SDRSAC: Semidefinite-Based Randomized Approach for Robust Point Cloud Registration without Correspondences

Huu M. Le¹, Thanh-Toan Do²,³, Tuan Hoang¹, and Ngai-Man Cheung¹
¹Singapore University of Technology and Design ²University of Liverpool ³AIOZ Pte Ltd

Abstract

This paper presents a novel randomized algorithm for robust point cloud registration without correspondences. Most existing registration approaches require a set of putative correspondences obtained by extracting invariant descriptors. However, such descriptors could become unreliable in noisy and contaminated settings. In these settings, methods that directly handle input point sets are preferable. Without correspondences, however, conventional randomized techniques require a very large number of samples in order to reach satisfactory solutions. In this paper, we propose a novel approach to address this problem. In particular, our work enables the use of randomized methods for point cloud registration without the need of putative correspondences. By considering point cloud alignment as a special instance of graph matching and employing an efficient semi-definite relaxation, we propose a novel sampling mechanism, in which the size of the sampled subsets can be larger-than-minimal. Our tight relaxation scheme enables fast rejection of the outliers in the sampled sets, resulting in high quality hypotheses. We conduct extensive experiments to demonstrate that our approach outperforms other state-of-the-art methods. Importantly, our proposed method serves as a generic framework which can be extended to problems with known correspondences.

1. Introduction

Point cloud registration is important in many computer vision applications, including range scan alignment [37], 3D object recognition and localization [18, 48], large scale reconstruction [2, 44]. Given two sets of points in three-dimensional (3D) Euclidean space, the objective is to search for the optimal rigid transformation, which comprises a rotation matrix \( \mathbf{R}^* \in SO(3) \) and a translation vector \( \mathbf{t}^* \in \mathbb{R}^3 \), that optimally aligns the two input point sets. In many practical applications, the input data contains a significant amount of noise. Moreover, the overlapping region between the two point sets can be small, resulting in a large number of outliers, i.e., non-overlapping points. Therefore, the registration needs to be conducted in a robust manner so that the final estimates are not affected by the contamination. Formally, let \( \mathcal{S} = \{ \mathbf{s}_i \in \mathbb{R}^3 \}_{i=1}^{N_s} \) and \( \mathcal{D} = \{ \mathbf{d}_j \in \mathbb{R}^3 \}_{j=1}^{N_d} \) denote the source and destination (target) point clouds, respectively, the problem of robust rigid registration can be formulated as

\[
\min_{\mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3} \sum_{i=1}^{N_s} \rho \left( \| \mathbf{Rs}_i + \mathbf{t} - \mathbf{d}_j \| \right), \quad (1)
\]

where the notation \( \| \cdot \| \) represents the \( \ell_2 \) norm, \( \mathbf{d}_j \) is a point in the target set \( \mathcal{D} \) that is closest to the transformed point \( \mathbf{Rs}_i + \mathbf{t} \), i.e.,

\[
\mathbf{d}_j = \arg \min_{\mathbf{d}_k \in \mathcal{D}} \| \mathbf{Rs}_i + \mathbf{t} - \mathbf{d}_k \|, \quad (2)
\]

and \( \rho \) is a robust loss function. Here, \( SO(3) \) denotes the space of rotation matrices. In order for the registration to be robust, \( \rho \) is typically chosen from a set of robust kernels [27, 1, 33]. In this work, we make use of the popular maximum consensus criterion [26], in which \( \rho \) is defined as

\[
\rho(x) = \begin{cases} 
0 & \text{if } x \leq \epsilon, \\
1 & \text{otherwise}. 
\end{cases} \quad (3)
\]

The threshold \( \epsilon > 0 \) is a user-defined parameter that specifies the maximum allowable distance for a correspondence pair to be considered as an inlier. Intuitively, by solving (1) with \( \rho \) defined as per (3), we search for the optimal alignment \( (\mathbf{R}^*, \mathbf{t}^*) \) that maximizes the set of overlapping points, where \( \mathbf{s}_i \in \mathcal{S} \) overlaps \( \mathbf{d}_j \in \mathcal{D} \) if the transformation \( (\mathbf{R}^*, \mathbf{t}^*) \) brings \( \mathbf{s}_i \) to a new location that lies within the ball \( (\mathbf{d}_j, \epsilon) \). The problem (1) is an active research topic in computer vision due to its computational complexity.

Many existing algorithms need to take as input a set of putative correspondences, which are usually obtained (as a pre-processing step) by extracting local invariant features on the given point sets, and executing multiple routines of nearest neighbor search to propose initial key-

¹Source code is available at: https://github.com/intellhave/SDRSAC.
point matches [5, 25]. Several types of 3D local features [45, 15, 52, 43] have demonstrated to provide promising results throughout a variety of challenging datasets. However, noise and contamination would degrade the quality of extracted features. Furthermore, in order for the local features to be precisely computed, many feature extractors require the surface representation to be dense; however, sparse point clouds are common in practice. In particular, it has been demonstrated in [4] that, for noisy datasets with high proportion of outliers, alignments using features have poorer results compared with using raw data. Therefore, there is interest to develop registration algorithms that directly align raw point cloud data without the need of a priori correspondences [11, 4, 9, 36, 21].

Due to the computational complexity of the problem under different input settings, randomized hypothesize-and-verify algorithms such as RANSAC [22] and its variants [17, 16, 47, 46, 42, 32] are popular approaches. Randomized techniques have been employed to address problems with known correspondences. On the other hand, similar random sampling strategy as that of RANSAC can also be applied to problems of without correspondences. Specifically, at each iteration, a minimal subset (of three points) on each point cloud can be sampled to form three pairs of correspondences, which are used for estimating and validating one hypothesis. Such procedure can then be repeated until a satisfactory solution is obtained. However, with noise and outliers, the likelihood of picking outlier-free subsets degrades rapidly. Therefore, much efforts have been devoted to develop better sampling strategies, notably the 4-Points Congruent Sets (4PCS) method proposed in [4] and its improved variant [38]. Although 4PCS provides considerable advantages over conventional randomized methods, the enumeration of (approximately) congruent sets that underpins this algorithm is the main issue when working with point clouds with large number of points and high outlier rates. In fact, for dense point clouds, 4PCS and its variants need to down-sample the point cloud before conducting the sampling process to reduce processing time.

In this paper, we address the above-mentioned limitations. Specifically, by employing a special instance of graph matching formulation, we propose a new larger-than-minimal sampling strategy for point cloud registration without correspondences. The advantage of our method is that the task of searching for correspondences is quickly approximated by solving a relaxed convex problem, rather than subset enumeration. This allows us to sample subsets with arbitrarily large size, in which sets of correspondences are obtained from solutions of convex semi-definite programming. These correspondences can then be used for estimating and validating hypotheses. A large subset of points in the source point cloud represent better its structure, and by identifying its corresponding points on the target set, the two point clouds can be coarsely align faster, which can then be refined by local methods, e.g., ICP [8]. Empirical results show that the proposed method is very competitive.

Our main contributions are:

- We apply graph matching for the registration problem without correspondences using a novel cost function to enforce robustness and a tight semidefinite (SDP) relaxation.
- From the SDP formulation, we then develop a new larger-than-minimal subset sampling scheme, leading to an effective randomized algorithm that outperforms other state-of-the-art methods.

2. Related Work

In practice, if the two input point clouds are coarsely aligned, the well-known Iterative Closest Point (ICP) [8] is commonly employed. Like other iterative algorithms, ICP alternates between establishing correspondences and estimating the transformation. The main drawback of this method lies in the fact that it requires a good starting point (initial pose) to prevent itself from converging to poor alignments. Additionally, in terms of robustness, ICP suffers from the same weakness as that of the least squares estimator, i.e., it is easily biased by erroneous outliers. Several works [13, 7, 40] have been proposed to improve the shortcomings of ICP. However, these variants still need to be bootstrapped by a descent initial pose. ICP is therefore commonly employed as a local refinement procedure, which is executed after the point clouds are roughly registered by some type of global alignment algorithms.

Algorithms that offer globally optimal solutions are also actively developed in the literature. To address the initialization-dependent issue of ICP, Go-ICP [50] employs the branch and bound strategy to search in the space of $\mathbb{R}^n$ and $t$ for the optimal transformation. However, due to the least squares objective of Go-ICP, its returned optimal solutions are still non-robust to outliers. The robustness of Go-ICP can be improved by incorporating robust loss functions in place of least squares. Another globally optimal algorithm to tackle the robust registration without correspondences was proposed by Bustos et al. [11]. Unlike Go-ICP, [11] solves (1) directly and its solution is robust to outliers. Branch and bound is also the mechanism behind [11], with a novel steographic projection implementation for fast matching query and tighter bound computation. Although the convergence to globally optimal solutions is guaranteed for these methods, they are still impractical for large datasets due to their expensive computational cost.

As previously mentioned, in some point cloud registration applications, randomized strategies – with the famous RANSAC [22] representative – are still the dominant approaches. Our work also belongs to this category. Gener-
ally speaking, for most randomized methods, the underlying sampling strategy significantly affects the run time, since the termination of a sampling mechanism depends largely on its ability to quickly identify outlier-free subsets. For the case with no correspondences, it is even harder since two outlier-free subsets, one in each input point cloud, need to be identified and the elements in these two subsets must form correct correspondences. Different sampling strategies have been proposed in the literature [17, 16, 49]. For instance, in problems with known correspondences, one of the notable improvements for RANSAC is LO-RANSAC (Locally Optimal RANSAC) [17], which proposes to sample larger-than-minimal subsets when RANSAC solution is updated. This improved strategy has been shown to significantly outperform conventional RANSAC. Randomized methods can also be improved using local refinement techniques [31, 12, 41].

In the context of robust point cloud alignment without correspondences, however, the idea of larger-than-minimal sampling has yet to be thoroughly explored. Unlike the case of known correspondences, the “inlier” set at each RANSAC iteration may not be true inliers, and purely applying LO-RANSAC to these subsets may not be of any help, while the run time is increased. Besides, even if true inliers reside in the subset, a larger-than-minimal subset may be contaminated with outliers, which can deteriorate the estimation since it is done solely by solving least squares over the sampled subsets. From the insight discussed above, it can be seen that from any larger-than-minimal subsets, if the outliers are efficiently rejected, the ability to discover good hypotheses can be accelerated. This idea is analyzed in our paper, in which we propose a new algorithm that enables the sampling of any arbitrarily large subsets. Such sampling scheme is incorporated with an oracle that allows the outliers in the sampled point sets to be efficiently discarded. Our experiments show that this newly-proposed algorithm outperforms previous approaches.

Our work is closely related to 4PCS [3], which is the state-of-the-art sampling approach that also solves the same problem as ours. Instead of randomly picking minimal subsets of three points, 4PCS works on sampled tuples of four co-planar points. This method is later improved by Super4PCS [38], where the complexity of congruent set extraction is reduced from quadratic to linear with respect to the number of points. Though efficient, especially with the improvements introduced in [38], 4PCS still suffers from the same weakness as RANSAC, e.g., the low likelihood of quickly sorting out the right subsets under the presence of high outliers. In fact, for each subset of 4 co-planar points on one point cloud, 4PCS needs to subsample the point set into a smaller subset before the sampling is conducted.

The technique developed in this work is also inspired by the class of methods that solve 3D registration problem using graph matching [21, 35, 34] and recent semi-definite relaxation for non-rigid registration [19, 28, 30]. These algorithms, however, can only work for a small number of data points, which is very inefficient if applied directly on large point clouds. We, on the other hand, propose to employ graph matching as sub-problems embedded in a random sampling framework. This allows us to combine the strengths of both classes of methods, i.e., randomized techniques and graph matching, to derive an efficient sampling mechanism for point cloud registration.

3. Semidefinite-Based Randomized Approach

3.1. The Correspondence Problem

When the putative correspondences are not given, the task of solving (1) can be viewed as jointly estimating the transformation \((R^*, t^*)\) and the best subset of correspondences \(C^*\). Hence, if \(C^*\) is provided, the optimal alignment can be computed in a straightforward manner, and vice versa. With that in mind, in this section, we first introduce the correspondence problem to search for \(C^*\). In the latter sections, we then discuss its semidefinite (SDP) relaxation, which can be employed as a fast hypothesis generation mechanism that lies at the core of our efficient randomized algorithm.

To simplify the formulation, let us for now assume that we are given two 3D point sets \(P = \{p_i\}_{i=1}^N\) and \(Q = \{q_j\}_{j=1}^N\), each contains \(N\) data points. The task is to find the optimal set of correspondences \(C_{PQ}\) that can be used to robustly align \(P\) and \(Q\). Note that we use different notations for input data here compared to (1), since this new problem will be used as sub-problem for solving (1), which will be clarified in the latter sections.

Let \(X \in \{0, 1\}^{N \times N}\) be the permutation matrix in which the element at the \(i\)-th row and \(j\)-th column (denoted by \(X_{i,j}\)) is assigned the value of 1 if the pair \(p_i \in P\) and \(q_j \in Q\) belongs to the set of correspondences and 0 otherwise. To account for outliers, let us further assume that the optimal solution contains \(m \leq N\) pairs of correspondences. The value of \(m\) can be chosen to be greater than or equal to 3 (the size of the minimal subset), or estimated based on the known outlier ratio of the problem at hand. Note that with the introduction of \(m\), \(X\) is now a sub-permutation matrix. Denote by \(\hat{X}\) the vectorization of the matrix \(X\) obtained by stacking its columns:

\[
\hat{X} = [X_{1,1}^T \ X_{1,2}^T \ \ldots \ X_{1,N}^T]^T,
\]

where \(X_{i,j}\) denotes the \(j\)-th column of \(X\). In order to search for the best correspondence assignments, consider
the following optimization problem:

\[
\begin{align*}
\max_{\mathbf{X}} & & \hat{\mathbf{X}}^T \mathbf{A} \hat{\mathbf{X}}, \\
\text{subject to} & & \mathbf{X}_{i,j} \in \{0, 1\} \quad \forall i, j \in \{1, \ldots, N\}, \\
& & \sum_{j=1}^{N} \mathbf{X}_{i,j} \leq 1 \quad \forall i \in \{1, \ldots, N\}, \\
& & \sum_{i=1}^{N} \mathbf{X}_{i,j} \leq 1 \quad \forall j \in \{1, \ldots, N\}, \\
& & \sum_{i=1}^{N} \sum_{j=1}^{N} \mathbf{X}_{i,j} = m, 
\end{align*}
\]  

(5a), (5b), (5c), (5d), (5e)

where \( \mathbf{A} \in \mathbb{R}^{N^2 \times N^2} \) is the symmetric matrix that characterizes the matching potentials between pairs of points (line segments) in \( \mathcal{P} \) and \( \mathcal{Q} \). In particular, we define the elements of the matrix \( \mathbf{A} \) as

\[
\mathbf{A}_{ab,cd} = \begin{cases} f(\mathbf{p}_a, \mathbf{p}_c, \mathbf{q}_b, \mathbf{q}_d) \text{ if } |\delta(\mathbf{p}_a, \mathbf{p}_c) - \delta(\mathbf{q}_b, \mathbf{q}_d)| \leq \gamma, \\
0 \text{ otherwise,}
\end{cases}
\]  

(6)

Note that in (7), the constraint (5b) can be removed without affecting the equivalence between (7) and (5). Indeed, since \( \text{trace}(\mathbf{Y}) = m \), it must hold that \( \sum_{i,j} \mathbf{X}_{i,j}^2 = m \). On the other hand, due to the constraint (5e), \( \sum_{i,j} \mathbf{X}_{i,j} = m \). These conditions result in \( \sum_{i,j} \mathbf{X}_{i,j}^2 = \sum_{i,j} \mathbf{X}_{i,j} \). Also, due to the condition \( 0 \leq \mathbf{X}_{i,j} \leq 1 \), \( \mathbf{X}_{i,j} \) can only be either 0 or 1. In other words, \( \mathbf{X} \) can now be constrained in the convex hull of the sub-permutation matrices [39].

3.2. Semidefinite Relaxation

Let us first consider the equivalent formulation of (5). Let \( \mathbf{Y} = \mathbf{X} \mathbf{X}^T \in \mathbb{R}^{N^2 \times N^2} \), the problem (5) becomes

\[
\begin{align*}
\max_{\mathbf{X}, \mathbf{Y}} & & \text{trace}(\mathbf{A} \mathbf{Y}), \\
\text{subject to} & & \mathbf{Y} = \mathbf{XX}^T, \\
& & \text{trace}(\mathbf{Y}) = m, \\
& & 0 \leq \mathbf{X}_{i,j} \leq 1 \quad \forall i, j, \\
& & (5c), (5d), (5e). \quad (7a), (7b), (7c), (7d), (7e)
\end{align*}
\]

The constraints in (7) are to assure that \( \mathbf{X} \) must lie in the space of permutation matrices. Specifically, besides the binary constraint (5b) to restrict \( \mathbf{X}_{i,j} \in \{0, 1\} \), it is also required that each point \( \mathbf{p}_a \in \mathcal{P} \) can only be assigned to at most one point in \( \mathcal{Q} \) and vice versa, which is reflected by the constraints (5c) and (5d). Finally, by enforcing (5e), the optimal solution of (5) will only contain \( m \) pairs of correspondences. The solution to (5) provides the optimal assignment such that the sum of matching potentials gained yield from the corresponding pairs is maximized.

While the use of graph matching has been explored in several rigid and non-rigid registration problems [21, 35, 30, 28], most previous work consider solving graph matching for the whole input data, which is infeasible for large datasets. This work proposes to solve graph matching on very small subsets of points, then embed them into a random sampling framework. Additionally, we also propose an better formulation of the matrix \( \mathbf{A} \) for the special case of robust rigid registration. Particularly, the matrix \( \mathbf{A} \) is designed based on the fact that if \( \mathbf{p}_a \in \mathcal{P} \) corresponds to \( \mathbf{q}_b \in \mathcal{Q} \) and \( \mathbf{p}_c \in \mathcal{P} \) corresponds to \( \mathbf{q}_d \in \mathcal{Q} \), since the transformation is rigid, the lengths of the two line segments \( \mathbf{p}_a \mathbf{p}_c \) and \( \mathbf{q}_b \mathbf{q}_d \) must be approximately the same (due to the effect of noise). With the new formulation (6), instead of assigning the matching potentials to all the elements of \( \mathbf{A} \), we only allow pairs of segments whose lengths differ by a small value \( \gamma \) to be considered as candidate for matching, while pairs having large gap are rejected.

3.2.1 Tightening the Relaxation

As (8) is solved as an approximation for (7) with rank-one constraint, one would expect that the two solutions to be as
close as possible. Inspired by [30], we add the following constraints to (8) to tighten the relaxation:

$$Y_{ab,cd} \leq \begin{cases} 0, & \text{if } a = c, b \neq d, \\ 0, & \text{if } b = d, a \neq c, \\ \min(X_{ab}, X_{cd}), & \text{otherwise}. \end{cases}$$ (9)

The intuition behind (9) is that one point in $S$ is not allowed to match with more than one point in $D$ and vice versa. Also, since $Y_{ab,cd} = X_{ab}X_{cd}$ and these are binary numbers, it must hold that $Y_{ab,cd} \leq \min(X_{ab}, X_{cd})$.

In addition, thanks to the special case of robust registration, (8) can be further tightened. Observe that, based on the discussion of formulating the matrix $A$, the following constraints can be enforced:

$$Y_{ab,cd} = 0 \text{ if } A_{ab,cd} = 0,$$ (10)

which means if the pairs $p_a p_c$ and $p_b p_d$ differ too much in length (more than $\gamma$, we directly disallow them to be matched.

Finally, with the addition of (9) and (10), our SDP relaxation becomes:

$$\max_{X,Y} \quad \text{trace}(AX)$$ (11a)
subject to (8b), (7c), (7d), (8c), (9), (10) (11b)

Note that (11) is still a convex SDP since the additional constraints (9) and (10) are linear.

### 3.3. Projection of Solutions to Permutation Matrix

After solving (11) up to its global optimality using a convex solver, the remaining task is to project its solution back to the space of permutation matrices. This task can be done using different strategies [23]. In this work, we apply the linear assignment problem [10], in which the projection can be computed efficiently by solving a linear program (LP).

Specifically, let $X$ be the optimal solution of (11). The LP for projection can be formulated as

$$\max_X \quad \langle X, \tilde{X} \rangle$$ subject to $0 \leq X_{i,j} \leq 1$ $\sum_i X_{i,j} = 1$, $\sum_j X_{i,j} = 1$ (12)

where $\langle \cdot, \cdot \rangle$ represents the inner product of two matrices.

The solution of (12) provides us with a set of $N$ correspondences. However, in (5), we only want to pick $m$ pairs. To tackle this with a simple heuristic, observe that in the optimal solution $X^*$ of (5), only $m$ rows/columns of $X^*$ contain the value of 1, while the rest of the rows/columns contain all zeros. Therefore, from correspondence set obtained from (12), we associate each pair of correspondence $(p_i, q_j)$ with a score that is equal to $X_{i,j}$. Finally, $m$ pairs with the highest scores are chosen as the approximate solution for (5). This approach has been empirically shown to be very efficient throughout our experiments.

### 3.4. Main Algorithm

**Algorithm 1** SDRSAC

Require: Input data $S$ and $D$, $\max_{\text{iter}}$, $\text{inner}_{\text{iter}}$s, size of sampled subsets $N_{\text{sample}}$  

1: iter $\leftarrow 0$; best_score $\leftarrow 0$;  
2: while iter $<$ $\max_{\text{iter}}$ do  
3: $S' \leftarrow$ Randomly sample $N_{\text{sample}}$ points from $S$  
4: for $t = 1$ to $\text{inner}_{\text{iter}}$s do  
5: $D' \leftarrow$ Randomly sample $N_{\text{sample}}$ points from $D$  
6: $\{M, R, t\} \leftarrow$ SDRMatching $(S, D, S', D')$ /*Described in Alg. 2 */  
7: if $|M| > \text{best}_{\text{score}}$ then  
8: best_score $\leftarrow |M|$; $R^* \leftarrow R$; $t^* \leftarrow t$  
9: end if  
10: end for  
11: iter $\leftarrow$ iter + $\text{inner}_{\text{iter}}$s  
12: $T$ $\leftarrow$ Number of iterations that satisfies the probabilistic stopping criteria (Sec. 3.5)  
13: if iter $\geq T$ then  
14: return  
15: end if  
16: end while  
17: return Best transformation ($R^*, t^*$), best_score

**Algorithm 2** SDRMatching

Require: Input data $S$ and $D$, sampled subsets $S'$ and $D'$, threshold $\epsilon$.  

1: $A \leftarrow$ Matrix generated using Eq. (6) with $P = S'$ and $Q = D'$  
2: $X \leftarrow$ Solve (11) with $A$ generated from Step 1.  
3: $X \leftarrow$ Solve (12) with $X$ obtained from Step 2.  
4: $M' \leftarrow \{(s'_{i} \in S', d'_{j} \in D') | X_{i,j} = 1\}$  
5: $(\tilde{R}, \tilde{t}) \leftarrow$ Estimate transformation based on correspondence set $M'$ [20, 29].  
6: $(R, t) \leftarrow$ Refinement using ICP, initialized by $(\tilde{R}, \tilde{t})$  
7: $M \leftarrow \{(s_{i} \in S, d_{j} \in D) | ||R s_{i} + t - d_{j}|| \leq \epsilon\}$ /*$d_{j}$ is defined in (2) */  
8: return $M$, $R$, $t$

Although (11) is convex, solving it for large value of $N$ is inefficient since the lifted variable $Y$ belongs to the domain of $\mathbb{R}^{N^2 \times N^2}$. However, for a small value of $N$ (typically, $N \leq 20$), this problem can be optimized efficiently and (11) provides a good approximation for the original problem (7). Taking advantage of this observation, one can
develop a sampling approach where the number of points in each sample can be of any size, up to the limit that the employed solver can handle (11) efficiently. More specifically, at each iteration, two subsets of points $S' \subseteq S$ and $D' \subseteq D$ are randomly sampled from $S$ and $D$, respectively. The cardinality of $S'$ and $D'$ can be controlled depending on the capability of the employed convex solver. By solving (11), the outliers in the sampled subsets can be rejected and the best subset in each sample is retained for estimating the transformation. Like other randomized paradigms, this process can be repeated after a fixed number of iterations or until the stopping criteria is satisfied. The insight behind this strategy is that by sampling large subsets at each iteration, one is more likely to encounter the right subsets that contains inliers, since if any sampled subset is contaminated, the outliers can be rejected efficiently by solving the SDP approximation discussed in Section 3.1. Algorithm 1 summarizes our method.

### 3.5. Stopping Criterion

We follow [4] to derive our stopping criterion. Let us denote by $p_I$ the probability of selecting one inlier (correct correspondence pair), and by $T$ the number of trials. Note that from statistics [24], the expected inlier rate in a sample of size $N_{\text{sample}}$ is also $p_I$. Moreover, after running Alg. 2, only $m$ pairs of correspondences remain, allowing us to compute the stopping criterion based on $m$. Denote by $p_f$ the probability that the algorithm fails to find an all-inlier correspondence set after $T$ trials, $p_f$ can be computed as

$$p_f = (1 - p_I^m)^T,$$

(13)

Therefore, in order to get a success probability to be greater than $p_s$, we must have the number of iterations greater than

$$T \geq \frac{\log(1 - p_s)}{\log(1 - p_f^m)}.$$  

(14)

Since the real inlier rate $p_I$ is not known in advance, following common practice of several randomized methods, we update this value during the sampling process, i.e., $p_f$ is iteratively updated using the inlier ratio of the current best-so-far solutions.

### 4. Experimental Results

To evaluate the performance of our proposed algorithm (SDRSAC), we conduct experiments on multiple sets of synthetic and real datasets and compare our approach against several state-of-the-art algorithms that can be used to solve point cloud registration without correspondences, including ICP [8], Trimmed ICP (TrICP) [13], Iteratively Re-weighted Least Squares (IRLS) [6], 4PCS [4] and its improvement Super4PCS [38]. Within the class of globally optimal algorithms, we also compare SDRSAC against Go-ICP [50] and its robust version with trimming (TrGoICP). Note that conventional RANSAC method [22] performs poorly for this type of problem, and in many cases it turns into a brute-force type algorithm. Thus, to save space, we only show results from established methods listed above.

As we focus on validating the effectiveness of robust global registration, throughout the experiments, we measure and report the number of matches (objective function of (1)) and run time for each method.

All experiments are executed on a standard Ubuntu machine with 16GB of RAM. SDRSAC, ICP, TrimmedICP were implemented using MATLAB. For 4PCS, Super4PCS [38], Go-ICP [50], we use the released C++ code and the parameters suggested by the authors. All results of randomized methods are reported by averaging the outcomes obtained from 20 different runs. In the following, we only report representative results. More results, implementation details and extensions to the case with known correspondences can be found in the supplementary material.

#### 4.1. Synthetic Data

We first evaluate the performance of SDRSAC on synthetically generated data. The bunny point cloud from the Stanford dataset\(^2\) is loaded and uniformly sampled to yield a source point cloud $S$ containing $N_s = 10,000$ points. To generate the target set $D$, we apply a random transformation $(R, t)$ to $S$. Each point in $D$ is then perturbed with a Gaussian noise of zero mean and variance of $\sigma_{\text{noise}} = 0.01$. To simulate partial overlapping, we randomly pick and remove $r\%$ of the points in $D$. In order to evaluate the performance of the algorithms with different outlier ratios, we repeat the experiments with $r = 10, 15, \ldots, 50\%$. The threshold $\varepsilon$ in (3) was chosen to be 0.01 for all the methods. For SDRSAC, we choose the sample size to be $N_{\text{sample}} = 16$, and $m = 4$. (The choice of $N_{\text{sample}}$ and $m$ will be studied in Sec. 4.3.) Figure 1 shows the number of matches (inliers) and run time for all the methods. It is evident that SDRSAC outperforms other methods in terms of correspondences obtained with faster (or comparable) execution time. Note that the run time of SDRSAC is very close to Super4PCS, though SDRSAC consistently attains higher solution quality. The performance of ICP, as anticipated, is unstable due to the effect of initialization (i.e., wrong initialization may lead to poor convergence), as shown in Fig. 1. Moreover, as the outlier ratio increases, the consensus sizes produced by most ICP-based methods are poor due to their non-robustness.

#### 4.2. Real Data

In this data, we evaluate the performance of our proposed method on real datasets and compare it with existing

\(^2\)http://graphics.stanford.edu/data/3Dscanrep/
Table 1. Results for real data experiments. For each pairs, the first row is the number of correspondences (#Corrs) and the second row shows the run time in second. Note that S-4PCS represent Super4PCS.

Figure 1. Results for experiments with synthetic data. Left: Number of correspondences (Objective of (1)). Right: Run time (in log scale).

Figure 2. Examples of alignments using SDRSAC. From top to bottom: Office 8 and Office 9; LivingRoom 1 and LivingRoom 2.

approaches. The input point clouds for this experiment are obtained from the challenging Redwood 3D dataset [14]. We randomly pick eight pairs of point clouds from the Office and Living Room repository, then uniformly sampled the point sets to obtain 10,000 points on each input set. The input threshold $\epsilon$ in (3) was chosen to be in the range of 0.01 to 0.05 and for each set of input data, $\epsilon$ was chosen to be the same for all benchmarking methods.

Table 1 shows the number of matches and run time for each method. The sample qualitative results are displayed in Figure 2. As can be seen, on average, SDRSAC obtains higher matches compared to other competitors. We also observe that local methods such as ICP or IRLS can sometimes converge fast to very good results, but are unstable due to the effect of bad initializations. The robust version TrGo-ICP performs slightly better than Go-ICP, but
also fails in some cases due to high outlier ratios residing in the input data.

4.3. Ablation Studies

This section analyzes the effect of different parameter settings to the performance of our proposed method, and suggest the choices for the hyperparameters.

4.3.1 Effect of Sample Size \( N_{\text{sample}} \)

Fig. 3 plots the number of inliers obtained at termination (using results are obtained as median over iterations (SD) of \( N \) may be contaminated with large numbers of outliers, which inflaterates is large among sampled subsets, thus some subsets.

20 median over runs). Our obtained results conform to statistical theory [24]. Specifically, the inlier rates in the subsets of \( N_{\text{sample}} \) points have a mean of \( p_f \) and standard deviation (SD) of \( \sqrt{\frac{p_f(1-p_f)}{N_{\text{sample}}}} \). From Fig. 3, we observe that for small \( N_{\text{sample}} \) (\( N_{\text{sample}} < 16 \)), the obtained consensus sizes are low and the algorithm requires a large number of iterations. In particular, if \( N_{\text{sample}} \) is small, then SD of the inlier rates is large among sampled subsets, thus some subsets may be contaminated with large numbers of outliers, which affects the quality and stability of our algorithm. When \( N_{\text{sample}} \) reaches around 16, SD is \( \sqrt{\frac{0.3+0.7}{16}} \approx 0.1 \), hence the inlier rates vary only slightly among subsets, leading to a stable performance, as shown in Fig. 3 (left, middle). While it is good to use large \( N_{\text{sample}} \) from the statistical point of view, SDP solver would take more time at each iteration for large \( N_{\text{sample}} \), resulting in longer overall run time as shown in Fig. 3 (right). Empirically, we found that \( N_{\text{sample}} = 16 \) provides a good trade-off between algorithm stability (accuracy) and run time in many settings.

4.3.2 Effect of \( m \)

This experiment is conducted to study on the effect of \( m \) (introduced in Sec. 3.1) on the solution quality and run time of our approach. We repeat the experiment for a synthetic dataset containing \( N = 5,000 \) points per point cloud with 10% outliers. The sample size \( N_{\text{sample}} \) is set to 16. All results are obtained as median over 20 runs. Fig. 4 (left) plots the number of inliers obtained at termination (using the stopping criterion (14)). Observe that the performance is quite stable at different values of \( m \), which demonstrates the effectiveness of SDP to reject outliers and the strengths of our proposed sampling scheme. In Fig. 4 (right), we also plot the run time required until termination. Apparently, although the solution qualities for different values of \( m \) are similar as discussed above, the run time is affected by \( m \). Specifically, as \( m \) increases, SDRSAC takes more iterations before the stopping criterion (14) is met. This can be explained by recalling from (13) that large \( p_f \) increase the failure probability \( p_f \). Moreover, for any sampled subset, the expected number of inliers is \( N_{\text{sample}} \times p_f \), hence if \( m \) is larger than this value, outliers may be included. In our experiments, the choice of \( m = 4 \) works best in most scenarios (the minimal case of \( m = 3 \) may not be very stable due to noise in 3D data points and the 3 correspondences could be near each other spatially).

5. Conclusions

We have presented a novel and efficient randomized approach for robust point cloud registration without correspondences. We apply graph matching formulation for the correspondence problem. We propose a novel cost function and a tight SDP relaxation scheme. We embed the formulation into a new sampling strategy which samples larger-than-minimal subsets. Extensive experiments with several synthetic and real datasets show that our algorithm is competitive. Our framework can also be extended for the problems with known correspondences.

Acknowledgement

This work was supported in part by both ST Electronics and the National Research Foundation(NRF), Prime Minister’s Office, Singapore under Corporate Laboratory @ University Scheme (Programme Title: STEE Infosec - SUTD Corporate Laboratory).

References


