AN UNSUPERVISED ITERATIVE $N$-DIMENSIONAL POINT-SET REGISTRATION ALGORITHM

P. Hosseinbor, R. Zhdanov, and A. Ushveridze

An unsupervised iterative $N$-dimensional point-set registration algorithm for unlabeled data (i.e., the correspondence between points is unknown) based on linear least squares is proposed. The algorithm considers all possible point pairings and iteratively aligns the two sets until the number of point pairs does not exceed the maximum number of allowable one-to-one pairings.

1. Introduction

Point-set registration seeks to align an $N$-dimensional query point-set (or point cloud) to some $N$-dimensional reference point-set via some mathematical transform and has been extensively studied in computer vision. For labeled data, an important class of solutions is given by linear least squares techniques [1, 2, 6, 11]. Unlike the cited works [1, 6, 11], the paper [2] treats a more general case of two point-sets of unequal size and does not assume correspondence. They first established the correspondence by numerically determining the matching pairs support between two point sets (i.e., finding an optimal subset of pairings between these two sets) and then derived (analytic) least squared solutions to the transformation parameters that optimally align the (labeled) optimal subset of pairings. However, their approach is computationally expensive and has quartic polynomial (average-case) complexity.

For the unlabeled data, the alignment of two point patterns is a two-part problem; both the correspondence and the optimal affine transform that minimizes some dissimilarity metric between two point sets need to be determined. One of the most important classes of solutions for the unlabeled case is the iterative closest point (ICP).

Various point pattern matching algorithms were proposed for unlabeled data in [3, 5, 7–10]. However, the necessary optimization for each algorithm is numerically based. The works [3, 8, 9] determined the optimal affine transform and, at the same time, the correspondence by numerically solving a constrained least-squares problem. The method proposed in [5] models each point-set as a Gaussian mixture and determines the appropriate alignment by the (nonlinear) optimization of the L2 distance between two Gaussian mixtures.

Unlike the numerical optimization schemes discussed above, a much more computationally efficient approach would be an analytic optimization scheme for the unlabeled data, which is proposed in this paper. Our alignment approach is similar to the labeled techniques of [1, 2, 6, 11] by the fact that we derive analytic solutions for the optimal affine transform via linear least squares. However, unlike these works, we do not assume or establish correspondence prior to registration but use registration to establish correspondence. To the best of our knowledge, our derivation of the (closed-form) linear least-square solutions for the registration of two unlabeled point-sets,
although remarkably simple, is absent from the available literature. The presented derivation is generalized to any $N$-dimensional point-set (i.e., each point in a point-set resides in $\mathbb{R}^N$), and the obtained solutions are then used to create an unsupervised iterative $N$-dimensional point-set registration algorithm. The case $N = 2$ was studied in [4] and utilized in the context of fingerprint matching.

The iterative closest point algorithm (ICP) is probably the method most closely related to our method (among the methods known from the literature) because both are iterative, pertinent to the unlabeled data, and employ linear least squares to estimate the optimal alignment parameters. However, there are major differences between these two methods: the ICP inherently assumes that the correspondence between points can be inferred from the distances between the points.

The paper is organized as follows. In Section 2, we lay the theoretical foundations of the proposed algorithm. In Section 3, we describe its numerical implementation and, finally, in Section 4, we conclude with a discussion of the algorithm and its potential applications.

2. Theory

Consider two $N$-dimensional point sets $U$ and $V$ comprising $N_U$ and $N_V$ singular points, respectively. We refer to $U$ and $V$ as the query and reference point-sets, respectively. The Cartesian coordinates of singular points are expressed in the form of an $N$-dimensional vector:

$$u_i \in \mathbb{R}^N \in U, \quad i = 1, \ldots, N_U,$$

$$v_k \in \mathbb{R}^N \in V, \quad k = 1, \ldots, N_V.$$

The elements of $u_i$ and $v_k$ are denoted by $u_i^j$ and $v_k^j$, respectively, where $j = 1, \ldots, N$. $U$ and $V$ can be interpreted as matrices whose columns are formed by the vectors $u_i$ and $v_k$, respectively, i.e.,

$$U \in \mathbb{R}^{N_U \times N} \quad \text{and} \quad V \in \mathbb{R}^{N_V \times N}.$$

At the same time, $u^j$ and $v^j$ can be interpreted as the features of $U$ and $V$, respectively.

We want to register the point-set $U$ to $V$. There are $N_U N_V$ possible matching (cross) pairs and at most $\min\{N_U, N_V\}$ one-to-one matching pairs. Let $m_{ik}$ denote the weight of a matching pair. The weight can be interpreted as a probability that the points $u_i$ and $v_k$ match locally. We assume that, for each (cross) pair, the (initial) $m_{ik}$ is determined prior to alignment. One way of computing $m_{ik}$ was discussed in [4] within the context of fingerprint matching.

2.1. Case I: No Scale. We apply a global rotation and translation to a point set $U$ such that

$$u_i' = Lu_i + t,$$

where $L$ is an $N \times N$ rotation matrix and $t$ is an $N$-d vector of translation parameters. Since $L$ is a rotation matrix, it is orthogonal, i.e.,

$$LL^T = L^TL = I_{N \times N},$$

and has a determinant of 1.
The measure of closeness of the transformed point set $\mathbf{U}$ and the template set $\mathbf{V}$ can be defined as the weighted sum of the squared distances between their points (i.e., the Euclidean-distance metric):

$$e(\mathbf{U}, \mathbf{V}; \mathbf{L}, \mathbf{t}) = \sum_{i=1}^{N_U} \sum_{k=1}^{N_V} m_{ik} (\mathbf{u}_i - \mathbf{v}_k)^T (\mathbf{u}_i - \mathbf{v}_k) \sum_{i=1}^{N_U} \sum_{k=1}^{N_V} m_{ik}.$$  \hfill (2)

Interpreting $m_{ik}$ as the probability of matching between $\mathbf{u}_i$ and $\mathbf{v}_k$ gives

$$\sum_{i=1}^{N_U} \sum_{k=1}^{N_V} m_{ik} = 1.$$  

Expanding the product, we obtain

$$e(\mathbf{U}, \mathbf{V}; \mathbf{L}, \mathbf{t}) = \sum_{i=1}^{N_U} \sum_{k=1}^{N_V} m_{ik} \mathbf{v}_k^T \mathbf{v}_k - 2 \text{Tr} (\mathbf{L} \mathbf{u}_i \mathbf{v}_k^T) + 2 \text{Tr} (\mathbf{L} \mathbf{u}_i \mathbf{t}^T) + \mathbf{u}_i^T \mathbf{u}_i - 2 \mathbf{t}^T \mathbf{v}_k + \mathbf{t}^T \mathbf{t},$$  \hfill (3)

where “Tr” denotes the trace operator.

We seek the optimal values of the parameters $\mathbf{L}$ and $\mathbf{t}$ that minimize $e(\mathbf{U}, \mathbf{V}; \mathbf{L}, \mathbf{t})$. Hence, the constrained optimization problem to be solved is

$$\begin{align*}
\text{minimize} & \quad e(\mathbf{U}, \mathbf{V}; \mathbf{L}, \mathbf{t}) \\
\text{subject to} & \quad \mathbf{L} \mathbf{L}^T = \mathbf{I}_{N \times N}, \\
& \quad \det \mathbf{L} = 1.
\end{align*}$$

The Lagrangian for the above (constrained) optimization problem is

$$\mathcal{L}(\mathbf{L}, \mathbf{t}, \alpha, \lambda) = e(\mathbf{U}, \mathbf{V}; \mathbf{L}, \mathbf{t}) + \text{Tr} (\alpha (\mathbf{L} \mathbf{L}^T - \mathbf{I}_{N \times N})) + \lambda (\det \mathbf{L} - 1),$$  \hfill (4)

where $\alpha$ is an $N \times N$ symmetric matrix of Lagrangian multipliers (it is symmetric because $\mathbf{L} \mathbf{L}^T$ is symmetric and, consequently, contains $N(N + 1)/2$ Lagrangian multipliers) and $\lambda$ is a scalar Lagrangian multiplier.

Prior to proceeding further, we define the weighted average coordinate vectors as follows:

$$\bar{\mathbf{u}} = \sum_{i=1}^{N_U} \sum_{k=1}^{N_V} m_{ik} \mathbf{u}_i,$$

$$\bar{\mathbf{v}} = \sum_{i=1}^{N_U} \sum_{k=1}^{N_V} m_{ik} \mathbf{v}_k.$$  

Optimizing Eq. (4) with respect to $\mathbf{t}$, we get

$$\hat{\mathbf{t}} = \bar{\mathbf{v}} - \mathbf{L} \bar{\mathbf{u}}.$$  \hfill (5)
Substituting Eq. (5) back into our Lagrangian and then taking its partial derivative with respect to $L$, we find

$$\frac{\partial L}{\partial L} = 2 \left( \overline{v} \overline{u}^T - \sum_{i=1}^{N_U} \sum_{k=1}^{N_V} m_{ik} v_k u_i^T + \alpha L \right) + \lambda L = 0_{N \times N}, \quad (6)$$

where $0_{N \times N}$ is an $N \times N$ matrix of zeros. Let

$$Z = \sum_{i=1}^{N_U} \sum_{k=1}^{N_V} m_{ik} v_k u_i^T - \overline{v} \overline{u}^T,$$

which is an $(N \times N)$ cross-covariance matrix assumes to be invertible. Specifically,

$$Z = \begin{pmatrix}
\text{cov}(u^1, v^1) & \text{cov}(u^1, v^2) & \text{cov}(u^1, v^3) & \cdots & \text{cov}(u^1, v^N) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\text{cov}(u^N, v^1) & \text{cov}(u^N, v^2) & \text{cov}(u^N, v^3) & \cdots & \text{cov}(u^N, v^N)
\end{pmatrix},$$

where $\text{cov}(u^j, v^{j'})$ denotes the covariance between the features $u^j$ and $v^{j'}$:

$$\text{cov}(u^j, v^{j'}) = \sum_{i=1}^{N_U} \sum_{k=1}^{N_V} m_{ik} u_i^j v_k^{j'},$$

$$- \left( \sum_{i=1}^{N_U} \sum_{k=1}^{N_V} m_{ik} u_i^j \right) \left( \sum_{i=1}^{N_U} \sum_{k=1}^{N_V} m_{ik} v_k^{j'} \right). \quad (7)$$

Solving Eq. (6) for $L$ yields

$$\hat{L} = \left( \alpha + \frac{\lambda}{2} I_{N \times N} \right)^{-1} Z.$$

Since $L$ is an orthogonal matrix, we have

$$LL^T = \left( \alpha + \frac{\lambda}{2} I_{N \times N} \right)^{-1} ZZ^T \left( \alpha + \frac{\lambda}{2} I_{N \times N} \right)^{-1} = I_{N \times N}.$$

Thus, isolating the cross-covariance $Z$, we obtain

$$\left( \alpha + \frac{\lambda}{2} I_{N \times N} \right)^2 = ZZ^T.$$

Hence,

$$\hat{L} = \pm \left( \sqrt{ZZ^T} \right)^{-1} Z. \quad (8)$$

The minimum of the error function corresponds to $\left( \sqrt{ZZ^T} \right)^{-1} Z$. Thus, we discard the solution $-\left( \sqrt{ZZ^T} \right)^{-1} Z$. 
The matrix $ZZ^T$ exhibits several interesting properties:

1. It is real and symmetric. It is real because the elements of $Z$ are covariances, which are always real by the definition of covariance. It is also symmetric because $ZZ^T = (ZZ^T)^T$.

2. It is diagonalizable as a consequence of being real and symmetric. We exploit this fact in what follows.

3. It is positive-definite. We now prove this property. First, $x^T ZZ^T x = (Z^T x)^T Z^T x = \|Z^T x\|^2 \geq 0 \quad \forall x \in \mathbb{R}^N \setminus 0$, which establishes that $ZZ^T$ is inherently positive semidefinite. Recall that we further assume that $Z$ is invertible, which means that $Zx = 0$ if and only if $x = 0$. Thus, $\|Z^T x\|^2 > 0 \quad \forall x \in \mathbb{R}^N \setminus 0$.

4. All its eigenvalues are positive as a consequence of its positive-definiteness.

Equation (8) requires taking the square root of the matrix $ZZ^T$. According to the spectral theorem, any real symmetric matrix can be diagonalized by an orthogonal matrix:

$$ZZ^T = PDP^T,$$

where $P$ is an $N \times N$ orthogonal matrix and $D$ is an $N \times N$ diagonal matrix; $P$ and $D$ are formed by the eigenvectors and eigenvalues of $ZZ^T$, respectively. The square root of $ZZ^T$ is then $PD^{1/2}P^T$ because

$$\left( PD^{1/2}P^T \right)^2 = PD^{1/2}P^T PD^{1/2}P^T = PDP^T = ZZ^T.$$

Thus, any $N \times N$ matrix with $N$ distinct eigenvalues has $2^N$ square roots because the square root of each eigenvalue can be either positive or negative. If a matrix of this kind is, in addition, positive-definite, then it has precisely one positive-definite square root. The positive-definite square root corresponds to the case where only the positive square root of each eigenvalue is taken. We use these properties of $ZZ^T$ to extract only its positive-definite square-root. Hence, $D^{1/2}$ is formed by the positive square roots of the eigenvalues of $ZZ^T$. To this end, we employ the notation $(\sqrt{\arg})_{PD}$ in order to denote the positive-definite square root of any square matrix. Taking all these observations into account, we can rewrite Eq. (8) as follows:

$$\hat{L} = P \left( \sqrt{D} \right)^{-1}_{PD} P^T Z.$$

Summarizing, we conclude that the optimal alignment parameters are

$$\hat{L} = \left( \sqrt{ZZ^T} \right)^{-1}_{PD} Z,$$

$$\hat{t} = \bar{v} - \left( \sqrt{ZZ^T} \right)^{-1}_{PD} Z \bar{u}.$$
or, equivalently,

\[
\hat{L} = P \left( \sqrt{D} \right)^{-1}_{PD} P^T Z, \\
\hat{t} = \bar{v} - P \left( \sqrt{D} \right)^{-1}_{PD} P^T \bar{u}.
\]

The minimum squared error is found by substituting the optimal alignment parameters back into Eq. (3). This yields

\[
e_{\text{min}} = \left( \sum_{i=1}^{N_U} \sum_{k=1}^{N_V} m_{ik} u_i^T u_i - \bar{u}^T \bar{u} \right) + \left( \sum_{i=1}^{N_U} \sum_{k=1}^{N_V} m_{ik} v_i^T v_i - \bar{v}^T \bar{v} \right) - 2 \text{Tr} \left( \left( \sqrt{Z^T} \right)_{PD} \right).
\]

Note that

\[
\sum_{i=1}^{N_U} \sum_{k=1}^{N_V} m_{ik} u_i^T u_i - \bar{u}^T \bar{u} = \sum_{j=1}^{N} \sigma_{u_j}^2,
\]

where

\[
\sigma_{u_j}^2 = \sum_{i=1}^{N_U} \sum_{k=1}^{N_V} m_{ik} (u_i^j)^2 - \left( \sum_{i=1}^{N_U} \sum_{k=1}^{N_V} m_{ik} u_i^j \right)^2.
\]

In other words, \( \sigma_{u_j}^2 \) is the variance of the feature \( u_j \) and, likewise, \( \sigma_{v_j}^2 \) is the variance of the feature \( v_j \). Therefore, we can rewrite the minimum squared error as follows:

\[
e_{\text{min}} = \sum_{j=1}^{N} \left( \sigma_{u_j}^2 + \sigma_{v_j}^2 \right) - 2 \text{Tr} \left( \left( \sqrt{Z^T} \right)_{PD} \right),
\]

where we have used the fact that

\[
\text{Tr} \left( P (\sqrt{D})_{PD} P^T \right) = \text{Tr} \left( (\sqrt{D})_{PD} \right).
\]

Note that if we use the relation

\[
\hat{L} = -\left( \sqrt{Z^T} \right)^{-1}_{PD} Z,
\]
then the generated error would be
\[ e' = \sum_{j=1}^{N} \left( \sigma_{u_j}^2 + \sigma_{v_j}^2 \right) + 2 \text{Tr} \left( \left( \sqrt{ZZ^T} \right)_{PD} \right). \]

All eigenvalues of a positive-definite matrix are always positive, which means that
\[ \text{Tr} \left( \left( \sqrt{ZZ^T} \right)_{PD} \right) > 0 \]
and, hence, \( e' > e_{\text{min}} \).

2.2. Case II: Uniform Scale. If we include a uniform scale \( s \) into our affine transform, then we get
\[ u'_i = sLu_i + t. \] (13)

Thus, our cost function becomes
\[
e(U, V; L, t, s) = \sum_{i=1}^{N_U} \sum_{k=1}^{N_V} m_{ik} \left[ v^T_k v_k - 2s \text{Tr} (Lu_i v^T_k) + 2s \text{Tr} (Lu_i t^T) + s^2 u^T_i u_i - 2t^T v_k + t^T t \right]
\] (14)

and our resulting optimization problem is
\[
\begin{align*}
\text{minimize} & \quad e(U, V; L, t, s) \\
\text{subject to} & \quad LL^T = I_{N \times N}, \\
& \quad \det L = 1.
\end{align*}
\]

The Lagrangian for the above (constrained) optimization problem is
\[
\mathcal{L}(L, t, s, \alpha, \lambda) = e(U, V; L, t, s) + \text{Tr} \left( \alpha (LL^T - I_{N \times N}) \right) + \lambda (\det L - 1).
\] (15)

Optimizing Eq. (15) with respect to \( t \), we find
\[ \hat{t} = \bar{v} - sL\bar{u}. \] (16)

Substituting Eq. (16) back into our Lagrangian and then taking its partial derivative with respect to \( L \), we obtain
\[ \frac{\partial \mathcal{L}}{\partial L} = 2 (\alpha L - sZ) + \lambda L = 0_{N \times N}. \]
Solving this equation for $L$, we get

$$\hat{L} = s \left( \alpha + \frac{\lambda}{2} I_{N \times N} \right)^{-1} Z.$$

Exploiting the fact that $L$ is an orthogonal matrix, we obtain

$$\hat{L} = \left( \sqrt{ZZ^T} \right)^{-1}_{PD} Z,$$

which is the same rotation matrix as in the nonscalar case. Thus, the inclusion of a fixed scale into our affine transform does not affect the rotation matrix.

Substituting both Eqs. (16) and (17) in the Lagrangian, Eq. (15), and optimizing it with respect to $s$, we can write

$$\frac{\partial L}{\partial s} = 2 \left( s \left( \sum_{i=1}^{N_U} \sum_{k=1}^{N_V} m_{ik} u_i^T u_i - \bar{u}^T \bar{u} \right) - \text{Tr}\left( \left( \sqrt{ZZ^T} \right)_{PD} \right) \right) = 0. \quad (18)$$

Note that

$$\sum_{i=1}^{N_U} \sum_{k=1}^{N_V} m_{ik} u_i^T u_i - \bar{u}^T \bar{u} = \sum_{j=1}^{N} \sigma_{u_j}^2,$$

where

$$\sigma_{u_j}^2 = \sum_{i=1}^{N_U} \sum_{k=1}^{N_V} m_{ik} (u_j^i)^2 - \left( \sum_{i=1}^{N_U} \sum_{k=1}^{N_V} m_{ik} u_j^i \right)^2.$$ 

In other words, $\sigma_{u_j}^2$ is the variance of the feature $u_j$. Solving Eq. (18) for $s$ yields

$$\hat{s} = \frac{\text{Tr}\left( \left( \sqrt{ZZ^T} \right)_{PD} \right)}{\sum_{j=1}^{N} \sigma_{u_j}^2}. \quad (20)$$

Since both the numerator and denominator of Eq. (20) are always positive, it is always guaranteed that $\hat{s} > 0$. This is reasonable because a negative scale defies physical interpretation.

Hence, the optimal alignment parameters are

$$\hat{L} = \left( \sqrt{ZZ^T} \right)^{-1}_{PD} Z,$$

$$\hat{s} = \frac{\text{Tr}\left( \left( \sqrt{ZZ^T} \right)_{PD} \right)}{\sum_{j=1}^{N} \sigma_{u_j}^2},$$

$$\hat{t} = \bar{u} - \frac{\text{Tr}\left( \left( \sqrt{ZZ^T} \right)_{PD} \right) \left( \sqrt{ZZ^T} \right)^{-1}_{PD} Z \bar{u}}{\sum_{j=1}^{N} \sigma_{u_j}^2}$$
or, equivalently,

\[
\hat{L} = P(\sqrt{D})^{-1}_{PD}P^T Z, \quad \hat{s} = \frac{\text{Tr} \left( (\sqrt{D})_{PD} \right)}{\sum_{j=1}^{N} \sigma^2_{u^j}}, \quad \hat{t} = \overline{v} - \frac{\text{Tr} \left( (\sqrt{D})_{PD} P(\sqrt{D})^{-1}_{PD} P^T Z \, \overline{u} \right)}{\sum_{j=1}^{N} \sigma^2_{u^j}}.
\]

The minimum squared error is found by substituting the optimal alignment parameters back into Eq. (14), which yields

\[
e_{\text{min}} = \sum_{j=1}^{N} \sigma^2_{v^j} - \frac{\left[ \text{Tr} \left( (\sqrt{ZZ^T})_{PD} \right) \right]^2}{\sum_{j=1}^{N} \sigma^2_{u^j}} = \sum_{j=1}^{N} \sigma^2_{v^j} - \frac{\left[ \text{Tr} \left( (\sqrt{D})_{PD} \right) \right]^2}{\sum_{j=1}^{N} \sigma^2_{u^j}}.
\]

(21)

2.3. Uncoupled Weights: An Ill-Posed Problem. We now examine a specific case that makes the minimization problem ill-posed, i.e., no solution exists. Assume that \( m_{ik} \) is separable, i.e., \( m_{ik} = \sigma_i \gamma_k \), which means that

\[
\overline{u} = \sum_{i=1}^{N_U} \sigma_i u_i, \quad \overline{v} = \sum_{k=1}^{N_V} \gamma_k v_k.
\]

Consequently, the cross-covariance matrix becomes

\[
Z = \sum_{k=1}^{N_V} \gamma_k v_k \sum_{i=1}^{N_U} \sigma_i u_i^T - \overline{v} \overline{u}^T = 0_{N \times N}.
\]

As a result, infinitely many rotation matrices are admissible. Thus, a unique solution for the minimization problem is guaranteed if and only if the weight term is coupled between the two point-sets. In a special subcase, the weight term is fixed for all possible pairings, i.e., \( m_{ik} = c \) for all \( i \) and \( k \), where \( c \) is a real constant.

2.4. Relation to Labeled Scenario. In the case where the correspondence is \textit{a priori} known, we have

\[
m_{ik} = w_{ik} \gamma_{ik},
\]
where
\[ \gamma_{ik} = \begin{cases} 
0, & \text{points } i \text{ and } k \text{ do not correspond,} \\
1, & \text{points } i \text{ and } k \text{ do correspond.} 
\end{cases} \]

Hence, our cost function reduces to
\[
D(U, V; a, b, s_x, s_y, \theta) = \sum_{n=1}^{N} w_n (u'_n - v_n)^T (u'_n - v_k), 
\tag{22}
\]
where \( N \leq \min\{N_U, N_V\} \) is the number of matching pairs between the two sets and \( w_n \) is the “strength” of matching between the two points forming a pair \( n \). The solution to Eq. (22) is the same as the solution obtained for the unlabeled scenario, except the fact that now the (weighted) averages, variances, and covariances are no longer coupled. Equation (22) is the cost function minimized in [1, 2, 6, 11].

3. Numerical Implementation

We have obtained the closed-form solutions for the optimal alignment parameters in two different cases: the absence of scale and the presence of a uniform scale. Without loss of generality, the presented discussion is focused on the case of uniform scale.

Let \( M \) be the data structure (say, an array) storing all \( N_U N_V \) potential pairings. After alignment, the Euclidean distance between any two (cross) points forming a pair is
\[
\Delta_{ik} = \sqrt{(\hat{s}\hat{L}u_i + \hat{t} - v_k)^T(\hat{s}\hat{L}u_i + \hat{t} - v_k)}.
\]

If a pair constitutes a genuine match, then ideally \( \Delta_{ik} \) is small, and it would be desirable to keep it. However, if the pair is spurious, then \( \Delta_{ik} \) would be large and it is desirable to discard it. To do this, it is necessary to compare \( \Delta_{ik} \) for each pair with some threshold \( T \). If, for a given pair,
\[
\Delta_{ik} > T,
\]
then the pair is an outlier, and we remove it from the array \( M \). Otherwise, we recompute its weight as follows:
\[
m_{ik} = 1 - \frac{\Delta_{ik}}{T}. \tag{23}
\]

Upon iterating across every pair in \( M \), we count the number of pairs left in the array. If no pairs have been removed, this indicates that the threshold is too large and, hence, we repeat Step 2 by using the threshold \( T := T - \epsilon \), where \( 0 < \epsilon < T \).

If some pairs have been removed, then it is necessary to check whether the convergence criterion is met. We define convergence as follows: the number of pairs should not exceed the maximum number of allowable one-to-one matches, i.e.,
\[
\text{length}(M) < \min(N_U, N_V).
\]
If this happens to be the case, then we are done. The remaining pairs in \( M \) form the optimal matching pairs. Otherwise, we repeat Stage II by using the updated \( M \) and new weights.
Note that the algorithm contains two hyperparameters: the max threshold $T$ and the threshold spacing $\varepsilon$. They should be appropriately tuned with respect to the data.

We now summarize the proposed algorithm in the form of a pseudocode:

```plaintext
1: while number of pairs > min($N_U, N_V$) and $T > 0$ do
2:   Align the query object to the template
3:   for each pair $M_j$ in the array $\mathbf{M}$ do
4:     compute the weighted sum $\Delta_j$ of radial displacements
5:     if $\Delta_j > T$, then
6:       remove pair $M_j$ from the array $\mathbf{M}$
7:     else
8:       compute the new weight of $M_j$
9:     end if
10:  end for
11:  if no pairs are removed, then
12:    $T := T - \varepsilon$
13:  end if
14: end while
```

### 3.1. Computational Complexity.

Denote by $I(T, \varepsilon)$ the number of iterations until convergence is reached; it is a function of the two hyperparameters. The computational complexity of the proposed algorithm is then $O(IN_U N_V)$. In the best-case scenario, the algorithm achieves convergence after a single iteration, i.e., $I = 1$.

We now consider a scenario in which no pairs are removed during each iteration of alignment. This is the worst-case scenario from the practical standpoint but not necessarily according to the algorithmic perspective (i.e., computational complexity). In this scenario, the total number of attempted alignments is $I(T, \varepsilon) = \left\lceil \frac{T}{\varepsilon} \right\rceil$ and, hence, its complexity is $O \left( \frac{T}{\varepsilon} N_U N_V \right)$. Note, the computational complexity worse than the indicated complexity could be attained; for example, the case where the number of iterations until convergence is reached is greater than $\left\lceil \frac{T}{\varepsilon} \right\rceil$ is completely plausible because any given threshold can be used more than once.

### 3.2. Similarity Score.

Based on the set of (optimal) matching point pairs outputted by the proposed algorithm, a similarity score measuring the strength of matching between two point-sets can be computed from the minimum squared error generated by the optimal alignment parameters; recall that, for the case of uniform scale, it is

$$e_{\text{min}} = \sum_{j=1}^{N} \sigma_{vj}^2 - \left[ \text{Tr} \left( \mathbf{P} (\sqrt{\mathbf{D}}) \mathbf{P}_D \mathbf{P}_D^T \right) \right]^2 \sum_{j=1}^{N} \sigma_{uj}^2.$$ 

Ideally, it should be small for genuine matches and large for dissimilar point-sets.

### 4. Discussion

We propose an unsupervised iterative $N$-dimensional point-set registration algorithm for the unlabeled data (i.e., the correspondence between points is unknown) based on linear least squares. The algorithm considers all possible point pairings and iteratively aligns two sets until the number of point pairs does not exceed the maximum number of allowable one-to-one pairings.
Note that the output of the algorithm, i.e., the optimal matching point pairs, may not necessarily be injective. In fact, a point in $U$ may match more than one point in $V$ or more than one point in $U$ may match the same point in $V$. Situations of this kind frequently arise in fingerprint matching in which, due to the presence of image noise and/or faulty image processing, a minutia in the query image may genuinely correspond to more than one minutia in the reference image.

The proposed algorithm can be utilized in a wide variety of matching problems: from 3D computer vision and graphics to face and fingerprint recognition. For example, assuming that $U$ and $V$ describe two different sets of individuals, we may be interested in identifying the pairs of individuals that are most likely to become friends. In another example, $U$ describes a set of individuals and $V$ describes a set of companies, and we seek a company for which a given individual is the best choice (match). In these two examples, injectivity may not be a realistic assumption; thus, there may be more than one company for which a given individual would fit well, or there may be one company for which more than one individual would be a great fit.

REFERENCES

1. K. S. Arun, T. S. Huang, and S. D. Blostein, “Least-squares fitting of two 3-D point sets,” *IEEE Trans. Pattern Anal. Machine Intell.*, 9, 698–700 (1987).
2. S. Chang, F. Cheng, W. Hsu, and G. Wu, “Fast algorithm for point pattern matching: Invariant to translations, rotations and scale changes,” *Pattern Recognit.*, 30, 311–320 (1997).
3. S. Gold, A. Rangarajan, C. P. Lu, S. Pappu, and E. Mjolsness, “New algorithms for 2d and 3d point matching: pose estimation and correspondence,” *Pattern Recognit.*, 31, 1019–1031 (1998).
4. A. P. Hosseinbor, R. Zhdanov, and A. Ushveridze, “An unsupervised 2d point-set registration algorithm for unlabeled feature points: Application to fingerprint matching,” *Pattern Recognit. Lett.*, 100, 137–143 (2017).
5. B. Jian and B. C. Vemuri, “Robust point set registration using Gaussian mixture models,” *IEEE PAMI*, 33, 1633–1645 (2010).
6. W. Kabsch, “A solution for the best rotation to relate two sets of vectors,” *Acta Crystallogr.*, 32, 922–923 (1976).
7. S. Lan, Z. Guo, and J. You, “A non-rigid registration method with application to distorted fingerprint matching,” *Pattern Recognit.*, 95, 48–57 (2019).
8. A. Rangarajan, H. Chui, and F. L. Bookstein, “The softassign procrustes matching algorithm,” *Inf. Process. Med. Imag.*, 29–42 (1997).
9. A. Rangarajan, H. Chui, E. Mjolsness, S. Pappu, L. Davachi, P. S. Goldman-Rakic, and J. S. Duncan, “A robust point matching algorithm for autoradiograph alignment,” *Med. Image Anal.*, 1, 379–398 (1997).
10. Y. Tsin and T. Kanade, “A correlation-based approach to robust point set registration,” *Europ. Conf. Comput. Vis.*, 558–569 (2004).
11. S. Umeyama, “Least-squares estimation of transformation parameters between two point sets,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 13, 376–380 (1991).