Isometric Correction for Manifold Learning

Behrouz Behmardi and Raviv Raich
School of Electrical Engineering and Computer Science
Oregon State University
Corvallis, OR, 97331
{behmardb,raich}@eecs.orst.edu

Abstract

In this paper, we present a method for isometric correction of manifold learning techniques. We first present an isometric nonlinear dimension reduction method. Our proposed method overcomes the issues associated with well-known isometric embedding techniques such as ISOMAP and maximum variance unfolding (MVU), i.e., computational complexity and the geodesic convexity requirement. Based on the proposed algorithm, we derive our isometric correction method. Our approach follows an isometric solution to the problem of local tangent space alignment. We provide a derivation of a fast iterative solution. The performance of our algorithm is illustrated on both synthetic and real datasets compared to other methods.

Recent advances in data acquisition and high rate information sources give rise to high volume and high dimensional data. For such data, dimension reduction provides means of visualization, compression, and feature extraction for clustering or classification. In the last decade, a variety of techniques for manifold learning and non-linear dimension reduction have been developed. ISOMAP (Tenenbaum, Silva, and Langford 2000) estimates a geodesic distance along the manifold and uses the multidimensional scaling method to embed the manifold into a low dimensional Euclidean space. Locally linear embedding (LLE) (Roweis and Saul 2000) was developed based on the linear dependence of tangent vectors on a tangent plane. Maximum variance unfolding (MVU) (Weinberger and Saul 2006a) maximizes the data spread in the embedding space while preserving the local geometry. More about the nonlinear data dimension reduction techniques can be found in (Van Der Maaten, Postma, and Van Den Herik 2007) and (Lee and Verleysen 2007).

With a handful of famous exception such as ISOMAP and MVU, none of the aforementioned techniques and other existing techniques perform an isometric embedding for a given manifold. Most of the algorithms yield a quadratic optimization objective in the form of $(T^TMT)$, where $T$ is the global coordinates in the low dimension and $M$ is a symmetric matrix which includes local geometry information specific to each method. The quadratic optimization problem without any constraints has degenerate solutions, e.g., $T = 0$ or $T = [t_1, t_1, \ldots, t_1]$ (i.e., embed into a single point). To circumvent this problem, a set of constraints is imposed to enforce the output to (i) be centered $(T^T1 = 0)$ and (ii) have a unitary covariance $(TT^T = I)$. This set of constraints violates the isometric assumption as it distorts
the metric along the embedded manifold.

In this paper, a precise approach is presented for the isometric embedding of a manifold. The isometric solution is motivated in several applications where true metric preservation is critical such as sensor network localization (Weinberger and Saul 2006b), and face recognition (Bronstein, Bronstein, and Kimmel 2003). The devised approach extends the framework of LTSA (Zhang and Zha 2004) to account for an isometric embedding. Furthermore, it provides a framework for isometric correction of all previously known spectral decomposition algorithms such as LLE, LE, and HILLE. Compared to ISOMAP and MVU, our method is significantly faster and (as with MVU) it is well-suited for datasets with non-convex regions (e.g., Swiss roll with hole). Unlike the previous methods, we form an unconstrained optimization problem which requires no regularization. The developed algorithm is validated with the experimental results for both synthetic and real datasets.

1 Isometric embedding

We consider the following manifold learning setting. A $d$ dimensional manifold $\mathcal{M}$ is embedded in an $m$ dimensional space with a defined mapping $f : \mathbb{R}^d \rightarrow \mathbb{R}^m$ ($d \leq m$). Suppose we are given a set of data points $x_1, \ldots, x_n$, sampled with noise from the manifold, i.e.,

$$x_i = f(\tau_i) + \epsilon_i, \quad i = 1, 2, \ldots, n,$$

(1)

where $\tau_i$ is the low dimensional representation of $x_i$ and $\epsilon_i$ is the construction error. Manifold learning involves finding either the $\tau_i$’s or the explicit mapping $f(\cdot)$. For a smooth function $f$ the Taylor series expansion is

$$f(\tau) = f(\tau^*) + \frac{df}{d\tau}(\tau - \tau^*) + O(\|\tau - \tau^*\|^2),$$

(2)

where $\frac{df}{d\tau}$ is the Jacobian of $f$ and $O(\|\cdot\|^2)$ denotes the contribution of higher order terms which becomes negligible as $\tau$ approaches $\tau^*$. Eq. (2) implies that every smooth manifold can be constructed locally by its tangent plane. Provided sufficient data samples, each point and its neighbors lie on or close to the tangent plane at that point. To find the coordinates of the tangent plane, we compute the singular value decomposition (SVD) on the local neighborhood. Let $X_i = [x_i^1, \ldots, x_i^k]$, be a matrix containing in order of proximity the $k$ neighboring points of $x_i$, $P_i = [X_i - x_i 1_k]^T$ is all vectors in the neighborhood of data point $i$ which lie on the tangent plane of that point. Calculating the SVD of the matrix $P_i$ yields $P_i = Q_i \Sigma_i V_i^T$ where $Q_i$, and $V_i$ are unitary and $\Sigma_i$ is a diagonal matrix with $d$ non-zero values, assuming that data is sampled from a $d$-dimensional manifold. Therefore, $\Theta_i = Q_i^T P_i$ constitutes the coordinates of the tangent plane at point $x_i$

$$\Theta_i = Q_i^T (X_i - x_i 1_k)^T = [\theta_i^1, \ldots, \theta_i^k],$$

$$\theta_i^j = Q_i^T (x_i^j - x_i),$$

(3)

while $Q_i$ forms the basis for the tangent plane. Every vector in the local neighborhood of point $i$ in low dimension can be written as a linear transformation of the coordinate bases of the tangent plane in high dimension plus a reconstruction noise. We consider a global representation of data points in low dimension as follows:

$$\tau_i^j - \tau_i = L_i \theta_i^j + \epsilon_i^j, \quad j = 1, \ldots, k, i = 1, \ldots, N,$$

(4)

where $L_i$ is a unitary transformation and $\theta_i^j$ is given in (3). In the absence of noise ($\epsilon_i^j = 0$), the dot product between every two vectors in the local neighborhood of point $i$ in low dimension can be obtained from (4) as

$$\langle \tau_i^j, \tau_i^l \rangle = \langle \theta_i^j, \theta_i^l \rangle,$$

(5)

since $L_i^{(N \times N)}$ is unitary. Hence, the approach can lead to an isometric embedding. Based on (4), we express the reconstruction error as

$$E_i = (T_i - \tau_i 1_k^T) - L_i \Theta_i = T_i \tilde{S}_i - L_i \Theta_i,$$

(6)

where $T_i^{(N \times k)} = [\tau_i^1, \ldots, \tau_i^k]$, $\tilde{S}_i^{(N \times k)} = \tilde{S}^i M$, $\tilde{S}_i^{(N \times k+1)} = [\epsilon_i \ 1_k]$, $M^{(k+1 \times k)} = \begin{bmatrix} -1^T & I \end{bmatrix}$, and $S_i^{N \times k}$ is the neighborhood selection matrix where the $(S)_{im}$ is one if $\tau_m$ is in the neighborhood of $\tau$. The global isometric embedding is the optimal solution to the optimization problem defined as

$$\begin{array}{ll}
\min_{T, \{L_1, \ldots, L_N\}} & J(T, L_1, \ldots, L_N) \\
\text{s.t.} & L_i^T L_i = I, \quad i = 1, \ldots, N,
\end{array}$$

(7)

where $J(T, L_1, \ldots, L_N) = \sum_i \|T \tilde{S}_i - L_i \Theta_i\|_2^2$. Expanding $J(T, L_1, \ldots, L_N)$, we have

$$J(T, L_1, \ldots, L_N) = \text{tr} (\mathcal{L} T T^T) - 2 \text{tr} \left( \sum_i L_i^T A_i \right) + \text{tr} \left( \sum_i \Theta_i^T \Theta_i \right),$$

(8)

where $\mathcal{L} = \sum_i \tilde{S}_i \tilde{S}_i^T$, and $A_i = T \tilde{S}_i \Theta_i^T$. The term $\sum_i \Theta_i^T \Theta_i$ is constant w.r.t. $T$ and $L_i$. The key distinction between the proposed criterion and the one in LTSA is the unitary constraint applied to the linear transformation, which leads to an isometric embedding. The value of $L_i$ which minimizes the objective function subject to $L_i^T L_i = I$ is obtained by solving the following optimization problem

$$\begin{array}{ll}
\max_{L_i} & \text{tr} \left( L_i^T A_i \right) \\
\text{s.t.} & L_i^T L_i = I.
\end{array}$$

(9)

The solution to Eq. (9) is $L_i^* = A_i (A_i^T A_i)^{-\frac{1}{2}}$. This can be verified by using the inequality $\text{tr} \left( L_i^T A_i \right) \leq \text{tr} \left( (A_i^T A_i)^{\frac{1}{2}} \right)$, which holds with equality if $L_i = L_i^*$, indicating that $L_i^*$ achieves the maximum of Eq. (9). The inequality can be derived by setting $X = L_i^T A_i$ in $\text{tr} (X) \leq \text{tr} \left( (X^T X)^{\frac{1}{2}} \right)$. The proof is given in Appendix A. Substituting $L_i = A_i (A_i^T A_i)^{-\frac{1}{2}}$ into (8), yields

$$\begin{array}{ll}
\min_{T} & \text{tr} \left( \mathcal{L} T T^T \right) - 2 \sum_i \|T \tilde{S}_i \Theta_i^T\|_2^2.
\end{array}$$

(10)
where \( \|Z\|_* = \text{tr} \left( (Z^T Z)^{\frac{1}{2}} \right) \) is the nuclear (or trace) norm of matrix \( Z \). The proposed criterion in (10) is non-convex and may not have a unique global solution. However, the problem is a special type of the non-convex optimization problem which is called \textit{d.c. optimization problem} where the objective function is the difference of two convex functions (Horst and Hoang 1996). The global optimality condition for this type of the problem has been developed in (Strekalovsky 1998). There are some proposed numerical approaches for finding the global optimum solution of these problems (Enkhbat, Barsbold, and Kamada 2006; Chinchuluun et al. 2005). However, this is an NP hard problem (Pardalos 1993) and none of the numerical approaches is feasible for a large scale problem. To address this issue, we provide a fast iterative algorithm to find the solution using the optimization transfer method. To avoid local minima, we consider multiple initializations. Moreover, we developed an isometric correction variation of our algorithm using the framework of (10), which improves the efficiency of the iterative algorithm in terms of computational complexity, global optimality, and convergence.

2 Isometric correction

In (10), we developed an unconstrained optimization objective. The minimization of the objective gives rise to an isometric embedding for manifold learning. We propose an alternative, which we denote by the term isometric correction. Suppose the basis derived from other non-isometric embedding approaches is \( \{V_1, \ldots, V_d\} \) where \( d \geq d \) is the dimension of the search space. The basis spans the \( \hat{d} \)-dimensional space where the solution for \( T \) resides. Therefore, we can write \( T = GV^T \) where \( G \) is a \( d \times \hat{d} \) linear transformation. Substituting \( T \) back into (10) yields

\[
\min_G \text{tr} \left( G \hat{L} G^T \right) - 2 \sum_i \|GV^T \hat{S}_i \Theta_i^T \|_* = 2 \sum_i \|XK_i\|_*,
\]

where \( \hat{L} = V^T \hat{L} V \). The main idea in (11) is by restricting the search from all the points to only the linear transformation \( G \), the problem in (10) becomes ‘easier’ to solve due to the reduced number of unknowns. Note that even with a large number of points, the dimensions of \( G \) remain fixed. Note that for most other methods the number of unknowns grows linearly with the number of data points. The basis obtained by spectral decomposition can approximate the global solution and hence the search space can be restricted to such basis. Note that the proposed approach is different from the one in \textit{ISDP} (Weinberger, Packer, and Saul 2005) where in \textit{ISDP} we learn some randomly chosen landmarks through the original \textit{SDP} and then a linear transformation is used to construct the data points in low dimension from the learned landmarks. Similarly, this approach differs from conformal mapping proposed in (Sha and Saul 2005), where the correction is angle preserving. Our approach goes beyond a simple rescaling of the basis of other algorithms. In essence the isometric correction searches among the meaningful basis and finds a linear combination of them which provides an isometric embedding. Note that the idea of representing matrix \( T \) as a linear combination of basis derived from spectral decomposition approach \( T = GV^T \) is similar to the approach in (Weinberger et al. 2007) where the authors used this approach to scale up the MVU. In the next section, we derive an iterative algorithm to solve the optimization problems in (10) and (11).

3 Algorithmic solution

In the Sections 1 and 2, we defined two related minimization problems with the general objective function in the form of

\[
J(X) = \text{tr} \left( XMX^T \right) - 2 \sum_i \|XK_i\|_*, \tag{12}
\]

where for the optimization problem in (10), \( M = \hat{L} \) and \( K_i = \hat{S}_i \Theta_i^T \) and in (11) \( M = \hat{L} \) and \( K_i = V^T \hat{S}_i \Theta_i^T \). An optimization transfer method is used to solve (12). Optimization transfer is an iterative algorithm replacing the minimization of a general function \( J(X) \) to that of a surrogate function \( H(X, X^{(n)}) \) which tends to simplify the optimization problem (Lange, Hunter, and Yang 2000). The surrogate function satisfies: 1) \( J(X) \leq H(X, X^{(n)}) \) and 2) \( J(X^{(n)}) = H(X^{(n)}, X^{(n)}) \) and the main optimization problem is replaced with the following iterations

\[
X^{(n+1)} = \arg \min_X H(X, X^{(n)}). \tag{13}
\]

These iterations guarantee convergence to a local minimum. To obtain a surrogate function for \( J(X) \) in (12), we use the following matrix inequality.

\[
\text{tr} \left( (A^T A)^{\frac{1}{2}} \right) \geq \text{tr} \left( (A^T B)(B^T B)^{-\frac{1}{2}} \right) \quad \forall B, \tag{14}
\]

where \( B \) is an arbitrary matrix. Applying the bound in (14) to (12) by setting \( A = XK_i \) and \( B = X^{(n)}K_i \), we have

\[
H(X, X^{(n)}) = \text{tr} \left( XMX^T \right) - 2\text{tr} \left( X^T Q(X^{(n)}) \right), \tag{15}
\]

where \( Q(X) = \sum_i XK_i(K_i^T X^T K_i)^{-\frac{1}{2}} K_i^T \). The RHS of (15) is quadratic and the optimum solution is found by setting its derivative w.r.t. \( X \) to zero:

\[
X^{(n+1)} M - Q(X^{(n)}) = 0. \tag{16}
\]

If matrix \( M \) is non-singular, then \( X^{(n+1)} = Q(X^{(n)})M^{-1} \).

For the \textbf{isometric correction} algorithm, we substitute matrices \( X \) and \( M \) with matrices \( G \) and \( \hat{L} \) respectively. Thus, the iterative algorithm for the problem in (11) is given by

\[
G^{(n+1)} = Q(G^{(n)})\hat{L}^{-1}, \tag{17}
\]

where \( Q(G) = \sum_i GK_i(K_i^T G^T GK_i)^{-\frac{1}{2}} K_i^T \) and \( K_i = V^T \hat{S}_i \Theta_i^T \).

For the \textbf{isometric embedding}, we replace matrices \( X \) and \( M \) with matrices \( T \) and \( \hat{L} \) respectively in (16):

\[
T^{(n+1)} \hat{L} = Q(T^{(n)}). \tag{18}
\]
data points, which are uniformly sampled with a missing

The Swiss roll dataset shown in Fig. 1 consists of 1600
thetic datasets to illustrate the effectiveness of our method.
begin with synthetic datasets on several high dimensional datasets. We begin with syn-

In this part, we evaluate the isometric correction algorithm
approach to solve this problem. The Landweber iteration for
(18). Landweber in (Landweber 1951) proposed an iterative
we consider an iterative approach for finding a solution to
(18). The accelerated Landweber method (Hanke 1991) can
be used to improve the convergence speed:
(19). The accelerated Landweber method (Hanke 1991) can
be used to improve the convergence speed:
(19) is

\[ \hat{T}^{(l+1)} = \hat{T}^{(l)} + \frac{1}{\lambda_{max}(\mathcal{L})} (Q(T^{(n)}) - \hat{T}^{(l)} \mathcal{L}), \]

where \( \lambda_{max}(\mathcal{L}) \) is the largest eigenvalue of the Laplacian
matrix \( \mathcal{L} \). The solution to (18) is \( T^{(n+1)} = T^{(\infty)} \) However,
in practical situations some termination criterion is used for
(19). The accelerated Landweber method (Hanke 1991) can
be used to improve the convergence speed:
\[ \hat{T}^{(l+1)} = a_k \hat{T}^{(l)} - b_k \hat{T}^{(l-1)} + \]
\[ c_k \frac{1}{\lambda_{max}(\mathcal{L})} (Q(T^{(n)}) - \hat{T}^{(l)} \mathcal{L}) \] (20)
where \( a_k = \frac{2k+1}{2k+2} \), \( b_k = \frac{2k}{2k+2} \), and \( c_k = \frac{4k+1}{2k+2} \). The
entire algorithm is summarized in Table 1.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>
| Tangent plane construction | (1) Construct matrix \( X_i = x_i^1, \ldots, x_i^d \) where \( x_i^j \) is the \( j \)th neighbor of point \( i \).
| | (2) Compute \( SVD(P) = S \Sigma V^T \) where \( P = X_i - x_i^1 \) and define \( \Theta_i = \Sigma_i V_i^T \) (coordinates of the tangent plane).
| | (3) Find neighborhood selection matrix \( S_i \) and set \( \tilde{S}_i = [e_i S_i] \left[-T_i^1 \right] \).
| | (4) Compute Laplacian matrix \( L = \sum_i \tilde{S}_i \tilde{S}_i^T \).
| Isometric correction | (1) Find \( V = [V_1, \ldots, V_d] \) from one of the spectral decomposition approaches.
| | (2) Compute \( K_i = V^T \tilde{S}_i \Theta_i^T \) and \( \hat{\mathcal{L}} = V^T \mathcal{L} V \).
| | (3) Initialize an arbitrary matrix \( G^{d \times d} \) (to avoid falling in local minima we initialized matrix \( G \) several times).
| | (4) Update \( G \) using (17) until convergence.
| | (5) \( T = GV^T \).
| Isometric embedding | (1) Initialize \( T \) as the output of isometric correction.
| | (2) Update \( T \) using (18) and (20) until convergence.

where \( Q(T) = \sum_i TK_i (K_i^T T K_i)^{-\frac{1}{2}} K_i^T \) and \( K_i = \tilde{S}_i \Theta_i^T \). Note that \( \mathcal{L} \) is a singular large matrix and hence we consider an iterative approach for finding a solution to
(18). Landweber in (Landweber 1951) proposed an iterative approach to solve this problem. The Landweber iteration for
(18) is

\[ \hat{T}^{(l+1)} = \hat{T}^{(l)} + \frac{1}{\lambda_{max}(\mathcal{L})} (Q(T^{(n)}) - \hat{T}^{(l)} \mathcal{L}), \]

4 Simulations and experimental results

In this part, we evaluate the isometric correction algorithm
on several high dimensional datasets. We begin with syn-
thetic datasets to illustrate the effectiveness of our method.
The Swiss roll dataset shown in Fig. 1 consists of 1600
data points, which are uniformly sampled with a missing
rectangle strip punched out of the center. The Swiss roll
is a 2D submanifold of \( \mathcal{R}^3 \). Fig. 3 shows the result of
implementing the isometric correction method on the basis
obtained from different spectral decomposition algorithms
including LLE, LE, LTSA, and Hlle with \( k = 8 \) neighbors.
Since the difference between isometric correction and
isometric embedding visually is unnoticeable, we omitted the
isometric embedding results from the paper. Note that the
algorithm was also implemented for Modified LLE (MLLE)
but due to space constraints the results were omitted. The
first column depicts the original embedding and columns 2
to 4 present the isometric correction method for different
values of \( d \) where \( d \geq d \) is the number of vectors in the basis.
Note that convergence time increases in \( d \). However, higher
\( d \) produces more accurate results. This can be observed by
comparing \( d = 2 \) and \( d = 6 \) for the isometric correction of
LE.

Figure 1: Swiss roll dataset with \( N = 1600 \) data points
sampled uniformly with a rectangle strip punched out of the
center.

Figure 2: Helix dataset with \( N=1600 \) data points.

The last row shows the effect of applying ISOMAP on
the same dataset. The non-convexity of the dataset causes
Figure 3: Isometric correction for the Swiss roll with hole dataset. We used the bases produced by 4 local embedding methods including LLE, LE, LTSA, and HLLE, and implemented our proposed isometric correction algorithm to convert them to an isometric embedding. The first column shows the original embedding where the columns 2 to 4 show the result of the isometric correction algorithm applied to the basis derived from the indicated techniques.

We applied our algorithm on sets of real images believed to come from a complex manifold with few degrees of freedom. Fig. 5 shows the results of the isometric correction technique on a set of 689 face images of an individual including variation in lighting condition and pose. The grayscale images of the face dataset are 64 × 64 and can be regarded as $D = 4096$ dimensional vectors. We separated four paths along the boundaries of 2D embedded set and indicated the image for each point in the boundary. It is shown that the 2D representation of images captures the variation in lighting condition and pose in a smooth way.

We applied our algorithm on sets of real images believed to come from a complex manifold with few degrees of freedom. Fig. 5 shows the results of the isometric correction technique on a set of 689 face images of an individual including variation in lighting condition and pose. The grayscale images of the face dataset are 64 × 64 and can be regarded as $D = 4096$ dimensional vectors. We separated four paths along the boundaries of 2D embedded set and indicated the image for each point in the boundary. It is shown that the 2D representation of images captures the variation in lighting condition and pose in a smooth way.

Fig. 6 shows the results of our algorithm applied to color images of a three dimensional solid object. The images have 76 × 101 pixels, with three bytes for color depth, giving rise to points of $D = 23028$ dimensions. The isometric correction technique was applied to $N = 100$ images spanning 360 degrees of rotation, with $k = 3$ nearest neighbors. Fig. 7 shows the results of our algorithm applied to $N = 1600$ images of digit number TWO from the MNIST data set. The images have 28 × 28 pixels giving rise to $D = 784$ dimensions. The low dimensional representation captures the variation in size, slant, and line thickness.

The computational complexity of ISOMAP is $O(N^3)$ associated with distance matrix and multi dimensional scaling (MDS) eigenvalue calculation. The memory complexity, by definition the storage requirement to solve a problem, for ISOMAP is $O(N^2)$. Computational complexity of MVU is $O(m^2)$ where $m$ is the number of constraints (Borchers and

---

Original embedding | $\hat{d} = 2$ | $\hat{d} = 4$ | $\hat{d} = 6$
---
LLE | ![LLE](image1) | ![LLE](image2) | ![LLE](image3)
LE | ![LE](image4) | ![LE](image5) | ![LE](image6)
LTSA | ![LTSA](image7) | ![LTSA](image8) | ![LTSA](image9)
HLLE | ![HLLE](image10) | ![HLLE](image11) | ![HLLE](image12)
ISOMAP | ![ISOMAP](image13) | ![ISOMAP](image14) | ![ISOMAP](image15)
Young 2007). The number of constraints in MVU is approximately $Nk$ where $k$ is the number of the neighborhood points (Van Der Maaten, Postma, and Van Den Herik 2007). Therefore, the complexity of MVU is $O((Nk)^3)$. Note that the memory complexity of MVU is $O((Nk)^3)$ which makes it computationally complex. For a given basis, the isometric correction algorithm complexity is of $O(N\tilde{d}^3)$ which is negligible compared to ISOMAP and MVU algorithms since $\tilde{d} \ll N$. However, the entire algorithm complexity is of $O(pN^2)$ due to eigendecomposition of a sparse matrix used to derive the basis for initializing the algorithm where $p$ denotes the ratio of the sparsity. Fig. 8 shows comparison among the run time of isometric embedding algorithms including ISOMAP, MVU, and our approach for different datasets. The run time comparison for our approach includes the entire process (spectral decomposition and isometric correction). As it is indicated, the isometric correction approach is computationally more efficient than ISOMAP and MVU by 2 and 3 orders of magnitude, respectively.

5 Summary
Most of the manifold learning methods reduce to an eigen-decomposition, for which a global solution is available. The
eigendecomposition leads to a non-isometric solution. In this paper, we have developed a new algorithm for isometric embedding of high dimensional datasets. Our approach extends the framework of the LTSA algorithm to account for isometric constraints. Our approach overcomes the shortcomings of previously known isometric embedding algorithms such as ISOMAP and MVU. The isometric embedding enhances the computational savings compared to ISOMAP and MVU. Moreover, our approach (as with MVU) resolves the issue of embedding distortion associated with geodesic convexity of ISOMAP. We introduced the isometric correction problem and provided an efficient iterative algorithm for solving the isometric correction. Since the dimension of the problem decreases by introducing the linear transformation matrix $G$, the problem becomes computationally more efficient. Another aspect that allows the speedup of our method lies in the fast convergence of the algorithm achieved by the accelerated Landweber iteration. In turn, more initializations can be used to increase the probability of finding the global solution to avoid local minima.

\section{Appendix}

Here we provide the proof for the inequality $\text{tr} (X) \leq \text{tr} \left( (X^T X)^{1/2} \right)$. Let $X$ be $n \times n$ matrix with SVD given by $X = USV^T$, where the columns of $U$ and $V$ are orthonormal and $S$ is an $l \times l$ diagonal matrix with all nonnegative elements. Therefore,

\begin{equation}
\text{tr} (X) = \text{tr} (USV^T).
\end{equation}

By circularity of the trace operator we have

\begin{equation}
\text{tr} (X) = \text{tr} (SV^TU).
\end{equation}

Let $s = [s_1, \ldots, s_n]^T$ and $\alpha = [(V^TU)_{11}, \ldots, (V^TU)_{nn}]^T$. We can write $\text{tr} (SV^TU) = s^T \alpha$. Using Hölder inequality,

\begin{equation}
\text{tr} (SV^TU) \leq \|s\|_1 \|\alpha\|_\infty
\end{equation}

Recalling that $V^TV = U^TU = I_l$ and using the Cauchy-Schwarz inequality we have

\begin{equation}
e_i^T V^TUe_i \leq \sqrt{e_i^TV^TVe_i} \sqrt{e_i^TU^TUe_i} = \sqrt{e_i^Te_i} \sqrt{e_i^Te_i} = 1.
\end{equation}

Hence,

\begin{equation}
\text{tr} (X) = \text{tr} (SV^TU) \leq \text{tr} (S).
\end{equation}

Since the nuclear norm of matrix $X$ is

\begin{equation}
\text{tr} \left( (X^T X)^{1/2} \right) = \text{tr} (S),
\end{equation}

we have

\begin{equation}
\text{tr} (X) \leq \text{tr} \left( (X^T X)^{1/2} \right).
\end{equation}

\section{References}


