Provably Approximated Point Cloud Registration

Ibrahim Jubran[1], Alaa Maalouf[1], Ron Kimmel[2], Dan Feldman[1]
[1]University of Haifa, [2]Technion
{ibrahim.jub, alaamalouf12, dannyf.post}@gmail.com
{ron}@cs.technion.ac.il

Abstract

The goal of the alignment problem *is to align a (given) point cloud $P = \{p_1, \cdots, p_n\}$ to another (observed) point cloud $Q = \{q_1, \cdots, q_n\}$. That is, to compute a rotation matrix $R \in \mathbb{R}^{3\times3}$ and a translation vector $t \in \mathbb{R}^3$ that minimize the sum of paired distances between every transformed point $Rp_i - t$, to its corresponding point q_i, over every $i \in \{1, \cdots, n\}$. A harder version is the* registration problem*, where the correspondence is unknown, and the minimum is also over all possible correspondence functions from P to Q. Algorithms such as the Iterative Closest Point (ICP) and its variants were suggested for these problems, but none yield a provable non-trivial approximation for the global optimum.*

We prove that there always *exists a "witness" set of 3 pairs in $P \times Q$ that, via novel alignment algorithm, defines a constant factor approximation (in the worst case) to this global optimum. We then provide algorithms that recover this witness set and yield the first provable constant factor approximation for the: (i) alignment problem in $O(n)$ expected time, and (ii) registration problem in polynomial time. Such small witness sets exist for many variants including points in d-dimensional space, outlier-resistant cost functions, and different correspondence types.*

Extensive experimental results on real and synthetic datasets show that, in practice, our approximation constants are close to 1 and our error is up to x10 times smaller than state-of-the-art algorithms.

1. Introduction

Consider the set P of known 3D landmarks mounted on a car, and the set Q of the same 3D landmarks as currently observed via an external 3D camera, say, a few seconds later. Suppose that we wish to compute the new car's position and orientation, relative to its starting point. These can be deduced by recovering the rigid transformation (rotation and translation) that align P to Q. In this *alignment problem*, we assume that the correspondence (matching) between every point in P to Q is known. When this matching is unknown, and needs to be computed, the problem is known as the *registration problem*. It is a fundamental problem in computer vision [33, 28, 40, 47] with many applications in robotics [27, 37, 13] and autonomous driving [51].

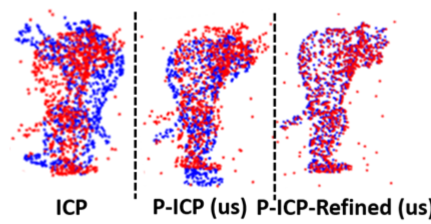

Figure 1: Registration visualization using the Armadillo model with $n = 1000$ points and $k = 20\%$ outliers. (Left) ICP(P, Q), (middle) P-ICP$(P, Q, \text{cost}, \gamma)$, (right) P-ICP-Refined$(P, Q, \gamma, \text{cost})$. cost is the SSD with a threshold M-estimator and $\gamma = 3000$; see Section 3.2.

Alignment. In the alignment problem the input consists of two ordered sets $P = \{p_1, \cdots, p_n\}$ and $Q = \{q_1, \cdots, q_n\}$ in $\mathbb{R}^d$, where $d = 3$ in the previous application, and the goal is to minimize

$$\sum_{i=1}^{n} D(Rp_i - t, q_i), \quad (1)$$

over every *alignment* (rigid transformation) (R, t) consisting of a rotation matrix $R \in \mathbb{R}^{d\times d}$ (an orthogonal matrix whose determinant is 1), and a translation vector $t \in \mathbb{R}^d$, and where $D(p, q) = \|p - q\|$ is the Euclidean (ℓ_2) distance between a pair of points $p, q \in \mathbb{R}^d$. Here, the sum is over the distance between every point $p_i \in P$ to its corresponding point $q_i \in Q$. This correspondence may be obtained using some auxiliary information, like point-wise descriptors e.g., SIFT [25], visual tracking of points [36, 44], or the use of predefined shapes and features [32, 39].

To our knowledge, the only provable approximation to the optimal *global minimum* of (1) is for its variant where $D(p, q)$ is replaced by $\ell(D(p, q)) = \|p - q\|^2$, i.e., *squared* Euclidean distance. In this special case, the optimal solution

Table 1: ***Example contributions.*** *Variants of the problems (1)–(3) that we approximate in this paper, either using: (i) Theorem 3 (known correspondence), or (ii) Theorem 5 (unknown correspondence). Let $P = \{p_1, \cdots, p_n\}$ and $Q = \{q_1, \cdots, q_n\}$ be two sets of points in $\mathbb{R}^d$, let $z, r, T > 0$, and let $w = d^{\left|\frac{1}{z} - \frac{1}{2}\right|}$. Formally, we wish to minimize $\text{cost}(P, Q, (R, t)) = f(\ell(D(Rp_1 - t, q_1)), \cdots, \ell(D(Rp_n - t, q_n)))$ for functions $D : \mathbb{R}^d \times \mathbb{R}^d \to [0, \infty)$, $\ell : [0, \infty) \to [0, \infty)$ and $f : \mathbb{R}^n \to [0, \infty)$ as in Definition 2. Rows marked with a $\star$ can also be approximated in linear time with high probability and bigger approximation factors, using Theorem 4.*

Use case	$f(v)$	$\ell(x)$	$D(p,q)$	Optimization Problem $\text{cost}(P, Q, (R, t))$	Approximation Factor	Matching m Necessary as input?
Sum of distances $\star$	$\|v\|_1$	x	$\|p-q\|_2$	$\sum_{i=1}^n \left\|Rp_i - t - q_{m(i)}\right\|$	$(1+\sqrt{2})^d$	No
Sum of squared distances $\star$	$\|v\|_1$	x^2	$\|p-q\|_2$	$\sum_{i=1}^n \left\|Rp_i - t - q_{m(i)}\right\|^2$	$(1+\sqrt{2})^{2d}$	No
Sum of distances with noisy data using M-estimators	$\|v\|_1$	$\min\{x, T\}$	$\|p-q\|_2$	$\sum_{i=1}^n \min\left\{\left\|Rp_i - t - q_{m(i)}\right\|, T\right\}$	$(1+\sqrt{2})^d$	No
Sum of ℓ_z distances to the power of r $\star$	$\|v\|_1$	x^r	$\|p-q\|_z$	$\sum_{i=1}^n \left\|Rp_i - t - q_{m(i)}\right\|_z^r$	$w^r(1+\sqrt{2})^{dr}$	No
Sum of ℓ_z distances to the power of r with $k \geq 1$ outliers	Sum of the $n-k$ smallest entries of v	x^r	$\|p-q\|_z$	$\sum_{i \in S \subset \{1, \cdots, n\}, \|S\| = n-k} \left\|Rp_i - t - q_{m(i)}\right\|_z^r$	$w^r(1+\sqrt{2})^{dr}$	Yes

is unique and easy to compute: t is simply the vector connecting the two centers of mass of P and Q, and $R \in \mathbb{R}^{d \times d}$ can be computed using Singular Value Decomposition [14] as described in [22]. There has been a long line of work to handle this problem also in the presence of outliers; see e.g., [6, 52, 49]. Many of those works are RANSAC-type algorithms [12]. This paper gives the first provable non-trivial approximation algorithm for (1), while also handling an even wider range of functions.

Registration. The registration problem does not assume the correspondence between P and Q is given, that is, we do not know which point in Q matches $p_i \in P$. Therefore, besides the rigid motion, the correspondence needs also to be extracted based solely on the two given point clouds, resulting in a much more complex problem with a large number of local minima; see Fig. 1. Formally, it aims to minimize

$$\sum_{i=1}^{n} \ell\left(D(Rp_i - t, q_{m(i)})\right), \tag{2}$$

over every alignment (R, t) and correspondence function $m : \{1, \ldots, n\} \to \{1, \ldots, n\}$; see recent survey [41]. Here, a natural selection for ℓ is $\ell(x) = x^2$. The set Q here is assumed to be of size n for simplicity only, but can be of any different size.

Unlike (1), we do not know a provable approximation to (2), even for $\ell(x) = x^2$. The most commonly used solution for this problem, both in academy and industry, is the Iterative Closest Point (ICP) heuristic [4]. Our main contribution is a provable alternative to the ICP which approximates the global optimum of this harder problem.

More complex cost functions. When dealing with real-world data, noise and outliers are inevitable. One may thus consider alternative cost functions, rather than the sum of squared distances (SSD) above, due to its sensitivity to such corrupted input. A natural more general cost function would be to pick e.g., $\ell(x) = x^r$ for $r > 0$, which is more robust to noise when $r \in (0, 1]$. Alternatively, for handling outliers, a more suitable function would be $\ell(x) = \min\{x, T\}$ for some threshold $T > 0$, or the common Tuckey or Huber losses, or any other robust statistics function [17].

To completely ignore these (unknown) faulty subsets of some paired data we may consider solving

$$\min_{(R,t)} \sum_{i \in S \subset \{1, \cdots, n\}, |S| = n-k} \ell\left(D(Rp_i - t, q_{m(i)})\right), \tag{3}$$

where $k \leq n$ is the number of outliers to ignore.

In this paper we suggest a general framework for provably approximating the global minimum of the alignment and registration problems, including formulations (1)–(3).

1.1. Related Work

The most common method for solving the registration problem in (2), for $\ell(x) = x^2$, is the ICP algorithm [7, 4]. The ICP is a local optimization technique, which alternates, until convergence, between solving the correspondence problem and the rigid alignment problem. Over the years, many variants of the ICP algorithm have been suggested; see survey in [38] and references therein. However, these methods usually converge to local and not global minimum if not initialized properly.

Estimation maximization approaches. To overcome the ICP limitations, probabilistic methods have been suggested, making use of GMMs, treating one point set as the GMM centroids, and the other as data points [18, 46, 8, 26, 29, 16, 5]. This category also includes the widely used Coherent Point Drift (CPD) method [31].

Learning-based approaches. Learning dedicated features for this task was shown to enhance the output alignment [45]. In [3], a deep learning model was combined with a modified version of the known Lukas & Kanade algorithm. Recently, an unsupervised deep learning based approach was proposed in [15].

Geometric and alternative approaches. Some works, e.g., [1, 34, 2], utilize techniques from computational geometry to devise a solution. [1, 34] also provide provable guarantees. Other results use a Branch and Bound scheme to compute the global minimum [35, 10, 50]. The work [30] tackles the problem using a smart indexing data organization. Some results use the Fourier domain [28], and use correlation of kernel density estimates (KDE) [42]. However, the above methods scale poorly as the input size increases.

Common limitations. The previously mentioned methods share similar properties and either (i) support only the simple sum of squared distances function or $d = 3$, (ii) they converge to a local minima due to bad initialization, (iii) give optimality guarantees, if any, only on a sub-task of the registration pipeline, and lack such guarantees relative to the global optimum of the registration problem, (iv) their convergence time is impractical or depends on the data itself, or (v) require a lot of training data. To our knowledge, no provable approximation algorithms have been suggested for tackling (2), even for $\ell(x) = x$.

Coresets. Some works suggest compressing the input point clouds into a small subset with a provable bound on the compression error. Existing and inefficient algorithms can thus run much faster. An error-less compression was suggested in [32] for the alignment problem; see more examples in [20]. While our method draw inspiration from the approximation techniques used in developing coresets, our paper suggests a "witness set", and not a coreset.

1.2. Our Contribution

(i) A novel alignment algorithm that given a specific set of d points from P and corresponding d points from Q, which we call a witness set, yields a provable constant factor approximation. We also prove that *every* input pair of point clouds in $\mathbb{R}^d$ admits such a witness set for all the versions of the alignment and registration problem, including problems (1)–(3); see Theorem 1.

(ii) RANSAC-style algorithms for recovering such witness sets for both the alignment and registration problems and their variants e.g., (1)–(3). Extensive experimental results on synthetic and real-world datasets demonstrate the effectiveness and accuracy of our suggested algorithms, as compared to state of the art methods; see Section 3. The results show that the approximation factor obtained in practice is much smaller than the theoretically predicted factor. We provide full open-source code for our algorithms [21].

(iii) A formal proof that running our suggested algorithms for a polynomial number of iterations yields a provable constant factor approximation for the alignment and registration problems, including sum of distances to the power of $r > 0$ and sum of M-estimators; see Theorems 3 and 5, Table 1, and Algorithms 2 and 4.

(iv) A probabilistic *linear time* algorithm for solving the alignment problem, which also supports e.g., sum of M-estimators; see Algorithm 3 and Theorem 4.

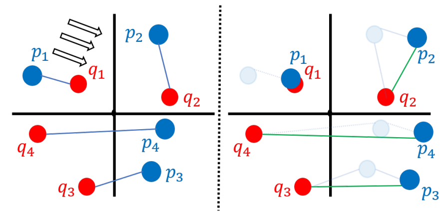

Figure 2: (Left): Two corresponding sets of points P (in blue) and Q (in red), where p_1 and q_1 have the smallest distance among all pairs. (Right): Translating P by $t = p_1 - q_1$ (i.e., p_1 now intersects q_1). By the triangle inequality, each distance $\|p_i - t - q_i\|$ (green lines) is at most $2 \cdot \|p_i - q_i\|$ (blue lines).

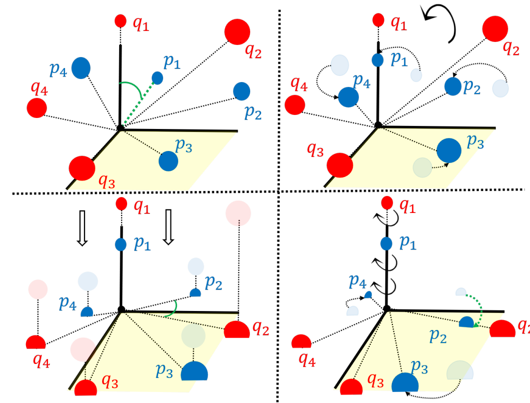

Figure 3: (Top left): Two sets of corresponding points P (in blue) and Q (in red). (Top right): Rotating P such that some $p_1 \in P$ aligns with its corresponding $q_1 \in Q$. (Bottom left): Projecting the rotated set P and the set Q onto the plane orthogonal to q_1. (Bottom right): Rotating the projected P such that one of its points aligns with its corresponding point from Q. Observe that the initial aligned pair of points (p_1, q_1) are not affected by the proceeding steps.

1.3. Novel Technique: Witness Set

We now introduce our novel technique. We first assume the correspondence between P and Q is given. We then generalize to the case with unknown correspondence.

Our main technical result is that for every *corresponding* ordered point sets $P = \{p_1, \cdots, p_n\}, Q = \{q_1, \cdots, q_n\} \subseteq \mathbb{R}^d$, every cost function cost which satisfies some set of properties (see Definition 2), and every possible alignment (R^*, t^*), there is a subset of P and a subset of Q, both of size equal to the dimension d, which we call a *witness set*. Using those subsets, our algorithm can determine an alignment (R', t'), that approximates the cost of (R^*, t^*), i.e., $\text{cost}(P, Q, (R', t')) \leq c \cdot \text{cost}(P, Q, (R^*, t^*))$, for small constant $c > 0$. Here, cost assigns a non-negative value for every pair of input point sets and alignment, and (R^*, t^*) can be the globally optimal (unknown) alignment.

For the sake of analysis only, we assume that (R^*, t^*) is known beforehand. The proof assumes an initial position of the point clouds where (R^*, t^*) has already been applied to P. It then applies a series of steps which alter this initial alignment (R^*, t^*), until a different alignment (R', t') is obtained, where a (witness) set of points from P and Q satisfies a sufficient number of known constraints, making it feasible (given this witness set) to recover (R', t'). Each step in this series is guaranteed to approximate the cost of its preceding step. Hence, the cost of (R', t') approximates the initial (optimal) cost of (R^*, t^*). The steps are as follows:

(i) Consider the set P' obtained by applying the optimal (unknown) alignment (R^*, t^*) to P. Now, consider the single corresponding pair of points $p' = R^* p - t^* \in P'$ and $q \in Q$ which have the closest distance $\|p' - q\|$ between them among all matched pairs. Using the triangle inequality, one can show that translating the set P' by $p' - q$ (that is, such that p' now intersects q) would not increase the pairwise distances of the other pairs of points by more than a multiplicative factor of 2; see Fig. 2. Hence, we proved the existence of a translation t' of P', where some $p \in P$ intersects its corresponding $q \in Q$, and where the cost is larger than the initial optimal cost by at most a constant factor. Now, assume that p' and q are located at the origin.

(ii) Similarly, we prove there is a corresponding pair of points $p' \in P'$, $q \in Q$ such that aligning their direction vectors via a rotation R', i.e., $R' \frac{p'}{\|p'\|} = \frac{q}{\|q\|}$, would increase the pairwise distances of the other pairs by at most a small factor. p' and q are the pair with the smallest angle between them.

(iii) We can repeat step (ii) above iteratively as follows: Find such a pair (p', q), align their direction vectors, project the two sets of points onto the hyperplane orthogonal to the direction vector of q, and repeat at most $d - 2$ times. Such a projection insures that the next uncovered rotation will maintain the alignment of (p', q); see Fig. 3. Each such step proves *the existence* of yet another corresponding pair of points which contribute at least 1 constraint on R', without damaging the cost by more than a constant factor. Hence, there exist $d - 1$ pairs of points which uniquely determine our approximated rotation R'. We call the d (unknown) pairs from the steps above a *witness set*. Given a witness set, the approximated alignment (R', t') can be recovered.

Recovering such a witness set requires recovering d points from P (where d is the dimension of P), which, using the known correspondence, uncover the d corresponding points from Q. When the correspondence is unknown, the minor difference is that we need to recover d points from P as well as d independent points from Q; see Theorem 5.

2. Provable Approximations

We now prove the existence of a witness set for the alignment and registration problems and their variants presented above. We then present algorithms that recover such a witness set for each of the problems. Due to lack of space, all our proofs are placed in the supplementary material.

Notation. We denote $[n] = \{1, \cdots, n\}$ for any integer $n \geq 1$. We assume every vector is a column vector. Let $\text{SO}(d)$ be the set of all rotation matrices in $\mathbb{R}^d$. For $t \in \mathbb{R}^d$ and $R \in \text{SO}(d)$, the pair (R, t) is called an *alignment*. We define $\text{ALIGNMENTS}(d)$ to be the union of all possible d-dimensional alignments. A *correspondence* (or matching) function is simply a function $m : [n] \to [n]$. For a correspondence function m and an ordered set $P = \{p_1, \cdots, p_n\}$, we define $P_{[m]} = \left\{p_{m(1)}, \cdots, p_{m(n)}\right\}$ as a new ordered set obtained from P after reordering its elements according to m.

2.1. Existence of a Witness Set

In what follows we present our main alignment algorithm and our main technical result; see Algorithm 1 and Theorem 1. Theorem 1 proves the existence of some witness set that, when plugged into Algorithm 1, produces an alignment with some provable guarantees.

Overview of Algorithm 1. Algorithm 1 gets as input two sets $\{p_1, \cdots, p_d\}$ and $\{q_1, \cdots, q_d\}$ in $\mathbb{R}^d$, and implements the scheme described in Section 1.3. At Line 1, we translate both sets so that p_d and q_d intersect at the origin. At Line 4 we compute a rotation matrix S that aligns the directions of p_1 and q_1. In Lines 5– 6 we compute an orthogonal matrix W whose column space spans the $(d-1)$-dimensional subspace π orthogonal to q_1 and project $Sp_2, \cdots, Sp_d, q_2, \cdots, q_d$ onto π, and repeat $d - 1$ times. Hence, at the i'th iteration, we compute a rotation S that aligns the directions of p_i (after $i - 1$ rotations and projections) and q_i (after $i - 1$ projections), but also maintains the alignment of the previously aligned pairs $(p_1, q_1), \cdots, (p_{i-1}, q_{i-1})$. Such a matrix exists since $p_i, \cdots, p_d, q_i, \cdots, q_d$ are, by construction, orthogonal to $p_1, \cdots, p_{i-1}, q_1, \cdots, q_{i-1}$; see Fig. 3. We output an alignment which replicates the composition of the steps above.

Overview of Theorem 1. Consider $P = \{p_1, \cdots, p_n\}$ and $Q = \{q_1, \cdots, q_n\}$ and any variant of either the alignment or registration problems, e.g., (1)–(3). Now, assume that (R^*, t^*) and m^* are respectively the globally optimal alignment and correspondence function for the task at hand. Observe that, in the alignment problem, m^* is given as input. Theorem 1 proves the existence of a set of d points

Algorithm 1: ALIGN$(\{p_1, \cdots, p_d\}, \{q_1, \cdots, q_d\})$

Input : Two sets of points that each spans a $d-1$ dimensional subspace in $\mathbb{R}^d$.

Output: An alignment (R, t); see Theorem 1

$p_i := p_i - p_d$ and $q_i := q_i - q_d$ for every $i \in [d]$
$R :=$ the identity matrix in $\mathbb{R}^d$
for *every* $z \in [d-1]$ **do**
    $S :=$ an arbitrary rotation matrix that satisfies $\frac{Sp_z}{\|p_z\|} = \frac{q_z}{\|q_z\|}$, and $Sp_i = p_i$ for every $i \in [z-1]$.
    $W :=$ an arbitrary matrix in $\mathbb{R}^{d \times (d-1)}$ such that $\left[W \mid \frac{q_z}{\|q_z\|}\right] \in \mathbb{R}^{d \times d}$ forms a basis of $\mathbb{R}^d$.
    $p_i := WW^T Sp_i$ and $q_i := WW^T q_i$, for every $i \in [d] \setminus [z]$
    $p_z := Sp_z$
    $R := SR$
$t := Rp_d - q_d$
return (R, t)

from P and d points from Q that, when plugged into Algorithm 1, produce an alignment (R, t) which guarantees:

$$\left\|Rp_i - t - q_{m^*(i)}\right\| \leq \sigma \cdot \left\|R^* p_i - t^* - q_{m^*(i)}\right\|, \quad \forall i \in [n]$$

for some small $\sigma \geq 1$. For $d = 3$ the constant is $\sigma < 15$. In other words, using (R, t) we can approximate each of the n pairwise distances of the optimal alignment (R^*, t^*). In the registration problem, m^* can be recovered afterwards, e.g., via nearest neighbour algorithm. Theorem 1 thus successfully decouples the two problems of recovering the alignment and recovering the correspondence function.

Theorem 1 (Witness sets)**.** *Let $P = \{p_1, \cdots, p_n\}$ and $Q = \{q_1, \cdots, q_n\}$ be two ordered sets each of n points in $\mathbb{R}^d$. Then, for every alignment (R^*, t^*) and matching function m^*, there exist $P' \subseteq P$ and $Q' \subseteq Q$ of size $|P'| = |Q'| = d$ such that the output (R, t) of the call* ALIGN(P', Q') *to Algorithm 1 satisfies the following for every $i \in [n]$:*

$$\left\|Rp_i - t - q_{m^*(i)}\right\| \leq (1 + \sqrt{2})^d \cdot \left\|R^* p_i - t^* - q_{m^*(i)}\right\|$$

Furthermore, (R, t) is computed in $O(d^3)$ time.

In Section 2.2 we prove that individually approximating the pairwise distances, as in Theorem 1, implies an immediate approximation to a wide range of cost functions.

While Theorem 1 guarantees the existence of at least one such witness set, empirically we have observed that many subsets of P and Q serve as good witness sets, in the sense that they produce approximation factors smaller than predicted in the theorem. Those factors are usually even close to 1. Hence, in Sections 2.3– 2.4 we apply RANSAC-type algorithms to recover a witness set. Theorem 1 also implies that running the suggested algorithms for a sufficient number of iterations produces a guaranteed constant factor approximation to the global optimum of the problems at hand.

2.2. Generalization

In what follows we define a wide family of cost functions which this work tackles, including the cost functions in (1)–(3). We then show that the approximation guarantees obtained in Theorem 1 suffice to approximate each such cost function; See Table 1 for examples. In what follows, for $r > 0$, an r-log-Lipschitz function is a function that, in every dimension individually, may be large but cannot increase too rapidly (in a rate that depends on r); see formal definition in Section B at the appendix.

Definition 2 (Cost function)**.** *Let $P = \{p_1, \cdots, p_n\} \subseteq \mathbb{R}^d$ and $Q = \{q_1, \cdots, q_n\} \subseteq \mathbb{R}^d$. Let $D : \mathbb{R}^d \times \mathbb{R}^d \to [0, \infty)$ be a function that assigns a non-negative weight for each pair of points in $\mathbb{R}^d$, $\ell : [0, \infty) \to [0, \infty)$ be an r-log-Lipschitz function and $f : [0, \infty)^n \to [0, \infty)$ be an s-log-Lipschitz function. Let (R, t) be an alignment. We define*

$$\begin{aligned}&\text{cost}(P, Q, (R, t)) \\ &\quad = f\left(\ell\left(D\left(Rp_1 - t, q_1\right)\right), \cdots, \ell\left(D\left(Rp_n - t, q_n\right)\right)\right).\end{aligned}$$

From pairwise distances to complex cost functions. Observation 5 in [19] (see Section B in the appendix) states that in order to approximate a given cost function from Definition 2, relative to the globally optimal alignment and correspondence (R^*, t^*) and m^* respectively, it is sufficient to approximate, simultaneously, each of the pairwise distances $\left\|R^* p_i - t^* - q_{m^*(i)}\right\|$ for every $i \in [n]$. By Theorem 1, there exists a witness set which provides the desired approximation for the above pairwise distances. Combining the above yields a provable approximation for any cost function from Definition 2. It is only left to recover a designated witness set, which is the goal of Sections 2.3 and 2.4.

2.3. Approximations for the Alignment Problem

We now provide an algorithm which computes a provable approximation for the alignment problem (1) and its variants, i.e., when the matching between P and Q *is given*. This is by recovering a designated witness set.

Overview of Algorithm 2. Using Theorem 1 and assuming the matching between P and Q is given, one can construct a RANSAC-type algorithm that iterates over γ subsets of P and Q of size d, applies Algorithm 1 to each two such corresponding subsets to obtain a candidate alignment, and returns the alignment that minimizes the cost function at hand. Algorithm 2 implements the scheme above.

Theorem 3. *Let P and Q be two ordered sets of n points in $\mathbb{R}^d$, $\gamma \in \Omega(n^d)$, $z > 0$, and $w = d^{\left|\frac{1}{z} - \frac{1}{2}\right|}$. Let* cost*, r and s be as defined in Definition 2 for $D(p, q) =$*

Algorithm 2: APPROX-ALIGNMENT$(P, Q, \gamma, \text{cost})$

Input : A pair of sets $P \subseteq \mathbb{R}^d$ and $Q \subseteq \mathbb{R}^d$, number of iterations $\gamma > 0$, and a cost function.

Output: An alignment (R, t); see Theorem 3

$M := \emptyset$.

$I :=$ randomly sample, with no repetition, a set of γ tuples of d distinct indices from $[n]$. // $|I| = \gamma$

for *every* $(i_1, \cdots, i_d) \in I$ **do**

  $(R', t') :=$ ALIGN$(\{p_{i_1}, \cdots, p_{i_d}\}, \{q_{i_1}, \cdots, q_{i_d}\})$ `// see Algorithm 1`

  $M := M \cup \{(R', t')\}$

$(R, t) \in \arg\min_{(R',t') \in M} \text{cost}(P, Q, (R', t'))$

return (R, t)

$\|p - q\|_z$. *Let* (R, t) *be the output of a call to* APPROX-ALIGNMENT$(P, Q, \gamma, \text{cost})$*; See Algorithm 2. Then,*

$$\text{cost}(P, Q, (R, t)) \leq w^{rs} \cdot (1+\sqrt{2})^{drs} \cdot \min_{(R',t')} \text{cost}(P, Q, (R', t')),$$

where the minimum is over every $(R', t') \in$ ALIGNMENTS. *Moreover,* (R, t) *is computed in* $n^{O(d)}$ *time.*

2.3.1 Run-Time Improvement

We now propose a randomized algorithm (see Algorithm 3) with the same goal as Algorithm 1, that succeeds with constant probability. By running this algorithm for a constant number of times, we can recover an alignment that, with probability approaching 1, has the same guarantees as the output of Algorithm 2. However, this new randomized algorithm requires linear, rather than polynomial, time.

Overview of Algorithm 3. Unlike Algorithm 1, which expects to receive a witness set as input, Algorithm 3 takes as input two full point clouds, and internally identifies a potential witness set. Intuitively, points in P with larger norm negatively affect our cost function more than points of smaller norm, when misaligned properly; see Fig. 11 at the appendix. Algorithm 3 thus samples a pair of corresponding points (p, q) with probability that depends on the norm of p and rotates P to align the direction vectors of p and q. Then, similarly to Algorithm 1, it projects the sets onto the hyperplane orthogonal to q and repeats.

Theorem 4. *Let* P *and* Q *be two ordered sets of* n *points in* $\mathbb{R}^d$ *and* $z > 0$. *Let* cost *be as in Definition 2 for* $f = \|v\|_1$, *some* r*-log Lipschitz function* ℓ *and* $D(p, q) = \|p - q\|_z$. *Let* (R, t) *be an output of a call to* PROB-ALIGN(P, Q, r)*; see Algorithm 3. Then, with probability at least* $\frac{1}{2^d}$,

$$\text{cost}(P, Q, (R, t)) \leq \sigma \cdot \min_{(R',t') \in \text{ALIGNMENTS}(d)} \text{cost}(P, Q, (R', t')),$$

Algorithm 3: PROB-ALIGN(P, Q, r)

Input : A pair of sets $P = \{p_1, \cdots, p_n\}$ and $Q = \{q_1, \cdots, q_n\}$ in $\mathbb{R}^d$ and $r > 0$.

Output: A rotation matrix; see Theorem 4

Sample an index $k \in [n]$ uniformly at random

$p := p - p_k$ for every $p \in P$

$q := q - q_k$ for every $q \in Q$

$J := \{k\}$ and $R :=$ the d-dimensional identity matrix

for *every* $z \in [d-1]$ **do**

  $w_i := \frac{\|p_i\|^r}{\sum_{j \in [n]} \|p_j\|^r}$ for every $i \in [n]$.

  Randomly sample an index $j \in [n] \setminus J$, where $j = i$ with probability w_i.

  $S :=$ an arbitrary rotation matrix that satisfies $\frac{Sp_j}{\|p_j\|} = \frac{q_j}{\|q_j\|}$ and $Sp_i = p_i$ for every $i \in J$.

  $W :=$ a matrix in $\mathbb{R}^{d \times (d-1)}$ such that $\left[W \mid \frac{q_j}{\|q_j\|}\right] \in \mathbb{R}^{d \times d}$ forms a basis of $\mathbb{R}^d$.

  $J := J \cup \{j\}$

  $p := Sp$ for every $p \in P$

  $p := WW^T p$ for every $p \in P \setminus \{p_i \mid i \in J\}$

  $q := WW^T q$ for every $q \in Q \setminus \{q_i \mid i \in J\}$

  $R := SR$

$t := Rp_k - q_k$

return (R, t)

for a constant σ *that depends on* d *and* r. *Furthermore,* (R, t) *is computed in* $O(nd^3)$ *time.*

Success probability. For the usual case of $d = 3$, the success probability of Algorithm 3 is at least $1/8$. Hence, repeating Algorithm 3 for less than 6 repetitions amplifies the success probability in Theorem 4 to more than $1/2$.

2.4. Approximations for the Registration Problem

As explained in Section 1.3, a witness set from P and Q also exists in the much harder variant where the matching between P and Q is unknown. We now provide an algorithm that, for any given variant of the registration problem, can recover a witness set. The formal statement is given in Theorem 5, which is one of our main contributions.

Overview of Algorithm 4. Unlike Algorithm 2 which samples d indices used to index both P and a Q, we now have to independently sample d indices for points in P as well as d indices for points in Q. Furthermore, we need to compute the nearest neighbour matching for every candidate alignment returned by Algorithm 1, before evaluating the cost function. Algorithm 4 applies the above scheme γ times and returns the alignment and matching function that minimize the given cost function.

Theorem 5. *Let* $P = \{p_1, \cdots, p_n\}$, $Q = \{q_1, \cdots, q_n\}$ *be two ordered sets of* n *points in* $\mathbb{R}^d$, $\gamma \in \Omega(n^{2d})$, $z > 0$,

Algorithm 4: ALIGN-AND-MATCH$(P, Q, \gamma, \text{cost})$

Input : A pair of sets $P \subseteq \mathbb{R}^d$ and $Q \subseteq \mathbb{R}^d$, number of iterations $\gamma > 0$, and a cost function.

Output: An alignment and a matching function; see Theorem 5

$M := \emptyset$.

$I :=$ randomly sample, with no repetition, a set of γ tuples of $2d$ indices $(i_1, \cdots, i_d, j_1, \cdots, j_d)$ from $[n]$, such that $i_1, \cdots, i_d$ are distinct and $j_1, \cdots, j_d$ are distinct. // $|I| = \gamma$

for *every* $(i_1, \cdots, i_d, j_1, \cdots, j_d) \in I$ **do**

  $(R', t') :=$ ALIGN$(\{p_{i_1}, \cdots, p_{i_d}\}, \{q_{j_1}, \cdots, q_{j_d}\})$ // see Algorithm 1

  $M := M \cup \{(R', t', \text{NN}(P, Q, (R', t')))\}$ /* $\text{NN}(P, Q, (R, t))$ is the nearest neighbour matching between Q, and P after applying (R, t). */

$(\tilde{R}, \tilde{t}, \tilde{m}) \in \underset{(R', t', m') \in M}{\arg\min} \ \text{cost}\left(P_{[m']}, Q, (R', t')\right)$.

return $(\tilde{R}, \tilde{t}, \tilde{m})$

and $w = d^{\left|\frac{1}{z} - \frac{1}{2}\right|}$. *Let* cost *and* r *be as in Definition 2 for* $D = \|p - q\|_z$ *and* $f(v) = \|v\|_1$. *Let* $(\tilde{R}, \tilde{t}, \tilde{m})$ *be the output of a call to* ALIGN-AND-MATCH$(P, Q, \gamma, \text{cost})$*; See Algorithm 4. Then, for* $c = w^r(1 + \sqrt{2})^{dr}$*, we have*

$$\text{cost}\left(P_{[\tilde{m}]}, Q, (\tilde{R}, \tilde{t})\right) \leq c \cdot \min_{(R,t,m)} \text{cost}\left(P_{[m]}, Q, (R, t)\right),$$

where the minimum is over every alignment (R, t) *and permutation* m*. Moreover,* $(\tilde{R}, \tilde{t}, \tilde{m})$ *is computed in* $n^{O(d)}$ *time.*

Substituting $z = r = 2$ in Theorem 5 yields a provable approximated alternative to ICP.

Comparison to RANSAC. RANSAC has some similarity to Algorithms 2 and 4 above in the sense that they both randomly sample points from P and Q. However, while RANSAC recovers a candidate alignment via common least squares on such candidate d pairs (see Section 1), our algorithms utilize a novel method (Algorithm 1) which is not necessarily optimal for those d pairs, but will be (almost) globally optimal for the overall cost of all the n pairs. In theory, unlike our algorithms, RANSAC does not guarantee global optimality; see comparison in Section 3.1.

Our approximation constants. While the approximation constants in Theorems 3- 5 above might seem large, they are only roughly < 14 in the pessimistic worst-case theory, they are smaller than 2 in practice, and can be obtained much faster than the suggested time; see Section 3.

3. Experimental Results

We now apply our algorithms to solve either the alignment or the registration problems. Additional experiments on real-world scans from the SUN3D dataset [48] are placed in Section F at the appendix.

Datasets. We used the Bunny, Armadillo, and Asian Dragon models from the Stanford 3D scanning repository [9, 23, 43]. Those models were scaled to $[-0.5, 0.5]^3$ due to the constraints of some competing methods (e.g., `GO-ICP`). We also used a synthetic dataset comprising uniformly sampled d-dimensional points in $[-0.5, 0.5]^d$.

Generating P and Q. In all experiments, given some data model (real or synthetic), we uniformly sample n points named Q. An alignment (R, t) is generated, where t is uniformly sampled such that $\|t\| \leq 0.1$ and R rotates the data around each axis by an angle uniformly sampled from $[-\pi, \pi]$. P is then obtained by applying (R, t) to Q and adding Gaussian noise with zero mean and σ^2 variance. In Section 3.2 we also apply a random shuffle to P.

Evaluation. We present two evaluation metrics: (i) The value of the minimized cost function itself, e.g., the value of (1) or (3). If not given, the optimal correspondence is trivially computed, after applying (R, t) to P, via nearest neighbor. (ii) Rotation and translation errors: $\left\|R^T R^* - I\right\|_F$ and $\|t - t^*\|_2$, where (R^*, t^*) and (R, t) are the ground truth and the recovered alignments, respectively. Every experiment in this section was conducted 20 times and averaged. The variance is presented in the graphs.

3.1. Alignment Experiments

Here, we assume the correspondences function is given. We applied three algorithms: (i) `P-RANSAC`: Provable RANSAC - an implementation in Python of Algorithm 2, (ii) `RANSAC`: A RANSAC scheme equipped with the common least squares solution to recover an alignment, and (iii) `TEASER++`: The state of the art TEASER++ [49].

In Fig. 4 we compare the algorithms above, and in Fig. 5 we demonstrate `P-RANSAC`'s fast recovery of an alignment with an approximation constant close to 1, even in high dimensions. In the latter test, the ground truth solution (R^*, t^*) is computed via SVD as explained in Section 1.1.

Discussion. Fig. 4 demonstrates that Algorithm 2 outperforms state of the art methods, in multiple common metrics. Fig. 5 shows that it suffices for our algorithm, in practice, to sample roughly 40 subsets (in $d = 3$) until an error of at most x1.5 the globally minimal cost is obtained. Furthermore, recall that Algorithm 3 provides a probabilistic alternative for Algorithm 2, by reducing number of iterations in the cost of larger approximation constants. However, Algorithm 2 was sufficient in practice as it produced very small approximation constants in very few iterations.

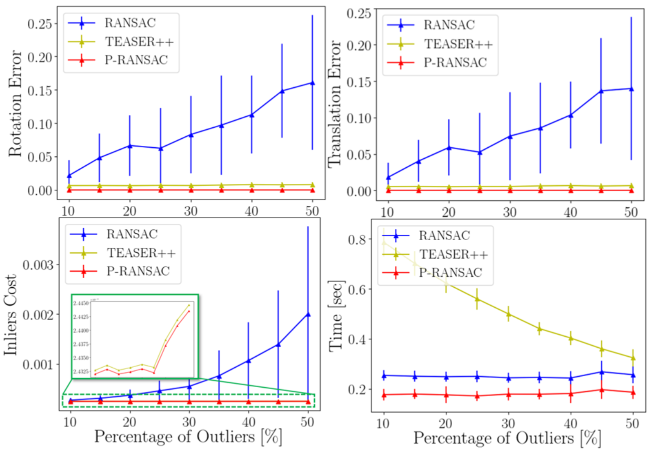

Figure 4: The Bunny model, $n = 2000$ points and $\sigma^2 = 0.009$ noise variance were used. The cost in our algorithm was SSD with M-estimator $\min\{\|p-q\|^2, 1\}$. The inliers cost is the mean error over the ground truth inlier pairs.

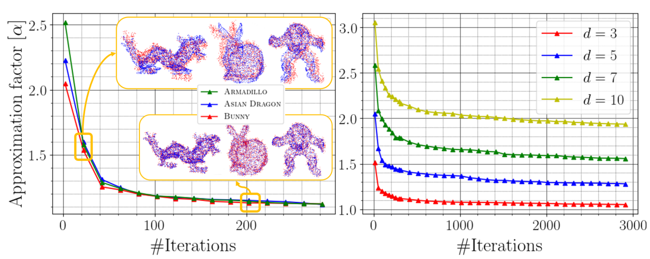

Figure 5: The approximation quality α (the cost of P-RANSAC *divided by the ground truth) as a function of the number of iteration γ. (Left): Bunny, Armadillo, and Asian Dragon models. The obtained alignment is also visualized. (Right): Synthetic data. In both figures $n = 2500$ and $\sigma^2 = 0.01$. The cost in our algorithm was the SSD.*

3.2. Registration Experiments

In this section we compared the following algorithms: **(i)** `ICP`(P, Q) - An implementation of the ICP algorithm [7]. **(ii)** `P-ICP`$(P, Q, \gamma, \text{cost})$ - Provable ICP; A parallelized implementation in Python of Algorithm 4. **(iii)** `P-ICP-Refined`$(P, Q, \gamma, \text{cost})$ - Applying the output alignment of `P-ICP` to the set P, then refining the alignment via single `ICP` run, on Q and the transformed P. As the ICP is guaranteed to converge to a local minimum, it can only help reduce the cost of our (approximately optimal) output result to the closest (hopefully global) minimum. **(iv)** `CPD`(P, Q) - The Coherent Point Drift algorithm [31]. **(v)** `GO-ICP`(P, Q) - The common ICP variant [50].

We tested multiple cost functions. Fig. 6 and Fig. 7 present the results for noisy input data, and data containing outliers respectively. Visual comparison is shown in Fig. 1.

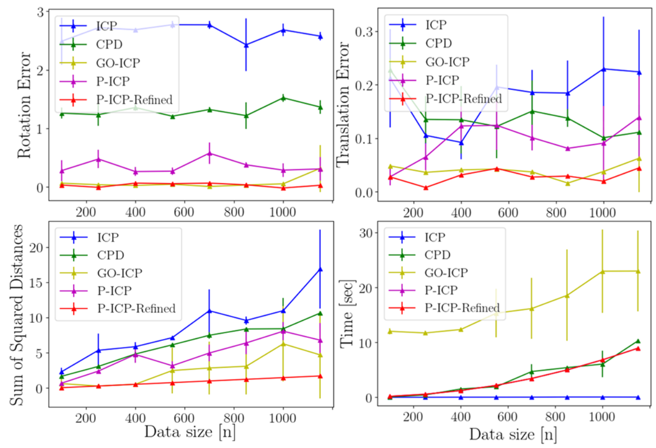

Figure 6: Armadillo model with $\sigma^2 = 0.01$ noise variance. The SSD cost function was used in our algorithms. The test was executed on the AWS platform, on a c5a.8xlarge machine with 32 CPUs.

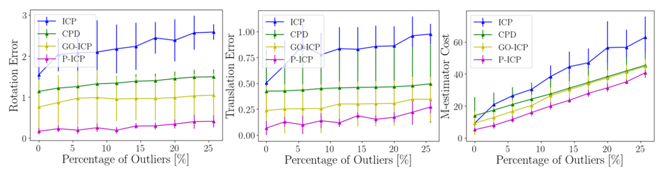

Figure 7: Robustness to outliers using the Armadillo model. $n = 800$ was used. Noise with variance $\sigma^2 = 1$ was added to k percentage of the points in P, which are considered as outliers. The SSD with M-estimator $\min\{\|p-q\|^2, 0.2\}$ was used in our algorithms. The computational time was roughly constant for each method for all tested k values, and is presented in Fig. 6 at $n = 800$.

Discussion. Fig. 6 demonstrates the accuracy of our algorithms, which yield an error smaller by x2-x10 than other methods, while also being among the fastest. Fig. 7 and 1 demonstrate our robustness to outliers in practice, due to our provable approximation to M-estimators cost functions.

4. Conclusions, Limitations, and Future Work

We present provable and practical non-trivial approximation algorithms for the alignment and registration problems and their hard variants. The algorithms rely on our proof that a witness set, which determines an approximated alignment, exists for both problems. Experiments show that our algorithms are efficient in practice, produce smaller errors, and are more stable than competing methods. The main limitation of our algorithm is their high running-time dependency on the dimension d. However, fortunately, for most applications d is a small constant. Future work includes: (i) generalizing to the non-rigid registration problem, and (ii) fast recovery of a witness set via deep learning.

References

[1] Pankaj K Agarwal and Jeff M Phillips. On bipartite matching under the rms distance. In *Proceedings of the Eighteenth Canadian Conference on Computational GeometryâĂŹ, Kingston, Canada*, pages 143–146, 2006. 3

[2] Dror Aiger, Niloy J Mitra, and Daniel Cohen-Or. 4-points congruent sets for robust pairwise surface registration. In *ACM SIGGRAPH 2008 papers*, pages 1–10. 2008. 3

[3] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7163–7172, 2019. 2

[4] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–607. International Society for Optics and Photonics, 1992. 2

[5] Peter Biber and Wolfgang Straßer. The normal distributions transform: A new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 3, pages 2743–2748. IEEE, 2003. 2

[6] Alvaro 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, 2017. 2

[7] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992. 2, 8

[8] Andrew DJ Cross and Edwin R Hancock. Graph matching with a dual-step em algorithm. *IEEE transactions on pattern analysis and machine intelligence*, 20(11):1236–1253, 1998. 2

[9] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996. 7

[10] Nadav Dym and Shahar Ziv Kovalsky. Linearly converging quasi branch and bound algorithms for global rigid registration. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1628–1636, 2019. 3

[11] Dan Feldman and Leonard J Schulman. Data reduction for weighted and outlier-resistant clustering. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 1343–1354. SIAM, 2012. 16, 17

[12] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2

[13] Miguel Angel Garcia and Agusti Solanas. 3d simultaneous localization and modeling from stereo vision. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 1, pages 847–853. IEEE, 2004. 1

[14] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. In *Linear Algebra*, pages 134–151. Springer, 1971. 2

[15] Oshri Halimi, Or Litany, Emanuele Rodola, Alex M Bronstein, and Ron Kimmel. Unsupervised learning of dense shape correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4370–4379, 2019. 3

[16] Jeroen Hermans, Dirk Smeets, Dirk Vandermeulen, and Paul Suetens. Robust point set registration using em-icp with information-theoretically optimal outlier handling. In *CVPR 2011*, pages 2465–2472. IEEE, 2011. 2

[17] Peter J Huber. *Robust statistics*. Springer, 2011. 2

[18] Anupam Joshi and C-H Lee. On the problem of correspondence in range data and some inelastic uses for elastic nets. *IEEE transactions on neural networks*, 6(3):716–723, 1995. 2

[19] Ibrahim Jubran and Dan Feldman. Aligning points to lines: Provable approximations. *IEEE Transactions on Knowledge and Data Engineering*, 2020. 5, 17

[20] Ibrahim Jubran, Alaa Maalouf, and Dan Feldman. Introduction to coresets: accurate coresets. *arXiv preprint arXiv:1910.08707*, 2019. 3

[21] Ibrahim Jubran, Alaa Maalouf, Ron Kimmel, and Dan Feldman. Open source code for the algorithms presented in this paper, 2021. Link for open-source code. 3

[22] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976. 2

[23] Venkat Krishnamurthy and Marc Levoy. Fitting smooth surfaces to dense polygon meshes. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 313–324, 1996. 7

[24] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 23

[25] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999. 1

[26] Bin Luo and Edwin R Hancock. A unified framework for alignment and correspondence. *Computer Vision and Image Understanding*, 92(1):26–55, 2003. 2

[27] Martin Magnusson, Achim Lilienthal, and Tom Duckett. Scan registration for autonomous mining vehicles using 3d-ndt. *Journal of Field Robotics*, 24(10):803–827, 2007. 1

[28] Ameesh Makadia, Alexander Patterson, and Kostas Daniilidis. Fully automatic registration of 3d point clouds. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 1297–1304. IEEE, 2006. 1, 3

[29] Graham McNeill and Sethu Vijayakumar. A probabilistic approach to robust shape matching. In *2006 International Conference on Image Processing*, pages 937–940. IEEE, 2006. 2

[30] Nicolas Mellado, Dror Aiger, and Niloy J Mitra. Super 4pcs fast global pointcloud registration via smart indexing. In *Computer Graphics Forum*, volume 33, pages 205–215. Wiley Online Library, 2014. 3

[31] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32(12):2262–2275, 2010. 2, 8

[32] Soliman Nasser, Ibrahim Jubran, and Dan Feldman. Coresets for kinematic data: From theorems to real-time systems.

arXiv preprint arXiv:1511.09120, 2015. 1, 3

[33] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136. IEEE, 2011. 1

[34] Carl Olsson, Olof Enqvist, and Fredrik Kahl. A polynomial-time bound for matching and registration with outliers. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 3

[35] Carl Olsson, Fredrik Kahl, and Magnus Oskarsson. Branch-and-bound methods for euclidean registration problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):783–794, 2008. 3

[36] OptiTrack. Motion capture systems. "https://www.optitrack.com". 1

[37] François Pomerleau, Francis Colas, Roland Siegwart, et al. A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends® in Robotics*, 4(1):1–104, 2015. 1

[38] François Pomerleau, Francis Colas, Roland Siegwart, and Stéphane Magnenat. Comparing icp variants on real-world data sets. *Autonomous Robots*, 34(3):133–148, 2013. 2

[39] Roman Rabinovich, Ibrahim Jubran, Aaron Wetzler, and Ron Kimmel. Cobe-coded beacons for localization, object tracking, and slam augmentation. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 1367–1374. IEEE, 2020. 1

[40] Joaquim Salvi, Carles Matabosch, David Fofi, and Josep Forest. A review of recent range image registration methods with accuracy evaluation. *Image and Vision computing*, 25(5):578–596, 2007. 1

[41] Gary KL Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank C Langbein, Yonghuai Liu, David Marshall, Ralph R Martin, Xian-Fang Sun, and Paul L Rosin. Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. *IEEE transactions on visualization and computer graphics*, 19(7):1199–1217, 2012. 2

[42] Yanghai Tsin and Takeo Kanade. A correlation-based approach to robust point set registration. In *European conference on computer vision*, pages 558–569. Springer, 2004. 3

[43] Greg Turk and Marc Levoy. Zippered polygon meshes from range images. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 311–318, 1994. 7

[44] Vicon. Motion systems. "https://www.vicon.com/". 1

[45] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3523–3532, 2019. 2

[46] William M Wells III. Statistical approaches to feature-based object recognition. *International Journal of Computer Vision*, 21(1-2):63–98, 1997. 2

[47] Thomas Whelan, Michael Kaess, Hordur Johannsson, Maurice Fallon, John J Leonard, and John McDonald. Real-time large-scale dense rgb-d slam with volumetric fusion. *The International Journal of Robotics Research*, 34(4-5):598–626, 2015. 1

[48] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *Proceedings of the IEEE international conference on computer vision*, pages 1625–1632, 2013. 7, 24

[49] Heng Yang, Jingnan Shi, and Luca Carlone. Teaser: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 2020. 2, 7

[50] 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, 2015. 3, 8

[51] Ji Zhang and Sanjiv Singh. Visual-lidar odometry and mapping: Low-drift, robust, and fast. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2174–2181. IEEE, 2015. 1

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

A. Existence of a Witness Set

In what follows, we denote by $\vec{0}$ the origin of $\mathbb{R}^d$. The d dimensional identity matrix is denoted by $I_d \in \mathbb{R}^{d\times d}$. For $p \in \mathbb{R}^d$ we denote by $\text{proj}(p, X)$ its projection on a set $X \subseteq \mathbb{R}^d$, that is, $\text{proj}(p, X) \in \arg\min_{x\in X} \text{dist}(p, x)$. For a set of vectors $P \subseteq \mathbb{R}^d$, we denote by $\text{sp}(P)$ the linear span of the set P. For a single vector $p \in \mathbb{R}^d$ we abuse notation and denote $\text{sp}(\{p\})$ by $\text{sp}(p)$ for short.

In what follows, given some Linear subspace X of $\mathbb{R}^d$, we define $\mathcal{R}_X$ to be the set of all rotation matrices R such that $p \in X$ if and only if $Rp \in X$; see Fig. 8.

Definition 6. *Let $\tau \in \{0, \cdots, d\}$, let X be a τ-dimensional subspace of $\mathbb{R}^d$, and let $V_X \in \mathbb{R}^{d\times d}$ be a unitary arbitrary matrix whose τ leftmost columns span X. We define*

$$\mathcal{R}_X = \left\{ V_X \begin{pmatrix} R & \mathbf{0} \\ \mathbf{0} & I_{d-\tau} \end{pmatrix} V_X^T \mid R \in \text{SO}(\tau) \right\},$$

for $\tau \geq 2$, and $\mathcal{R}_X = \{I_d\}$ otherwise.

For example, if X is spanned by the first τ axis of $\mathbb{R}^d$ then $\mathcal{R}_X$ contains all the rotation matrices in $\text{SO}(d)$ which, when multiplied by any point in $p \in \mathbb{R}^d$, affect only the first τ coordinates of p. If not, V_X^T aligns X with the first τ axis of $\mathbb{R}^d$ and V_X does the inverse rotation; see Fig. 8.

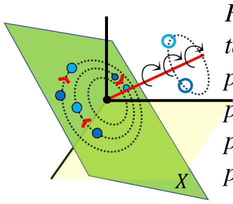

Figure 8: A plane X in $\mathbb{R}^3$. $\mathcal{R}_X$ contains all the rotation matrices that map points $p \in X$ (solid dark blue) to other points $p' \in X$ (solid light blue), and points $q \notin X$ (dark blue circles) to points $q' \notin X$ (light blue circles).

Overview of Claim 6.1. Let $R \in \mathcal{R}_X$ be an arbitrary rotation matrix that rotates a 2-dimensional plane X in $\mathbb{R}^d$, for example, a rotation matrix that affects only the first two coordinates of the points it multiplies. In this example, X is the xy-plane. In what follows we formally prove that displacement of a unit vector $\hat{p} \in X$, after multiplication with R (i.e., $\|R\hat{p} - \hat{p}\|$), must be greater than or equal to the displacement of a unit vector $\hat{q} \notin X$ after multiplication with the same R (i.e., $\|R\hat{q} - \hat{q}\|$).

Claim 6.1. *Let $d \geq 2$ be an integer. Let X be a 2-dimensional subspace (plane) of $\mathbb{R}^d$, let $p \in X \setminus \left\{\vec{0}\right\}$, and let $R \in \mathcal{R}_X$ be a rotation matrix that rotates every $x \in X$ by at most $\theta \in [-\pi/2, \pi/2]$ radians around the origin. Then for every $q \in \mathbb{R}^d \setminus \left\{\vec{0}\right\}$*

$$\frac{\|Rq - q\|}{\|q\|} \leq \frac{\|Rp - p\|}{\|p\|}.$$

Proof. Without loss of generality, assume that X is spanned by the standard vectors $e_1, e_2 \in \mathbb{R}^d$. Otherwise rotate the coordinates system. Since $R \in \mathcal{R}_X$ and X is spanned by the first two standard basis vectors, R can be expressed as

$$R = \begin{pmatrix} R' & \mathbf{0} \\ \mathbf{0} & I_{d-2} \end{pmatrix},$$

where $R' = \begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix}$ is a two dimensional rotation matrix, for some $\alpha \in [-\pi/2, \pi/2]$; see Definition 6.

Let $p' \in \mathbb{R}^2$ denote the first two entries of p, and $q' \in \mathbb{R}^2$ denote the first two entries of q. Notice that

$$\|p\| = \|p'\| \tag{4}$$

since $p \in X$ and X is spanned by e_1 and e_2. By the definition of the matrix R, we have that

$$\|Rp - p\| = \|R'p' - p'\| \text{ and } \|Rq - q\| = \|R'q' - q'\|. \tag{5}$$

Therefore,

$$\frac{\|Rq - q\|^2}{\|q\|^2} = \frac{\|R'q' - q'\|^2}{\|q\|^2} = \frac{2\|q'\|^2 - 2q'^T(R'q')}{\|q\|^2} \tag{6}$$

$$\leq \frac{2\|q'\|^2 - 2q'^T(R'q')}{\|q'\|^2} = \frac{2\|p'\|^2 - 2p'^T(R'p')}{\|p'\|^2} \tag{7}$$

$$= \frac{\|R'p' - p'\|^2}{\|p'\|^2} = \frac{\|R'p' - p'\|^2}{\|p\|^2} \tag{8}$$

$$= \frac{\|Rp - p\|^2}{\|p\|^2}, \tag{9}$$

where the first derivation in (6) is by (5), the first derivation in (7) holds since $\|q'\| \leq \|q\|$, the second derivation in (7) holds since $\frac{q'^T(R'q')}{\|q'\|^2} = \frac{p'^T(R'p')}{\|p'\|^2}$, the first inequality in (8) holds is by (4), and (9) is by (5).

Claim 6.1 now holds by taking the squared root of (8). □

Overview of Lemma 7. For two vectors $p, q \in \mathbb{R}^d$, we define the *angle between p and q* as the smallest angle between them when considering the 2-dimensional subspace that they span.

Now, consider two ordered sets of points P and Q contained in some subspace π of $\mathbb{R}^d$, and let $p \in P$ and $q \in Q$ be a pair of corresponding points, whose angle between them is the smallest among all n corresponding pairs. Let $R \in \text{SO}(d)$ be a rotation matrix that aligns the direction of p with the direction of q, i.e., $Rp \in \text{sp}(q)$. Then the following lemma proves that after rotating P by R, the distance from every point in P to its corresponding point in Q

will at most increase by a multiplicative factor of $(1+\sqrt{2})$. Furthermore, R does not affect any point outside of π, i.e., $R \in \mathcal{R}_\pi$.

Lemma 7. *Put $r \in [d]$. Let π be an r-dimensional subspace of $\mathbb{R}^d$, $P = \{p_1, \cdots, p_n\} \subset \pi$ and $Q = \{q_1, \cdots, q_n\} \subset \pi$ be two ordered sets of points. Put $\mathcal{R}^*$ in $\mathrm{SO}(d)$. Then there is an index $j \in [n]$ and a rotation matrix $R' \in \mathcal{R}_\pi$ that satisfy the following properties:*

(i) For every $i \in [n]$,

$$\|R'p_i - q_i\| \leq (1+\sqrt{2}) \cdot \|R^* p_i - q_i\| \tag{10}$$

and

(ii) $R'p_j \in \mathrm{sp}(q_j)$ and $\|p_j\|, \|q_j\| \neq 0$.

Proof. Throughout this proof, for simplicity of notation, we assume that the points of P have already been rotated by the rotation matrix R^*, i.e., we assume that R^* is the identity matrix. Therefore, (10) reduces to

$$\|R'p_i - q_i\| \leq (1+\sqrt{2}) \cdot \|p_i - q_i\|$$

for every $i \in [n]$.

We first observe that if $\|p_i\| = 0$ or $\|q_i\| = 0$ for some $i \in [n]$, then for any rotation matrix R we have that $\|Rp_i - q_i\| = \|p_i - q_i\|$. Hence, for the rest of the proof, we assume for simplicity that $\|p_i\|, \|q_i\| \neq 0$ for every $i \in [n]$.

In what follows we pick a pair of corresponding input points p_j, q_j which have the smallest angle between them around the origin, among all the input pairs (ties broken arbitrarily). We then show that there is a rotation matrix R_j that aligns the direction vectors of p_j and q_j and satisfies the requirements of the lemma.

For every $i \in [n]$, let $X_i = \mathrm{sp}(\{p_i, q_i\})$ be the plane spanned by p_i and q_i, and $R_i \in \mathcal{R}_{X_i}$ be a rotation matrix that satisfies $R_i p_i \in \mathrm{sp}(q_i)$, i.e., aligns the directions of the vectors p_i and q_i by a rotation in the 2-dimensional subspace (plane) X_i that these pair of vectors span. If there is more than one such rotation matrix, pick an arbitrary one among the (possible two) which rotate p_i with the smallest angle of rotation.

Put $i \in [n]$. Let π_i be a 2-dimensional subspace that contains p_i and $R_i p_i$. By the definition of R_i, we have that $R_i p_i \in \mathrm{sp}(q_i) \subseteq \pi_i$. Let $j \in \arg\min_{k \in [n]} \frac{\|R_k p_k - p_k\|}{\|p_k\|}$, i.e., j is the index of the corresponding pair that have the smallest angle among all pair of corresponding points.

We now prove that the distance $\|R_j p_i - q_i\|$ between the corresponding pair $p_i \in P$ and $q_i \in Q$ after applying R_j is larger by at most a multiplicative factor of $(1+\sqrt{2})$ compared to their original distance $\|p_i - q_i\|$. We have that

$$\frac{\|R_j p_i - p_i\|}{\|p_i\|} \leq \frac{\|R_j p_j - p_j\|}{\|p_j\|} \leq \frac{\|R_i p_i - p_i\|}{\|p_i\|}, \tag{11}$$

where the first inequality holds by substituting $p = p_j, q = p_i$ and $R = R_j$ in Claim 6.1, and the second inequality holds by the definition of j. Multiplying (11) by $\|p_i\|$ yields

$$\|R_j p_i - p_i\| \leq \|R_i p_i - p_i\|. \tag{12}$$

We now prove Lemma 7 (i) for $R' = R_j$, i.e., we prove that

$$\|R_j p_i - q_i\| \leq (1+\sqrt{2}) \cdot \|p_i - q_i\|. \tag{13}$$

Lemma 7 (ii) then follows by the definition of R_j.

We prove (13) by the following case analysis: **(i)** R_i is the identity matrix, **(ii)** $\|q_i\| > \|p_i\|$ and R_i is not the identity matrix, and **(iii)** $\|q_i\| \leq \|p_i\|$ and R_i is not the identity matrix.

Case (i): R_i is the identity matrix. In this case we have that $\|R_i p_i - p_i\| = 0$. Therefore, by the definition of j, we have that $\|R_j p_j - p_j\| = 0$, i.e., R_j is also the identity matrix. Hence, Case (i) trivially holds as $\|R_j p_i - q_i\| = \|p_i - q_i\| \leq (1+\sqrt{2}) \cdot \|p_i - q_i\|$.

Case (ii): $\|q_i\| > \|p_i\|$ and R_i is not the identity matrix. By the definition of proj, we have that $p_i - \mathrm{proj}(p_i, \mathrm{sp}(q_i))$ is orthogonal to q_i. Combining this with the fact that the vector $(\mathrm{proj}(p_i, \mathrm{sp}(q_i)) - q_i)$ has the same direction as q_i yields that $(p_i - \mathrm{proj}(p_i, \mathrm{sp}(q_i)))$ is orthogonal to $(\mathrm{proj}(p_i, \mathrm{sp}(q_i)) - q_i)$, i.e.,

$$(p_i - \mathrm{proj}(p_i, \mathrm{sp}(q_i)))^T (\mathrm{proj}(p_i, \mathrm{sp}(q_i)) - q_i) = 0. \tag{14}$$

Similarly, we have that $(p_i - \mathrm{proj}(p_i, \mathrm{sp}(q_i)))$ is orthogonal to $(\mathrm{proj}(p_i, \mathrm{sp}(q_i)) - R_i p_i)$. Therefore, by applying the Pythagorean theorem in the right triangle $\Delta(p_i, \mathrm{proj}(p_i, \mathrm{sp}(q_i)), R_i p_i)$ we obtain

$$\begin{aligned}\|p_i - \mathrm{proj}(p_i, \mathrm{sp}(q_i))\|^2 + \|\mathrm{proj}(p_i, \mathrm{sp}(q_i)) - R_i p_i\|^2 \\ = \|p_i - R_i p_i\|^2.\end{aligned} \tag{15}$$

Observer that

$$\|\mathrm{proj}(p_i, \mathrm{sp}(q_i))\| \leq \|p_i\| = \|R_i p_i\| < \|q_i\|, \tag{16}$$

where the last derivation is by the assumption of Case (ii). Combining (16) with the fact that $\mathrm{proj}(p_i, \mathrm{sp}(q_i)), R_i p_i$ and q_i lie on the straight line $\mathrm{sp}(q_i)$, we obtain that

$$\begin{aligned}&\|\mathrm{proj}(p_i, \mathrm{sp}(q_i)) - q_i\| \\ &= \|\mathrm{proj}(p_i, \mathrm{sp}(q_i)) - R_i p_i\| + \|R_i p_i - q_i\|.\end{aligned} \tag{17}$$

Therefore

$$\begin{aligned}\|p_i - q_i\|^2 &= \|p_i - \text{proj}(p_i, \text{sp}(q_i)) + \text{proj}(p_i, \text{sp}(q_i)) - q_i\|^2 \\ &= \|p_i - \text{proj}(p_i, \text{sp}(q_i))\|^2 \\ &\quad + \|\text{proj}(p_i, \text{sp}(q_i)) - q_i\|^2 \qquad (18) \\ &= \|p_i - \text{proj}(p_i, \text{sp}(q_i))\|^2 \\ &\quad + (\|\text{proj}(p_i, \text{sp}(q_i)) - R_i p_i\| + \|R_i p_i - q_i\|)^2 \qquad (19) \\ &= \|p_i - \text{proj}(p_i, \text{sp}(q_i))\|^2 \\ &\quad + \|\text{proj}(p_i, \text{sp}(q_i)) - R_i p_i\|^2 + \|R_i p_i - q_i\|^2 \\ &\quad + 2\,\|\text{proj}(p_i, \text{sp}(q_i)) - R_i p_i\| \cdot \|R_i p_i - q_i\| \\ &= \|R_i p_i - p_i\|^2 + \|R_i p_i - q_i\|^2 \\ &\quad + 2\,\|\text{proj}(p_i, \text{sp}(q_i)) - R_i p_i\| \cdot \|R_i p_i - q_i\| \qquad (20) \\ &\geq \|R_i p_i - p_i\|^2 , \qquad (21)\end{aligned}$$

where (18) is by (14), (19) is by (17), and (20) is by (15). Combining (12) and (21) yields

$$\|R_j p_i - p_i\| \leq \|p_i - q_i\| . \qquad (22)$$

Hence, (13) holds for Case (ii) as

$$\|R_j p_i - q_i\| \leq \|R_j p_i - p_i\| + \|p_i - q_i\| \leq 2 \cdot \|p_i - q_i\| ,$$

where the first derivation holds by the triangle inequality, and the second derivation by (22).

Case (iii): $\|q_i\| \leq \|p_i\|$ and R_i is not the identity matrix. Since R_i is not the identity matrix, then p_i and $R_i p_i$ are distinct points. Let $q^* = \text{proj}(p_i, \text{sp}(R_i p_i))$. Combining the definition of q^* with the fact that p_i and $R_i p_i$ are distinct points, we obtain that all three points $p_i, R_i p_i$ and q^* are distinct.

Consider the triangle $\Delta(p_i, R_i p_i, \vec{0})$. Let $\alpha \in [0, \pi/2]$ be the angle at vertex $\vec{0}$ of the triangle; see Fig. 9 for illustration. Since $\|p_i\| = \|R_i p_i\|$ we obtain that the angles at vertices p_i and $R_i p_i$ is $\beta = \frac{\pi - \alpha}{2}$. Consider triangle $\Delta(p_i, R_i p_i, q^*)$, and observe that the angle at vertex q^* is $\pi/2$. It therefore holds that

$$\begin{aligned}\|R_i p_i - p_i\| &= \frac{\|p_i - q^*\| \sin \pi/2}{\sin \frac{\pi - \alpha}{2}} = \frac{\|p_i - q^*\|}{\sin \frac{\pi - \alpha}{2}} \\ &\leq \frac{\|p_i - q^*\|}{\sin \frac{\pi - \pi/2}{2}} = \sqrt{2} \cdot \|p_i - q^*\| \leq \sqrt{2} \cdot \|p_i - q_i\| , \qquad (23)\end{aligned}$$

where the firsst derivation is by applying the law of sines in triangle $\Delta(p_i, R_i p_i, q^*)$, and the last derivation is by the definition of q^*.

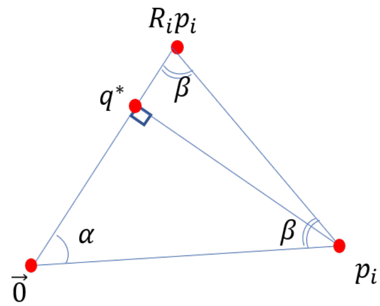

Figure 9: A triangle $\Delta(p_i, R_i p_i, \vec{0})$, a point $q^* = \text{proj}(p_i, \text{sp}(R_i p_i))$ and a right triangle $\Delta(p_i, q^*, \vec{0})$.

Hence, (13) holds for Case (iii) as

$$\begin{aligned}\|R_j p_i - q_i\| &\leq \|R_j p_i - p_i\| + \|p_i - q_i\| \\ &\leq \|R_i p_i - p_i\| + \|p_i - q_i\| \qquad (24) \\ &\leq (1 + \sqrt{2}) \cdot \|p_i - q_i\| ,\end{aligned}$$

where the fist derivation holds by the triangle inequality, the second derivation is by (12), and the last derivation holds by 23.

□

Theorem 8 (Witness sets)**.** *Let $P = \{p_1, \cdots, p_n\}$ and $Q = \{q_1, \cdots, q_n\}$ be two ordered sets each of n points in $\mathbb{R}^d$. Then, for every alignment (R^*, t^*) and matching function m^*, there exist $P' \subseteq P$ and $Q' \subseteq Q$ of size $|P'| = |Q'| = d$ such that the output (R, t) of the call* ALIGN(P', Q') *to Algorithm 1 satisfies the following for every $i \in [n]$:*

$$\left\|R p_i - t - q_{m^*(i)}\right\| \leq (1 + \sqrt{2})^d \cdot \left\|R^* p_i - t^* - q_{m^*(i)}\right\|$$

Furthermore, (R, t) is computed in $O(d^3)$ time.

Proof. Let m^* be a matching function. Without loss of generality assume that $m^*(i) = i$. Otherwise, shuffle the points of Q according to m^*.

Put $(R^*, t^*) \in$ ALIGNMENTS(d). Without loss of generality assume that R^* is the identity matrix and that t^* is a vector of zeros, otherwise rotate and translate the set P by (R^*, t^*).

Recovering an approximated translation.

Claim 8.1. *There is an index $k \in [n]$ such that for $t = p_k - q_k$ we have that*

$$\|p_i - t_k - q_i\| \leq 2 \cdot \|p_i - q_i\| ,$$

for every $i \in [n]$.

Proof. For every $i \in [n]$, let $t_i = p_i - q_i$ and let $k \in \arg\min_{i \in [n]} \|t_i\|$. Put $i \in [n]$. Since $\|t_k\| \leq \|t_i\| =$

$\|p_i - q_i\|$, it holds that

$$\begin{aligned}\|p_i - t_k - q_i\| &\le \|p_i - t_k - p_i\| + \|p_i - q_i\| \\ &= \|t_k\| + \|p_i - q_i\| \\ &\le 2 \cdot \|p_i - q_i\|,\end{aligned} \tag{25}$$

where the first derivation is by the triangle inequality. □

Hence, there exists an index $k \in [n]$ and a vector $t = p_k - q_k$ such that $\|p_i - t_k - q_i\| \le 2 \cdot \|p_i - q_i\|$ for every $i \in [n]$.

Observe that, by definition of t_k in the above claim, $p_k - t_k = q_k$. Hence, p_k and q_k intersect after applying the translation t_k to P. This proves the existence of a translation that aligns a corresponding pair of points from P and Q, and yields a provable constant factor approximation. This translation can afterwards be easily recovered.

Recovering an approximated rotation. Let k be the index from Claim 13.2. As discussed above, p_k and q_k intersect after applying t_k to P. Translating both sets by the same translation does not change the pairwise distances. Hence, we will translate both sets again such that q_k and $p_k - t_k$ intersect the origin. In other words, we redefine the original (untranslated) P and Q as follows:

$$P := \{p - t_k - q_k \mid p \in P\}, Q := \{q - q_k \mid q \in Q\}. \tag{26}$$

We now prove the existence of some rotation matrix R, which provides a constant factor approximation to the optimal rotation matrix, using an iterative scheme with $d-1$ iterations.

We denote by $P^{(0)} = P$ and $Q^{(0)} = Q$, and for every $p \in P$ and $q \in Q$ we denote by $p^{(0)} = p$ and $q^{(0)} = q$. We denote by $p^{(j)}$ and $q^{(j)}$ the points p and q after the j'th iteration of the above $d-1$ iterations, and by $P^{(j)}$ and $Q^{(j)}$ the set P and Q after the j'th iteration respectively.

At the jth iteration ($j \in [d-1]$) we apply the following steps:

(i) We use Lemma 7 to prove the existence of an index i_j and some rotation matrix $R^{(j)}$ which aligns the direction vectors of at least one point $p_{i_j}^{(j-1)} \in P^{(j-1)}$ to its corresponding $q_{i_j}^{(j-1)}$. Lemma 7 also guarantees that $\left\|R^{(j)}p_i^{(j-1)} - q_i^{(j-1)}\right\| \le (1+\sqrt{2})\left\|p_i^{(j-1)} - q_i^{(j-1)}\right\|$ for every $i \in [n]$, i.e., applying this rotation to the points of the current $P^{(j-1)}$ does not increase each of the pairwise distances $\left\|p_i^{(j-1)} - q_i^{(j-1)}\right\|$ by more than a multiplicative constant. Furthermore, it guarantees that $\left\|p_{i_j}^{(j-1)}\right\|, \left\|q_{i_j}^{(j-1)}\right\| \neq 0$.

(ii) Let $\pi^{(j)}$ be a $(d-j)$-dimensional subspace orthogonal to $q_{i_j}^{(j-1)}$. We define

$$p^{(j)} = \text{proj}(R^{(j)}p^{(j-1)}, \pi^{(j)}) \tag{27}$$

and

$$q^{(j)} = \text{proj}(q^{(j-1)}, \pi^{(j)}). \tag{28}$$

We continue to the $(j+1)$'th iteration with the sets

$$P^{(j)} = \left\{p^{(j)} \mid p \in P \setminus \{p_{i_1}, \cdots, p_{i_j}\}\right\}$$

and

$$Q^{(j)} = \left\{q^{(j)} \mid q \in Q \setminus \{q_{i_1}, \cdots, q_{i_j}\}\right\},$$

i.e., we rotate the points of $P^{(j-1)}$ and then project the points of $P^{(j-1)}$ and $Q^{(j-1)}$, without the previously aligned points, onto a subspace orthogonal to $q_{i_j}^{(j-1)}$.

The projection at the end of each iteration ensures that the points of $P^{(j)}$ and $Q^{(j)}$ after the (j)'th iteration are orthogonal to $q_{i_1}^{(0)}, \cdots, q_{i_j}^{(j-1)}$. Hence, the matrix $R^{(j+1)}$ from the $(j+1)$'th iterations, which aligns $p_{i_{j+1}}^{(j)}$ and $q_{i_{j+1}}^{(j)}$, does not affect the prior alignment of the pairs $(p_{i_1}^{(0)}, q_{i_1}^{(0)}), \cdots, (p_{i_j}^{(j-1)}, q_{i_j}^{(j-1)})$.

After the above $d-1$ iterations, we end up with a series of $d-1$ rotation matrices $R^{(1)}, \cdots, R^{(d-1)}$ and $d-1$ corresponding pairs $(p_{i_1}, q_{i_1}), \cdots, (p_{i_{d-1}}, q_{i_{d-1}})$. Thos pairs determine a unique rotation matrix $R = R^{(d-1)} \cdots R^{(1)}$ since $q_{i_1}^{(0)}, \cdots, q_{i_{d-1}}^{(d-2)}$ are $d-1$ orthogonal points. Given those $d-1$ pairs, recovering this rotation matrix is trivial, as implemented in Algorithm 1.

It is left to prove that the matrix $R^{(d-1)} \cdots R^{(1)}$ yields a constant factor approximation, as follows.

Claim 8.2. *Put $r \in [d-1]$. Let $R'' = R^{(d-1)} \cdots R^{(r)}$ and $j = r-1$. Then $R'' \in \mathcal{R}_{\pi^{(j)}}$ and*

$$\|R''p - q\| \le (1+\sqrt{2})^{d-r} \|p - q\|$$

for every $p \in P^{(j)}$ and its corresponding $q \in Q^{(j)}$.

Proof. The proof is by induction on $r \in [d-1]$, where the base case is $r = d-1$.

Base case for $r = d-1$. By substituting $P^{(d-2)}$, $Q^{(d-2)}$, and $R^* = I$ in Lemma 7 (i) and (ii), we obtain, as required, that there is $i_1 \in [n]$ and $R' \in \mathcal{R}_{\pi^{(1)}}$ that satisfy the following pair of properties.

(i) $\|R'p - q\| \le (1+\sqrt{2}) \cdot \|p - q\|$ for every $p \in P^{(d-2)}$ and its corresponding $q \in Q^{(d-2)}$.

(ii) $R'p_{i_1} \in \text{sp}(q_{i_1})$ and $\|p_{i_1}\|, \|q_{i_1}\| \neq 0$.

Induction step for $r \in [d-2]$. We assume that Claim 8.2 holds for $r' \in [d-1] \setminus \{1\}$ and prove that it holds for $r = r'-1$, i.e., we assume that the matrix $R'' = R^{(d-1)} \cdots R^{(r')}$ satisfies that $R'' \in \mathcal{R}_{\pi^{r'-1}}$ and

$$\|R''p - q\| \le (1+\sqrt{2})^{d-r'} \|p - q\|, \tag{29}$$

for every $p \in P^{(r'-1)} = P^{(r)}$ and its corresponding $q \in Q^{(r'-1)} = Q^{(r)}$.

Recall that $j = r - 1$. We now prove that the matrix $R''R^{(r)} = R^{(d-1)} \cdots R^{(r)}$ satisfies that $R''R^{(r)} \in \mathcal{R}_{\pi^{(j)}}$ and

$$\left\| R''R^{(r)}p - q \right\| \leq (1+\sqrt{2})^{d-r} \left\| p - q \right\|$$

for every $p \in P^{(j)}$ and its corresponding $q \in Q^{(j)}$.

By plugging $P^{(j)}$ and $Q^{(j)}$ in Lemma 7, we obtain that there exists $i_r \in [n]$ and $R' \in \mathcal{R}_{\pi^{(j)}}$ that satisfy the following:

$$\left\| R'p - q \right\| \leq (1+\sqrt{2}) \cdot \left\| p - q \right\|, \tag{30}$$

for every $p \in P^{(j)}$ and its corresponding $q \in Q^{(j)}$.

Assume without loss of generality that $q_{i_r}^{(j)}$ spans the x-axis, otherwise rotate the coordinates system.

Let $\hat{\pi}^{(j)} \subset \pi^{(j)}$ be a j-dimensional subspace that is orthogonal to the x-axis and passes through the origin. Let $\pi_i^{(j)} \subset \pi^{(j)}$ be a j-dimensional affine subspace that is orthogonal to the x-axis and passes through $R'p_i^{(j)}$ and let $\hat{q}_i^{(j)} = \text{proj}(q_i^{(j)}, \pi_i^{(j)})$; see Fig. 10 for visualization.

For every $p \in \pi_i^{(j)}$ and rotation matrix $R \in \mathcal{R}_{\hat{\pi}^{(j)}}$, we have

$$\begin{aligned}
&\left\| Rp - q_i^{(j)} \right\|^2 = \left\| Rp - \hat{q}_i^{(j)} \right\|^2 + \left\| \hat{q}_i^{(j)} - q_i^{(j)} \right\|^2 \\
&= \left\| Rp - \hat{q}_i^{(j)} \right\|^2 + \text{dist}(q_i^{(j)}, \pi_i^{(j)})^2 \\
&= \left\| \text{proj}(Rp, \hat{\pi}^{(j)}) - \text{proj}(\hat{q}_i^{(j)}, \hat{\pi}^{(j)}) \right\|^2 + \text{dist}(q_i^{(j)}, \pi_i^{(j)})^2 \\
&= \left\| \text{proj}(Rp, \hat{\pi}^{(j)}) - \text{proj}(q_i^{(j)}, \hat{\pi}^{(j)}) \right\|^2 + \text{dist}(q_i^{(j)}, \pi_i^{(j)})^2 \\
&= \left\| R\text{proj}(p, \hat{\pi}^{(j)}) - \text{proj}(q_i^{(j)}, \hat{\pi}^{(j)}) \right\|^2 + \text{dist}(q_i^{(j)}, \pi_i^{(j)})^2,
\end{aligned} \tag{31}$$

where the first derivation holds by the Pythagorean theorem, the second derivation holds by the definition of $\hat{q}_i^{(j)}$, the third derivation holds by combining that $Rp, \hat{q}_i^{(j)} \in \pi_i^{(j)}$ and that $\pi_i^{(j)}$ and $\hat{\pi}^{(j)}$ are parallel, the fourth equality holds since $\text{proj}(\hat{q}_i^{(j)}, \hat{\pi}^{(j)}) = \text{proj}(q_i^{(j)}, \hat{\pi}^{(j)})$ and the last derivation holds by combining that $R \in \mathcal{R}_{\hat{\pi}^{(j)}}$ and that $\pi_i^{(j)}$ and $\hat{\pi}^{(j)}$ are parallel; see Fig. 10 for a visualization of the derivations in (31).

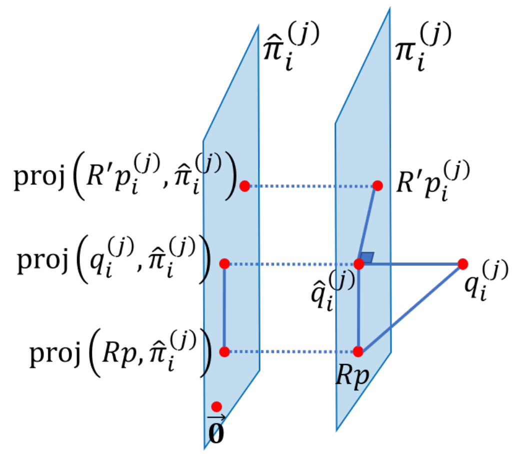

Figure 10: Illustration of the derivations in Eq. (31).

We now have that

$$\left\| R''R'p_i^{(j)} - q_i^{(j)} \right\|^2 \tag{32}$$

$$= \left\| R''\text{proj}(R'p_i^{(j)}, \hat{\pi}^{(j)}) - \text{proj}(q_i^{(j)}, \hat{\pi}^{(j)}) \right\|^2 + \text{dist}^2(q_i^{(j)}, \pi_i^{(j)}) \tag{33}$$

$$= \left\| R''p_i^{(r)} - q_i^{(r)} \right\|^2 + \text{dist}^2(q_i^{(j)}, \pi_i^{(j)}) \tag{34}$$

$$\leq (1+\sqrt{2})^{2(d-r-1)} \cdot \left\| p_i^{(r)} - q_i^{(r)} \right\|^2 + \text{dist}^2(q_i^{(j)}, \pi_i^{(j)}) \tag{35}$$

$$= (1+\sqrt{2})^{2(d-r-1)} \cdot \left\| \text{proj}(R'p_i^{(j)}, \pi_i^{(j)}) - \text{proj}(q_i^{(j)}, \pi_i^{(j)})) \right\|^2 + (1+\sqrt{2})^{2(d-r-1)} \cdot \text{dist}^2(q_i^{(j)}, \pi_i^{(j)}) \tag{36}$$

$$= (1+\sqrt{2})^{2(d-r-1)} \cdot \left\| R'p_i^{(j)} - \text{proj}(q_i^{(j)}, \pi_i^{(j)})) \right\|^2 + (1+\sqrt{2})^{2(d-r-1)} \cdot \text{dist}^2(q_i^{(j)}, \pi_i^{(j)}) \tag{37}$$

$$= (1+\sqrt{2})^{2(d-r-1)} \cdot \left\| R'p_i^{(j)} - q_i \right\|^2 \tag{38}$$

$$\leq (1+\sqrt{2})^{2(d-r)} \cdot \left\| p_i - q_i \right\|^2, \tag{39}$$

where (33) holds by substituting $p = R'p_i^{(j)}$ and $R = R''$ in (31), (34) is by combining that $j = r - 1$ with the definitions of $p^{(r)}$ and $q^{(r)}$ in (27) and (28) respectively, (35) holds by squaring both sides of (29), (36) holds by combining that that $\hat{\pi}$ and π_i are parallel with definitions of $p_i^{(r)}$ and $q_i^{(r)}$ in (27) and (28) respectively, (37) holds since $Rp_i^{(j)} \in \pi_i$, (38) is by the Pythagorean theorem, and (39) holds by (30); see Fig. 10.

Let $R = R''R'$. Hence, by (39), it follows that

$$\left\| Rp_i - q_i \right\| \leq (1+\sqrt{2})^{d-r} \cdot \left\| p_i - q_i \right\|. \tag{40}$$

By combining that $R' \in \mathcal{R}_{\pi^{(j)}}$ and $R'' \in \mathcal{R}_{\hat{\pi}^{(r'-1)}} \subseteq$

$\mathcal{R}_{\pi(j)}$, it also follows that

$$\hat{R} \in \mathcal{R}_{\pi(j)}. \tag{41}$$

By applying the induction step above $d-1$ times, we end up with a rotation matrix $R = R^{(d-1)} \cdots R^{(1)}$ that satisfies

$$\|Rp - q\| \leq (1+\sqrt{2})^{d-1} \|p - q\|,$$

for every $p \in P$ and its corresponding $q \in Q$. Furthermore, R is guaranteed to align the pairs $(p_{i_1}, q_{i_1}), \cdots, (p_{i_d}^{d-1}, q_{i_d}^{(d-1)})$. Hence, R satisfies Claim 8.2. □

Combining the (re-)definitions of P and Q in (26) with Claim 8.2 proves that

$$\|Rp' - q'\| \leq (1+\sqrt{2})^{d-1} \|p' - q'\|, \tag{42}$$

where $p' = p - t_k - q_k = p - p_k$ and $q' = q - q_k$ for every $p \in P$ and $q \in Q$.

Combining the approximated translation and rotation. Let $k \in [n]$ and $t_k = p_k - q_k$ be the index and approximated translation vector from Claim 8.1. Let R be the approximated rotation as in (8.2) for $r = 1$, and let $t = R'p_k - q_k$. Let $p' = p - t_k - q_k = p - p_k$ and $q' = q - q_k$ for every $p \in P$ and $q \in Q$. Now, consider the alignment (R, t). Observe that for every $i \in [n]$ we have

$$\|Rp_i - t - q_i\| = \|R(p_i - p_k) + q_k - q_i\| = \|Rp'_i - q'_i\|. \tag{43}$$

Put $i \in [n]$. We now obtain that

$$\|Rp_i - t - q_i\| = \|Rp'_i - q'_i\| \tag{44}$$
$$\leq (1+\sqrt{2})^{d-1} \|p'_i - q'_i\| \tag{45}$$
$$= (1+\sqrt{2})^{d-1} \|p_i - t_k - q_i\| \tag{46}$$
$$\leq (1+\sqrt{2})^{d} \|p_i - q_i\| \tag{47}$$

where (44) is by (43), (45) holds by (42), (46) holds similarly to (43), and (47) is by Claim 13.2.

Hence, each pairwise distance $\|Rp_i - t - q_i\|$ after applying alignment (R, t) is a constant factor approximation to the original optimal pairwise distance $\|p_i - q_i\| = \|R^*p_i - t^* - q_i\|$

Computing (R, t)**.** By the claims above, given the indices $i_1, \cdots, i_{d-1}$, it is straight forward to compute the rotation matrix R. Given R and the index k, it is straight forward to compute $t = Rp_k - q_k$. Algorithm 1 is a direct implementation of the alignment (R, t) above given two sets of d corresponding points from P and Q. Hence, there exists a subset $P' = \left\{p_k, p_{i_1}, \cdots, p_{i_{d-1}}\right\} \subseteq P$ and its corresponding $Q' = \left\{q_k, q_{i_1}, \cdots, q_{i_{d-1}}\right\} \subseteq Q$ that, when plugged into Algorithm 1, produces the desired approximated alignment (R, t) which satisfies

$$\begin{aligned}\left\|Rp_i - t - q_{m^*(i)}\right\| &= \|Rp_i - t - q_i\| \\ &\leq (1+\sqrt{2})^d \cdot \|p_i - q_i\| \\ &= (1+\sqrt{2})^d \cdot \left\|R^*p_i - t^* - q_{m^*(i)}\right\|.\end{aligned}$$

for every $i \in [n]$. Here, the first derivation is by our assumption that m^* is the identity function, the second derivation is by (47), and the last derivation is by our assumption that R^* is the identity matrix, t^* is a zeros vector, and m^* is the identity function.

By step (i) of the iterative scheme above, Lemma 7 guarantees that $\left\|p_{i_j}^{(j-1)}\right\|, \left\|q_{i_j}^{(j-1)}\right\| \neq 0$. In other words, the pair of aligned vectors at the j'th iteration are non-zero vectors. Recall that the two sets have been initially translated such that p_k and q_k intersected the original. Then, for every $j \in [d-1]$, the j'th aligned pair has been projected $j-1$ times onto subspaces orthogonal to the prior $j-1$ aligned pairs. Hence, we obtain that the original points $p_k, p_{i_1}, \cdots, p_{i_{d-1}}$ must span $\mathbb{R}^d$, and similarly $q_k, q_{i_1}, \cdots, q_{i_{d-1}}$ must also span $\mathbb{R}^d$.

Given some d indices $k, i_1, \cdots, i_{d-1}$ where $p_k, p_{i_1}, \cdots, p_{i_{d-1}}$ and $q_k, q_{i_1}, \cdots, q_{i_{d-1}}$ each span $\mathbb{R}^d$, Algorithm 1 computes the unique alignment (R, t) for which p_k intersects with q_k, p_{i_1} aligns with q_{i_1}, p_{i_2} aligns with q_{i_2} after the projection onto the hyperplane orthogonal to q_{i_1}, and so on. If there is no such sets of points which span $\mathbb{R}^d$, then the original data is entirely contained in a subspace of dimension smaller than d. In such case, we can simply reduce the dimensionality of the input sets, then apply the algorithm to the data of smaller dimension.

The running time of Algorithm 1 is $O(d^3)$ since there are at most d iterations, where each iteration takes $O(d^2)$ time to execute.

□

B. Generalization

In what follows is a formal definition of an r-log-Lipschitz function, which is a generalization of the definition in [11] from 1 to n dimensional functions. Intuitively, an r-log-Lipschitz function is a function whose derivative may be large but cannot increase too rapidly (in a rate that depends on r). For a higher dimensional function, we demand the previous constraint over every dimension individually.

For every pair of vectors $v = (v_1, \cdots, v_n)$ and $u = (u_1, \cdots, u_n)$ in $\mathbb{R}^n$ we denote $v \leq u$ if $v_i \leq u_i$ for every $i \in [n]$. Similarly, $f : \mathbb{R}^n \to [0, \infty)$ is non-decreasing if $f(v) \leq f(u)$ for every $v \leq u \in \mathbb{R}^d$.

Definition 9 (Log-Lipschitz function)**.** *Let $n \geq 1$, $r > 0$ and let $h : [0, \infty)^n \to [0, \infty)$ be a non-decreasing function.*

*Then $h(x)$ is r-*log-Lipschitz *if for every $x \in [0,\infty)$ and $c>0$ we have that $h(cx) \leq c^r h(x)$. For $n \geq 2$, a function $h:[0,\infty)^n \to [0,\infty)$ is called a The parameter r is called the* log-Lipschitz constant.

The following observation states that if we find an alignment $(R,t) \in \textsc{Alignments}(d)$ that approximates the function D for every input element, then it also approximates the more complex function cost as defined in Definition 2.

Observation 10 (Observation 5 in [19])**.** *Let $P = \{p_1, \cdots, p_n\}$ and $Q = \{q_1, \cdots, q_n\}$ and* cost *be as defined in Definition 2. Let $(R^*, t^*), (R,t) \in \textsc{Alignments}(d)$, let m^* be a correspondence function, and let $c \geq 1$. If $D\left(Rp_i - t, q_{m^*(i)}\right) \leq c \cdot D\left(R^* p_i - t^*, q_{m^*(i)}\right)$ for every $i \in [n]$, then*

$$\text{cost}\left(P_{[m^*]}, Q, (R,t)\right) \leq c^{rs} \cdot \text{cost}\left(P_{[m^*]}, Q, (R^*, t^*)\right).$$

C. Approximation for the Alignment Problem

Theorem 11 (Theorem 3)**.** *Let P and Q be two ordered sets of n points in $\mathbb{R}^d$, $\gamma \in \Omega(n^d)$, $z>0$, and $w = d^{\left|\frac{1}{z}-\frac{1}{2}\right|}$. Let* cost*, r and s be as defined in Definition 2 for $D(p,q) = \|p-q\|_z$. Let (R,t) be the output of a call to* $\textsc{Approx-Alignment}(P,Q,\gamma,\text{cost})$*; See Algorithm 2. Then,*

$$\text{cost}(P,Q,(R,t)) \leq w^{rs} \cdot (1+\sqrt{2})^{drs} \cdot \min_{(R',t')} \text{cost}(P,Q,(R',t')),$$

where the minimum is over every $(R',t') \in \textsc{Alignments}$. Moreover, (R,t) is computed in $n^{O(d)}$ time.

Proof. Let $(R^*, t^*) \in \min_{(R,t)} \text{cost}(P,Q,(R,t))$ be the optimal alignment for the cost at hand.

Theorem 8 proves the existence of a set $P' \subseteq P$ and a corresponding set $Q' \subseteq Q$ of size $|P'| = |Q'| = d$, such that the output (R,t) of the call $\textsc{Align}(P',Q')$ to Algorithm 1 satisfies the following for every $i \in [n]$:

$$\|Rp_i - t - q_i\| \leq (1+\sqrt{2})^d \cdot \|R^* p_i - t^* - q_i\|. \quad (48)$$

Here, every point $p_i \in P$ is assumed to correspond to $q_i \in Q$ since the matching function is given.

By (48) and since the ℓ_2-norm of every vector in $\mathbb{R}^d$ is approximated up to a multiplicative factor of $w = d^{\left|\frac{1}{z}-\frac{1}{2}\right|}$ by its ℓ_z-norm, for every $i \in [n]$ we have that

$$\|Rp_i - t - q_i\|_z \leq w(1+\sqrt{2})^d \cdot \|R^* p_i - t^* - q_i\|_z.$$

Combining the last equation, the definition of cost and D, and Observation 10 yields that

$$\begin{aligned}\text{cost}(P,Q,(R,t)) &\leq w^{rs} \cdot (1+\sqrt{2})^{drs} \cdot \text{cost}(P,Q,(R^*,t^*)) \\ &= w^{rs} \cdot (1+\sqrt{2})^{drs} \cdot \min_{(R,t)} \text{cost}(P,Q,(R,t)).\end{aligned}$$

To recover the above (R,t), we must recover the subsets P' and Q' and plug them into Algorithm 1. This can be done via exhaustive search over all $\theta(n^d)$ possible subsets of P of size d. Every such subset immediately dictates the corresponding subset $Q' \subseteq Q$ since the correspondence is given.

Observe that Algorithm 2 iterates over γ possible *distinct* corresponding subsets P', Q' of size d from P and q, plugs P', Q' into Algorithm 1, and returns the alignment which minimizes the given cost function (over all γ candidate alignments). Hence, plugging $\gamma \in \Omega(n^d)$ into Algorithm 2 yields an exhaustive search over all possible subsets from P and Q. Combining the fact that one of the γ computed alignments gives our desired approximation with the fact that Algorithm 2 returns the alignment with the smallest cost yields that the output is guaranteed to satisfy the requirements of Theorem 11.

The running time of Algorithm 2 is $n^{O(d)}$ since we make $O(n^d)$ calls to Algorithm 1, each call takes $O(d^3)$ time by Theorem 8. □

D. Run-Time Improvements

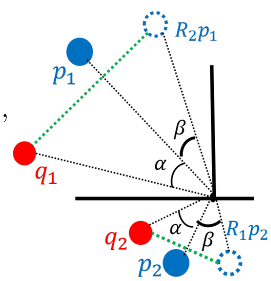

Figure 11: Corresponding pairs (p_1,q_1) and (p_2,q_2), where $\|p_1\| > \|p_2\|$. R_1 and R_2 are rotations that align (p_1,q_1) and (p_2,q_2) respectively. The cost of applying R_2 while damaging (p_1,q_1) is bigger than the cost of applying R_1 while damaging (p_2,q_2).

The following claim proves a weak version of the triangle inequality, for the cost functions we define in Definition 2.

Claim 11.1 (Weak triangle inequality.)**.** *Let $z>0$ be a constant. Let ℓ be an r-log-Lipschitz functions and $\mathrm{D}: \mathbb{R}^d \times \mathbb{R}^d \to [0,\infty)$ be a function such that $\mathrm{D}(p,q) = \|p-q\|_z$. Then, for every $p,q,v \in \mathbb{R}^d$,*

$$\ell\left(D(p,q)\right) \leq \rho c^r \left(\ell\left(D(p,v)\right) + \ell\left(D(v,q)\right)\right),$$

where $\rho = \max\left\{2^{r-1}, 1\right\}$ and $c = d^{\left|\frac{1}{z}-\frac{1}{2}\right|}$.

Proof. The case of $z=2$ immediately holds by substituting $\tilde{D} = \ell$ and dist $= D$ in Lemma 2.1 of [11]. The case $z=2$ can be trivially extended to any constant $z>0$ by combining the following property of vector norms: For every vector $u \in \mathbb{R}^d$ and constant $a>0$ it holds that

$$\begin{cases} \|u\|_2 \leq \|u\|_a \leq c\,\|u\|_2 & \text{if } a \in (0,2] \\ \|u\|_a \leq \|u\|_2 \leq c\,\|u\|_a & \text{if } a > 2 \end{cases}.$$

□

In what follows, for simplicity, we denote by $\mathrm{cost}(P, Q, R)$ the cost $\mathrm{cost}(P, Q, (R, \vec{0}))$ and by $\mathrm{cost}(P, Q) = \mathrm{cost}(P, Q, (I_d, \vec{0}))$.

The following lemma proves that there is an index $j \in [n]$ that can be recovered via non-uniform sampling, and a rotation matrix R' that aligns the directions of p_j and q_j, and with high probability also approximates the total initial cost up to some constant factor.

Lemma 12. *Put $\tau \in [d]$ and $z > 0$. Let π be an τ-dimensional subspace of $\mathbb{R}^d$, and $P = \{p_1, \cdots, p_n\} \subset \pi$ and $Q = \{q_1, \cdots, q_n\} \subset \pi$ be two ordered sets of points. Let cost be as defined in Definition 2 for $f = \|v\|_1$, an r-log-Lipschitz function ℓ and $D(p, q) = \|p - q\|_z$. Let $R^* \in \mathrm{SO}(d)$ and let $j \in [n]$ be an index sampled randomly, where $j = i$ with probability $w_i = \frac{\|p_i\|^r}{\sum_{j\in[n]} \|p_j\|^r}$ if $\|q_i\| \neq 0$ and $w_i = 0$ otherwise. Then there is $R' \in \mathcal{R}_\pi$ that satisfy the following properties:*

(i) $R'p_j \in \mathrm{sp}(q_j)$.

(ii) $\mathrm{cost}(P, Q, (R', \vec{0})) \leq 6\rho^2 c^{2r} \cdot \mathrm{cost}(P, Q, (R^*, \vec{0}))$ *with probability at least 1/2, where* $\rho = \max\left\{2^{r-1}, 1\right\}$ *and* $c = d^{\left|\frac{1}{z} - \frac{1}{2}\right|}$.

Proof. Without loss of generality assume that R^* is the identity matrix and that π is spanned by $e_1, \cdots, e_\tau$, otherwise rotate the set P of points and rotate the coordinates system respectively. Furthermore, we remove all pairs (p_i, q_i) of corresponding points from P and Q where $\|p_i\| = 0$ or $\|q_i\| = 0$. The distance $D(p_i, q_i)$ between such pairs is unaffected by a rotation of p_i, i.e., $D(Rp_i, q_i) = D(p_i, q_i)$ for every rotation matrix R. We can therefore ignore such pairs. The sampling probabilities of other pairs will not be affected by removing such (p_i, q_i) since $w_i = 0$ by definition.

For every $i \in [n]$, let $R_i \in \mathcal{R}_{\mathrm{sp}(\{p_i, q_i\})}$ be a rotation matrix that satisfies $q_i \in \mathrm{sp}(R_i p_i)$, i.e., aligns the directions of the vectors p_i and q_i by a rotation in the 2-dimensional subspace (plane) that those two vectors span. If there is more than one such rotation matrix, pick the one that rotates p_i with the smallest angle of rotation.

Now, by the definition of R_j we have that

$$\ell(D(R_j p_j, q_j)) \leq \ell(D(p_j, q_j)) \tag{49}$$

for every $j \in [n]$. Therefore, the following holds

$$\begin{aligned}\sum_{j=1}^n \ell(D(R_j p_j, p_j)) &\leq \sum_{j=1}^n \rho c^r \left(\ell(D(R_j p_j, q_j)) + \ell(D(q_j, p_j))\right) \\ &\leq 2\rho c^r \sum_{j=1}^n \ell(D(q_j, p_j)) \\ &= 2\rho c^r \mathrm{cost}(P, Q),\end{aligned} \tag{50}$$

where the first derivation is by Claim 11.1 and the second derivation is by (49).

We now prove that if we sample an index $j \in [n]$ according to the distribution $w = (w_1, \cdots, w_n)$, then the expected cost $\mathrm{cost}(P, Q, R_j)$ between the points of P after a rotation by R_j and their correspond points in Q is at most a constant times the original cost $\mathrm{cost}(P, Q)$. We observe that

$$\sum_{j\in[n]} w_j \cdot \mathrm{cost}(P, Q, R_j) = \sum_{j\in[n]} w_j \sum_{i=1}^n \ell(D(R_j p_i, q_i) \tag{51}$$

$$\leq \sum_{j\in[n]} w_j \sum_{i=1}^n 2\rho c^r \left(\ell(D(R_j p_i, p_i)) + \ell(D(p_i, q_i))\right) \tag{52}$$

$$= \rho c^r \sum_{j\in[n]} w_j \sum_{i=1}^n \ell(D(R_j p_i, p_i)) + \sum_{j\in[n]} w_j \sum_{i=1}^n \ell(D(p_i, q_i))$$

$$= \rho c^r \sum_{j\in[n]} w_j \sum_{i=1}^n \ell(D(R_j p_i, p_i)) + \sum_{i=1}^n \ell(D(p_i, q_i)) \tag{53}$$

$$= \rho c^r \sum_{j\in[n]} w_j \sum_{i=1}^n \ell(D(R_j p_i, p_i)) + \mathrm{cost}(P, Q), \tag{54}$$

where (52) is by substituting in Claim 11.1 and (53) holds since w is a distribution vector.

We now bound the leftmost term of (54).

$$\sum_{j\in[n]} w_j \sum_{i=1}^{n} \ell(D(R_j p_i, p_i))$$

$$= \sum_{j\in[n]} w_j \sum_{i=1}^{n} \ell\left(\|p_i\| \, D\left(\frac{R_j p_i}{\|p_i\|}, \frac{p_i}{\|p_i\|}\right)\right) \tag{55}$$

$$\leq \sum_{j\in[n]} w_j \sum_{i=1}^{n} \|p_i\|^r \, \ell\left(D\left(\frac{R_j p_i}{\|p_i\|}, \frac{p_i}{\|p_i\|}\right)\right) \tag{56}$$

$$= \sum_{j\in[n]} \frac{\|p_j\|^r}{\sum_{k\in[n]} \|p_k\|^r} \sum_{i=1}^{n} \|p_i\|^r \, \ell\left(D\left(\frac{R_j p_i}{\|p_i\|}, \frac{p_i}{\|p_i\|}\right)\right) \tag{57}$$

$$\leq \sum_{j\in[n]} \frac{\|p_j\|^r}{\sum_{k\in[n]} \|p_k\|^r} \sum_{i=1}^{n} \|p_i\|^r \, \ell\left(D\left(\frac{R_j p_j}{\|p_j\|}, \frac{p_j}{\|p_j\|}\right)\right) \tag{58}$$

$$= \sum_{j\in[n]} \|p_j\|^r \, \ell\left(D\left(\frac{R_j p_j}{\|p_j\|}, \frac{p_j}{\|p_j\|}\right)\right)$$

$$\leq \sum_{j\in[n]} \ell\left(D\left(R_j p_j, p_j\right)\right) \tag{59}$$

$$= 2\rho c^r \text{cost}(P, Q) \tag{60}$$

where (55) holds since D is simply a norm function, (56) holds since ℓ is an r-log Lipschitz function, (57) is simply by substituting the value of w_j, (58) holds by Claim 6.1, 59 holds since D is a norm and ℓ is an r-log Lipschitz function, and (60) is by (50).

Combining (60) and (54) yields that

$$\sum_{j\in[n]} w_j \cdot \text{cost}(P, Q, R_j) \leq 3\rho^2 c^{2r} \cdot \text{cost}(P, Q). \tag{61}$$

For a random variable X, a positive constant $a > 0$, the Markov inequality states that

$$Pr(X \geq a) \leq \frac{E(X)}{a}, \tag{62}$$

where $E(X)$ is the expectation of X.

Define the random variable $X := \text{cost}(P, Q, R_j)$, where the randomness is over the choice of the index $j \in [n]$, and let $a = 6\rho^2 c^{2r} \cdot \text{cost}(P, Q)$. By (61), the expectation of X is $E(X) = 3\rho^2 c^{2r} \cdot \text{cost}(P, Q)$. Plugging into the Markov equality (62) yields that

$$\text{cost}(P, Q, R_j) \leq 6\rho^2 c^{2r} \cdot \text{cost}(P, Q).$$

with probability at least $1/2$. □

The following lemma proves the correctness of Algorithm 3.

Theorem 13 (Theorem 4)**.** *Let P and Q be two ordered sets of n points in $\mathbb{R}^d$ and let $z > 0$. Let* cost *be as defined in Definition 2 for $f = \|v\|_1$, some r-log Lipschitz function ℓ and $D(p, q) = \|p - q\|_z$. Let (R, t) be an output of a call to* PROB-ALIGN(P, Q, r)*; see Algorithm 3. Then, with probability at least $\frac{1}{2^d}$,*

$$\text{cost}(P, Q, (R, t)) \leq \sigma \cdot \min_{(R',t')\in \text{ALIGNMENTS}(d)} \text{cost}(P, Q, (R', t')),$$

for a constant σ that depends on d and r. Furthermore, (R, t) is computed in $O(nd^3)$ time.

Proof. Let

$$(R^*, t^*) \in \operatorname*{arg\,min}_{(R,t)\in \text{ALIGNMENTS}(d)} \text{cost}(P, Q, (R, t))$$

be the optimal alignment, and let $\rho = \max\left\{2^{r-1}, 1\right\}$ and $c = d^{\left|\frac{1}{z} - \frac{1}{2}\right|}$. Without loss of generality assume that R^* is the identity matrix and t^* is a zeros vector. Otherwise rotate and translate the set P by (R^*, t^*). In other words, we assume the set P is already optimally aligned to Q.

Let OPT be the minimal cost, i.e.,

$$\text{OPT} = \text{cost}(P, Q, (I_d, \vec{0})) = \sum_{i=1}^{n} \ell\left(D(p_i, q_i)\right). \tag{63}$$

Recovering an approximated translation. We first aim to recover some approximated translation.

Claim 13.1. *There are at least $n/2$ corresponding pairs (p_i, q_i) whose cost in the optimal alignment is smaller than $\frac{2\text{OPT}}{n}$, i.e.,*

$$\ell\left(D(p_i, q_i)\right) \leq \frac{2\text{OPT}}{n}.$$

Proof. Falsely assume that there are less than $n/2$ such pairs. This implies that there are at least $n/2$ pairs which satisfy

$$\ell\left(D(p_i, q_i)\right) > \frac{2\text{OPT}}{n}.$$

The cost of those (at least) $n/2$ pairs would thus be greater than $\frac{n}{2} \cdot \frac{2\text{OPT}}{n} = \text{OPT}$, which contradicts the definition of OPT as the cost of the whole n pairs. □

The following claim states that translating the set P, from its optimal position, such that p_k intersects q_k for a randomly selected index $k \in [n]$, yields a constant factor approximation to the current (optimal) cost, with high probability.

Claim 13.2. *Let $\rho = \max\left\{2^{r-1}, 1\right\}$ and $c = d^{\left|\frac{1}{z} - \frac{1}{2}\right|}$. Then, there is an index $k \in [n]$ such that, with probability at least $1/2$,*

$$\text{cost}(P, Q, (I_d, p_k - q_k)) \leq 3\rho c^r \cdot \text{OPT}.$$

Proof. Let $k \in [n]$ be an index selected uniformly at random. Then,

$$\text{cost}(P, Q, (I_d, (p_k - q_k))) = \sum_{i=1}^{n} \ell\left(D(p_i - (p_k - q_k), q_i)\right) \tag{64}$$

$$\leq \rho c^r \left(\sum_{i=1}^{n} \left(\ell\left(D(p_i - (p_k - q_k), p_i)\right) + \ell\left(D(p_i, q_i)\right) \right) \right) \tag{65}$$

$$= \rho c^r \left(\sum_{i=1}^{n} \ell\left(D(p_k, q_k)\right) + \sum_{i=1}^{n} \ell\left(D(p_i, q_i)\right) \right)$$

$$\leq \rho c^r \left(\sum_{i=1}^{n} \frac{2\text{OPT}}{n} + \sum_{i=1}^{n} \ell\left(D(p_i, q_i)\right) \right) \tag{66}$$

$$= \rho c^r \left(2\text{OPT} + \sum_{i=1}^{n} \ell\left(D(p_i, q_i)\right) \right)$$

$$= 3\rho c^r \cdot \text{OPT}, \tag{67}$$

where (64) is by the definition of cost, (65) is by the weak triangle inequality in Claim 11.1, (66) holds with probability at least $1/2$ by combining Claim 13.1 with the random pick of the index k, and (67) holds by the definition of OPT in (63).

Therefore, a translation of P by $t_k = p_j - q_j$ where $k \in [n]$ is selected uniformly at random yields a constant factor approximation to OPT. □

Observe that, by definition of t_k in the above claim, $p_k - t_k = q_k$. Hence, p_k and q_k intersect after applying the translation t_k to P. This proves the existence of a translation that aligns a corresponding pair of points from P and Q, and yields a provable constant factor approximation to OPT. This translation can afterwards be easily recovered.

Recovering an approximated rotation. Let k be the index from Claim 13.2. As discussed above, p_k and q_k intersect after applying t_k to P. Translating both sets by the same translation does not change the pairwise distances. Hence, we will translate both sets again such that q_k and $p_k - t_k$ intersect the origin. In other words, we redefine the original (untranslated) P and Q as follows:

$$P := \{p - t_k - q_k \mid p \in P\}, Q := \{q - q_k \mid q \in Q\}. \tag{68}$$

We now aim prove the existence of a rotation matrix that: (i) can be afterwards easily recovered, and (ii) when applied to P yields a constant factor approximation to the cost of the initial (optimal) alignment. After recovering such a rotation matrix, we will rotate the set P, and then undo the translation applied above by translating both sets by $-q_k$.

Hence, we now aim to find a rotation matrix $\hat{R}$ such that $\text{cost}(P, Q, (\hat{R}, \vec{0})) \leq \sigma \cdot \text{cost}(P, Q, (R^*, \vec{0}))$ for some constant $\sigma > 0$. Observe that R^* can be any rotation matrix For simplicity of notation, we denote by $\text{cost}(P, Q, R)$ the cost $\text{cost}(P, Q, (R, \vec{0}))$, and by $\text{cost}(P, Q)$ the cost $\text{cost}(P, Q, (I_d, \vec{0}))$.

Let $j_1 \in [n]$ be an index sampled randomly, where $j = i$ with probability $w_i = \frac{\|p_i\|^r}{\sum_{j \in [n]} \|p_j\|^r}$ if $\|q_i\| \neq 0$ and $w_i = 0$ otherwise. Observe that the probabilities w_i are independent of R^* (which is assumed to be the identity matrix) since a rotation matrix does not change the norms of the points, i.e., $\|R^* p_i\| = \|p_i\|$ for every $i \in [n]$. By Lemma 12, there exists a matrix R_1 that aligns the direction vectors of p_{j_1} and q_{j_1}, and with probability at least $1/2$ satisfies:

$$\text{cost}(P, Q, R_1) \leq 6\rho^2 c^{2r} \cdot \text{cost}(P, Q). \tag{69}$$

However, there might be an infinite set A_1 of such rotation matrices which align the direction vectors of p_{j_1} and q_{j_1}. Let $R_1 \in A_1$ be an arbitrary such rotation matrix.

Let P' be the set P after applying R_1, and let $\hat{P}'$ and $\hat{Q}$ be the sets P' and Q respectively after projecting their points onto the hyperplane orthogonal to q_{j_1} i.e.,

$$P' = \{p_i' := R_1 p_i \mid i \in [n]\},$$

$$\hat{P}' = \left\{\hat{p}_i := WW^T p_i' \mid i \in [n]\right\},$$

and

$$\hat{Q} = \left\{\hat{q}_i := WW^T q_i \mid i \in [n]\right\},$$

where $W \in \mathbb{R}^{d \times (d-1)}$ is an orthogonal matrix whose columns span the hyperplane H orthogonal to q_{j_1}.

Let $j_2 \in [n]$ be an index sampled randomly, where $j = i$ with probability $w_i = \frac{\|\hat{p}_i\|^r}{\sum_{j \in [n]} \|\hat{p}_j\|^r}$ if $\|\hat{q}_i\| \neq 0$ and $w_i = 0$ otherwise. By applying Lemma 12 again on using $\hat{P}'$ and $\hat{Q}$, there is a matrix R_2 that aligns the direction vectors of $\hat{p}_{j_2}$ and $\hat{q}_{j_2}$, and with probability at least $1/2$ satisfies:

$$\text{cost}(\hat{P}', \hat{Q}, R_2) \leq 6\rho^2 c^{2r} \cdot \text{cost}(\hat{P}', \hat{Q}, I_d). \tag{70}$$

However, again, there might be an infinite set A_2 of such rotation matrices which align the direction vectors of $\hat{p}_{j_2}$ and $\hat{q}_{j_2}$. Let $R_2 \in A_2$ be an arbitrary such rotation matrix.

We now prove the following claims: (i) the cost $\text{cost}(P', Q, R_2)$ of applying the rotation matrix R_2 to the (unprojected) sets P' and Q will approximate the cost $\text{cost}(P', Q, I_d)$, (ii): the choice of j_2 is independent of the choice of R_1, and (iii) the vectors q_{j_1} and $\hat{q}_{j_2}$ are orthogonal.

Claim 13.3. *It holds that*

$$\text{cost}(P, Q, R_2 R_1) \leq 12\rho^4 c^{5r} \text{cost}(P, Q, R_1). \tag{71}$$

Proof. Recall that $P' = \{p'_1, \cdots, p'_n\}$. Consider the hyperplane H orthogonal to q_{j_1} (the hyperplane the points are projected on after the first step). Let H_i be a hyperplane parallel to H but passes through q_i. Let $v_i = \text{proj}(R_2 p'_i, H_i)$ be the projection of $R_2 p'_i$ onto the hyperplane H_i for every $i \in [n]$.

Observe that, by construction, the rotation matrix R_2 rotates every point p'_i around the rotation axis q_{j_1}, which is orthogonal to H_i. Therefore, the distance between $R_2 p'_i$ to its (orthogonal) projection onto H_i equals the distance between p'_i and its (orthogonal) projection onto H_i. Formally,

$$D(R_2 p'_i, \text{proj}(R_2 p'_i, H_i)) = D(p'_i, \text{proj}(p'_i, H_i)). \quad (72)$$

Let $v_i = \text{proj}(R_2 p'_i, H_i)$ be the projection of $R_2 p'_i$ onto the hyperplane H_i for every $i \in [n]$. We now have that

$$\text{cost}(P', Q, R_2) = \sum_{i=1}^{n} \ell\left(D(R_2 p'_i, q_i)\right) \quad (73)$$

$$\leq \rho c^r \sum_{i=1}^{n} \ell\left(D(R_2 p'_i, v_i)\right) + \rho c^r \sum_{i=1}^{n} \ell\left(D(v_i, q_i)\right) \quad (74)$$

$$= \rho c^r \sum_{i=1}^{n} \ell\left(D(p'_i, \text{proj}(p'_i, H_i))\right) + \rho c^r \sum_{i=1}^{n} \ell\left(D(v_i, q_i)\right) \quad (75)$$

where (73) is by the definition of cost, (74) is by the weak triangle inequality in Claim 11.1, and (75) is by combining (72) with the definition of v_i.

We now bound the rightmost term of (75) as follows:

$$\rho c^r \sum_{i=1}^{n} \ell\left(D(v_i, q_i)\right)$$

$$= \rho c^r \sum_{i=1}^{n} \ell\left(D(\text{proj}(v_i, H), \text{proj}(q_i, H))\right) \quad (76)$$

$$= \rho c^r \cdot \text{cost}(\hat{P}', \hat{Q}, R_2) \quad (77)$$

$$\leq 6\rho^3 c^{3r} \cdot \text{cost}(\hat{P}', \hat{Q}, I_d) \quad (78)$$

$$= 6\rho^3 c^{3r} \cdot \sum_{i=1}^{n} \ell\left(D(\text{proj}(p'_i, H), \text{proj}(q_i, H))\right) \quad (79)$$

$$= 6\rho^3 c^{3r} \sum_{i=1}^{n} \ell\left(D(\text{proj}(p'_i, H_i), q_i)\right), \quad (80)$$

where (76) holds by combining that $v_i, q_i \in H_i$ and that H_i and H are two parallel hyperplanes, (77) holds by the definitions of v_i and R_2, (78) is by (70), (79) is by the definition of $\hat{P}'$ and $\hat{Q}$, and (80) holds by combining that $q_i \in H_i$ and that H and H_i are parallel.

Now, consider the triangle $\Delta(p'_i, \text{proj}(p'_i, H_i), q_i)$. This triangle is a right triangle since $q_i \in H_i$. Hence, $D_2(p'_i, \text{proj}(p'_i, H_i)) \leq D_2(p'_i, q_i)$ and $D_2(\text{proj}(p'_i, H_i), q_i) \leq D_2(p'_i, q_i)$. By the properties of vector norms and since ℓ is an r-log-Lipschitz function, we obtain that

$$\begin{aligned}&\ell\left(D(p'_i, \text{proj}(p'_i, H_i))\right) + \ell\left(D(\text{proj}(p'_i, H_i), q_i)\right)\\ &\leq 2c^r \ell\left(D(p'_i, q_i)\right).\end{aligned} \quad (81)$$

Combining the above yields that

$$\text{cost}(P, Q, R_2 R_1) = \text{cost}(P', Q, R_2) \quad (82)$$

$$\leq \rho c^r \sum_{i=1}^{n} \ell\left(D(p'_i, \text{proj}(p'_i, H_i))\right) + \rho c^r \sum_{i=1}^{n} \ell\left(D(v_i, q_i)\right) \quad (83)$$

$$\leq \rho c^r \sum_{i=1}^{n} \ell\left(D(p'_i, \text{proj}(p'_i, H_i))\right) + 6\rho^4 c^{4r} \sum_{i=1}^{n} \ell\left(D(\text{proj}(p'_i, H_i), q_i)\right) \quad (84)$$

$$\leq 6\rho^4 c^{4r} \sum_{i=1}^{n} \ell\left(D(p'_i, \text{proj}(p'_i, H_i))\right) + 6\rho^4 c^{4r} \sum_{i=1}^{n} \ell\left(D(\text{proj}(p'_i, H_i), q_i)\right)$$

$$\leq 12\rho^4 c^{5r} \sum_{i=1}^{n} \left(\ell\left(D(p'_i, q_i)\right)\right) \quad (85)$$

$$= 12\rho^4 c^{5r} \cdot \text{cost}(P', Q, I_d) \quad (86)$$

$$= 12\rho^4 c^{5r} \cdot \text{cost}(P, Q, R_1), \quad (87)$$

where (82) is by the definition of P, (83) is by (75), (84) is by (80), (85) is by (81), and (87) is by the definition of cost. □

Claim 13.4. *The choice of j_2 is independent of the choice of R_1.*

Proof. The choice of j_2 depends on the ℓ_2 norms $\|\hat{p}_i\|$ of the points in $\hat{P}'$. Observe that the rotation matrices in the set A_1 rotate the set P around the axis q_{j_1}, and that the projection matrix $W_{j_1} W_{j_1}^T$ projects any point onto the hyperplane orthogonal to q_{j_1}. Therefore, for any two matrices $M_1, M_2 \in A_1$, the norms of the vectors $WW^T M_1 p$ and $WW^T M_2 p$ are the same, for every $p \in P'$. Hence, the distribution from which j_2 is drawn is identical for any two matrices in A_1. □

Claim 13.5. *The vectors q_{j_1} and $\hat{q}_{j_2}$ are orthogonal.*

Proof. By construction, the vector $\hat{q}_{j_2}$ is obtained by a projection of $q_{j_2} \in Q$ onto the subspace orthogonal to q_{j_1}. Therefore, they are orthogonal. □

The above claims prove the existence of 2 indices, j_1 and j_2, and two rotation matrices R_1 and R_2 that align the direction vectors of p_{j_1} with q_{j_1} and $\hat{p}_{j_2}$ with $\hat{q}_{j_2}$ respectively. The choice of j_1 and j_2 is independent of any initial rotation R^* of P and independent of the choice of R_1 respectively. Consider the rotation matrix $\hat{R} = R_2 R_1$. Since q_{j_1} and $\hat{q}_{j_2}$ are orthogonal, then $\hat{R}$ can simultaneously align both pairs of vectors, i.e., $\hat{R}$ satisfies both constraints. By combining (71) and (69) we obtain that with probability at least $1/4$,

$$\text{cost}(P, Q, \hat{R}) \leq (12\rho^4 c^{5r})^2 \text{cost}(P, Q).$$

Repeating the above steps $d-1$ times yields that there is a set of $d-1$ indices $j_1, \cdots, j_{d-1}$, corresponding rotation matrices $R^{(1)}, \cdots, R^{(d-1)}$, and a rotation matrix $R' = R_{d-1} \cdots R_1$ that satisfies

(i) R' aligns the direction vectors of p_{j_1} and q_{j_1}, i.e., $R' p_{j_1} \in \text{sp}(q_{j_1})$.

(ii) For every $i \in \{2, \cdots, d-1\}$, R' aligns the direction vectors of p_{j_i} and q_{j_i} after their projection onto the hyperplane orthogonal to q_{j_1}, then onto the hyperplane orthogonal to q_{j_2}, and so on until the projection onto the hyperplane orthogonal to $q_{j_{i-1}}$.

(iii) The indices $j_1, \cdots, j_{d-1}$ are independent of the initial R^*, and also independent of the arbitrary choice of rotation matrices throughout the $d-1$ steps.

(iv) Identically to 71, for every $k \in \{2, \cdots, d-1\}$ we can prove that with probability at least $1/2$,

$$\text{cost}(P, Q, R_k \cdots R_1) \leq 12\rho^4 c^{5r} \cdot \text{cost}(P, Q, R_{k-1} \cdots R_1).$$

By combining the last inequality for every $k \in \{1, \cdots, d-1\}$, we obtain that with probability at least $1/2^{d-1}$,

$$\text{cost}(P, Q, R') \leq (12\rho^4 c^{5r})^{d-1} \text{cost}(P, Q). \tag{88}$$

Combining the (re-)definitions of P and Q in (68) with (88) proves that with probability at least $1/2^{d-1}$ we have that

$$\text{cost}(P', Q', R') \leq (12\rho^4 c^{5r})^{d-1} \text{cost}(P', Q') \tag{89}$$

where $P' = \{p - p_k \mid p \in P\}$ and $Q' = \{q - q_k \mid q \in Q\}$.

Furthermore, since the pair of aligned direction vectors at the i'th step are orthogonal to all the previous $i-1$ aligned direction vectors by construction, the obtained set of $d-1$ constraints on the output rotation are Linearly independent. They thus determine a single rotation matrix R', which can be recovered regardless of the initial R^*, or $R_1, \cdots, R_{d-1}$. We can thus recover those indices via simple exhaustive search, and then recover the rotation matrix R' using those indices and the constraints they determine.

Combining the approximated translation and rotation. Let t_k be the approximated translation vector from Claim 13.2. Let R' be the approximated rotation as in (88) and let $t' = R' p_k - q_k$. Now, consider the alignment (R', t'). Observe that

$$R' p_i - t' - q_i = R'(p_i - p_k) + q_k - q_i = R' p'_i - q'_i.$$

Hence, we have that

$$\text{cost}(P, Q, (R', t')) = \text{cost}(P', Q', (R', \vec{0})). \tag{90}$$

We now obtain that

$$\begin{aligned}
\text{cost}(P, Q, (R', t')) &= \text{cost}(P', Q', (R', \vec{0})) && (91)\\
&\leq (12\rho^4 c^{5r})^{d-1} \text{cost}(P', Q', (I_d, \vec{0})) && (92)\\
&= (12\rho^4 c^{5r})^{d-1} \text{cost}(P, Q, (I_d, t')) && (93)\\
&\leq (36\rho^5 c^{6r})^{d-1} \cdot \text{OPT} && (94)\\
&\leq (36\rho^5 c^{6r})^{d-1} \text{cost}(P, Q, (I_d, \vec{0})) && (95)
\end{aligned}$$

where (91) is by (90), (92) holds with probability at least $1/2^{d-1}$ by (89), (93) holds similarly to (90), and (94) holds with probability at least $1/2$ by combining the definition of t' with Claim 13.2, and (95) is by the definition of OPT.

Hence, the cost $\text{cost}(P, Q, (R', t'))$ of the alignment (R', t') is a constant factor approximation to the original optimal cost $\text{cost}(P, Q) = \text{cost}(P, Q, (R^*, t^*))$, with probability at least $1/2^d$.

Computing (R', t'). By the claims above, t' can be recovered as $t' = p_k - q_k$ with a randomly sampled index $k \in [n]$. Similarly, R' can be recovered from a randomly sampled set of indices $j_1, \cdots, j_{d-1}$. Algorithm 3 is an implementation of the scheme above. It computes the set of indices $k, j_1, \cdots, j_{d-1}$ discussed above, and returns the alignment that they determine.

The algorithm contains at most d iterations. Each iteration takes at most $O(nd^2)$ time. Therefore, the total running time is $O(nd^3)$.

□

E. Approximation for the Registration Problem

Theorem 14 (Theorem 5). *Let $P = \{p_1, \cdots, p_n\}$, $Q = \{q_1, \cdots, q_n\}$ be two ordered sets of n points in $\mathbb{R}^d$, $\gamma \in \Omega(n^{2d})$, $z > 0$, and $w = d^{\left|\frac{1}{z} - \frac{1}{2}\right|}$. Let* cost *and r be as in Definition 2 for $D = \|p - q\|_z$ and $f(v) =$*

$\|v\|_1$. *Let* $(\tilde{R}, \tilde{t}, \tilde{m})$ *be the output of a call to* ALIGN-AND-MATCH$(P, Q, \gamma, \text{cost})$*; See Algorithm 4. Then,*

$$\text{cost}\left(P_{[\tilde{m}]}, Q, (\tilde{R}, \tilde{t})\right) \leq w^r(1+\sqrt{2})^{dr} \cdot \min_{(R,t,m)} \text{cost}\left(P_{[m]}, Q, (R,t)\right), \tag{96}$$

where the minimum is over every alignment (R, t) *and permutation* m*. Moreover,* $(\tilde{R}, \tilde{t}, \tilde{m})$ *is computed in* $n^{O(d)}$ *time.*

Proof. Let $(R^*, t^*, m^*) \in \arg\min_{(R,t,m)} \text{cost}(P_{[m]}, Q, (R,t))$.

Theorem 8 proves the existence of a set $P' \subseteq P$ and a corresponding set $Q' \subseteq Q$ of size $|P'| = |Q'| = d$, such that the output (R, t) of the call ALIGN(P', Q') to Algorithm 1 satisfies the following for every $i \in [n]$:

$$\left\|Rp_i - t - q_{m^*(i)}\right\| \leq (1+\sqrt{2})^d \cdot \left\|R^*p_i - t^* - q_{m^*(i)}\right\|. \tag{97}$$

By (97) and since the ℓ_2-norm of every vector in $\mathbb{R}^d$ is approximated up to a multiplicative factor of $w = d^{\left|\frac{1}{z} - \frac{1}{2}\right|}$ by its ℓ_z-norm, for every $i \in [n]$ we have that

$$\left\|Rp_i - t - q_{m^*(i)}\right\|_z \leq w(1+\sqrt{2})^d \cdot \left\|R^*p_i - t^* - q_{m^*(i)}\right\|_z.$$

Combining the last equation, the definition of cost and D, and Observation 10 yields that

$$\text{cost}(P_{[m^*]}, Q, (R,t)) \leq w^r \cdot (1+\sqrt{2})^{dr} \cdot \text{cost}(P_{[m^*]}, Q, (R^*, t^*)). \tag{98}$$

To recover the above (R, t), we must recover the subsets P' and Q' and plug them into Algorithm 1. This will be done via exhaustive search over all $\theta(n^d)$ possible subsets from P of size d as well as all $\theta(n^d)$ possible subsets of size d from Q.

However, to pick the alignment (R', t') with the smallest cost among all candidate alignments computed above (via many calls to Algorithm 1), we must evaluate its cost. To do so, we must first recover the correspondence between P and Q. Fortunately, given an alignment (R', t'), solving for the optimal correspondence is now trivial: We can apply (R', t') to the set P, and then compute, for every transformed point in P, its nearest neighbor in Q. This is the best possible correspondence for the given cost function and the specific alignment (R', t'). Recovering such correspondence was made possible since we have successfully decoupled the two problems of recovering an alignment and recovering a correspondence function.

Let $\text{NN}(P, Q, (R', t'))$ be the nearest neighbor matching between the points of P after applying (R', t'), and the points of Q.

Recall that the alignment (R, t) satisfies (98). Let $m = \text{NN}(P, Q, (R,t))$. Since m is an optimal matching function for P, Q, the alignment (R, t), and the function cost, it satisfies that

$$\text{cost}\left(P_{[m]}, Q, (R,t)\right) \leq \text{cost}\left(P_{[m^*]}, Q, (R,t)\right). \tag{99}$$

By plugging $\gamma \in \Omega(n^{2d})$ into Algorithm 4, we are guaranteed, by Line 2, to iterate over all possible subsets $P' \subseteq P$ and $Q' \subseteq Q$ of size d. For every such subset, the alignment $(R', t') :=$ APPROX-ALIGNMENT(P', Q') is computed at Line 4. The neighbour matching $\text{NN}(R', t')$ of (R', t') is then computed, and the triplet $(R', t', \text{NN}(R', t'))$ is added to the set M at Line 5. By iterating over all possible P' and Q', we are guaranteed to recover the desired alignment (R, t) above. Hence, we are also guaranteed that $(R, t, m) \in M$.

Algorithm 4 then computes, at Line 6, the triplet $(\tilde{R}, \tilde{t}, \tilde{m}) \in M$ which yields the smallest value for the cost function at hand. Combining $(R, t, m) \in M$ with the definition of $(\tilde{R}, \tilde{t}, \tilde{m})$ yields

$$\text{cost}\left(P_{[\tilde{m}]}, Q, (\tilde{R}, \tilde{t})\right) \leq \text{cost}\left(P_{[m]}, Q, (R,t)\right). \tag{100}$$

Hence, the following holds

$$\begin{aligned}\text{cost}\left(P_{[\tilde{m}]}, Q, (\tilde{R}, \tilde{t})\right) &\leq \text{cost}\left(P_{[m]}, Q, (R,t)\right) \\ &\leq \text{cost}\left(P_{[m^*]}, Q, (R,t)\right) \\ &\leq w^r(1+\sqrt{2})^{dr} \cdot \text{cost}\left(P_{[m^*]}, Q, (R^*, t^*)\right),\end{aligned}$$

where the first derivation holds by (100), the second derivation holds by (99) and the third derivation is by (98). Furthermore, the running time of the algorithm is $n^{O(d)}$ since there are $n^{O(d)}$ iterations, each iteration takes time independent of n ($O(d^3)$ time). Afterwards, we compute the optimal matching $\text{NN}(P, Q, (R,t))$ for every $(R, t) \in M$. There are $n^{O(d)}$ alignments in M, and computing such an optimal matching for each alignment takes $n^{O(1)}$ time. Hence, the total running time is $n^{O(d)}$.

Constrained correspondence. Assume we wish to solve the registration problem under constraints on the correspondence function, for example that the correspondence function is a bijection function. Then the computation of the set M remains unchanged. Afterwards, the only change required is to compute, for every $(R', t') \in M$, the optimal bijective function between the transformed P and Q, rather than the nearest neighbor correspondence.

Kuhn and Harold suggested in [24] an algorithm that given the pairwise distances (fitting loss) between two sets of n elements P and Q, it finds an assignment for every $p \in P$ to an element $q \in Q$ that minimizes the sum of distances between every assigned pair. This algorithm takes $O(n^3)$ time. We can use this algorithm to compute an optimal matching function $\hat{m}(P, Q, (R', t'), \text{cost})$ for every $(R', t') \in M$ in Line 5 of Algorithm 4. The proof above remains unchanged except that the optimal correspondence function m^* is assumed to be a bijection. □

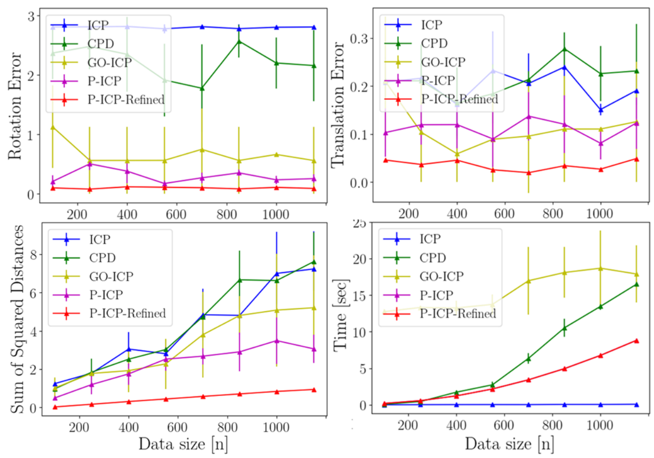

Figure 12: Asian Dragon model with $\sigma^2 = 0.01$ noise variance. The SSD cost function was used in our algorithms. The test was executed on the AWS platform, on a c5a.8*xlarge machine with* 32 *CPUs.*

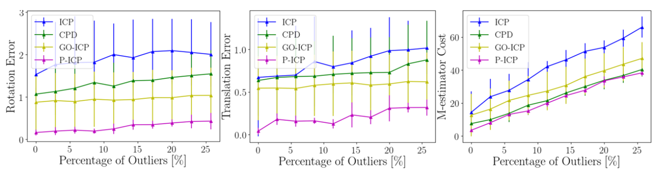

Figure 13: Robustness to outliers using the Asian Dragon model. $n = 800$ was used. Noise with variance $\sigma^2 = 1$ was added to k percentage of the points in P, which are considered as outliers. The SSD with M-estimator $\min\{\|p-q\|^2, 0.2\}$ was used in our algorithms. The computational time was roughly constant for each method for all tested k values, and is presented in Fig. 12 at $n = 800$.

F. Additional Experiments

In this section, we provide additional experimental results. We have conducted the same experiment depicted in Section 3.2, but using different models; see Fig 12-15.

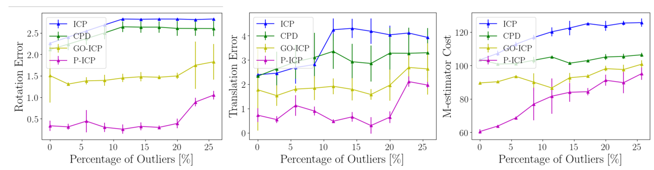

Figure 14: Robustness to outliers using the SUN3D 76-1studyroom1 dataset [48]. $n = 800$ was used. Noise with variance $\sigma^2 = 1$ was added to k percentage of the points in P, which are considered as outliers. The SSD with M-estimator $\min\{\|p-q\|^2, 0.2\}$ was used in our algorithms. The computational time was roughly constant for each method for all tested k values, and is presented in Fig. 12 at $n = 800$.

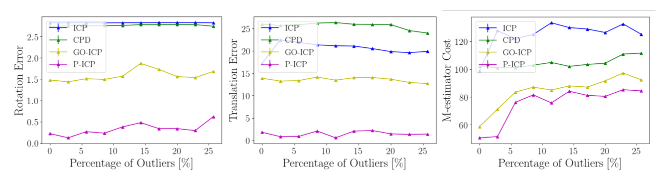

Figure 15: Robustness to outliers using the SUN3D hv_corridor1_1 dataset [48]. $n = 800$ was used. Noise with variance $\sigma^2 = 1$ was added to k percentage of the points in P, which are considered as outliers. The SSD with M-estimator $\min\{\|p-q\|^2, 0.2\}$ was used in our algorithms. The computational time was roughly constant for each method for all tested k values, and is presented in Fig. 12 at $n = 800$.