A Polyhedral Annexation Algorithm for Aligning Partially Overlapping Point Sets

WEI LIAN¹, WANGMENG ZUO², ZHESEN CUI¹
¹Department of Computer Science, Changzhi University, Changzhi, 046011, Shanxi, China (e-mail: lianwei3@gmail.com, cuizhesen@gmail.com)
²School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China (e-mail: cswmzuo@gmail.com)

ABSTRACT Point set registration aims to find a spatial transformation that best aligns two point sets. Algorithms which can handle partial overlap and are invariant to the corresponding transformations are particularly desirable. To this end, we first reduce the objective of the robust point matching (RPM) algorithm to a function of a low dimensional variable. The resulting function is nevertheless only concave over a finite region including the feasible region, which prohibits the use of the popular branch-and-bound (BnB) algorithm. To address this issue, we propose to use the polyhedral annexation (PA) algorithm for optimization, which enjoys the merit of only operating within the concavity region of the objective function. The proposed algorithm does not need regularization on transformation and thus is invariant to the corresponding transformation. It is also approximately globally optimal and thus is guaranteed to be robust. Moreover, its most computationally expensive subroutine is a linear assignment problem which can be efficiently solved. Experimental results demonstrate better robustness of the proposed method over the state-of-the-art algorithms. Our method’s matching error is on average 44% (resp. 65%) lower than that of Go-ICP in 2D (resp. 3D) synthesized tests. It is also efficient when the number of transformation parameters is small.

INDEX TERMS point set registration, branch-and-bound, concave optimization, polyhedral annexation

I. INTRODUCTION
Point set registration aims to find a spatial transformation that best aligns two point sets to a common coordinate system. It has extensive applications in 3D reconstruction, 3D localization, pose estimation, panorama stitching and medical imaging. Disturbances such as non-rigid deformation, positional noise, partial overlap and pose variation cause this problem difficult to solve. One way of achieving point set registration is by minimizing the objective of the robust point matching (RPM) algorithm [5]. By eliminating the transformation variable, [21] reduces the objective of RPM to a concave quadratic function of point correspondence with a low rank Hessian matrix. Then the branch-and-bound (BnB) algorithm with a low dimensional branch space is employed for optimization. But the method requires that one point set can be embedded in another point set, which may not be satisfied in many applications.

By relaxing this requirement, [19] reduces the objective of RPM to a concave function of point correspondence, which, albeit not quadratic, still has a low rank structure. [19] then employs the normal simplex algorithm, a variant of the BnB algorithm, for optimization. But this algorithm requires that the to-be-optimized function is concave over a set of simplices whose union includes the feasible region, whereas the function at hand is not necessarily concave outside the feasible region. To address this issue, [19] enlarges the concavity region of the objective function by enforcing regularization on transformation where prior information about the values of the transformation needs to be supplied. Consequently, the method tends to yield transformation parameters biased towards the prior values. So the method is not invariant to the corresponding transformation and can not be used in applications where, e.g., rotation invariance is required.

To address this issue, the BnB based approach of [20] utilizes the constraints of 2D/3D rigid transformations to develop a new objective function of point correspondence which is always concave. Thus, no regularization on transformation is needed and the method is invariant to the corresponding transformation. Computation of the lower bound is the most time consuming subroutine of this algorithm. To render this subroutine efficient, [20] relaxes the original lower bounding problem into a linear assignment problem.
by dropping the branching constraints. The latter can be efficiently solved. However, this has the negative effect that the computed lower bound is loose and the method fails to converge to the upper bound. Consequently, early stopping is adopted and the method is only heuristic.

Aiming at aligning partially overlapping point sets, in this paper, we propose an alternative approach to optimizing the objective of RPM. Instead of employing the BnB algorithm for optimization as in [19], [20] which can lead to various types of difficulties as described previously, we employ the polyhedral annexation (PA) algorithm [11] for optimization. It can directly optimize the RPM objective function by only operating within the concavity region of the function. Therefore, no regularization on transformation is needed and the proposed method is invariant to the corresponding transformation. Also, unlike the BnB algorithm where branching constraints need to be taken into account to ensure tightness of the generated lower bounds, in the proposed PA algorithm, no such constraints exist and the core subroutine has a linear assignment formulation which can be efficiently solved. Finally, the proposed method is approximately globally optimal and thus is guaranteed to be robust. Our contributions are summarized as follows:

1) It is the first time that PA is introduced into the computer vision field.
2) Our method can directly optimize within the concavity region of the RPM objective function.
3) The core subroutine of the proposed method has a linear assignment formulation which can be efficiently solved.
4) Our method is versatile. It can handle partial overlap and is invariant to the corresponding transformation. It is also approximately globally optimal.
5) Experiments on 2D/3D synthetic and real data demonstrated favorable robustness of the proposed method over the state-of-the-art approaches.

The rest of the paper is arranged as follows: In Sec. II, we review related work. In Sec. III and IV, we discuss our objective functions under two different types of transformations. In Sec. V, we present the PA algorithm. In Sec. VI, we present experimental results.

II. RELATED WORK
A. HEURISTIC METHODS

ICP based methods. ICP [2], [40] iterates between estimating point correspondence and transformation. It is fast but prone to be trapped in local minima due to the discrete nature of point correspondence. RPM [5] improves ICP by relaxing point correspondence to be fuzzily valued and uses deterministic annealing (DA) for optimization. But the employed DA is biased towards aligning the mass centers of two point sets. CDC [28] uses the covariance matrix of the transformation parameters to help recover point correspondence, resulting in better robustness to missing or extraneous structures. But the method is limited to simple transformations since the size of the covariance matrix is square times the number of transformation parameters.

GMM-based methods. These methods are robust to noise and outliers since they align distributions. CPD [24] casts point matching as a problem of fitting a Gaussian Mixture Model (GMM) representing one point set to another point set. This method was later extended in a Bayesian setting in [10]. GMMREG [15] uses GMMs to represent both point sets and minimizes the $L_2$-norm distance between the resulting GMMs. Support vector is used to generate Gaussian mixtures from point sets in [3], resulting in sparse Gaussian components. The efficiency of GMM based methods is improved by using filtering to solve the correspondence problem in [8]. The density variation problem of point sets is addressed in [17] by modeling the structure of scene as well. Hierarchical Gaussian mixture representation is used in [6] to improve the speed and accuracy of registration.

B. GLOBAL OPTIMIZATION METHODS

BnB based methods. BnB is a popular global optimization technique widely used in computer vision. It is used to align 3D shapes based on the Lipschitz optimization theory [18]. But the method assumes no occlusion or outliers. Go-ICP [14] uses BnB to optimize the ICP objective by exploiting the structure of the geometry of 3D rigid motions. The fast rotation search (FRS) method [1], [25] recovers rotation between 3D point sets by using stereographic projections. The general 6 degree rigid registration is accomplished by using a nested BnB algorithm. GOGMA registers two point sets by aligning the GMMs constructed from the original point sets [4]. A new type of rotation-invariant feature was proposed in [22], leading to a more efficient BnB based registration algorithm. Bayesian nonparametric point set representation and a new way of tessellating rotation space were proposed in [29]. BnB is used to optimize the RPM objective in [26], where branching over the correspondence variable and over the transformation variable are both considered. But the methods do not scale well due to lack of good structure for optimization.

Non-BnB based methods. FGR [41] optimizes a robust objective based on Black-Rangarajan duality between robust estimation and line processes. Mixture integer programming [13] and semidefinite programming [16], [23] have also been proposed for 2D/3D registration.

C. DEEP LEARNING METHODS

Correspondences-free methods. PointNetLK [37] uses the Lucas–Kanade (LK) algorithm to align global features of two point sets generated via a Pointnet [27] network. In a similar vein, FMR [12] uses an encoder to generate global features of two point sets and a decoder to convert the features back into point sets. LK is used to align the features and fidelity of the features is guaranteed by minimizing the Chamfer distances between the decoded point sets and the original point sets.

Correspondences-based methods. DCP [33] uses DGCNN [35] and Transformer [32] to extract features for each point set. Similarities of the features are used to determine point correspondences and SVD is used to recover rigid
transformation. PRNet [34] uses keypoint-based solutions to handle partial overlap. But the ability to handle partial overlap highly relies on the quality of keypoint detection. RPM-Net [38] uses the differentiable Sinkhorn layer and annealing to get soft assignments of point correspondences from hybrid features learned from both spatial coordinates and local geometry. DeepGMR [39] designs a neural network to extract pose-invariant correspondences between point sets and GMM parameters.

III. CASE ONE: TRANSFORMATION IS LINEAR W.R.T. ITS PARAMETERS

Given two point sets \( \mathcal{S} = \{x_i, i = 1, \ldots, n_x\} \) and \( \mathcal{S}' = \{y_j, j = 1, \ldots, n_y\} \) in \( \mathbb{R}^{d \times n} \) to be registered, where the column vectors \( x_i \) and \( y_j \) denote coordinates of points, following [19], we model point set registration as a mixed linearization–least square problem:

\[
\min E(P, \theta) = \sum_{i,j} p_{ij} \| y_j - J(x_i)\theta \|^2 = 1^T_{n_x} P \tilde{y}
\]

where \( T(x_i) = J(x_i)\theta \) is chosen to be linear w.r.t. its parameters \( \theta \). Here \( J(x_i) \) is called the Jacobian matrix of \( x_i \), \( \text{vec}(I_{n_d}) \) denotes the \( n_d \times n_d \) identity matrix. \( \text{vec}(P) \) denotes the \( n_d \times n_d \) matrix vector of all ones. \( \otimes \) denotes the Kronecker product. \( \text{diag}(\cdot) \) converts a vector into a diagonal matrix. For the correspondence matrix \( P \), its element \( p_{ij} = 1 \) indicates that there is a correspondence between \( x_i \) and \( y_j \) and \( p_{ij} = 0 \) otherwise. The constraint \( 1^T_{n_x} P 1_{n_y} = n_p, P \geq 0 \) indicates that the number of matches is fixed to \( n_p \), a predefined positive integer.

Given \( P, E \) is a convex quadratic function of \( \theta \). Thus, the optimal \( \theta \) minimizing \( E \) can be obtained by solving \( \frac{\partial E}{\partial \theta} = 0 \). After substituting the optimal \( \theta \) into \( E, \theta \) is eliminated and consequently we get a function in only one variable \( P \):

\[
E(P) = 1^T_{n_x} P \tilde{y} - y^T (P \otimes I_{n_d})^T J^T (\text{diag}(P_{1_{n_y}}) \otimes I_{n_d}) J^{-1} (P \otimes I_{n_d}) y
\]

where the matrix \( J \triangleq \begin{bmatrix} J^T(x_1), \ldots, J^T(x_{n_x}) \end{bmatrix}^T \) and the vectors \( y \triangleq \begin{bmatrix} y_1, \ldots, y_{n_y} \end{bmatrix}^T, \tilde{y} \triangleq \begin{bmatrix} \| y_1 \|^2, \ldots, \| y_{n_y} \|^2 \end{bmatrix}^T \).

To facilitate optimization of \( E(P) \), \( P \) needs to be vectorized. We define the vectorization of a matrix as the concatenation of its rows into a column vector, denoted by vec(\( \cdot \)). Let \( p \triangleq \text{vec}(P) \). To obtain a concise form of \( E(p) \), new denotations need to be introduced. Let

\[
1^T_{n_x} P \tilde{y} \equiv a^T p, J^T (P \otimes I_{n_d}) y \equiv A^T p, \text{vec}(J^T (\text{diag}(P_{1_{n_y}}) \otimes I_{n_d}) J) = \text{vec}(J^T_2 (P_{1_{n_y}} \otimes I_{n_d})) \equiv Bp
\]

where \( n_\theta \) denotes the dimension of \( \theta \) and matrix \( J_2 \triangleq \begin{bmatrix} J(x_1)^T, \ldots, J(x_{n_x})^T \end{bmatrix}^T \). Based on the formula

\[
\text{vec}(M_1 M_2 M_3) = (M_1 \otimes M_2^T) \text{vec}(M_3)
\]

for any multiplicable matrices \( M_1, M_2 \) and \( M_3 \), we have

\[
a = 1_{n_x} \otimes \tilde{y}, A = (J^T \otimes y^T) W_{n_y,n_y}, B = (J^T_2 \otimes I_{n_y}) W_{n_y,n_y} (1_{n_x} \otimes 1_{n_y})
\]

where the \( m \times n \) constant matrix \( W_{m,n} \) satisfies \( \text{vec}(C_{m,n} \otimes I_d) = W_{m,n} \) and \( C_{m,n} \) is a symmetric matrix.

Since \( 1_{n_x} \otimes 1_{n_y} = n_p \), a constant, for rows of \( B \) containing identical elements, the result of them multiplied by \( p \) can be replaced by constants. Besides, redundant rows can be removed. Since \( \text{mat}(Bp) \) is a symmetric matrix, \( B \) will contain redundant rows. Based on the above analysis, we hereby denote \( B_2 \) as the matrix formed as a result of \( B \) removing such rows. (Please refer to Sec. VI for examples of \( B_2 \)). Consequently, \( E(p) \) can be rewritten as

\[
E(p) = a^T p - p^T A^T \text{mat}(Bp)^{-1} A p
\]

where the nonzero elements of the constant matrix \( \Delta \) correspond to the rows of \( B \) containing identical elements. The elements of the constant matrix \( K \) take values in \( \{0, 1\} \) and record the correspondences between the rows of \( B_2 \) not containing identical elements and the rows of \( B \).

In view of the form of \( E(p) \), we can see that \( E(p) \) is determined by the variable \( [B_2^T, A^T, a]^T = \Gamma^T Q^T p \), which in turn is determined by a low dimensional variable \( t \triangleq Q^T p \). Here \( QT \) denotes the QR decomposition of matrix \( [B_2^T, A^T, a]^T \) with \( \Gamma \) being an upper triangular matrix and the columns of \( Q \) being orthogonal unity vectors. The specific form of \( E \) in terms of variable \( t \) is:

\[
E(t) = (\Gamma^T t)_A - (t^T A^T) \text{mat}((\Gamma^T t)_{B_2}) + \Delta^{-1} (\Gamma^T t)_A
\]

where \( (\Gamma^T t)_{B_2} \) denotes the vector formed by the elements of vector \( \Gamma^T t \) with indices equal to the row indices of the submatrix \( B_2 \) in matrix \( [B_2^T, A^T, a]^T \). Vectors \( (\Gamma^T t)_A \) and \( (\Gamma^T t)_B \) are similarly defined.
IV. CASE TWO: 3D SIMILARITY TRANSFORMATION

If the framework presented in the previous section is directly applied to 3D registration problem, 3D affine transformation needs to be employed. However, Its high number of parameters will cause the algorithm to become rather inefficient. To address this issue, we instead focus on 3D similarity transformation due to its relatively low number of parameters and derive the corresponding objective function.

Following [20], we model point matching as a linear assignment—least square problem:

\[
\min E(P, s, R, b) = \sum p_{ij} \|y_j - sR x_i - b\|^2 \\
= \|n^T P \tilde{y} + \tilde{x}^T P1_n + n_p\|b\|^2 - 2tr(sR^TPY)
\]

\[
-2t^T (Y^TP^T1_n - sRX^TP1_n) \quad \text{s.t.} \quad P \in \Omega, \ R \in SO_3, \ \tilde{x} \leq s \leq \tilde{\pi}
\]

where \( \Omega \) denotes the feasible region of \( p \), as determined by (2). Here the spatial transformation \( T(x_i|s,R,b) = sR x_i + b \) is chosen to be a similarity transformation with scaling factor \( s \), rotation matrix \( R \) and translation \( b \). The matrices \( X \triangleq [x_1, \ldots, x_n]^T \) and \( Y \triangleq [y_1, \ldots, y_n]^T \) and the vector \( \tilde{x} \triangleq [||x_1||^2_2, \ldots, ||x_m||^2_2]^T \).

Given \( P, s \) and \( R, E \) is a convex quadratic function of \( b \). Thus the optimal \( b \) minimizing \( E \) can be obtained by solving \( \frac{\partial E}{\partial b} = 0 \). After substituting the optimal \( b \) into \( E \), \( b \) is eliminated and we obtain a new function without the variable \( b \):

\[
E(P, s, R) = s^2(\tilde{x}^T P1_n - \frac{1}{n_p} \|X^TP1_n\|^2) + 1_m^T \tilde{y}^T
\]

\[
-2s tr[RX^TP(\frac{1}{n_p}P1_n^m)PY] - \frac{1}{n_p} \|1_m^T PY\|^2
\]

(14)

where the vector \( \tilde{x} \triangleq [||x_1||^2_2, \ldots, ||x_m||^2_2]^T \).

It is apparent that

\[
\min_{P,s,R} E(P, s, R) = \min_{P,s,R} \{ \min_p E(P, s, R) \} = \min_P E(P)
\]

where the objective function

\[
E(P) \triangleq \min_{s,R} E(P, s, R)
\]

Thus, the minimization of \( E(P,s,R) \) boils down to the minimization of \( E(P) \). Here the optimal \( R \) can be computed by first computing the singular value decomposition USV* of \( [X^T(P - \frac{1}{n_p}P11^T)PY]^T \), where \( S \) is a diagonal matrix and the columns of \( U \) and \( V \) are orthogonal unity vectors. Then the optimal \( R \) is \( R^* = \text{Udiag}(1, \ldots, 1, \det(UV^T))V^T \) [24]. By substituting \( R^* \) back into (14), \( R \) is eliminated and we get a (possibly concave) quadratic program only in one variable \( s \). Given the range of \( s \) as \( s \leq s \leq \tilde{\pi} \), one can easily solve this univariate quadratic program by comparing the function values at the two boundary points \( s, \tilde{\pi} \) and the extreme point to obtain the optimal \( s \).

\[ E(P) \] is characterized by the following proposition:

**Proposition 1.** \( E(P) \) is concave.

Please refer to [20] for the proof.

To facilitate optimization of \( E(P) \), \( E(P) \) needs to be expressed in terms of vector \( p \). To obtain a concise form of \( E(P) \), new denotations needed to be introduced. Let

\[
\begin{align*}
\vec{X}^TPY & \triangleq B_1 p, \\
X^TP1_n & \triangleq B_2 p, \\
\tilde{x}^TP1_n & \triangleq a_1 p
\end{align*}
\]

(16)

(17)

(18)

Based on formula (6), we have

\[
B_1 = X^T \otimes Y^T, \quad B_2 = X^T \otimes 1_n, \quad B_3 = 1_m \otimes Y^T, \quad a_1 = \tilde{x} \otimes 1_n, \quad a_2 = 1_m \otimes \tilde{y}
\]

(19)

(20)

With the above preparation, \( E(P) \) can be rewritten in terms of vector \( p \) as

\[
E(p) = a_2^T p - \frac{1}{n_p} \|B_3 p\|^2 + \min_{s,R} \{ s^2(a_1^T p - \frac{1}{n_p} \|B_2 p\|^2) \\
- 2s tr[R[\text{mat}(B_1 p) - \frac{1}{n_p} B_2 pp^T B_3]] \}
\]

(21)

In view of the form of \( E(p) \) in (21), we can see that \( E(p) \) is determined by a low dimensional variable \( B_1, B_2, B_3, a_1, a_2 \) and \( p = \Gamma^T Q p \), which in turn is determined by a variable \( t \triangleq Q p \). Here \( Q \Gamma \) denotes the QR decomposition of matrix \( \{B_1, B_2, B_3, a_1, a_2\} \) with \( \Gamma \)

being an upper triangular matrix and the columns of \( Q \) being orthogonal unity vectors. The specific form of \( E \) in terms of variable \( t \) is:

\[
E(t) = (\Gamma^T t)_{a_2} - \frac{1}{n_p} \|(\Gamma^T t)_{B_2}\|^2 \\
+ \min_{s,R} \{ s^2((\Gamma^T t)_{a_1} - \frac{1}{n_p} \|(\Gamma^T t)_{B_2}\|^2) \\
- 2s tr[R[\text{mat}((\Gamma^T t)_{B_1}) - \frac{1}{n_p} (\Gamma^T t)_{B_2} (\Gamma^T t)_{B_1}]) \}
\]

(22)

where \( (\Gamma^T t)_{B_1} \) denotes the vector formed by the elements of \( \Gamma^T t \) with indices equal to the row indices of the submatrix \( B \) in matrix \( \{B_1, B_2, B_3, a_1, a_2\}^T \). Vectors \( (\Gamma^T t)_{B_2}, (\Gamma^T t)_{B_1}, (\Gamma^T t)_{a_1}, \) and \( (\Gamma^T t)_{a_2} \) are similarly defined. One can deduce that the dimension of \( t \) is \( n_d^2 + 2n_d + 2 \).

V. OPTIMIZATION

The analysis in the previous section indicates that \( E(p) \) is linear w.r.t. the subspace \( L \triangleq \{p|Q^T p = 0\} \), i.e.,

\[
Q^T p = Q^T p' \implies E(p) = E(p')
\]

For the case when the transformation is linear w.r.t. its parameters, by Proposition 1 in [19], we have that \( E(p) \) is concave over the spectrahedra \( \Psi \triangleq \{p|\text{mat}(KBp) + \Delta > 0\} \supset \Omega \).

For the case when the transformation is 3D similarity, we have that \( E(p) \) is always concave. Based on the above analysis, it is natural to use a decompositional version of the PA algorithm [31], a global optimization algorithm suitable for functions which have low rank structures and are concave over the feasible region, to optimize \( E(p) \).
To use this algorithm, we first give the definition of polars.

**Definition 1.** For any set \( D \) in \( \mathbb{R}^n \), the set

\[
\mathcal{D} = \{ r \mid \langle r, p \rangle \leq 1, \forall p \in D \}
\]

is called the polar of \( D \).

One can deduce that the polar of a subspace is the same as its orthogonal complement. Some properties of polars are listed below.

**Proposition 2.** (i) The polar \( \mathcal{D} \) of any set \( D \) is a closed convex set containing the origin. If \( D \subset C \) then \( C^\circ \subset \mathcal{D} \).

(ii) \( \mathcal{D} \) is bounded if and only if \( 0 \in \text{int}D \), where \( \text{int}(\cdot) \) denote the interior of a set.

**Proposition 3.** (vertex-face duality) Let \( D = \{ p \mid \langle a_i, p \rangle \leq 1, a_i \neq 0, i = 1, \ldots, m \} \) be a polytope whose facets are defined by \( a_i \). Let \( \mathcal{D} \) be the polar of \( D \) with vertex set \( V(\mathcal{D}) \).

\[
V(\mathcal{D}) = \{ a_i, i = 1, \ldots, m \} \tag{23}
\]

Please refer to [31] for the proofs. Figure 1 illustrates the polar of a triangle.

**FIGURE 1.** A triangle \( ABC \) and its polar which is triangle \( DEF \).

Let us choose a feasible point \( p_0 \) of \( \Omega \). (In this work, \( p_0 \) is chosen as \( \frac{n_p}{n_{x+n}} 1_{n_{x+n}} \), corresponding to the furthest point correspondence.) For the convenience of derivation in the sequel, let us set \( \Omega \leftarrow \Omega - p_0, \Psi \leftarrow \Psi - p_0 \) and function \( E(p) \leftarrow E(p + p_0) \). Then \( 0 \) is a feasible point. Let \( \mathcal{P} \) be a feasible solution such that \( E(0) > E(\mathcal{P}) \).

\[
C \triangleq \{ p \mid p \in \Psi, E(p) \geq E(\mathcal{P}) \} \tag{24}
\]

for the case when the transformation is linear w.r.t its parameters or

\[
C \triangleq \{ p \mid E(p) \geq E(\mathcal{P}) \} \tag{25}
\]

for the case when the transformation is 3D similarity. By concavity and continuity of \( E(p) \) (over \( \Psi \) in the case when the transformation is linear w.r.t. its parameters), \( C \) is a closed convex set. Since \( 0 \in \text{int}C \), so \( C^\circ \) is bounded.

The essential idea of PA is to construct, starting from a polyhedron \( D_1 \supset C^\circ \), a nested sequence of polyhedrons \( D_1 \supset D_2 \supset \ldots \supset C^\circ \) that yields eventually either a polyhedron \( D_k \subset \Omega^\circ \) or a point \( p^k \in \Omega \setminus C \). For the former case, \( C^\circ \subset \Omega^\circ \), hence \( \Omega \subset C \) and the optimal solution is found. For the latter case, \( C \) is updated by resetting \( \mathcal{P} = p^k \) and the above procedure is restarted. Let

\[
\mu(v) = \max \{ \langle v, p \rangle \mid p \in \Omega \},
\]

we have the following result.

**Proposition 4.** If \( V \) is the vertex set of any polytope \( D \) such that \( C^\circ \subset D \), then

\[
\mu(v) \leq 1, \forall v \in V \Rightarrow C^\circ \subset \Omega^\circ \tag{26}
\]

**Proof.** If \( \max \{ \langle v, p \rangle \mid p \in \Omega \} \leq 1 \), \( \forall v \in V \), then \( C^\circ \subset \Omega^\circ \) because \( C^\circ \subset D \).

Let \( V_k \) be the vertex set of polytope \( D_k \). Let \( v^k = \arg \max \{ \langle v, p \rangle \mid v \in V_k \} \). By Proposition 4, if \( \mu(v^k) \leq 1 \), then \( C^\circ \subset \Omega^\circ \) and the optimal solution is found.

Otherwise, \( \mu(v^k) > 1 \). Let \( \mu(v^k) = \langle v^k, p^k \rangle, p^k \in \Omega \).

Then either \( p^k \in C \) or \( p^k \notin C \). For the latter case, \( C \) is updated by resetting \( \mathcal{P} = p^k \).

We next construct polytope \( D^{k+1} \) by pruning \( v^k \) from \( D^k \). Denote by \( \mathcal{P}^k \) the point where the half line from \( 0 \) through \( p^k \) meets \( \partial C \). \( \mathcal{P}^k \) satisfies \( \langle v^k, \mathcal{P}^k \rangle \geq \mu(v^k) > 1 \). So the inequality

\[
\langle \mathcal{P}^k, r \rangle \leq 1 \tag{27}
\]

holds for all \( r \in C^\circ \) but not for \( r = v^k \). Therefore, the cut (27) determines a polytope \( D_{k+1} = \{ r \in D_k \mid \langle \mathcal{P}^k, r \rangle \leq 1 \} \) such that \( v^k \notin D_{k+1} \).

**A. INITIAL POLYTOPE**

In order to exploit the low rank structure of \( E(p) \), the initial polyhedron \( D_1 \) is chosen in such a way that \( D_1 \subset L^1 \), where \( L^1 \) denotes the orthogonal complement of the linear subspace \( L \). If this can be done, then all \( D_k \subset L^k \) since \( D_k \subset D_1 \). Thus, the whole procedure will operate in \( L^k \), which will permit considerable computational saving since generally \( \text{dim}L^k = n_t < n_x n_p \).

A polyhedron \( D_1 \) satisfying the above requirement can be built as follows: We first construct a full dimensional polytope \( S_1 \) in subspace \( L^1 \) such that \( S_1 \subset C \) and \( 0 \in \text{ri}S_1 \), where \( \text{ri}(\cdot) \) denotes the interior of \( S_1 \) relative to \( \text{aff}S_1 \), the affine hull of \( S_1 \). We then let \( D_1 = (S_1 + L)^\circ \). Since \( 0 \in \text{ri}S_1 \) and \( S_1 \subset L^1 \), so \( 0 \in \text{int}(S_1 + L) \). Hence \( D_1 \) is compact. Since \( (S_1 + L)^\circ = S_1^\circ \cap L^1 \), so \( D_1 \subset L^1 \).

**B. THE PA ALGORITHM FOR MINIMIZING \( E(p) \)**

Based on the above preparation, we can now present the PA algorithm for minimizing \( E(p) \), whose pseudo-code is listed in Algorithm 1. Please also refer to Figure 2 for an illustration.

Note that in Algorithm 1, the problem

\[
\theta_k = \sup \{ \theta | E(\theta p^k) \geq E(\mathcal{P}), \theta p^k \in \Psi \} \tag{29}
\]

can be solved in two steps. First, we solve the following semidefinite problem

\[
\hat{\theta}_k = \arg \sup \{ \theta | \theta p^k \in \Psi \} \tag{30}
\]
FIGURE 2. An iteration of the PA algorithm. $\theta_k p^k \in \partial C$, $D_k = (S_k + L)^\circ$ and $D_k$ is updated as $D_{k+1} = D_k \cap \{x | \langle x, \theta_k p^k \rangle \leq 1 \}$.

via solvers such as Sedumi [30]. Then, we solve the following univariant optimization problem

$$
\theta_k = \arg \max \{ \theta | E(\theta p^k) \geq E(\overline{p}), 0 \leq \theta < \theta_k \} 
$$

(31)

via solvers such as fminbnd in Matlab (by minimizing $|E(\theta p^k) - E(\overline{p})|$). Note that the search can be further limited to the range [1, $\theta_k$] instead of [0, $\theta_k$] since $E(p^k) \geq E(\overline{p})$ and $E(\theta p^k)$ is concave. Also note that since $\theta_k \land \theta_k \Rightarrow E(\theta p^k) \land = \infty$, so $\theta_k$ will not be a solution of (31).

Parallel speedup. Note that in step 1, the linear assignment problem for each new vertex solved independently and repetitively. Thus, we can achieve speedup by running these problems in parallel. In this work, the parallel implementation is done on a GTX1080Ti graphics card.

C. CHOICE OF THE INITIAL POLYTOPE

It is apparent that if the polytope $S_1$ is chosen in such a way that it is a good approximation of $C$ (consequently $D_1$ is a good approximation of $C^\circ$), the PA algorithm will require less number of iterations to converge. In [31], $S_1$ is chosen as a simplex which has few facets and thus is a poor approximation of $C$, causing the resulting algorithm to converge slowly. To address this issue, in this work, we choose $S_1$ as a polytope with more facets, which is constructed as follows.

We first compute the vectors $p_i^k = \arg \max \{ \pm \langle q_i, p \rangle | p \in \Omega \}$, for $i = 1, \ldots, n_t$ (32) by solving linear assignment problems. The projection of $p_i^k$, $1 \leq i \leq n_t$, $j \in \{ +, - \}$ onto the subspace $L^\top$ are $p_i^j = Q Q^\top p_i^k$. Since $E$ is linear w.r.t. the subspace $L$, so $E(p_i^j) = E(p_i^k)$. After the above computation, we define the set $C$ according to (24) or (25) by choosing $p$ as the best $p_i^j$ for $1 \leq i \leq n_t$, $j \in \{ +, - \}$.

We next define $z_i^j = \alpha_i^j p_i^j$ to be the positive numbers such that $z_i^j = \alpha_i^j p_i^j \in \partial C$. We now let the polytope $S_1 = \text{conv}\{z_i^j, i = 1, \ldots, n_t, j = +, -\}$, where $\text{conv}(\cdot)$ denotes the convex hull of a point set. It is apparent that $S_1$ is full dimensional in subspace $L^\perp$, $S_1 \subset C$ and $0 \in \text{ri}S$. One can see that any facet of $S_1$ passes through the vertices $z_{j_k}^i$, where $j_k \in \{ +, - \}$. Thus, there is a total of $2n_t$ facets, which is much more than the $r + 1$ facets of a simplex. Therefore, $S_1$ chosen in this way can better approximate $C$ than a simplex.

Denote by $D_1$ the polar of $S_1 + L$. Since $0 \in \text{ri}S_1$ and $S_1 \subset L^\perp$, so $0 \in \text{int}(S_1 + L)$. Thus $D_1$ is bounded. Since $(S_1 + L)^\circ = S_1^\circ \cap L^\perp$, so $D_1 \subset L^\perp$, hence a point of $D_1$ is of the form $\sum t_i q_i$ and we can let polytope $\tilde{D}_1 = \pi(D_1)$. Since a facet of $S_1$ passes through the vertices $z_{j_k}^i$, where $j_k \in \{ +, - \}$, so based on Proposition 3, a vertex $t$ of $\tilde{D}_1$ satisfies

$$
[z_{j_k}^1, \ldots, z_{j_k}^{n_t}]^\top Q t = 1
$$

(33)

where $j_k \in \{ +, - \}$. Based on this formula, all the vertices of $\tilde{D}_1$ can be recovered.

Algorithm 1: PA algorithm for minimizing $E(p)$

1 Initialization: Let $\overline{p}$ be the best feasible solution so far available. Define the convex set $C$ according to Eq. (24) or (25). Let $q_1^1, \ldots, q_1^{n_t}$ be the columns of $Q$ (so that $L^\perp$ is the subspace generated by these vectors). Let $D_1 \equiv \pi(D_1)$ be a polytope in space $\mathbb{R}^{n_t}$, where $\pi : L^\perp \rightarrow \mathbb{R}$ denotes the linear map such that $y = \sum_{i=1}^{n_t} t_i q_i \Leftrightarrow \pi(y) = t$. Let $V_1$ be the vertex set of $D_1$.

2 for $k = 1, 2, \ldots$

3 For every new $t = (t_1, \ldots, t_{n_t}) \in V_k$, solve the linear assignment problem

$$
\max \left\{ \sum_{i=1}^{n_t} t_i \langle q_i, p \rangle | p \in \Omega \right\}
$$

(28)

to obtain its optimal value $\mu(t)$ and a basic optimal solution $p(t)$.

4 Let $t^k \in \arg \max \{ \mu(t) | t \in V_k \}$. If $\mu(t^k) \leq 1$ then terminate: $p$ is an optimal solution.

5 If $\mu(t^k) > 1$ and $p^k \neq p(t^k) \notin C$, then update the current best feasible solution and the set $C$ by resetting $\overline{p} = p^k$.

6 Compute

$$
\theta_k = \sup \{ \theta | E(\theta p^k) \geq E(\overline{p}), \theta p^k \in \Omega \}
$$

in the case that transformation is linear w.r.t. its parameters or

$$
\theta_k = \max \{ \theta | E(\theta p^k) \geq E(\overline{p}) \}
$$

in the case that transformation is 3D similarity.

Define

$$
\tilde{D}_{k+1} = \tilde{D}_k \cap \left\{ t | \sum_{i=1}^{n_t} t_i \langle q_i, p \rangle \leq \frac{1}{\theta_k} \right\}
$$

7 From $V_k$ derive the vertex set $V_{k+1}$ of $\tilde{D}_{k+1}$.

end
D. VERTEX ENUMERATION

In step 7 of the PA algorithm, we need to compute the intersection of a hyperplane with a polytope. This is the vertex enumeration (VE) problem [11], whose definition is as follows:

Let $D = \{t \mid \langle a_i, t \rangle - b_i \leq 0, i = 1, \ldots, m\}$ be a polytope with known vertex set $V(D)$ and let $H = \{t \mid \langle a_{m+1}, t \rangle - b_{m+1} = 0\}$ be a hyperplane such that $D \cap H$ is neither empty nor a facet of $D$, determine the vertex set $V(D)$ of $D$, as illustrated in Figure 3.

![Figure 3](image)

**FIGURE 3.** Vertex enumeration aims to determine the vertex set of $D \cap H$.

Let

$$V^+(D) = \{t \in V(D) : \langle a_{m+1}, t \rangle - b_{m+1} > 0\}$$  \hspace{1cm} (34)

$$V^-(D) = \{t \in V(D) : \langle a_{m+1}, t \rangle - b_{m+1} < 0\}$$  \hspace{1cm} (35)

Without loss of generality, let us assume $|V^-| \leq |V^+|$, where $|\cdot|$ denotes the cardinality of a set. For each $t \in V^-$, denote by $J(t)$ the set of constraints of $D$ which are active at $t$. Because of the way $D$ is constructed, vertex $t$ is nondegenerate, thus, we have $|J(t)| = n_t$ and linear independence of the corresponding system of linear equations

$$\langle a_i, t \rangle - b_i = 0 \quad (i \in J(t))$$  \hspace{1cm} (36)

Moreover, $t$ has $n_t$ neighboring vertices in $D$. That is, $n_t$ edges of $D$ are incident with $t$. Each line through $t$ in the direction of such an edge is the solution set of a system of $n_t - 1$ linear equations which can be obtained from (36) by dropping one equation. The set of new vertices in $D \cap H$ which are adjacent to $t$ contains the intersection points of these lines with the hyperplane $H$.

Without loss of generality, for simplicity of notation, we assume $J(t) = \{1, \ldots, n_t\}$. Then, for each $t \in V^-$, we have to consider the $n_t$ systems of $n_t$ linear equations

$$\langle a_i, t \rangle - b_i = 0 \quad (i \in \{1, \ldots, n_t\} \setminus \{l\})$$

$$\langle a_{m+1}, t \rangle - b_{m+1} = 0$$  \hspace{1cm} (37)

which arise when $l$ runs from $1$ to $n$. When a system in (37) has a solution, we have to check whether this solution satisfies the remaining inequalities of $D$.

Instead of directly solving (37), which is cumbersome, in the following, the simplex pivoting algorithm is employed to solve this problem. It works by introducing slack variables $s \in \mathbb{R}^{n_t}$ to write the binding inequalities of $J(t)$ in the form $At + I_{n_t}s = b$  \hspace{1cm} (38)

and the equation of $H$ in the form

$$a^T_{m+1}t + 0^T s = b_{m+1}$$  \hspace{1cm} (39)

One can transform (38) into

$$I_{n_t}t + A^{-1}s = A^{-1}b$$  \hspace{1cm} (40)

and transform (39) (by adding to (39) multiples of the rows of (38)) into

$$0^T t + \pi^T s = \bar{b}$$  \hspace{1cm} (41)

from which all possible new vertices neighboring $t$ can be obtained by pivoting on all the current nonbasic variables $s$ in the row (41).

E. TERMINATION CRITERION

The termination criterion $\max_{i} \mu_i \leq 1$ guarantees that the PA algorithm is globally optimal. Nevertheless, meeting this criterion is generally computationally expensive and often not necessary since satisfactory solutions can often be generated before this condition is met. Based on the above analysis, we therefore relax this criterion into $\max_{i} \mu_j \leq 1 + \epsilon$, where $\epsilon$ is a preset small positive value. Consequently, our algorithm becomes approximately globally optimal.

Since higher dimensional space of $t$ tends to lead to slower convergence, so instead of directly setting $\epsilon$, we let $\epsilon = n_t \epsilon_0$ by also taking into account the dimension $n_t$ of the space of $t$ and set $\epsilon_0$ instead.

VI. EXPERIMENTS

We implement our method under Matlab 2019b and compare it with other methods on a PC with 3.5 GHz CPU and 32G RAM. For the comparison methods which only output point correspondences, the generated correspondences are used to find the best affine transformations between two point sets. We define error as the root mean squared squared difference between the coordinates of transformed ground truth model inliers and those of their corresponding scene inliers. For our algorithm, we set the parameter $\epsilon_0 = 0.3$.

A. 2D SYNTHESIZED DATASETS

We compare our method with RPM-CAV [20] and Go-ICP [14]. Both of them are based on global optimization, can handle partial overlap and allow arbitrary rotation and translation between two point sets, making them good candidates for comparison.

2D similarity and affine transformations are respectively considered for our method. For the former, we have its formulation $T(x_i|\theta) = [x_i^1 - \theta_1 x_i^2 - \theta_3, \theta_2 x_i^1 + \theta_1 x_i^2 + \theta_4]^T$ where $\theta = [\theta_1, \ldots, \theta_4]^T$. Therefore, the Jacobian matrix $J(x_i) = \begin{bmatrix} x_i^1 & -x_i^2 & 1 & 0 \\ x_i^2 & x_i^1 & 0 & 1 \end{bmatrix}$. It can be verified that the rows of $B_2 = B([1, 3, 4], :) \) constitute the unique rows of $B$ not equal to scaled versions of $1_{n \times n}$.

For the latter, we have its formulation $T(x_i|\theta) = \begin{bmatrix} \theta_1 x_i^1 + \theta_2 x_i^2 + \theta_5, \theta_3 x_i^1 + \theta_4 x_i^2 + \theta_6 \end{bmatrix}^T$ where $\theta = \begin{bmatrix} \theta_1, \ldots, \theta_6 \end{bmatrix}^T$. Therefore, the Jacobian matrix $J(x_i) = \begin{bmatrix} x_i^1 & -x_i^2 & 1 & 0 \\ x_i^2 & x_i^1 & 0 & 1 \end{bmatrix}$. It can be verified that the rows of $B_2 = B([1, 3, 4], :) \) constitute the unique rows of $B$ not equal to scaled versions of $1_{n \times n}$.

VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
First 3 columns: model point sets (left column) and examples of scene point sets in the deformation and noise tests, respectively (column 2 to 3). Last 4 columns: examples of model and scene point sets in the outlier (column 4, 5) and occlusion + outlier (column 6, 7) tests.

FIGURE 4. First 3 columns: model point sets (left column) and examples of scene point sets in the deformation and noise tests, respectively (column 2 to 3). Last 4 columns: examples of model and scene point sets in the outlier (column 4, 5) and occlusion + outlier (column 6, 7) tests.

![Graphs showing error rates for various tests](image)

FIGURE 5. Average matching errors by our method, RPM-CAV [20] and Go-ICP [14] with different \( n_p \) values (chosen from 1/2 to 1/1 the ground truth value) over 100 random trials for the 2D deformation, noise, outlier and occlusion+outlier tests.

![Graphs showing error rates for various methods](image)
formation runs quite efficiently across all types of disturbances. In comparison, Go-ICP runs slowly for the outlier test. This demonstrates high efficiency of the proposed method using similarity transformation. In terms of different choices of transformations, our method using affine transformation is slower than our method using similarity transformation, particularly for the outlier and occlusion+outlier tests. This is because affine transformation has more number of parameters, causing our algorithm to converge slowly.

TABLE 1. Average running time (in seconds).  

| Method         | Deform. | Noise | Outliers | Occl.+Outl. |
|----------------|---------|-------|----------|-------------|
| ours (simi, 1/1) | 8.99    | 9.25  | 9.79     | 5.84        |
| ours (simi, 1/2) | 7.18    | 7.33  | 9.27     | 5.84        |
| ours (aff, 1/1)  | 18.98   | 20.30 | 304.83   | 141.04      |
| ours (aff, 1/2)  | 33.82   | 35.42 | 105.39   | 60.40       |
| RPM-cav [20] (1/1) | 2.35    | 2.27  | 3.40     | 1.93        |
| RPM-cav [20] (1/2) | 3.09    | 3.31  | 2.34     | 1.34        |
| Go-ICP [14] (1/1) | 6.62    | 6.69  | 42.98    | 2.02        |
| Go-ICP [14] (1/2) | 0.06    | 0.06  | 0.17     | 0.06        |

B. 2D POINT SETS EXTRACTED FROM IMAGES

Point sets extracted from images are a more realistic type of data for testing methods. We test different methods on 2D point sets extracted via the Canny edge detector from several images in the Caltech-256 [9] and VOC2007 [7] datasets, as illustrated in Fig. 6. To test a method’s ability at handling rotations, model point sets are rotated 180 degree before being matched to scene point sets.

The registration results by different methods are presented in Fig. 6. Our method using similarity transformation performs much better than our method using affine transformation. This is because the data set contains significant clutters, causing transformations with high degree of transformation freedom (e.g. affine transformation) to easily lead to unconstrained registration results. Another reason is that for our method, the tolerance error $\epsilon = n_t \epsilon_0$ for affine transformation is actually set larger than that of similarity transformation given that $\epsilon_0$ is set the same for both types of transformations, whereas $n_t$ is large for affine transformation and small for similarity transformation. Go-ICP’s performance is between our method using similarity transformation or affine transformation.

C. 3D SYNTHESIZED DATASETS

In this section, we compare our method with Go-ICP [14], FRS [25] and FGR [41], all of which are based on global optimization and allow arbitrary rigid transformations between two point sets, making them good candidates for comparison. We also compare with PRNet [34], which can handle partial overlap. PRNet is pretrained on ModelNet40 [36].

Similar to the experimental setup in Sec. VI-A, we conduct 4 categories of tests to evaluate different methods’ robustness to various types of disturbances: i) Deformation test, ii) Noise test, iii) Outlier test and iv) Occlusion + Outlier test. Fig. 7 illustrates these tests.

The matching errors by different methods are presented in Fig. 8. Overall, the proposed method performs much better than other methods. It is insensitive to different degrees of non-rigid deformation and positional noise. It is also less sensitive to different choices of $n_p$ than Go-ICP. In comparison, Go-ICP only performs well when the $n_p$ value is close to the ground truth. FGR and PRNet perform poorly for all the tests. The reason for FGR’s poor performance is that our dataset contains significant non-rigid deformation, to which FGR is quite sensitive. The reason for PRNet’s poor performance is that our dataset contains arbitrary rotations between two point sets, of which PRNet is incapable of handling. The average matching errors across all degrees of disturbance and all types of disturbances are 0.11 for our method (1/1), 0.17 for our method (1/2), 0.17 for Go-ICP (1/1), 0.30 for Go-ICP (1/2), 0.17 for FRS, 0.31 for FGR and 0.29 for PRNet.

The average running time (in seconds) by different methods are listed in Table 2. Our method’s running time is higher than those of other methods. This is because 3D similarity transformation has high number of parameters, causing our method to converge slowly.

TABLE 2. Average running time (in seconds).  

| Method         | Deform. | Noise | Outliers | Occl.+Outl. |
|----------------|---------|-------|----------|-------------|
| ours (1/1)     | 312.1   | 417.8 | 1640     | 1729        |
| ours (1/2)     | 5494.9  | 5544.9| 11709    | 11723       |
| Go-ICP [14] (1/1) | 9.6    | 5.7   | 31.9     | 4.6         |
| Go-ICP [14] (1/2) | 1.3    | 1.3   | 0.8      | 0.8         |
| FRS [25]       | 0.8     | 0.3   | 73.1     | 88.8        |
| FGR [41]       | 0.002   | 0.002 | 0.002    | 0.002       |
| PRNet [34]     | 0.03    | 0.03  | 0.03     | 0.03        |

VII. CONCLUSION

We proposed an approximately global optimization-based algorithm for aligning partially overlapping point sets with the property that it is invariant to the corresponding transformation. The method works by reducing the RPM objective to a function of a low dimensional variable and then using the PA algorithm to optimize the objective over its concavity region. Two cases of transformations, transformation is linear w.r.t. its parameters and 3D similarity transformation, are discussed, to deal with 2D/3D registration problems respectively. Experiments on 2D/3D data sets demonstrated better robustness of the proposed method over state-of-the-art algorithms for tasks involving various types of disturbances. It is also efficient when the number of transformation parameters is small.

REFERENCES

[1] J.-C. Bazin, Y. Seo, and M. Pollefeys. Globally optimal consensus set maximization through rotation search. In Asian Conference on Computer Vision, 2012.
[2] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. IEEE Trans. Pattern Analysis and Machine Intelligence, 14(2):239–256, 1992.
[3] Dylan Campbell and Lars Petersson. An adaptive data representation for robust point-set registration and merging. In ICCV, 2015.
FIGURE 6. Registration on point sets extracted from images. Subfigure (a): model images with extracted point sets superimposed. Subfigure (b) to (f): scene images with extracted point sets superimposed, registration results by our method using similarity or affine transformations and Go-ICP [14]. The $n_p$ value for every method is chosen as $0.9$ the minimum of the cardinalities of two point sets.
Computer Vision and Pattern Recognition, pages 352–359, 2014.
[20] Wei Lian and Lei Zhang. A concave optimization algorithm for matching partially overlapping point sets. Pattern Recognition, 103, 7 2020.
[21] Wei Lian, Lei Zhang, and Ming-Hsuan Yang. An efficient globally optimal algorithm for asymmetric point matching. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016.
[22] Yinlong Liu, Chen Wang, Zhijian Song, and Manning Wang. Efficient global point cloud registration by matching rotation invariant features through translation search. In The European Conference on Computer Vision (ECCV).
[23] Haggar Maron, Nadav Dym, Itay Kezurer, Shahr Kovalsky, and Yaron Lipman. Point registration via efficient convex relaxation. ACM Trans. Graph., 35(4), July 2016.
[24] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(12):2262–2275, 2010.
[25] Álvaro Parra, Tat-Jun Chin, Anders Eriksson, Hongdong Li, and David Suter. Fast rotation search with stereographic projections for 3d registration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(11):2227–2240, 2016.
[26] Frank Pfeuffer, Michael Stiglmayr, and Kathrin Klamroth. Discrete and geometric branch and bound algorithms for medical image registration. Annals of Operations Research, 196(1):737–765, 2012.
[27] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. In CVPR, 2017.
[28] Michal Softa, Gehua Yang, and Charles V. Stewart. Simultaneous covariance driven correspondence (cdc) and transformation estimation in the expectation maximization framework. In IEEE Conf. Computer Vision and Pattern Recognition, pages 1–8, 2007.
[29] J. Straub, T. Campbell, J. P. How, and J. W. Fisher. Efficient global point cloud alignment using bayesian nonparametric mixtures. In Proc. IEEE Conf. Comput. Vis. Pattern Recog., pages 2403–2412, 2017.
[30] J.F. Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. Optimization Methods and Software, pages 625–653, 1999.
[31] Hoang Tuy. Convex Analysis and Global Optimization. Springer, 2016.
[32] Ashish Voswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, unclearukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17, pages 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.
[33] Y. Wang and J. Solomon. Deep closest point: Learning representations for point cloud registration. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pages 3522–3531, 2019.
[34] Yue Wang and Justin M Solomon. Prnet: Self-supervised learning for partial-to-partial registration. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019.
[35] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. ACM Trans. Graph., 38(5), Oct. 2019.
[36] Zhirong Wu, Shuran Song, Aditya Khosla, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shape modeling. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
[37] Aoki Yasuhiro, Hunter Goforth, Ranagaprasad Arun Sridivasan, and Simon Lucey. Pointnet: Robust and efficient point cloud registration using pointnet. In CVPR, 2019.
[38] Z. J. Yew and C. H. Lee. Rpm-net: Robust point matching using learned features. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 11821–11830, 2020.
[39] Wentao Yuan, Benjamin Eckart, Kihwan Kim, Varun Jampani, Dieter Fox,
and Jan Kautz. Deepgmr: Learning latent gaussian mixture models for registration. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, Computer Vision – ECCV 2020, pages 733–750, Cham, 2020. Springer International Publishing.

[40] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. International Journal of Computer Vision, 13(2):119–152, 1994.

[41] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, Computer Vision – ECCV 2016, pages 766–782, Cham, 2016. Springer International Publishing.

***