Guaranteed Outlier Removal for Rotation Search

Álvaro Parra Bustos   Tat-Jun Chin

School of Computer Science, The University of Adelaide, South Australia

{aparra, tjchin}@cs.adelaide.edu.au

Abstract

Rotation search has become a core routine for solving many computer vision problems. The aim is to rotationally align two input point sets with correspondences. Recently, there is significant interest in developing globally optimal rotation search algorithms. A notable weakness of global algorithms, however, is their relatively high computational cost, especially on large problem sizes and data with a high proportion of outliers. In this paper, we propose a novel outlier removal technique for rotation search. Our method guarantees that any correspondence it discards as an outlier does not exist in the inlier set of the globally optimal rotation for the original data. Based on simple geometric operations, our algorithm is deterministic and fast. Used as a preprocessor to prune a large portion of the outliers from the input data, our method enables substantial speed-up of rotation search algorithms without compromising global optimality. We demonstrate the efficacy of our method in various synthetic and real data experiments[1].

1. Introduction

Given two point sets $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ and $\mathcal{Y} = \{\mathbf{y}_i\}_{i=1}^N$ in 3D, we aim to find the 3D rotation $\mathbf{R}$ that aligns them, i.e., such that $\mathbf{R}\mathbf{x}_i \approx \mathbf{y}_i$ for all i. Here, each $(\mathbf{x}_i, \mathbf{y}_i)$ is a pair of matching points. If there are no false matching points or outliers, the best rotation in the least squares sense can be obtained analytically [10, 1]. Otherwise, we seek the rotation that agrees with as many of the pairs as possible

$$\begin{aligned} \underset{\mathbf{R},\, \mathcal{I} \subseteq \mathcal{H}}{\text{maximize}} \quad & |\mathcal{I}| \\ \text{subject to} \quad & \angle(\mathbf{R}\mathbf{x}_i, \mathbf{y}_i) \le \epsilon,\ \forall i \in \mathcal{I}, \end{aligned} \qquad (1)$$

where agreement is up to the inlier threshold ϵ. Here, $\mathcal{H} = \{1, \ldots, N\}$ indexes the set of all point matches, and $\angle(\cdot, \cdot)$ denotes the angular distance. The optimal $\mathbf{R}^*$ is consistent with the largest possible subset $\mathcal{I}^* \subseteq \mathcal{H}$ of the data. Note that given $\mathcal{I}^*$ we can easily find $\mathbf{R}^*$ and vice versa, thus, we may quote $\mathcal{I}^*$ or $\mathbf{R}^*$ as the solution without ambiguity.

RANSAC [6] can be applied to approximately solve (1). Candidate rotations are hypothesized from randomly sampled minimal subsets of two point matches [10] and evaluated. Although RANSAC is very efficient, in general it does not provide the optimal solution $\mathcal{I}^*$. Formally, let $\tilde{\mathcal{I}} \subseteq \mathcal{H}$ be the result of RANSAC. We have that $|\tilde{\mathcal{I}}| \le |\mathcal{I}^*|$, and in general $\tilde{\mathcal{I}} \not\subseteq \mathcal{I}^*$, i.e., genuine inliers may be discarded.

Hartley and Kahl [7] pioneered branch-and-bound (BnB) as a viable technique for rotation search. BnB systematically partitions and prunes the rotation space until the solution is found. Their algorithm was extended to include a robust formulation [2] such as the one we use in (1). Unlike RANSAC, BnB is guaranteed to find the globally optimal result. The solution of many computer vision problems have benefited from BnB rotation search as a subroutine [5, 7, 15, 9, 13]; such as essential matrix and camera pose estimation, hand-eye calibration, panoramic image stitching, and point cloud registration.

Another class of global algorithms [12, 4] leverage on the fact that the solution to (1) is equal to the solution of the same problem on a subset of $\mathcal{H}$ of size at most d, where d is 3 for rotation search. The result $\mathbf{R}^*$ is found by enumerating all $\binom{N}{p}$ subsets of $\mathcal{H}$ for all $p \le d$ and solving each subset for $\mathbf{R}$ analytically (note that this differs from "standard" RANSAC which solves for $\mathbf{R}$ via least squares on subsets of size two). These algorithms have been demonstrated successfully on similar applications.

A general weakness of global algorithms, however, is their high computational cost, especially for data with large sizes N and high outlier contamination rates. In the case of [4], the number of unique subsets to test is enormous even for moderate N (e.g., for $N = 500$ there are ≥ 20 million 3-subsets). An outlier rate in excess of 95% is also frequently encountered in practice, e.g., in point cloud registration where 3D keypoint detection and matching techniques [17, 14, 18] are much less accurate than their 2D counterparts such as SIFT and SURF. These factors lead to significant runtimes of BnB rotation search.

Our contribution is a novel *guaranteed* outlier removal technique for rotation search. Specifically, our method is able to reduce $\mathcal{H}$ to a subset $\mathcal{H}'$ of point matches, in a way

---

[1]Implementation is provided in the supplementary material.

that any $(\mathbf{x}_i, \mathbf{y}_i)$ discarded by reducing $\mathcal{H}$ to $\mathcal{H}'$ is a genuine outlier, i.e., any $(\mathbf{x}_i, \mathbf{y}_i)$ that is removed does not belong to $\mathcal{I}^*$. More formally, our method ensures that $\mathcal{I}^* \subseteq \mathcal{H}' \subseteq \mathcal{H}$, which is a result RANSAC cannot guarantee.

We pose our technique as an efficient *preprocessor* to the rotation search problem (1). Based on simple geometric operations, our method is deterministic and fast. By aggressively reducing the population of true outliers (almost 90% can be eliminated), our method significantly accelerates the global algorithms. For example, using our method before BnB reduces the overall runtime by an order of magnitude. Note that the global solution to the reduced data $\mathcal{H}'$ equals the global solution $\mathcal{I}^*$ to the original $\mathcal{H}$.

Our work is closest in spirit to Svärm et al. [16], who proposed a technique for camera localization from 2D-3D correspondences. In their work, the usage of gravitational sensors reduces camera localization to a 3DOF problem (2D translation and 1D rotation). Their approach also conducts a guaranteed outlier rejection scheme for the 2D-3D point matches, before a globally optimal algorithm is invoked. Since our target problem (3DOF rotation search) differs from Svärm et al.'s, the core geometric motivations and operations of the two works are vastly different.

2. Guaranteed outlier removal

Using the angular distance renders the norm of the points irrelevant. Henceforth, we take all the points to have unit norm. The rotation search problem (1) can be rewritten as

$$\underset{k \in \mathcal{H}}{\text{maximize}} \ f_k, \qquad (2)$$

where f_k is defined as the maximum objective value of the subproblem P_k, with $k = 1, \ldots, N$:

$$\begin{aligned} \underset{\mathbf{R}_k, \, \mathcal{I}_k \subseteq \mathcal{H} \backslash \{k\}}{\text{maximize}} \quad & |\mathcal{I}_k| + 1 \\ \text{subject to} \quad & \angle(\mathbf{R}_k \mathbf{x}_i, \mathbf{y}_i) \le \epsilon, \ \forall i \in \mathcal{I}_k, \qquad (P_k) \\ & \angle(\mathbf{R}_k \mathbf{x}_k, \mathbf{y}_k) \le \epsilon. \end{aligned}$$

In words, P_k seeks the rotation $\mathbf{R}_k$ that agrees with as many of the data as possible, given that $\mathbf{R}_k$ *must align* $(\mathbf{x}_k, \mathbf{y}_k)$. Our reformulation (2) does not make the original problem (1) any easier - its utility derives from clarifying how an upper bound on f_k allows to identify outliers.

Let $l \le |\mathcal{I}^*|$ be a lower bound for the solution of the rotation search problem (1). Our outlier removal technique depends on the ability to calculate an upper bound $\hat{f}_k$ for the result of each P_k, i.e., $\hat{f}_k \ge f_k$. Given the lower and upper bound values, the following result can be established.

Proposition 1 *If $\hat{f}_k < l$, then $(\mathbf{x}_k, \mathbf{y}_k)$ is a true outlier, i.e., k does not exist in the solution $\mathcal{I}^*$ to (1).*

Proof The proof is by contradiction. If k is in $\mathcal{I}^*$, then we must have that $f_k = |\mathcal{I}^*|$. However, if we are given that $\hat{f}_k < l$, then $f_k < l \le |\mathcal{I}^*|$, which contradicts the previous condition. Hence, k cannot exist in $\mathcal{I}^*$. $\square$

Our main algorithm (Sec. 4) applies Proposition 1 iteratively for $k = 1, \ldots, N$ to remove outliers. Our main contribution is an efficient algorithm to calculate a tight upper bound $\hat{f}_k$ for P_k (Sec. 3) for each k. As a by-product, our upper bound algorithm also computes a tight lower bound l for (1) to enable efficient removal of true outliers.

3. Efficient algorithm for upper bound

Recall that any candidate rotation $\mathbf{R}_k$ to solve P_k must bring $\mathbf{x}_k$ within angular distance ϵ from $\mathbf{y}_k$, i.e.,

$$\angle(\mathbf{R}_k \mathbf{x}_k, \mathbf{y}_k) \le \epsilon. \qquad (3)$$

We interpret $\mathbf{R}_k$ by decomposing it into two rotations

$$\mathbf{R}_k = \mathbf{A}\mathbf{B} \qquad (4)$$

where we define $\mathbf{B}$ as a rotation that honors the condition

$$\angle(\mathbf{B}\mathbf{x}_k, \mathbf{y}_k) \le \epsilon, \qquad (5)$$

and $\mathbf{A}$ as a rotation about axis $\mathbf{B}\mathbf{x}_k$. Since $\mathbf{A}$ leaves $\mathbf{B}\mathbf{x}_k$ unchanged, the condition (5) and hence constraint (3) are always satisfied. Fig. 1(a) illustrates this interpretation.

Solving P_k thus amounts to finding the combination of the rotation $\mathbf{B}$ (a 2DOF problem, given (5)) and the rotation angle of $\mathbf{A}$ (a total of 3DOF) that maximize the objective.

3.1. The ideal case

In the absence of noise and outliers, $\mathbf{x}_i$ can be aligned exactly with $\mathbf{y}_i$ for all i. Based on (4), we denote the rotation that solves P_k under this ideal case as

$$\hat{\mathbf{R}}_k = \hat{\mathbf{A}}\hat{\mathbf{B}}, \qquad (6)$$

which can be solved as follows (refer also to Fig. 1(a)). First, find a rotation $\hat{\mathbf{B}}$ that aligns $\mathbf{x}_k$ exactly with $\mathbf{y}_k$, i.e.,

$$\hat{\mathbf{B}}\mathbf{x}_k = \mathbf{y}_k. \qquad (7)$$

For example, take $\hat{\mathbf{B}}$ as the rotation that maps $\mathbf{x}_k$ to $\mathbf{y}_k$ with the minimum geodesic motion. To solve for $\hat{\mathbf{A}}$, take any $i \ne k$, then find the angle $\hat{\theta}$ of rotation about axis $\hat{\mathbf{B}}\mathbf{x}_k$ that maps $\hat{\mathbf{B}}\mathbf{x}_i$ to $\mathbf{y}_i$. Then $\hat{\mathbf{A}} = \exp(\hat{\theta}\hat{\mathbf{B}}\mathbf{x}_k)$, where $\exp(\cdot)$ is the exponential map as defined in [8, Eq. (3)]. The above steps affirm that rotation estimation requires a minimum of two point matches [10].
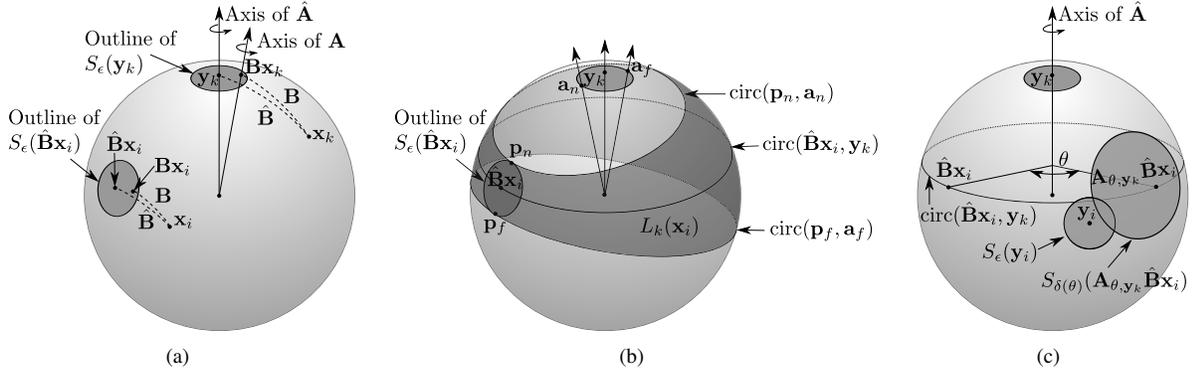
Figure 1. (a) Interpreting rotation $\mathbf{R}_k$ according to (4). (b) The uncertainty region $L_k(\mathbf{x}_i)$ (15). (c) This figure shows $S_{\delta(\theta)}(\mathbf{A}_{\theta,\mathbf{y}_k}\hat{\mathbf{B}}\mathbf{x}_i)$ intersecting with $S_\epsilon(\mathbf{y}_i)$ for a particular θ. We wish to find a bounding interval $\Theta_i \subset [-\pi, \pi]$ on θ for which the intersection is non-empty.

3.2. Uncertainty bound

In the usual case, we must contend with noise and outliers. The aim of this section is to establish a bound on the position of $\mathbf{x}_i$ when acted upon by the set of feasible rotations $\mathbf{R}_k$, i.e., those that satisfy (3) for P_k.

The set of $\mathbf{B}$ that maintain (5) cause $\mathbf{B}\mathbf{x}_k$ to lie within a spherical region of angular radius ϵ centered at $\mathbf{y}_k$, i.e.,

$$\mathbf{B}\mathbf{x}_k \in S_\epsilon(\mathbf{y}_k), \tag{8}$$

where $S_\epsilon(\mathbf{y}_k) := \{\mathbf{x} \mid \angle(\mathbf{x}, \mathbf{y}_k) \leq \epsilon\}$ and $\|\mathbf{x}\| = 1$. (9)

Since $\mathbf{B}\mathbf{x}_k$ is the rotation axis of $\mathbf{A}$, the interior of $S_\epsilon(\mathbf{y}_k)$ also represents the set of possible rotation axes for $\mathbf{A}$. Further, for any $i \neq k$, we can establish

$$\angle(\mathbf{B}\mathbf{x}_i, \hat{\mathbf{B}}\mathbf{x}_i) = \angle(\mathbf{B}\mathbf{x}_k, \hat{\mathbf{B}}\mathbf{x}_k) \tag{10}$$
$$= \angle(\mathbf{B}\mathbf{x}_k, \mathbf{y}_k) \leq \epsilon, \tag{11}$$

where (10) is based on the fact that applying the same pair of rotations on different points will transport the points across the same angular distance. Hence, (11) also shows that the set of feasible $\mathbf{B}$ cause $\mathbf{B}\mathbf{x}_i$ to lie in a spherical region, i.e.,

$$\mathbf{B}\mathbf{x}_i \in S_\epsilon(\hat{\mathbf{B}}\mathbf{x}_i). \tag{12}$$

Fig. 1(a) also shows $S_\epsilon(\mathbf{y}_k)$ and $S_\epsilon(\hat{\mathbf{B}}\mathbf{x}_i)$. The bound on $\mathbf{R}_k\mathbf{x}_i$ can thus be analysed based on these two regions.

To make explicit the dependence of $\mathbf{A}$ on a rotation axis $\mathbf{a}$ and angle θ, we now denote it as $\mathbf{A}_{\theta,\mathbf{a}}$, where

$$\mathbf{A}_{\theta,\mathbf{a}} = \exp(\theta\mathbf{a}). \tag{13}$$

Let $\mathbf{p}$ be an arbitrary unit-norm point. Define

$$\text{circ}(\mathbf{p}, \mathbf{a}) := \{\mathbf{A}_{\theta,\mathbf{a}}\mathbf{p} \mid \theta \in [-\pi, \pi]\} \tag{14}$$

as the circle traced by $\mathbf{p}$ when acted upon by rotation $\mathbf{A}_{\theta,\mathbf{a}}$ for all θ at a particular axis $\mathbf{a}$.

The set of possible positions of $\mathbf{R}_k\mathbf{x}_i$ is then defined by

$$L_k(\mathbf{x}_i) := \{\text{circ}(\mathbf{p}, \mathbf{a}) \mid \mathbf{p} \in S_\epsilon(\hat{\mathbf{B}}\mathbf{x}_i), \mathbf{a} \in S_\epsilon(\mathbf{y}_k)\}. \tag{15}$$

Fig. 1(b) illustrates this feasible region, which exists on the unit sphere. The region is bounded within the two circles

$$\text{circ}(\mathbf{p}_n, \mathbf{a}_n) \quad \text{and} \quad \text{circ}(\mathbf{p}_f, \mathbf{a}_f), \tag{16}$$

which are highlighted in Fig. 1(b). Intuitively, $\mathbf{p}_n$ and $\mathbf{a}_n$ (resp. $\mathbf{p}_f$ and $\mathbf{a}_f$) are the closest (resp. farthest) pair of points from $S_\epsilon(\hat{\mathbf{B}}\mathbf{x}_i)$ and $S_\epsilon(\mathbf{y}_k)$. Mathematically,

$$\mathbf{p}_n = \exp\left(\epsilon\hat{\mathbf{B}}\mathbf{x}_i \times \mathbf{y}_k / \|\hat{\mathbf{B}}\mathbf{x}_i \times \mathbf{y}_k\|\right)\hat{\mathbf{B}}\mathbf{x}_i; \tag{17}$$

$$\mathbf{a}_n = \exp\left(\epsilon\mathbf{y}_k \times \hat{\mathbf{B}}\mathbf{x}_i / \|\mathbf{y}_k \times \hat{\mathbf{B}}\mathbf{x}_i\|\right)\mathbf{y}_k; \tag{18}$$

$$\mathbf{p}_f = \exp\left(\epsilon\mathbf{y}_k \times \hat{\mathbf{B}}\mathbf{x}_i / \|\mathbf{y}_k \times \hat{\mathbf{B}}\mathbf{x}_i\|\right)\hat{\mathbf{B}}\mathbf{x}_i; \text{ and} \tag{19}$$

$$\mathbf{a}_f = \exp\left(\epsilon\hat{\mathbf{B}}\mathbf{x}_i \times \mathbf{y}_k / \|\hat{\mathbf{B}}\mathbf{x}_i \times \mathbf{y}_k\|\right)\mathbf{y}_k. \tag{20}$$

Note that if $\hat{\mathbf{B}}\mathbf{x}_i$ is antipodal to $\mathbf{y}_k$, the feasible region reduces to the spherical region $S_{3\epsilon}(\hat{\mathbf{B}}\mathbf{x}_i)$.

Result 1 *For any $i \neq k$, if $S_\epsilon(\mathbf{y}_i)$ does not intersect with $L_k(\mathbf{x}_i)$, then $(\mathbf{x}_i, \mathbf{y}_i)$ cannot be aligned by any rotation $\mathbf{R}_k$ that satisfies (3). The correspondence $(\mathbf{x}_i, \mathbf{y}_i)$ can then be safely removed without affecting the result f_k of P_k.*

3.3. Reducing the uncertainty

For each point match $(\mathbf{x}_i, \mathbf{y}_i)$ that survives the pruning by Result 1, we reduce its uncertainty bound (15) into an *angular interval*. This reduction is crucial for our efficient upper bound algorithm to be introduced in Sec. 3.4.

Consider rotating an arbitrary unit-norm point $\mathbf{p}$ with $\mathbf{A}_{\theta,\mathbf{u}}$ for a fixed angle θ and an axis $\mathbf{u} \in S_\epsilon(\mathbf{y}_k)$. We wish to bound the possible locations of $\mathbf{A}_{\theta,\mathbf{u}}\mathbf{p}$ given the uncertainty in $\mathbf{u}$. To this end, we establish

$$\max_{\mathbf{u} \in S_\epsilon(\mathbf{y}_k)} \angle(\mathbf{A}_{\theta,\mathbf{u}}\mathbf{p}, \mathbf{A}_{\theta,\mathbf{y}_k}\mathbf{p}) \leq \max_{\mathbf{u} \in S_\epsilon(\mathbf{y}_k)} \|\theta\mathbf{u} - \theta\mathbf{y}_k\|_2$$
$$= 2|\theta|\sin(\epsilon/2), \tag{21}$$
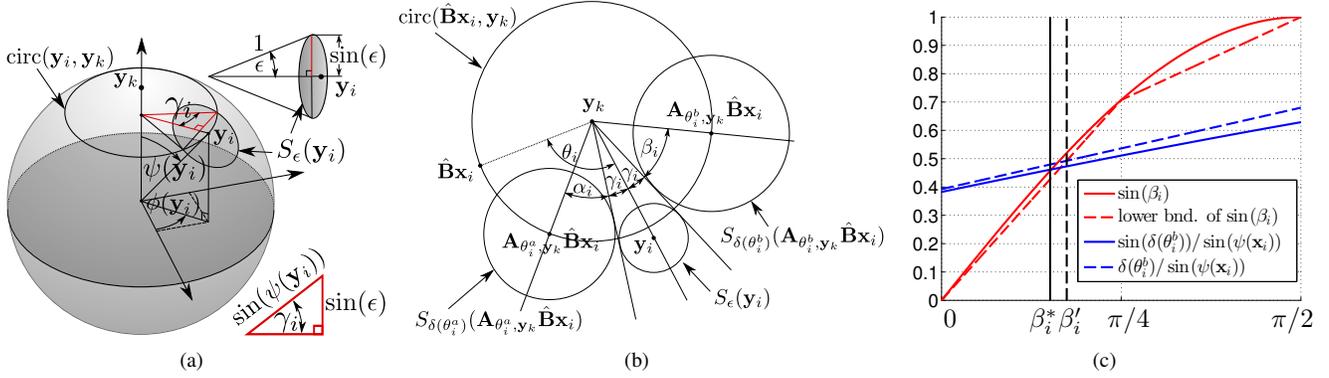
Figure 2. (a) Solving for γ_i in the red triangle. Its cathetus is half of the longest segment connecting points in $S_\epsilon(\mathbf{y}_i)$ and its hypotenuse is the radius of circ$(\mathbf{y}_i, \mathbf{y}_k)$. (b) To simplify the diagram and to aid intuition, the sphere in Fig. 1 is stereographically projected to the 2D plane using the North Pole $(\mathbf{y}_k)$ as the projection pole. Recall that the stereographic projection preserves circles [11], thus the shapes of all the circles and spherical regions on the sphere are preserved. Note that stereographic projection is only for presentation and is not required in practice. (c) Solving for β_i in (31) using the proposed linear approximations. Note that the obtained solution β_i' is always greater than the exact solution β_i^*, thus guaranteeing that the upper limit θ_i^b of Θ_i is a valid bound.

where the first line is based on a well-known result of the axis-angle representation (see [7, Lemma 2]), and the second line occurs since $S_\epsilon(\mathbf{y}_k)$ has an angular radius of ϵ.

Now we extend (21) to accommodate the uncertainty of $\mathbf{p}$ itself as a point from $S_\epsilon(\hat{\mathbf{B}}\mathbf{x}_i)$. We thus establish

$$\max_{\substack{\mathbf{p} \in S_\epsilon(\hat{\mathbf{B}}\mathbf{x}_i) \\ \mathbf{u} \in S_\epsilon(\mathbf{y}_k)}} \angle(\mathbf{A}_{\theta,\mathbf{u}}\mathbf{p}, \mathbf{A}_{\theta,\mathbf{y}_k}\hat{\mathbf{B}}\mathbf{x}_i)$$

$$\leq \max_{\substack{\mathbf{p} \in S_\epsilon(\hat{\mathbf{B}}\mathbf{x}_i) \\ \mathbf{u} \in S_\epsilon(\mathbf{y}_k)}} \angle(\mathbf{A}_{\theta,\mathbf{u}}\mathbf{p}, \mathbf{A}_{\theta,\mathbf{y}_k}\mathbf{p}) + \angle(\mathbf{A}_{\theta,\mathbf{y}_k}\mathbf{p}, \mathbf{A}_{\theta,\mathbf{y}_k}\hat{\mathbf{B}}\mathbf{x}_i)$$

$$\leq 2|\theta|\sin(\epsilon/2) + \epsilon. \tag{22}$$

The 2nd line is due to the triangle inequality, while the 3rd line applies (21) on the 1st term of the 2nd line. Define

$$\delta(\theta) = 2|\theta|\sin(\epsilon/2) + \epsilon. \tag{23}$$

The inequality (22) states that for a fixed θ and for all $\mathbf{u} \in S_\epsilon(\mathbf{y}_k)$ and $\mathbf{B}\mathbf{x}_i \in S_\epsilon(\hat{\mathbf{B}}\mathbf{x}_i)$, the point $\mathbf{A}_{\theta,\mathbf{u}}\mathbf{B}\mathbf{x}_i$ lies in

$$S_{\delta(\theta)}(\mathbf{A}_{\theta,\mathbf{y}_k}\hat{\mathbf{B}}\mathbf{x}_i). \tag{24}$$

Fig. 1(c) depicts this spherical region. Observe that for all $\theta \in [-\pi, \pi]$, the center of the region lies in circ$(\hat{\mathbf{B}}\mathbf{x}_i, \mathbf{y}_k)$. Intuitively, this is a circle of a fixed latitude on the globe when $\mathbf{y}_k$ is the "North Pole". Further, the spherical region attains the largest angular radius at $\theta = \pm\pi$.

For a pair $(\mathbf{x}_i, \mathbf{y}_i)$, we wish to obtain a bound Θ_i (an interval) on the range of θ that enable $\mathbf{A}_{\theta,\mathbf{u}}\mathbf{B}\mathbf{x}_i$ to align with $\mathbf{y}_i$, given the uncertainties $\mathbf{u} \in S_\epsilon(\mathbf{y}_k)$ and $\mathbf{B}\mathbf{x}_i \in S_\epsilon(\hat{\mathbf{B}}\mathbf{x}_i)$. This is analogous to seeking a bound on the θ that allows $S_{\delta(\theta)}(\mathbf{A}_{\theta,\mathbf{y}_k}\hat{\mathbf{B}}\mathbf{x}_i)$ to "touch" $S_\epsilon(\mathbf{y}_i)$; see Fig. 1(c).

Henceforth, concepts from the spherical coordinate system are used with reference to $\mathbf{y}_k$ as the North Pole.

3.3.1 Degenerate cases

If $\hat{\mathbf{B}}\mathbf{x}_i$ is close to $\mathbf{y}_k$, the North Pole may lie in $L_k(\mathbf{x}_i)$. If this occurs, we take $\Theta_i = [-\pi, \pi]$.

3.3.2 Non-degenerate cases

Define $\phi(\mathbf{y}_i)$ and $\psi(\mathbf{y}_i)$ respectively as the azimuth and inclination of $\mathbf{y}_i$. The spherical region $S_\epsilon(\mathbf{y}_i)$ is contained between the meridians $\phi(\mathbf{y}_i) - \gamma_i$ and $\phi(\mathbf{y}_i) + \gamma_i$, where

$$\gamma_i = \arcsin\left(\frac{\sin(\epsilon)}{\sin(\psi(\mathbf{y}_i))}\right) \tag{25}$$

following the geometric considerations in Fig. 2(a). Let $\theta_i \in [-\pi, \pi]$ be the rotation angle such that the point $\mathbf{A}_{\theta_i,\mathbf{y}_k}\hat{\mathbf{B}}\mathbf{x}_i$ is on the meridian $\phi(\mathbf{y}_i)$. Refer to Fig. 2(b).

Case 1: $\theta_i \in [0, \pi]$

This case is shown in Fig. 2(b). Define $\Theta_i = [\theta_i^a, \theta_i^b]$. The desired bounding interval Θ_i can be obtained by taking

$$\theta_i^a = \theta_i - \gamma_i - \alpha_i \quad \text{and} \quad \theta_i^b = \theta_i + \gamma_i + \beta_i, \tag{26}$$

where α_i is the largest value such that the spherical region

$$S_{\delta(\theta_i^a)}(\mathbf{A}_{\theta_i^a,\mathbf{y}_k}\hat{\mathbf{B}}\mathbf{x}_i) \tag{27}$$

still touches the meridian $\phi(\mathbf{y}_i) - \gamma_i$, and β_i is the largest value such that the spherical region

$$S_{\delta(\theta_i^b)}(\mathbf{A}_{\theta_i^b,\mathbf{y}_k}\hat{\mathbf{B}}\mathbf{x}_i) \tag{28}$$

still touches the meridian $\phi(\mathbf{y}_i) + \gamma_i$. Refer to Fig. 2(b). To determine Θ_i, we must find α_i and β_i. From (23),

$$\delta(\theta_i^a) = 2|\theta_i - \gamma_i - \alpha_i|\sin(\epsilon/2) + \epsilon \quad \text{and} \tag{29}$$

$$\delta(\theta_i^b) = 2|\theta_i + \gamma_i + \beta_i|\sin(\epsilon/2) + \epsilon. \tag{30}$$

Applying the same geometric considerations in Fig. 2(a) on the spherical regions (27) and (28), we have

$$\sin(\alpha_i) = \frac{\sin(\delta(\theta_i^a))}{\sin(\psi(\mathbf{x}_i))}, \quad \sin(\beta_i) = \frac{\sin(\delta(\theta_i^b))}{\sin(\psi(\mathbf{x}_i))}. \quad (31)$$

Note that the functions on both sides of each equation have the unknowns α_i and β_i respectively.

Fig. 2(c) plots the two sine functions $\sin(\beta_i)$ and $\sin(\delta(\theta_i^b))/\sin(\psi(\mathbf{x}_i))$. We consider only $\beta_i \in [0, \pi/2]$, since the condition where the two functions do not intersect before $\beta_i \leq \pi/2$ corresponds to the degeneracies in Sec. 3.3.1; see supplementary material for proof. Further, since usually $\epsilon \ll \pi$, the period of the second sine function

$$\frac{2\pi}{2\sin(\epsilon/2)} \gg 2\pi \quad (32)$$

is much greater than 2π, thus explaining the almost linear trend of the second sine function for $\beta_i \in [0, \pi/2]$. A largely identical plot occurs for the functions involving α_i.

Analytically solving the equations in (31) is non-trivial. However, since all that we require is a bounding interval Θ_i, we can replace the sine functions with more amenable approximations that yield a valid bounding interval. An identical technique is used to solve for α_i and β_i respectively, thus we describe our solution only for β_i.

We replace $\sin(\beta_i)$ with a lower-bounding two-piece linear function; see Fig. 2(c). To obtain an upper-bounding line to $\sin(\delta(\theta_i^b))/\sin(\psi(\mathbf{x}_i))$, we use Jordan's inequality

$$\sin(t) \leq t \text{ for } t \leq \pi/2, \quad (33)$$

which enables us to replace the second sine function with

$$\frac{\delta(\theta_i^b)}{\sin(\psi(\mathbf{x}_i))} = \frac{2|\theta_i + \gamma_i + \beta_i|\sin(\epsilon/2) + \epsilon}{\sin(\psi(\mathbf{x}_i))}. \quad (34)$$

This upper-bounding line is legitimate for

$$2|\theta_i + \gamma_i + \beta_i|\sin(\epsilon/2) + \epsilon \leq \pi/2, \quad (35)$$

where in the worst case requires

$$2\pi\sin(\epsilon/2) + \epsilon \leq \pi/2 \quad (36)$$

or $\epsilon \leq \pi/(2\pi+2) \equiv 21.7°$, which is more than adequate for practical applications. Solving for β_i in the manner above allows us to compute the upper limit θ_i^b in constant time.

Note that the resulting upper limit θ_i^b may extend beyond π; to "wrap around" the interval, we break $\Theta_i = [\theta_i^a, \theta_i^b]$ into two connected intervals $[\theta_i^a, \pi]$ and $[\theta_i^b - 2\pi, -\pi]$.

Case 2: $\theta_i \in [-\pi, 0]$

Case 2 is simply a mirror of **Case 1** and the same steps apply with the "directions" reversed.

Result 2 *For any $i \neq k$, if $S_\epsilon(\mathbf{y}_i)$ intersects with $L_k(\mathbf{x}_i)$, the range of angles θ such that $\angle(\mathbf{A}_{\theta,\mathbf{u}}\mathbf{B}\mathbf{x}_i, \mathbf{y}_i) \leq \epsilon$ for all $\mathbf{u} \in S_\epsilon(\mathbf{y}_k)$ and $\mathbf{B}\mathbf{x}_i \in S_\epsilon(\hat{\mathbf{B}}\mathbf{x}_i)$ is bounded by Θ_i computed according to Sec. 3.3.*

3.4. Interval stabbing

For problem P_k, on the input point matches that remain after pruning by the application of Result 1, we use Result 2 to convert them into a set of angular intervals $\{\Theta_j\}$, where each $\Theta_j = [\theta_j^a, \theta_j^b]$. We aim to find the largest number of point matches that can be aligned by the same rotation angle θ. More formally, we seek the solution

$$O_k = \underset{\theta \in [-\pi, \pi]}{\text{maximize}} \sum_j \mathbb{I}(\theta \in [\theta_j^a, \theta_j^b]) \quad (37)$$

where $\mathbb{I}(\cdot)$ is an indicator function that returns 1 if the input predicate is true and 0 otherwise. This is the well-known *interval stabbing* problem, for which efficient deterministic algorithms exist [3, Chap. 10]. We take $\hat{f}_k := O_k + 1$ as an upper bound to the solution f_k to P_k.

Proposition 2 $\hat{f}_k := O_k + 1 \geq f_k$.

Proof By Result 2, each interval Θ_j is an over-estimation of the range of angles of rotation $\mathbf{A}_{\theta,\mathbf{u}}$ that permit the associated point match to be aligned. The number $O_k + 1$ must thus be greater than or equal to the maximum number of point matches that can be aligned under problem P_k. $\square$

As a by-product of interval stabbing, we derive

$$\tilde{\mathbf{R}}_k = \mathbf{A}_{\tilde{\theta}, \hat{\mathbf{B}}\mathbf{x}_k}\hat{\mathbf{B}}, \quad (38)$$

where $\tilde{\theta}$ is an angle that globally solves (37). Aligning the input data with $\tilde{\mathbf{R}}_k$ thus provides a lower bound to the original rotation search problem (1).

4. Main algorithm

We develop a **g**uaranteed **o**utlier **re**moval algorithm (GORE) for the rotation search problem (1). Algorithm 1 summarizes our method. Given a set of input point matches $\mathcal{H}$, our method iterates over each point match $(\mathbf{x}_k, \mathbf{y}_k)$ and performs two operations: seek an improved lower bound l to problem (1) and an upper bound $\hat{f}_k$ to subproblem P_k; both steps are conducted simultaneously using our techniques in Sec. 3. Both values are then compared to attempt to reject the current point match as an outlier. The output is a reduced set of point matches $\mathcal{H}' \subseteq \mathcal{H}$ guaranteed to include the globally optimal solution $\mathcal{I}^*$ to (1).

GORE is a deterministic algorithm, unlike RANSAC. The worst case time complexity can be established as follows: for each k, the bounding interval Θ_i for each $i \neq k$ is obtained in constant time. Given N intervals, the stabbing problem (37) can be solved in $\mathcal{O}(N \log N)$ time [3, Chap. 10]. Thus, Line 5 in Algorithm 1 takes $\mathcal{O}(N \log N)$ time. In the worst case, Line 5 is performed N times, and GORE thus consumes $\mathcal{O}(N^2 \log N)$ time.

As a whole, GORE contains only very simple geometric operations. In Sec. 5, we demonstrate the extreme efficiency of GORE in processing large input data sizes.

Algorithm 1 Guaranteed outlier removal for rotation search

Require: Point matches $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, inlier threshold ϵ.
1: $\mathcal{H} \leftarrow \{1, 2, \ldots, N\}$.
2: $\mathcal{H}' \leftarrow \mathcal{H}, \mathcal{O} \leftarrow \mathcal{H}, \mathcal{V} \leftarrow \emptyset$, and $l \leftarrow 0$.
3: **for all** $k \in \mathcal{O}$ **do**
4:    $\mathcal{V} \leftarrow \mathcal{V} \cup \{k\}$.
5:    Compute upper bound $\hat{f}_k$ and suboptimal rotation $\tilde{\mathbf{R}}_k$ (Sec. 3) for problem P_k on data indexed by $\mathcal{H}'$.
6:    $\mathcal{C}_k \leftarrow \{i \mid i \in \mathcal{H}', \angle(\tilde{\mathbf{R}}_k \mathbf{x}_i, \mathbf{y}_i) \leq \epsilon\}$.
7:    $l_k \leftarrow |\mathcal{C}_k|$.
8:    **if** $l_k > l$ **then**
9:       $l \leftarrow l_k$.
10:      $\mathcal{O} \leftarrow \mathcal{H}' \setminus \mathcal{C}_k$.
11:   **end if**
12:   **if** $\hat{f}_k < l$ **then**
13:      $\mathcal{H}' \leftarrow \mathcal{H}' \setminus \{k\}$.
14:   **end if**
15:   $\mathcal{O} \leftarrow \mathcal{O} \setminus \mathcal{V}$.
16: **end for**
17: **return** $\{(\mathbf{x}_i, \mathbf{y}_i) \mid i \in \mathcal{H}'\}$.

5. Results

All algorithms were implemented in C++. Experiments were conducted on a standard PC with a 2.70GHz CPU.

Our implementation of GORE is provided in the supplementary material if the reader wishes to verify our results.

5.1. Synthetic data

A data instance was generated as follows: N points on the unit-sphere were randomly produced to obtain set $\mathcal{X}$. Set $\mathcal{X}$ was randomly rotated to produce set $\mathcal{Y}$, which was then added with Gaussian noise of $\sigma = 0.5°$ (recall that we use the angular distance here). For a given outlier rate ρ, ρN point matches $(\mathbf{x}_i, \mathbf{y}_i)$ were randomly chosen from $(\mathcal{X}, \mathcal{Y})$ and resampled uniformly on the sphere to create outliers. In our experiments, $N \in \{100, 250, 500\}$ and $\rho = \{0, 0.05, \ldots, 0.9\}$ were used. For each (N, ρ) combination, 1000 data instances were generated and $\epsilon = 0.5°$ was used in (1). The following approaches were tested:

- RANSAC: A confidence level of 0.99 was used for the stopping criterion [6]. For each data instance, median runtime over 100 runs were taken.
- GORE: Algorithm 1. No particular ordering for the data was conducted beyond the order of generation.
- BnB: Following the method of [2, 7].
- GORE+BnB: Data remaining after GORE was fed to BnB. The lower bound of BnB was also initialized as the value of l at the termination of Algorithm 1.
- GORE+RANSAC: Data remaining after GORE was fed to RANSAC. Global optimality is not guaranteed.

- RGORE+BnB: Same as GORE+BnB, but the initial value of l in Algorithm 1 for GORE was obtained by first running RANSAC to yield a suboptimal result $\tilde{\mathcal{I}}$.
- GORE+aBnB: Same as GORE+BnB, but all the original data was given to BnB (GORE was only used to initialize the lower bound of BnB).

Since [4] was much slower even with very efficient solvers (2 ms for each 3-subset), we do not show its results here.

Runtime comparisons The first row of Fig. 3 shows the median *total* runtime over all data instances for the methods.

While RANSAC was faster than GORE at low outlier rates, as the outlier rate increased, the runtime of RANSAC increased exponentially. In contrast, the runtime of GORE grew at a much lower rate. As expected, the runtime of BnB also grew rapidly as the outlier rate increased. This contrasts with the trend exhibited by GORE+BnB - as the outlier rate increased, the total runtime decreased! This is because at higher outlier rates, GORE removed more outliers and reduced the overall data population more aggressively, hence BnB was able to find the global solution using less time. Since the bulk of the runtime in GORE+BnB was due to BnB, the total runtime decreased with outlier rate.

At lower outlier rates, GORE+BnB took longer than "raw" BnB since there were fewer outliers to remove, but steadily GORE+BnB began to outperform BnB. The performance gain became significant at $\approx 70\%$ outliers. At 90% outliers, GORE+BnB was an order of magnitude faster than BnB. As we will show in Sec. 5.2, outlier rates greater than 95% are actually very common in real data.

The results of RGORE+BnB at low outlier rates show that initializing GORE with RANSAC only marginally reduced the total runtime. However, at high outlier rates, the total runtime increased dramatically following the slowing down of RANSAC. Crucially, the trend of GORE+aBnB shows clearly that the dominating factor in speeding up BnB is in reducing the data amount and outlier rate, not in initializing BnB with a good lower bound. Hence, hot starting BnB with the suboptimal result $|\tilde{\mathcal{I}}|$ of RANSAC will not reduce runtime (not to mention that at high outlier rates, the computation of RANSAC itself is a major burden).

Evaluation of suboptimal rotation Here we provide empirical evidence that, although GORE cannot completely eliminate all outliers, the best suboptimal rotation $\tilde{\mathbf{R}}_k$ calculated by Algorithm 1 is actually a good approximate solution. On each data instance generated above, we calculated the error of the best $\tilde{\mathbf{R}}_k$ to the globally optimal solution $\mathbf{R}^*$, where the error is measured by $d\angle(\tilde{\mathbf{R}}_k, \mathbf{R}^*) = \|\log(\tilde{\mathbf{R}}_k(\mathbf{R}^*)^T)\|_2$ with $\log(\cdot)$ the inverse of the exponential map. The distance is interpreted as the minimum geodesic motion between $\tilde{\mathbf{R}}_k \mathbf{p}$ and $\mathbf{R}^* \mathbf{p}$ where $\mathbf{p}$ is an arbitrary point [8]. We evaluated in the same way the rotations estimated using SVD [1] from the raw data, and from
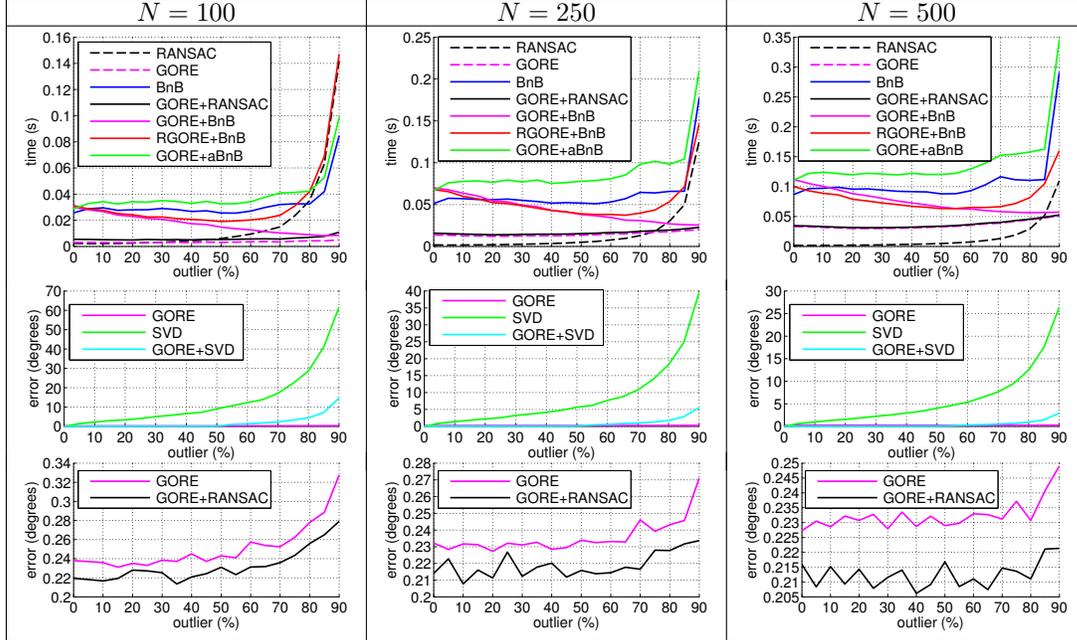
Figure 3. Results on synthetic data. Row 1: Runtimes for different outlier ratios. Rows 2 and 3: Angular error of estimated rotations. To avoid clutter, GORE+RANSAC's error is not plotted in Row 2.

the data after outlier removal with GORE. The second row in Fig. 3 shows error from all three rotations for increasing outlier rate, while the third row shows error from $\tilde{\mathbf{R}}_k$ and GORE+RANSAC's result.

As expected, SVD (least squares) rotation estimation is easily biased by outliers. Also, the non-negligible error of GORE+SVD points to the presence of remaining outliers after GORE. The error of $\tilde{\mathbf{R}}_k$, however, remains low ($\leq 0.5°$) even for high outlier rates. This indicates the efficacy of GORE as a suboptimal rotation search method. Results also show that a further improvement to $\tilde{\mathbf{R}}_k$ can be achieved by GORE+RANSAC at a small additional runtime.

5.2. Point cloud registration

Although practical settings usually demand full 6DOF registration, rotation search for 3D point cloud registration serves as an important subroutine in 6DOF methods (see, e.g., [5]). In this experiment, we test the use of GORE for rotational registration. We use data from the Stanford repository, namely *buddha*, *bunny*, *armadillo* and *dragon*. Two partially overlapping scans S_1 and S_2 were selected for each object. Sizes of S_1 and S_2 are listed on the Column 1 in Table 1. S_1 and S_2 were centered and scaled such as their centroids coincided with the origin and both point sets were contained in the cube $[-50, 50]^3$. Point matches between S_1 and S_2 were obtained using ISS3D [18] keypoint detector and matching with the PFH [14] descriptor as available in Point Cloud Library (http://pointclouds.org/).

A correspondence set was created by retaining N of the
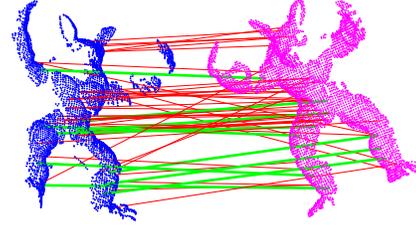
Figure 4. Data instance for *armadillo* for $N = 100$. Green lines represent the 10 inlier matches found by BnB. To avoid excessive clutter, only half of the outlier matches (red lines) are displayed.

best point matches based on the L_2-norm of the PFH descriptors. For $N \in \{100, 250, 500\}$, the obtained inlier ratios based on the threshold $\epsilon = 0.5°$ are listed in Column 3 in Table 1. Observe that the outlier rates in this problem are extremely high, even reaching 99% in some data instances. For each correspondence set, 10 different randomized rotations were applied on S_1 to produce 10 data instances for rotation search; Fig. 4 depicts one such instance.

For GORE, a straightforward variant was used; the main loop in Algorithm 1 was iterated until no more outliers could be removed. Typically this required 3 to 10 passes through the data. While this increased the duration of our method, the total runtime was still relatively minuscule, as evidenced in Table 1. We also executed RANSAC, BnB, RANSAC+BnB (the suboptimal RANSAC result $|\tilde{\mathcal{I}}|$ was used to initialize the lower bound of BnB) and GORE+BnB. We recorded the following measures:

- lwbnd: objective value (1) of best suboptimal solution.

Object	N	irat	GORE				RANSAC			BnB			RANSAC +BnB	GORE +BnB
			lwbnd	err (°)	out	time (s)	lwbnd	err (°)	time (s)	opt	err (°)	time (s)	time (s)	time (s)
	100	0.09	6	0.23	53	0.009	7	0.36	0.164	9	0.37	0.225	0.389	0.074
buddha	250	0.05	9	0.31	178	0.040	10	0.24	0.583	12	0.22	0.980	1.561	0.116
$|S_1| = 4151$	500	0.03	13	0.35	390	0.112	14	0.31	1.366	17	0.27	2.875	4.211	0.237
$|S_2| = 3901$	750	0.02	13	0.34	590	0.304	14	0.32	4.127	17	0.27	7.565	11.827	0.630
	1000	0.01	13	0.32	807	0.447	14	0.30	6.494	17	0.27	12.610	19.470	1.018
	100	0.18	16	0.19	74	0.003	16	0.20	0.032	18	0.13	0.030	0.062	0.003
bunny	250	0.10	20	0.27	209	0.015	21	0.24	0.133	24	0.13	0.145	0.278	0.024
$|S_1| = 6533$	500	0.06	27	0.23	442	0.056	26	0.23	0.342	30	0.22	0.520	0.881	0.076
$|S_2| = 6226$	750	0.04	31	0.18	684	0.127	29	0.25	0.659	32	0.23	1.245	1.946	0.147
	1000	0.04	32	0.19	924	0.219	30	0.24	1.220	35	0.14	2.445	3.764	0.269
	100	0.10	7	0.17	80	0.003	8	0.30	0.125	10	0.21	0.095	0.215	0.013
armadillo	250	0.06	10	0.17	229	0.014	12	0.31	0.501	14	0.26	0.350	0.875	0.021
$|S_1| = 4508$	500	0.03	10	0.69	469	0.055	12	0.31	1.783	15	0.24	1.430	3.198	0.066
$|S_2| = 4362$	750	0.02	13	0.34	713	0.146	13	0.29	3.270	16	0.24	3.435	7.002	0.161
	1000	0.01	13	0.34	958	0.233	13	0.31	6.843	16	0.50	7.150	14.505	0.264
	100	0.20	19	0.22	71	0.004	18	0.20	0.024	20	0.24	0.060	0.079	0.014
dragon	250	0.12	29	0.11	205	0.016	29	0.15	0.068	30	0.25	0.175	0.241	0.034
$|S_1| = 5332$	500	0.07	30	0.18	446	0.055	31	0.17	0.257	33	0.22	0.565	0.827	0.065
$|S_2| = 4683$	750	0.05	33	0.15	693	0.167	33	0.16	0.506	35	0.17	1.340	1.908	0.184
	1000	0.04	36	0.12	939	0.226	36	0.16	0.870	38	0.14	2.635	3.557	0.283

Table 1. Point cloud registration results.

- err (°): angular error in degrees of best suboptimal rotation to true rotation.
- time (s): total runtime in seconds.
- opt: objective value (1) of global solution.

Table 1 lists the median values over all 10 data instances.

Due to the extremely high outlier rates, RANSAC was an order of magnitude slower than GORE. On all the data instances, GORE was able to terminate within 1 second, even with multiple passes over the data. The most crucial outcome is that the combination GORE+BnB was able to find the globally optimal result with an order of magnitude less time than raw BnB. This was due to the massive reduction of true outliers before BnB - in this experiment, after GORE the median problem size to BnB was just 50. Additionally, the fact that RANSAC+BnB was slower than raw BnB indicates the ineffectiveness of hot starting using RANSAC.

5.3. Image stitching

We follow the image stitching experiment in [4]. SIFT correspondences are obtained across an image pair taken with known camera intrinsics $\mathbf{K}_1$ and $\mathbf{K}_2$. The scene is sufficiently far away to justify a homography $\mathbf{H} = \mathbf{K}_2 \mathbf{R} \mathbf{K}_1^{-1}$ as an alignment function, where $\mathbf{R}$ is the rotation between the views. The rotation $\mathbf{R}$ can be estimated by registering the matching vectors backprojected from the SIFT keypoint coordinates using the inverse calibration matrix.

Fig. 5 presents a challenging image pair where there is a very small overlapping area. A total of 154 SIFT matches were detected and 64 of them correspond to inliers. The inlier threshold used was $\epsilon = 0.05°$. GORE with multiple repetitions eliminated all outliers in 10 ms; running BnB after GORE would thus terminate immediately, since the

Figure 5. SIFT correspondences and stitching result of GORE.

best solution found by GORE equals to the global solution.

Due to limited space, we can only show one image stitching result here; see supplementary material for more results.

6. Conclusions

We have presented a guaranteed outlier removal technique for rotation search, in the sense that any datum it removes cannot be in the globally optimal solution. Based on simple geometric operations, our algorithm is deterministic and efficient. Experiments show that, by significantly reducing a significant amount of the outliers, our method greatly speeds up globally optimal rotation search.

References

[1] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-D point sets. *IEEE TPAMI*, 9(5):698–700, 1987. 1, 6

[2] J.-C. Bazin, Y. Seo, and M. Pollefeys. Globally optimal consensus set maximization through rotation search. In *ACCV*, 2012. 1, 6

[3] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational geometry*. Springer, 3rd edition, 2008. 5

[4] O. Enqvist, E. Ask, F. Kahl, and K. Åström. Robust fitting for multiple view geometry. In *ECCV*, 2012. 1, 6, 8

[5] O. Enqvist and F. Kahl. Robust optimal pose estimation. In *ECCV*, 2008. 1, 7

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

[7] R. I. Hartley and F. Kahl. Global optimization through rotation space search. *IJCV*, 82(1):64–79, 2009. 1, 4, 6

[8] R. I. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *IJCV*, 103(3):267–305, 2013. 2, 6

[9] J. Heller, M. Havlena, and T. Pajdla. A branch-and-bound algorithm for globally optimal hand-eye calibration. In *CVPR*, 2012. 1

[10] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *JOSA*, A4:629–642, 1987. 1, 2

[11] T. Needham. *Visual complex analysis*. Clarendon Press, 1997. 4

[12] C. Olsson, O. Enqvist, and F. Kahl. A polynomial-time bound for matching and registration with outliers. In *CVPR*, 2008. 1

[13] T. Ruland, T. Pajdla, and L. Kruger. Globally optimal hand-eye calibration. In *CVPR*, 2012. 1

[14] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz. Aligning point cloud views using persistent feature histograms. In *IROS*, 2008. 1, 7

[15] Y. Seo, Y.-J. Choi, and S. W. Lee. A branch-and-bound algorithm for globally optimal calibration of a camera-and-rotation-sensor system. In *ICCV*, 2009. 1

[16] L. Svärm, O. Enqvist, M. Oskarsson, and F. Kahl. Accurate localization and pose estimation for large 3d models. In *CVPR*, 2014. 2

[17] F. Tombari, S. Salti, and L. D. Stefano. Performance evaluation of 3D keypoint detectors. *IJCV*, 102(1–3):198–220, 2013. 1

[18] Y. Zhong. A shape descriptor for 3d object recognition. In *Proceedings ICCV 2009 Workshop 3DRR*, 2009. 1, 7