Analysis of Alignment Algorithms with Mixed Dimensions for Dimensionality Reduction

Qiang Ye, Weifeng Zhi∗,†

Department of Mathematics, University of Kentucky, Lexington, KY 40506, USA.

SUMMARY

We consider an alignment algorithm for reconstructing global coordinates of a given data set from coordinates constructed for data points in small local neighborhoods through computing a spectral subspace of an alignment matrix. We show that, under certain conditions, the null space of the alignment matrix recovers global coordinates even when local point sets have different dimensions. This result generalizes a previous analysis to allow alignment of local coordinates of mixed dimensions. We also extend this result to the setting of a semi-supervised learning problem and we present several examples to illustrate our results.

1. INTRODUCTION

Manifold-based nonlinear dimensionality reduction has attracted significant interests in recent years. Mathematically, it can be described as follows. Consider a d-dimensional parameterized manifold \( M \) embedded in \( \mathbb{R}^m \) \( (d < m) \) defined by a nonlinear mapping, \( f : C \subset \mathbb{R}^d \rightarrow \mathbb{R}^m \), where \( C \) is a compact and connected subset of \( \mathbb{R}^d \). Here \( \mathbb{R}^m \) represents the high-dimensional data space and \( \mathbb{R}^d \) represents the low-dimensional parameter space. Given a set of data points \( x_1, \ldots, x_N \in \mathbb{R}^m \) with

\[
x_i = f(\tau_i), \quad i = 1, \ldots, N,
\]

where \( \tau_i \in C \), the problem of dimensionality reduction is to recover low dimensional coordinates (parametrization) \( \tau_i \)'s from the \( x_i \)'s. For the theoretical purpose, we consider noise-free data (1) and we follow Donoho and Grimes [4] to assume that \( f \) is a local isometry.

Traditionally, linear dimensionality reduction methods such as Principal Component Analysis (PCA) have been considered, where the map \( f \) is assumed to be linear. In recent years, nonlinear dimensionality reduction methods have attracted more attentions. Since the publications of LLE [9] and Isomap [12], several competitive algorithms have been proposed, which include Laplacian Eigenmaps [1], Hessian Eigenmaps [4], and LTSA (Local Tangent Space Alignment) [17] among many others; see [10] for a thorough review. One idea underlying several of these methods is to reconstruct global coordinates \( \tau_i \) from their local relations as defined by data points in small neighborhoods. For example, the LTSA method [17] partitions the data points into subsets consisting of points in small local neighborhoods and, for points in each local neighborhood, it constructs
approximate local coordinates by the PCA method or a projection on the local tangent space. It then constructs global coordinates from the local coordinates using an alignment algorithm. The idea of global alignment was also discussed in [11].

A crucial step in the LTSA method is the alignment process of reconstructing global coordinates from local ones, which is carried out by computing a spectral subspace of a matrix, called an alignment matrix. It has been proved in [14] under a condition called fully overlap on local neighborhoods that the null space of the alignment matrix consists of \( e \) (the vector of all ones) and \( d \) vectors with their components defined by the global coordinates, provided that the local coordinates are constructed correctly. Thus, the global coordinates can be obtained up to an affine transformation from computing the spectral subspace of the alignment matrix. This provides a theoretical basis for the alignment algorithm.

One common assumption for methods such as LLE, Isomap and Hessian Eigenmaps that are based on local relations is that local neighborhoods all have the same dimension \( d \). (Here, we say a set of data points \( (1) \) is of dimension \( p \) if the corresponding set of coordinates \( \tau_i \) spans, after being centered, a \( p \)-dimensional space.) For example, the fully overlapped condition for the LTSA method [14] necessarily requires this. There are many situations, however, where such an assumption may not hold. For example, the data points may lie on several manifolds of different dimensions or they may be sampled from a \( d \)-dimensional manifold with lower dimensional branches. Then the ability of dimensionality reduction algorithms to detect and work with change of dimension in the data set is very important.

In this paper, we first consider the alignment algorithm of the LTSA method for reconstructing global coordinates when local neighborhoods have different dimensions. We shall show that, under a generalized fully overlap condition, the null space of the alignment matrix still recover the global coordinate vectors. Our result generalizes the analysis of [14] to allow existence of neighborhoods of mixed dimensions. We shall also extend this result to the setting of a semi-supervised learning problem [5] where one wishes to find full association of two data sets that are partially associated through an alignment matrix. Numerical examples will be presented to illustrate our results.

The paper is organized as follows. We first review the LTSA method and present related notation in Section 2. We present an analysis of the alignment algorithm for local neighborhoods of different dimensions in Section 3. We discuss a semi-supervised learning problem with an analysis in Section 4. We finally present several examples to demonstrate effectiveness of these methods in section 5 and conclude with some remarks in section 6.

**NOTATION.** We use \( e \) to denote a column vector of all ones of appropriate dimension to be determined from the context. For a matrix \( M \), \( \text{null}(M) \) denotes the null space of \( M \), \( \text{span}(M) \) denotes the subspace spanned by all the columns of \( M \), and \( M^\dagger \) denotes the pseudo inverse of \( M \).

### 2. ALIGNMENT ALGORITHM

Consider the data set \( (1) \). Let \( X = \{x_1, \ldots, x_N\} \) and let \( \{X_i, i = 1, \ldots, s\} \) be a collection of subsets of \( X \) with \( X_i = \{x_{i_1}, \ldots, x_{i_{k_i}}\} \) \( (i_1 < i_2 < \cdots < i_{k_i}) \). Assume that \( \cup_i X_i = X \), in which case we say \( \{X_i, i = 1, \ldots, s\} \) is a covering of \( X \). In the context of LTSA, each \( X_i \) is a small local neighborhood so that a coordinate system on the local tangent space can be approximately constructed. In general, we assume that \( X_i \) is any subset such that an isometric coordinate \( \{s_{1}^{(i)}, \ldots, s_{k_i}^{(i)}\} \subset \mathbb{R}^d \) can be constructed, i.e. \( \|s_{p}^{(i)} - s_{q}^{(i)}\|_2 = d_M(x_{i_p}, x_{i_q}) \) (for any \( 1 \leq p, q \leq k_i \)) where \( d_M(\cdot, \cdot) \) is the geodesic distance along \( \mathcal{M} \). In practice, only an approximate isometric coordinate can be computed.

It has been shown [16, 14] that the global coordinates \( \tau_i \)'s can be constructed from the local coordinates through an alignment process as follows. Let

\[
S_i = \{s_{1}^{(i)}, \ldots, s_{k_i}^{(i)}\}, \quad S_i = [s_{1}^{(i)}, \ldots, s_{k_i}^{(i)}]
\]
and define $Q_i$ to be the orthogonal projection onto the orthogonal complement of $\text{span}\{e, S_i^T\}$ in $\mathbb{R}^{k_i}$. Let
\[ E_i = \begin{bmatrix} e_{i1}, \ldots, e_{ik_i} \end{bmatrix} \in \mathbb{R}^{N \times k_i}, \]
where $e_i \in \mathbb{R}^N$ is the $i$-th column of $I_N$ (the $N \times N$ identity matrix). ($E_i$ is called a selection matrix.) Let
\[ \Psi_i = E_i Q_i E_i^T \quad \text{and} \quad \Psi = \sum_{i=1}^s \Psi_i. \]

Note that $\Psi_i$ is the embedding of $Q_i$ into an $N \times N$ matrix such that the $i$th element of $\Psi_i$ is the $(p, q)$th element of $Q_i$. $\Psi$ is called the alignment matrix for $\{S_i\}$.

Under a condition called fully overlap for the covering $\{X_i\}$, it is shown in Theorem 2.7 of [14] that
\[ \text{null}(\Psi) = \text{span}\{[e, T]^T\} \]
where $T = [\tau_1, \ldots, \tau_N]$. Hence, the global coordinates $\tau_i$'s can be obtained from computing $\text{null}(\Psi)$, up to an orthogonal transformation (a rigid motion). For the ease of references, we state the LTSA algorithm as follows. See [14] for the post-processing step (step 5 of the algorithm).

**Algorithm 2.1**

**Local Tangent Space Alignment (LTSA) Algorithm**

1. Construct a fully overlapped covering $\{X_i, i = 1, \ldots, s\}$ with $X_i = \{x_{i1}, \ldots, x_{ik_i}\}$;
2. For each $X_i$, construct its local coordinates $s_1^{(i)}, \ldots, s_{k_i}^{(i)}$;
3. Construct $\Psi$ from $S_i = [s_1^{(i)}, \ldots, s_{k_i}^{(i)}]$ as in (4);
4. Compute $[e/\sqrt{N}, Q^T]$ as an orthonormal basis of the spectral subspace of $\Psi$ corresponding to the $(d + 1)$ smallest eigenvalues, where $Q^T \in \mathbb{R}^{N \times d}$;
5. Recover $T$ as $T = WQ$, where $W = [S_1, \ldots, S_s][Q_1, \ldots, Q_s]$, and $Q_i = Q E_i$.

The fully overlapped condition guarantees sufficient intersection (overlap) among $X_i$’s to allow alignments. In the case of two subsets $X_1$ and $X_2$, it requires that the intersection $X_1 \cap X_2$ is of dimension $d$ (see Definition 3.1 below or [14] for details). This immediately requires that all subsets $X_i$ have the same dimension $d$. However, the structure of the data set $X$ may contain lower or nearly lower dimensional branches. In the next section, we consider this situation.

**Remark 2.1**

We briefly comment on computational complexity of Algorithm 2.1. In the implementation of LTSA [17], a local neighborhood $X_i$ is constructed by finding $k_i = k$ nearest points of each point $x_i$ (for $i = 1, \ldots, N$). Noting that $\Psi$ is a sparse matrix, its spectral subspace corresponding to the $(d + 1)$ smallest eigenvalues can be efficiently computed by the Lanczos method [3, p.366]. The operation counts are $O(mN^2)$ for computing pairwise distances between the $N$ data points, $O(mk^2N)$ for constructing local coordinates, and $O(pkN)$ for finding the spectral subspace of $\Psi$ by the Lanczos method, where $p$ is the number of iterations used in the Lanczos method and is usually much less than $N$. Thus, the cost is primarily dominated by computing the pairwise distances.

### 3. NULL SPACE OF ALIGNMENT MATRIX

In this section, we generalize the analysis of [14] to allow local neighborhoods of mixed dimensions. Specifically, we show that the alignment algorithm still works as long as a generalized fully overlapped condition holds. First we give a precise definition of the dimension of a data set or its coordinate set.

**Definition 3.1**

A data set $X_0 = \{x_1, \ldots, x_k\}$ and the corresponding coordinate set $T_0 = \{\tau_1, \ldots, \tau_k\}$ are said to be of dimension $p$ if
\[ \text{rank}[\tau_1 - \bar{\tau}, \tau_2 - \bar{\tau}, \ldots, \tau_k - \bar{\tau}] = p \]
where $\bar{\tau} = (\Sigma_{j=1}^{k}\tau_j)/k$. We write $\dim(X_0) = \dim(T_0) = p$.

Clearly, this definition is equivalent to that the centralized point set $\{\tau_1 - \bar{\tau}, \tau_2 - \bar{\tau}, \ldots, \tau_k - \bar{\tau}\}$ spans a $p$ dimensional subspace. The following lemma is already shown in [14].

**Lemma 3.1**
Let $T_0 = \{\tau_1, \ldots, \tau_k\}$ and $T_0 = [\tau_1, \ldots, \tau_k]$. Then, $\dim(T_0) = p$ if and only if

$$\text{rank}(\begin{bmatrix} e, T_0^T \end{bmatrix}) = 1 + p.$$ 

As in [14], our analysis begins with the alignment matrix based on $\tau_i$’s, which is constructed in the same way as in (4), but from $T_i$. Let $T = \{\tau_1, \tau_2, \cdots, \tau_N\} \subset \mathbb{R}^d$ and let $\{T_i, 1 \leq i \leq s\}$ be the collection corresponding to $\{X_i, 1 \leq i \leq s\}$, i.e.,

$T_i = \{\tau_{i_1}, \cdots, \tau_{i_k}\}$, $i_1 < i_2 < \cdots < i_k$. \hspace{1cm} (6)

Set

$$T = [\tau_1, \cdots, \tau_N] \in \mathbb{R}^{d \times N}, \hspace{0.2cm} T_i = [\tau_{i_1}, \cdots, \tau_{i_k}].$$ \hspace{1cm} (7)

Let $P_i$ be the orthogonal projection onto the orthogonal complement of $\text{span}(\begin{bmatrix} e, T_i^T \end{bmatrix})$, i.e.,

$$\text{null}(P_i) = \text{span}(\begin{bmatrix} e, T_i^T \end{bmatrix}).$$

Define

$$\Phi_i = E_i P_i E_i^T \hspace{0.2cm} \text{and} \hspace{0.2cm} \Phi = \sum_{i=1}^{s} \Phi_i.$$ \hspace{1cm} (8)

Then $\Phi$ is called the alignment matrix for the collection $\{T_i\}$. Note that $\Phi$ is defined from the original coordinates $\tau_i$’s while $\Psi$ is defined from the local coordinates. However, if $S_i$ is isometric to $X_i$ (and hence to $T_i$), then it can be shown that $S_i$ is an affine transformation of $T_i$, and hence $\Psi_i = \Phi_i$, which leads to $\Psi = \Phi$, see [14].

First, we extend the original definition of fully overlap to sets with different dimensions.

**Definition 3.2**
Let $T_1$ and $T_2$ be two subsets of $T \subset \mathbb{R}^d$. We say $T_1$ and $T_2$ are fully overlapped if

$$\min\{\dim(T_1), \dim(T_2)\} = \dim(T_1 \cap T_2).$$ \hspace{1cm} (9)

This definition allows both $T_1$ and $T_2$ to have dimension less than $d$ and is significantly more general than the original definition [14]. Next, we extend the definition to several subsets as in [14].

**Definition 3.3**
This definition is recursive. Let $T_i, 1 \leq i \leq s$, be $s$ subsets of $\mathbb{R}^d$. The collection $\{T_i, 1 \leq i \leq s\}$ is fully overlapped if it can be partitioned into two nonempty disjoint collections, say, $\{T_{i, i = 1, \ldots, p}\}$ and $\{T_{i, i = p + 1, \ldots, s}\}$ each of which is a fully overlapped collection, and if the union sets of the two collections $T_1 := \cup_{i=1}^{p} T_i$ and $T_2 := \cup_{i=p+1}^{s} T_i$ are fully overlapped.

**Definition 3.4**
The collection $\{T_i, 1 \leq i \leq s\}$ is a covering of $T$ if $\cup_{i=1}^{s} T_i = T$, and a fully overlapped covering if the collection is a covering and is fully overlapped.

We now show that this fully overlapped condition is sufficient to guarantee reconstruction of $T$ from $\Phi$ or $\Psi$. First, the following is a lemma from [14].

**Lemma 3.2**
Let $\{T_i, 1 \leq i \leq s\}$ be a covering of $T$, and let $\Phi_i$ and $\Phi$ be defined as in (8). Then

$$\text{null}(\Phi_i) = \{x | E_i^T x \in \text{span}(\begin{bmatrix} e, T_i^T \end{bmatrix})\},$$

$$\text{null}(\Phi) = \cap_{i=1}^{s} \text{null}(\Phi_i).$$
We now present the first theorem concerning the null space of $\Phi$ in the case of two subsets.

**Lemma 3.3**

Let $\{T_1, T_2\}$ be a fully overlapped covering of $T$ and let $\Phi_i$ and $\Phi$ be defined as in (8). We have $\text{null}(\Phi) = \text{span}\{ [e, T_i^T]\}$.

**Proof:** Without loss of generality, we assume that $d_2 := \dim(T_2) < d_1 := \dim(T_1)$. Then $ank([e, T_2^T]) = d_2 + 1$. There is a nonsingular matrix $U$ such that $[e, T_2^T]U = [e, \tilde{T}_2^T]$ with the last $d - d_2$ columns of $\tilde{T}_2^T$ being all zero. Let $[e, T_i^T]U = [e, \tilde{T}_i^T]$. Suppose there are $k$ vectors in $T_1 \cap T_2$. Without loss of generality, we assume that the last $k$ columns of $T_1$ and the first $k$ columns of $T_2$ are the vectors in $T_1 \cap T_2$. Then we write

$$
\tilde{T}_1 = d_2 \begin{pmatrix}
\tilde{T}_1^{(1)} & \tilde{T}_1^{(2)} \\
\tilde{T}_1^{(2)} & 0
\end{pmatrix} ; \tilde{T}_2 = d_2 \begin{pmatrix}
\tilde{T}_2^{(1)} & \tilde{T}_2^{(2)} \\
0 & 0
\end{pmatrix}
$$

where $\tilde{T}_{12} = \tilde{T}_{21}$. Since $\dim(T_1 \cap T_2) = d_2$, we have

$$
\rank([e, \tilde{T}_2^T]) = d_2 + 1. \tag{10}
$$

Next, let $Q$ be such that its columns form a basis of $\text{null}(\Phi)$. We have $\text{span}(Q) \subset \{ x \mid E_i^T x \in \text{span}([e, \tilde{T}_i^T]) \}$ for each $i$. Then we can find a matrix $W_i \in R^{(d+1) \times m}$, where $m = \dim(\text{null}(\Phi))$, such that $E_i^TQ = [e, \tilde{T}_i^T]W_i$. Let

$$
W_i = d_2+1 \begin{pmatrix}
W_i^{(1)} \\
W_i^{(2)}
\end{pmatrix}.
$$

Comparing the common rows of $E_1^TQ$ and $E_2^TQ$, we have

$$
[e, \begin{pmatrix}
\tilde{T}_1^{(1)} & 0 \\
\tilde{T}_1^{(2)} & 0
\end{pmatrix}]W_1 = [e, \begin{pmatrix}
\tilde{T}_2^{(1)} & 0 \\
\tilde{T}_2^{(2)} & 0
\end{pmatrix}]W_2.
$$

From the first $d_2 + 1$ columns of last equation, we obtain that

$$
[e, \tilde{T}_{12}^T]W_1^{(1)} = [e, \tilde{T}_{21}^T]W_2^{(1)}.
$$

By (10), we have that $[e, \tilde{T}_{12}^T]$ has full column rank. From

$$
[e, \tilde{T}_{12}^T](W_1^{(1)} - W_2^{(1)}) = 0,
$$

it follows that

$$
W_1^{(1)} - W_2^{(1)} = 0.
$$

Noting that

$$
[e, \tilde{T}_2^T]W_2 = \begin{pmatrix}
[e, \begin{pmatrix}
\tilde{T}_{21}^T & 0 \\
\tilde{T}_{22}^T & 0
\end{pmatrix}] \begin{pmatrix}
W_1^{(1)} \\
W_1^{(2)}
\end{pmatrix}
\end{pmatrix}
= \begin{pmatrix}
[e, \begin{pmatrix}
\tilde{T}_{21}^T & 0 \\
\tilde{T}_{22}^T & 0
\end{pmatrix}] \begin{pmatrix}
W_1^{(1)} \\
W_1^{(2)}
\end{pmatrix}
\end{pmatrix}
= [e, \tilde{T}_2^T]W_1,
$$

we have

$$
E_i^TQ = [e, T_i^T]UW_1.
$$

So we can write $Q$ as

$$
Q = [e, T^T]UW_1.
$$
Thus
\[
\text{null}(\Phi) = \text{span}\left([e, T^T]\right).
\]

We note that a similar result has been obtained in [15] for the case of mixed dimensional neighborhoods under a slightly more restrictive definition of fully overlap than Definition 3.2, i.e., with \(\max\{\text{dim}(T_1), \text{dim}(T_2)\} = d\) in addition to (9). In [15], spectral bounds of the alignment matrix are obtained that generalize the earlier results of [7]. As a by-product of the spectral characterization, the dimension of the null space is obtained. The next theorem concerns the case of several subsets.

**Theorem 3.1**
Let \(\{T_i, i = 1, \ldots, s\}\) be a fully overlapped covering of \(T\) and let \(\Phi_i\) and \(\Phi\) be defined as in (8). Then \(\text{null}(\Phi) = \text{span}\left([e, T^T]\right).

**Proof:** This is proved by virtually the same induction as in the proof of Theorem 2.6 [14] using Lemma 3.3 and Definition 3.3. We omit the details.

We now present an example to illustrate that the fully overlap condition is necessary.

**Example 3.1**
Consider
\[
T = \left\{\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}\right\}
\]
with two subsets
\[
T_1 = \left\{\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right\} \quad \text{and} \quad T_2 = \left\{\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}\right\}.
\]

Then \(\text{dim}(T_1) = 2, \text{dim}(T_2) = 1\) and \(\text{dim}(T_1 \cap T_2) = \text{dim}([1, 0]^T) = 0\). Let \(P_1\) be the orthogonal projection onto the orthogonal complement of \(\text{span}([e, T_1^T])\), where \(T_1\) is the matrix consisting of the vectors in \(T_1\). Then \(P_2 = 0\). So \(\Phi = E_1P_1E_1^T\), where \(E_1 = [e_1, e_2, e_3, e_4] \in \mathbb{R}^{5 \times 4}\). It can be verified that
\[
e, [1, 2, 2, 1, 0]^T, [1, 1, 0, 0, 0]^T, [0, 0, 0, 0, 1]^T
\]
form a basis for \(\text{null}(\Phi)\). Noticing that \(\text{dim}(\text{span}([e, T^T])) = 3\), \(\text{null}(\Phi)\) is not sufficient to recover \(\text{span}([e, T^T])\).

This fact can be easily understood geometrically. We plot the points of \(T\) on the left panel of Figure 1 with the points on two neighborhoods defined by \(T_1\) and \(T_2\) outlined in the dash-dotted line. The unit square \(T_2\) intersect with the 1-dimensional branch \(T_2\) at only one point. Now, consider another coordinate set consisting of the same unit square \(T_2\) but a 1-dimensional branch connected at the same point but making some angle as shown in the right panel of Figure 1. With the coordinate set in this 1-dimensional branch isometric to \(T_2\), the alignment matrix that the coordinate points in Figure 1 (right) define is identical to the one defined by Figure 1 (left). Thus \(T\) can not be uniquely determined by the alignment matrix. The reason is clear geometrically because \(T_1\) and \(T_2\) only intersect at one point which is not sufficient to fix the other points (0 in this case) in the 1-dimensional branch \(T_2\).

We note that the fully overlapped condition will be necessary not just for the alignment algorithm, but will also be necessary for other methods that are based on constructions of local neighborhoods. Indeed, for the two cases of Example 1 (the left and the right panel of Figure 1), other methods such as LLE [9], Isomap [12], and Hessian Eigenmaps [4] will result in the same matrix for reconstructing the global coordinates and will not be able to distinguish them.

As pointed out earlier, if we assume that we can construct local isometric coordinates \(S_i\) for each local neighborhood \(X_i\), then we have \(\Psi = \Phi\). Then \(T\) can be reconstructed by computing the null space of \(\Psi\). In practice, however, when we have a neighborhood consisting of points lying on a lower dimensional branch, their coordinates are likely computed with large errors in the components that are supposed to be zero (see Example 3.2 below). Amazingly, with a slightly stronger condition,
this does not change the null space of the alignment matrix. Before we present an analysis, we first illustrate with an example.

**Example 3.2**

Let \( a, b, c, d, f, g, u, v, w, x \) be distinct numbers such that \((c-b)(w-u)-(d-b)(v-u) \neq 0\), \(b \neq 0\) and \(g \neq 0\), and let 

\[
T = \{T_1, T_2, T_3, T_4, T_5\},
\]

where

\[
T_1 = \begin{bmatrix} a \\ g \end{bmatrix}, \; T_2 = \begin{bmatrix} b \\ 0 \end{bmatrix}, \; T_3 = \begin{bmatrix} c \\ 0 \end{bmatrix}, \; T_4 = \begin{bmatrix} d \\ 0 \end{bmatrix}, \; T_5 = \begin{bmatrix} f \\ 0 \end{bmatrix}.
\]

Consider the following two subsets

\[
T_1 = \{T_1, T_2, T_3, T_4\}, \; T_2 = \{T_2, T_3, T_4, T_5\}.
\]

Then \(\dim(T_1) = 2\) and \(\dim(T_2) = \dim(T_1 \cap T_2) = 1\). By Theorem 3.1, we can recover \(T\) from \(\Phi\) as constructed from \(T_1\) and \(T_2\).

In practice, we can only compute two coordinate sets \(S_1\) and \(S_2\) that are (approximately) isometric to \(T_1\) and \(T_2\) respectively. Then, large errors could be present in the second components of the vectors in \(T_2\). For example, when computing \(T_2^T\) from local first order approximation [17], they are computed as 2 singular vectors and the second components derived from a singular vector corresponding to a tiny singular value may effectively be random (see [3]). \(T_1\) can be computed accurately, however. To illustrate, we consider the map \(f(\tau) = \begin{bmatrix} \cos(s) \\ t \\ \sin(s) \end{bmatrix}^T\), where \(\tau = [s, t]^T\) and the data set \(X = \{f(\tau_1), \ldots, f(\tau_5)\}\) with \(\tau_i\) generated with the following values

\[
a = 0.00, \; b = 0.01, \; c = 0.02, \; d = 0.03, \; f = 0.04 \text{ and } g = 0.01.
\]

Now, to construct the alignment matrix, we construct \(S_1\) and \(S_2\) from \(X_1 = \{f(\tau_1), \ldots, f(\tau_4)\}\) and \(X_2 = \{f(\tau_2), \ldots, f(\tau_5)\}\) respectively using PCA. The computed \(S_2\) is

\[
S_2 = \begin{pmatrix} 0.6708 & 0.2236 & -0.2236 & -0.6708 \\
-0.5000 & 0.5000 & 0.5000 & -0.5000 \end{pmatrix},
\]

where we note that the second row is obtained from a singular vector corresponding to a singular value that is nearly zero due to roundoff errors as \(\dim(T_2) = 1\) (see [3]). Thus, the second row of \(S_2\) could be completely different from \(T_2\). Note that the first row may appear different as well but it only differs by an affine transformation, i.e.

\[
-0.0224S_2 + \begin{pmatrix} 0.0250 \\ 0 \end{pmatrix} e^T = \begin{pmatrix} 0.0100 & 0.0200 & 0.0300 & 0.0400 \\
0.0112 & -0.0112 & -0.0112 & 0.0112 \end{pmatrix}.
\]
So, it is not realistic to assume that the second component of the coordinates in $T_2$ can be correctly computed in $S_2$. We therefore consider

$$S_1 = \begin{bmatrix} a & b & c & d \\ g & 0 & 0 & 0 \end{bmatrix}; \quad S_2 = \begin{bmatrix} b & c & d & f \\ u & v & w & x \end{bmatrix}$$

where $u, v, w$ and $x$ are arbitrary. Now, constructing $\Psi$ from $S_i$ as in (4). Using Maple, we can compute $\Psi$ and verify that $\text{null}(\Psi) = \text{span}\{[e, T^T]\}$. Hence, even when the second components in $S_2$ are computed incorrectly, original $T$ can still be recovered from $\Psi$!

The phenomenon explained in the example above is true in general as shown in the following theorem.

**Theorem 3.2**

Let $\{T_1, T_2\}$ be a fully overlapped covering of $T \subset \mathbb{R}^d$ with $\dim(T_1) = d_1$ and $\dim(T_2) = \dim(T_1 \cap T_2) = d_0 < d_1$. Assume that the vectors in $T_2$ have vanishing last $d - d_0$ components. Let $S_1 = T_1$ and $S_2$ be a set of vectors that are the same as $T_2$ except at the last $d - d_0$ components, where they are arbitrary, i.e.

$$S_2 = \left\{ \begin{array}{c} \tilde{\tau}_1 \\ \tilde{\rho}_1 \end{array} \right\} : \begin{array}{c} \tilde{\tau}_1 \\ \tilde{\rho}_1 \end{array} \in T_2 \right\}.$$

Let $\Psi_i$ and $\Psi$ be defined from $S_i$ as in (4). If the points of $S_2$ that correspond to $T_1 \cap T_2$ form a $d$-dimensional set, i.e.

$$\dim\left( \left\{ \begin{array}{c} \tilde{\tau}_i \\ \tilde{\rho}_i \end{array} \right\} \subset S_2 : \begin{array}{c} \tilde{\tau}_i \\ \tilde{\rho}_i \end{array} \in T_1 \cap T_2 \right\} \right) = d,$$

then we have $\text{null}(\Psi) = \text{span}\{[e, T^T]\}$.

**Proof:** Without loss of generality, we assume that the last $k$ columns of $T_1$ and the first $k$ columns of $T_2$ are the vectors in $T_1 \cap T_2$. Write

$$T_1 = \begin{array}{c} \tilde{\tau}_1 \\ \tilde{\rho}_1 \end{array} = \begin{array}{c} T^{(1)}_{11} \\ T^{(2)}_{11} \end{array}, \quad T_2 = \begin{array}{c} \tilde{\tau}_2 \\ \tilde{\rho}_2 \end{array} = \begin{array}{c} T^{(1)}_{21} \\ T^{(2)}_{21} \end{array},$$

where $T^{(1)}_{12} = T^{(1)}_{21}$. Let $S_1$ and $S_2$ be the matrices whose columns are the vectors in $S_1$ and $S_2$ respectively, i.e. we write

$$S_1 = T_1 = \begin{array}{c} \tilde{\tau}_1 \\ \tilde{\rho}_1 \end{array} = \begin{array}{c} T^{(1)}_{11} \\ T^{(2)}_{11} \end{array}, \quad S_2 = \begin{array}{c} \tilde{\tau}_2 \\ \tilde{\rho}_2 \end{array} = \begin{array}{c} T^{(1)}_{21} \\ T^{(2)}_{21} \end{array}.$$

Let $Q$ be such that its columns form a basis for $\text{null}(\Psi)$. We have $\text{span}(Q) \subset \{ x | E^T_i x \in \text{span}\{[e, S_i^T]\} \}$. Then we can find a matrix $W_i \in \mathbb{R}^{(d+1) \times m}$, where $m = \dim(\text{null}(\Psi))$, such that $E^T_i Q = [e, S_i^T]W_i$. Let

$$W_i = \begin{array}{c} \tilde{\tau}_i \\ \tilde{\rho}_i \end{array} = \begin{array}{c} W^{(1)}_1 \\ W^{(2)}_1 \end{array}.$$
Noting \( T_{12}^{(1)^T} = T_{21}^{(1)^T} \), we have
\[
[e, T_{21}^{(1)^T}] (W_1^{(1)} - W_2^{(1)}) = T_{21}^{(2)^T} W_2^{(2)}. 
\]

Noticing \( T_1 \cap T_2 \) consists of the columns of \( \begin{pmatrix} T_{12}^{(1)} \\ 0 \end{pmatrix} \), we have that the set
\[
\left\{ \left( \frac{\hat{t}_i}{\hat{p}_i} \right) : \left( \frac{\hat{t}_i}{\hat{p}_i} \right) \in T_1 \cap T_2 \right\}
\]
consists of the columns of the matrix \( \begin{pmatrix} T_{12}^{(1)} \\ T_{21}^{(2)} \end{pmatrix} \). Using (11), we see that \( [e, T_{21}^{(1)^T}, T_{21}^{(2)^T}] \) has full column rank. It follows from
\[
[e, T_{21}^{(1)^T}] (W_1^{(1)} - W_2^{(1)}) - T_{21}^{(2)^T} W_2^{(2)} = 0
\]
that
\[
[e, T_{21}^{(1)^T}] (W_1^{(1)} - W_2^{(1)}) = 0, \quad T_{21}^{(2)^T} W_2^{(2)} = 0.
\]
This further implies that
\[
W_1^{(1)} - W_2^{(1)} = 0, \quad W_2^{(2)} = 0.
\]
Thus \( W_1^{(1)} = W_2^{(1)} \). Then we can write
\[
E_2^T Q = [e, S_2^T] W_2 = [e, T_2^T] W_1.
\]
This together with \( E_1^T Q = [e, T_1^T] W_1 \) implies that
\[
Q = [e, T^T] W_1.
\]
Therefore, \( \text{null}(\Psi) = \text{span}\left( [e, T^T] \right) \).

\[\text{Remark 3.1}\]
For Example 3.2, we see \( \dim(T_2) = 1 < \dim(T_1) = 2 \) and
\[
\left\{ \left( \frac{\hat{t}_i}{\hat{p}_i} \right) : \left( \frac{\hat{t}_i}{\hat{p}_i} \right) \in T_1 \cap T_2 \right\} = \left\{ \left[ \begin{array}{c} b \\ u \end{array} \right], \left[ \begin{array}{c} c \\ v \end{array} \right], \left[ \begin{array}{c} d \\ w \end{array} \right] \right\}.
\]
We notice \( \dim(\left\{ \left[ \begin{array}{c} b \\ u \end{array} \right], \left[ \begin{array}{c} c \\ v \end{array} \right], \left[ \begin{array}{c} d \\ w \end{array} \right] \right\}) = 2 \). Then \( S_1 \) and \( S_2 \) satisfy the condition of the theorem and we have \( \text{null}(\Psi) = \text{span}\left( [e, T^T] \right) \), which verifies the theorem. However, if \( u = v = w = 0 \) and \( x \neq 0 \), then we have
\[
\dim(\left\{ \left( \frac{\hat{t}_i}{\hat{p}_i} \right) : \left( \frac{\hat{t}_i}{\hat{p}_i} \right) \in T_1 \cap T_2 \right\}) = 1.
\]
So the condition (11) is not satisfied. Indeed, \( e, [a, b, c, d, f]^T, [g, 0, 0, 0, 0]^T \) and \( [0, 0, 0, 0, 1]^T \) are in the null space of \( \Psi \). It follows that \( \text{span}\{[e, T^T]\} \) is a proper subspace of \( \text{null}(\Psi) \). Therefore, the condition (11) in the theorem is necessary.

\[\text{Remark 3.2}\]
In the implementations of the LTSA method or other dimensionality reduction algorithms, the intrinsic dimension \( d \) of the manifold is often assumed to be known. When \( d \) is not known, it can be determined for each local neighborhood by inspecting the gap in the singular values when computing local coordinates by the PCA method. However, the dimension so determined may not be consistent across different neighborhoods if two local neighborhoods have or are close to having different dimensions. Our present result shows that the alignment algorithm is not affected if we just use a larger \( d \) in such a situation.
4. SEMI-SUPERVISED ALIGNMENT OF MANIFOLDS

The results in the previous section show that the alignment algorithm is capable of dealing with sections of different dimensions. Other than reconstruction of global coordinates, this result has application in other contexts. Here we consider a problem in semi-supervised learning of manifolds that has been discussed in [5] and [13].

Assume that there are two different data sets that may admit pairwise correspondences, some of which are known. The objective is to generate the full association (correspondence) of the data sets from the partial association of samples. One approach to this problem is to first generate a common low-dimensional embedding for those two data sets. From the common embedding, we can associate samples between the two data sets.

This semi-supervised learning problem has many applications, including the image comparison [5], cross-lingual information retrieval, bioinformatics [13], and speech analysis [8]. For example, in the image comparison, we have several sets of pictures of different objects taken by a camera from various positions and angles and we wish to match images taken from the same or similar positions/angles, provided matching of the samples is available. Example 5.2 in Section 5 illustrates one such problem.

Mathematically, consider two data sets \( \mathbf{X}_1, \mathbf{X}_2 \subset \mathbb{R}^m \) that are sampled from two \( d \)-dimensional manifolds and are represented by vectors in two matrices \( \mathbf{X}_1 \in \mathbb{R}^{m \times k_1} \) and \( \mathbf{X}_2 \in \mathbb{R}^{m \times k_2} \). Let \( \mathbf{X}_1 = [\mathbf{X}'_1, \mathbf{X}''_1] \) and \( \mathbf{X}_2 = [\mathbf{X}'_2, \mathbf{X}''_2] \) and assume that \( \mathbf{X}'_1 \in \mathbb{R}^{m \times k_0} \) and \( \mathbf{X}'_2 \in \mathbb{R}^{m \times k_0} \) are already known to be in pairwise association. We want to determine any possible pairwise associations between \( \mathbf{X}''_1 \) and \( \mathbf{X}''_2 \). This can be done by finding a joint low dimensional parametrization (embedding) for \( \mathbf{X}_1 \) and \( \mathbf{X}_2 \), in which the corresponding vectors in \( \mathbf{X}'_1 \) and \( \mathbf{X}'_2 \) are mapped to the same coordinates.

First, assume that a \( d \)-dimensional parametrization for each of \( \mathbf{X}_i \) has been obtained by, say, the LTSA method, and let \( Z_1 = [Z_1^1, Z_1^2] \in \mathbb{R}^{d \times k_0} \) and \( Z_2 = [Z_2^1, Z_2^2] \in \mathbb{R}^{d \times k_0} \) be the parameterizations, where \( Z_1^1 \in \mathbb{R}^{d \times k_0} \) and \( Z_2^1 \in \mathbb{R}^{d \times k_0} \) are then the parameterizations for \( \mathbf{X}'_1 \) and \( \mathbf{X}'_2 \) respectively. The assumption that a pairwise association between \( \mathbf{X}'_1 \) and \( \mathbf{X}'_2 \) exist can be mathematically expressed as there exists an affine transformation between \( Z_1^1 \) and \( Z_2^1 \), i.e.

\[
Z_2^1 = RZ_1^1 + ce^T
\]

for some nonsingular matrix \( R \) and a vector \( c \). If the pairwise association is isometric, i.e. the pairwise distances between vectors in \( Z_1^1 \) are the same as those in \( Z_2^1 \), then \( R \) is orthogonal.

To find a joint parametrization, we find \( T_1 = [T_{11}^1, T_{11}^2] \) and \( T_2 = [T_{21}^1, T_{21}^2] \) that are respectively affine transformations of \( Z_1 \) and \( Z_2 \) such that \( T_{11}^1 = T_{21}^1 \). Then, \( T = [T_1, T_1^2, T_2^2] \) is a parametrization for \( [\mathbf{X}'_1, \mathbf{X}''_1, \mathbf{X}''_2] \) and defines a joint parametrization for \( \mathbf{X}_1 \) and \( \mathbf{X}_2 \), where \( T_{21}^1 = T_{21}^2 \). Hence any association between vectors in \( \mathbf{X}''_1 \) and \( \mathbf{X}''_2 \) can be derived from that between \( T_{21}^1 \) and \( T_{21}^2 \).

It turns out that a joint parametrization \( T \) can be obtained from an alignment process. Define the alignment matrix

\[
\Psi = E_1 Q_1 E_1^T + E_2 Q_2 E_2^T,
\]

where \( Q_i \) is the orthogonal projection onto the orthogonal complement of \( \text{span}\{e, Z_i^1\} \) and \( E_i \) is the selection matrix as in (3) such that \( T E_i = T_i \) for \( i = 1, 2 \). The following theorem demonstrates that the null space of \( \Psi \) recovers a joint parametrization \( T \). Here, for the ease of notation, we have used the notation \( \text{dim}(Z) \) for a matrix \( Z \) to denote the dimension of the coordinate set consisting of the columns of \( Z \).

**Theorem 4.1**
Let \( Z_1 = [Z_1^1, Z_1^2] \in \mathbb{R}^{d \times k_1} \) and \( Z_2 = [Z_2^1, Z_2^2] \in \mathbb{R}^{d \times k_2} \) with \( Z_1^1 \in \mathbb{R}^{d \times k_0} \) and \( Z_2^1 \in \mathbb{R}^{d \times k_0} \) such that \( Z_2^1 = RZ_1^1 + ce^T \). Suppose \( R \) is a nonsingular matrix and \( c \) is a vector. Let \( \Psi \) be the alignment matrix as defined in (12). If \( \text{dim}(Z_1) \cap \text{dim}(Z_2) = \text{dim}(Z_1^1) \), then we have \( \text{null}(\Psi) = \text{span}\{e, T^1T^2\} \) where \( T = [T_1, T_2^1, T_2^2] \) with \( [T_1^1, T_2^1] = Z_1 \) and \( T_{22}^1 = R^{-1}Z_2^1 - R^{-1}ce^T \). In particular, \( Z_2 = R[T_1, T_2^1] + ce^T \).

**Proof:** Let \( P_i \) be the orthogonal projection onto the orthogonal complement of \( \text{span}\{e, T_i^1\} \). From the definition of \( T \), we have

\[
RT_{22} + ce^T = [RZ_1^1, Z_2^0 - ce^T] + ce^T = [RZ_1^1 + ce^T, Z_2^0] = [Z_1^1, Z_2^0] = Z_2.
\]
It follows that $\text{span}\{[e, T_2^T]\} = \text{span}\{[e, Z_2^T]\}$ and hence $P_2 = Q_2$. Since $T_1 = Z_1$, we also have $P_1 = Q_1$. Thus,
\[ \Phi := E_1 P_1 E_1^T + E_2 P_2 E_2^T = E_1 Q_1 E_1^T + E_2 Q_2 E_2^T. \]

By Lemma 3.1, $\dim(Z_2) = \dim(T_2)$. Therefore $\min\{\dim(T_1), \dim(T_2)\} = \dim(T^\ell)$. Noting that $\Phi$ is the alignment matrix for $T_1$ and $T_2$, we have $\text{null}(\Phi) = \text{span}\{[e, T^\ell]\}$ by Theorem 3.1. Now the theorem follows from $\Phi = \Psi$.

From the theorem, we can obtain from $\text{null}(\Psi)$ a matrix $T = [T^\ell, T_1^u, T_2^u]$ or its affine transformation such that $T_1 = [T^\ell, T_1^u] = Z_1$ and $T_2 = [T^\ell, T_2^u] = R^{-1}Z_2 - R^{-1}ce^T$. Thus $T = [T^\ell, T_1^u, T_2^u]$ defines a joint parametrization.

In practice, if we do not have a low dimensional parametrization $Z_i$ available, we can first compute it by the LTSA algorithm, but this can be combined with the alignment process as follows.

We partition $X_1$ and $X_2$ into several small neighborhoods $\{X_i^{(1)}, i = 1, \ldots, s_1\}$ and $\{X_i^{(2)}, i = 1, \ldots, s_2\}$. For each small neighborhood, we compute a local parametrization and let $S_i^{(j)}$ be the matrix of local coordinate vectors for $X_i^{(j)}$. Then a joint parametrization $T = [T^\ell, T_1^u, T_2^u]$ can be found by aligning all $\{S_i^{(1)}, 1 \leq i \leq s_1\}$ and $\{S_i^{(2)}, 1 \leq i \leq s_2\}$ together using the alignment algorithm. Specifically, let $Q_i^{(j)}$ be orthogonal projection onto the orthogonal complement of $\text{span}\{[e, S_i^{(j)}]\}$ and let
\[ \Psi_i^{(j)} = E_i^{(j)} Q_i^{(j)} E_i^{(j)T} \]
where $E_i^{(j)}$ is the selection matrix, such that $TE_i^{(j)} = T_i^{(j)}$ with the columns of $T_i^{(j)}$ corresponding to the vectors in $X_i^{(j)}$. Then, $T$ is computed from the null space (or the $d + 1$ smallest eigenvectors) of
\[ \Psi = \sum_{i=1}^{s_1} \Psi_i^{(1)} + \sum_{i=1}^{s_2} \Psi_i^{(2)}. \]

Since we have shown in Section 3 that the alignment algorithm works for neighborhoods of different dimensions, the above alignment process for semi-supervised learning is applicable to the manifolds or neighborhoods of different dimensions. Furthermore, the idea can be easily generalized to matching of $n > 2$ data sets. We state it as the following algorithm.

Algorithm 4.1
Semi-supervised Alignment Algorithm for $n$ Manifolds
Given $X_j \subset \mathbb{R}^m$, $j = 1, \ldots, n$.
1. Construct a fully overlapped covering $\{X_i^{(j)}, i = 1, \ldots, s_j\}$ for $X_j, j = 1, \ldots, n$;
2. For each $X_i^{(j)}$, construct its local coordinates $S_i^{(j)}$;
3. Construct $\Psi_i^{(j)}$ from $S_i^{(j)}, i = 1, \ldots, s_j$ as in (13);
4. Construct the matrix $\Psi$ as in (14);
5. Compute $[e, T^\ell]$ as an orthogonal basis of the spectral subspace of $\Psi$ corresponding to the smallest $d + 1$ eigenvalues.

Remark 4.1
Several algorithms have been proposed for this semi-supervised learning problem, such as [5] and [13]. However, they do not work well with manifolds of different dimensions, as shown by examples in Section 5. The alignment algorithm that we propose has some other advantages as well. For example, it appears that the method of [13] can only work with two data sets. It is also slightly more computationally intensive since it requires computing SVD for the data points already in pairwise correspondences. We note that the dominant computational cost for Algorithm 4.1 is in computing pairwise distances between points within each data set $X_j$, which uses $O(nm \bar{N}^2)$ operations, where $\bar{N} = \max_j N_j$; see Remark 2.1 for other operational counts.
Figure 2. The generating coordinates of the data set.

Figure 3. The reconstructed coordinates of the data set by Algorithm 2.1 with $k_i = 15$ and $d = 2$.

5. NUMERICAL EXAMPLES

In this section, we present two examples to show the alignment algorithm works well with neighborhoods with different dimensions. The first example is a manifold learning problem. We have a set of face images generated from a 3D face model depending on two parameters. We try to find the low-dimensional parametrization for this image set.

Example 5.1
Consider a data set consisting of $N = 2715$ face images generated based on the 3D face model in [2]. The set contains $64 \times 64$ images of the same face, which are obtained by varying pan and tilt angles for the observer. For 2700 images, they vary from $-30$ to 45 degrees of pan angles and $-10$ to 10 degrees for tilt angles. For 15 images, they vary from $-45$ to $-30$ degrees of pan angles and have 0 degree of tilt angles. We are interested in finding the low-dimensional parametrization for these face images. The original coordinates of all those 2715 pictures are shown in Figure 2, where the x-axis is the pan angle and the y-axis is the tilt angle. From Figure 2, we can see that those 2700 images lie on a manifold with dimensionality $d = 2$, whereas the other 15 images with the same tilt angles lie on a branch with dimensionality $d = 1$.

We implement Algorithm 2.1 with fifteen neighbors of each $x_i$ ($k_i = 15$) and dimension two ($d = 2$) to recover the parameters of the images. We notice that for those 15 images with the same tilt angles, their local coordinates should have intrinsic dimensionality one with this example, but our algorithm will treat it as if it were dimension two, having the second components derived from a singular vector corresponding to a tiny singular value. The reconstructed coordinates of all these 2715 images after Algorithm 2.1 are shown in Figure 3. Though one set of these data points is of intrinsic dimensionality one and the other set of data points is of intrinsic dimensionality two, Algorithm 2.1 recover the parametrization correctly. The lower dimensional branch is clearly identified in Figure 3.
Our second example is a semi-supervised learning problem concerning two sets of face images generated from different face models. We are interested in matching the face images shot from the same tilt and pan angles.

Example 5.2
We have two sets of pictures generated from two different 3D face models in [2]. The pictures of two different persons are taken from different pan and tilt angles. We are interested in matching the images with the same pan and tilt angles from different image sets. This problem can be solved by computing a joint low dimensional parametrization (Algorithm 4.1) that we discussed in the previous section.

The first data set \( X \) consists of 100 pictures coming from face model A and all these pictures have the same tilt angle of 0 degree and pan angles varying from \(-45\) to \(45\) degrees. The second data set \( Y \) contains 2720 pictures generated from face model B. These pictures have pan angles varying from \(-45\) to \(45\) degrees and tilt angles varying from \(-10\) to \(10\) degrees. The goal is to match the images with the same tilt angle and the same pan angle. First, 20 matching pairs of pictures in \( X \) and \( Y \) are manually chosen so that each pair of images are shot from the same tilt angle and pan angle. These 20 pictures are labeled samples. We implement the alignment algorithms in [5], [13] and Algorithm 4. Each of these algorithms requires setting the number of points in neighborhoods. For our test, we have used ten or twelve points (\( k_i = 10 \) or \( k_i = 12 \)) and the dimension is set to 2.

For the data set \( X \), we notice that the intrinsic dimensionality should be one, whereas we treat them as having dimension two which is necessary in order to carry out alignment. In the left three plots of Figure 4, we show the computed joint parametrization of data sets \( X \) and \( Y \) with 10 neighbors (\( k_i = 10 \)) and dimensions two (\( d = 2 \)) by the algorithms in [5] (top), in [13] (center) and Algorithm 4.1 (bottom). The right three plots of Figure 4 show the corresponding results when we use 12 neighbors (\( k_i = 12 \)) and dimensions two (\( d = 2 \)). The red circle line are the joint parametrization of the images in \( X \). The blue points are the joint parametrization of the images in \( Y \). Our semi-supervised alignment algorithm works well with data sets of different dimensions.

Given one unlabeled sample picture \( x_i \) from the data set \( X \) as input, which has a parameter \( z_{x_i} \), from joint parametrization, we find an image \( y_j \in Y \) that associates to \( x_i \) by solving

\[
y_j = \arg \min_j \| z_{x_i} - z_{y_j} \|_2,
\]

where \( z_{y_j} \) is the parameter for \( y_j \). For this test, we take five unlabeled sample pictures from \( X \) data as the input, which are shown in Figure 5(a). The best matching data for \( Y \) found by algorithms in [5], in [13] and Algorithm 4.1 are shown in Figure 5(b), Figure 5(c) and Figure 5(d). There is a clear match in the pan and tilt angles for the pairs by our algorithm, while the other two methods obviously have at least some of the pictures mismatched.

6. CONCLUSION
We have analyzed the null space of the alignment matrix when local neighborhoods have different intrinsic dimensions. Our result demonstrates that the alignment algorithm can still recover the global coordinates in this situation. This is a property not shown for other manifold learning algorithms and would be an advantage of the alignment algorithm. Our examples confirm our theoretical finding.

We have also analyzed the use of alignment algorithm in an application to a semi-supervised learning problem. Our result provides a theoretical basis for this application. Our examples demonstrate that the alignment algorithm compares very favorably with several other methods that have been proposed for semi-supervised learning.

One notable issue that we have not addressed in this paper concerns performance of the algorithms for noisy data. There are some discussions on this issue for the LTSA method in [6]. For the future works, it will be interesting to study the behavior of the LTSA method working with local neighborhoods of mixed dimensions with noises.
Figure 4. **Top:** algorithm in [5]. **Center:** algorithm in [13]. **Bottom:** Algorithm 4.1. **Left:** $k_i = 10$. **Right:** $k_i = 12$.

REFERENCES

Figure 5. (a) The five unlabeled pictures from data set $X$ with different pan angles. (b) The matched pictures by algorithm in [5]. (c) The matched pictures by algorithm in [13]. (d) The matched pictures by Algorithm 4.1.
