling Liang, Kim-Chuan Toh, and Luca Carlone. STRIDE along spectrahedral vertices for solving large-scale rank-one semidefinite relaxations. Technical report, arXiv:2105.14033 [math.OC], 2021. 1

[80] Heng Yang, Jingnan Shi, and Luca Carlone. TEASER: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 37(2):314–333, 2021. 1, 2, 3, 7, 8

[81] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Go-ICP: A globally optimal solution to 3D ICP point-set registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2241–2254, 2016. 1, 2

[82] Jiaolong Yang, Hongdong Li, and Yunde Jia. Go-ICP: Solving 3D registration efficiently and globally optimally. In *IEEE International Conference on Computer Vision*, pages 1457–1464, 2013. 2

[83] Yunzhen Yao, Liangzu Peng, and Manolis Tsakiris. Unlabeled principal component analysis. *Advances in Neural Information Processing Systems*, 2021. 5

[84] 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. 7, 8

[85] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *European Conference on Computer Vision*, pages 766–782, 2016. 1, 2, 7

[86] Zhihui Zhu, Tianyu Ding, Daniel Robinson, Manolis Tsakiris, and René Vidal. A linearly convergent method for non-smooth non-convex optimization on the grassmannian with applications to robust subspace and dictionary learning. In *Advances in Neural Information Processing Systems*, 2019. 5

[87] 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. 5