International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:10, 2008

# Manifold Analysis by Topologically Constrained Isometric Embedding

Guy Rosman, Alexander M. Bronstein, Michael M. Bronstein and Ron Kimmel

***Abstract*— We present a new algorithm for nonlinear dimensionality reduction that consistently uses global information, and that enables understanding the intrinsic geometry of non-convex manifolds. Compared to methods that consider only local information, our method appears to be more robust to noise. Unlike most methods that incorporate global information, the proposed approach automatically handles non-convexity of the data manifold. We demonstrate the performance of our algorithm and compare it to state-of-the-art methods on synthetic as well as real data.**

***Keywords*— Dimensionality reduction, manifold learning, multidimensional scaling, geodesic distance, boundary detection.**

## I. Introduction

*NONLINEAR dimensionality reduction* (NLDR) algorithms explain a given data set of high dimensionality, in terms of a small number of variables or coordinates. Such methods are used in various pattern recognition problems, including pathology tissue analysis [1], motion understanding [2], lip reading [3], speech recognition [4], enhancement of MRI images [5], and face recognition [6].

Most NLDR algorithms map the data to a coordinate system of given dimensionality that represents the given data while minimizing some error measure. Unlike classical dimensionality reduction methods such as principal component analysis (PCA) [7], the map is non-linear.

The data is usually assumed to arise from a manifold $\mathcal{M}$, embedded into a high-dimensional Euclidean space $\mathbb{R}^M$. The manifold $\mathcal{M}$ is assumed to have a low intrinsic dimension $m$ ($m \ll M$), i.e., it has a parametrization in a subset $\mathcal{C}$ of $\mathbb{R}^m$, represented by the smooth bijective map $\varphi : \mathcal{C} \subset \mathbb{R}^m \to \mathcal{M}$. The *geodesic distances* $\delta : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$, defined as the lengths of the shortest paths on $\mathcal{M}$ (called *geodesics*), represent the intrinsic structure of the data. The goal of NLDR is, given $\mathcal{M}$, to recover the parametrization in $\mathbb{R}^m$.

The intrinsic dimension $m$ is usually assumed to be known *a priori*. We denote the data samples by $\mathbf{z}_i$, $i \in I$ a point on $\mathcal{M}$, where $I$ is a set of continuous indices.

In the discrete setting, the data is represented as a graph whose vertices $\mathbf{z}_1, ..., \mathbf{z}_N$ are finite samples of the manifold, and the connectivity matrix $\mathbf{A} = (a_{ij})$, where $a_{ij} = 1$ if $\mathbf{z}_i$ and $\mathbf{z}_j$ are neighbors and zero otherwise. Hereinafter, we write $\mathcal{M}$ referring to both the discrete and the continuous manifold. NLDR algorithms usually approximate *local* (short) distances on the data manifold by the Euclidean distances in the embedding space, $\delta(\mathbf{z}_i, \mathbf{z}_j) = \|\mathbf{z}_i - \mathbf{z}_j\|_2$, for $i, j$ such that $a_{ij} = 1$. The geodesic distances are approximated as graph distances, which can be expressed as a sum of local distances. The NLDR problem can be formulated as finding a set of coordinates $\{\mathbf{x}_1, ..., \mathbf{x}_N\} = \varphi^{-1}(\{\mathbf{z}_1, ..., \mathbf{z}_N\})$ in $\mathbb{R}^m$ that describe the data.

Most NLDR methods minimize criteria that consider the relationship of each point and its nearest neighbors. For example, the *locally linear embedding* (LLE) algorithm [8] attempts to express each point as a linear combination of its neighbors. The deviation of each point from this linear combination is summed over the manifold and used as a penalty function. The coordinates that minimize the penalty are then computed by solving an eigenvalue problem.

The *Laplacian eigenmaps* algorithm [9] uses as intrinsic coordinate functions the minimal eigenfunctions of the Laplace-Beltrami operator. This is done by constructing the Laplacian matrix of the proximity graph, finding its smallest $m$ non-zero eigenvectors and using them as the coordinates of the data points. *Diffusion maps* have been recently proposed as an extension of Laplacian eigenmaps, able to compensate for non-uniform sampling of the manifold [1].

The *Hessian eigenmaps* algorithm [10], computes coordinate functions that minimize the Frobenius norm of the Hessian, summed over the manifold. The algorithm expresses, for each coordinate function, the sum of the quadratic components at each point. The minimization result in an eigenvalue problem, whose minimal vectors provide the desired coordinate vectors, similarly to Laplacian eigenmaps.

The *semidefinite embedding* algorithm [11], takes a different approach, trying to maximize the variance of the data set in its new coordinates, while preserving short distances. This is done by solving a semidefinite programming (SDP) problem, while preservation of local distances imposed as constraints. Solving the resulting SDP problem, however, still involves high computational cost. Attempts to lower the complexity have been made in [12].

Unlike local methods, the *Isomap* algorithm [13], [27] , tries to preserve a *global* invariant – the geodesic distances on the data manifold. While the geodesics may change dramatically even in case of small noise, for well-sampled manifolds, their lengths (i.e., the geodesic distances) hardly change even in the presence of high level of noise. This property may be useful

Manuscript received September 10, 2006; revised October 1, 2006.

G. Rosman is with the department of Computer Science, Technion – Israel Institute of Technology, Haifa 32000, Israel (email: rosman@cs.technion.ac.il).

A. M. Bronstein is with the department of Computer Science, Technion – Israel Institute of Technology, Haifa 32000, Israel (email: bron@cs.technion.ac.il).

M. M. Bronstein is with the department of Computer Science, Technion – Israel Institute of Technology, Haifa 32000, Israel (email: mbron@cs.technion.ac.il).

R. Kimmel is with the department of Computer Science, Technion – Israel Institute of Technology, Haifa 32000, Israel (email: ron@cs.technion.ac.il).

in analysis of noisy data, in which local methods often fail. A multidimensional scaling (MDS) algorithm is used to find a set of coordinates whose Euclidean distances approximate the geodesic distances. The *least squares* MDS (LSMDS) algorithm, for example, minimizes the *stress* [14],

$$\mathbf{X}^* = \underset{\mathbf{X}\in\mathbb{R}^{N\times m}}{\operatorname{argmin}} \sum_{i<j} w_{ij}\left(d_{ij}(\mathbf{X}) - \delta_{ij}\right)^2,$$

Here $\mathbf{X}^* = (x_{ij})$ is an $N \times m$ matrix whose rows are the coordinate vectors in the low-dimensional Euclidean space $\mathbb{R}^m$, $\delta_{ij} = \delta(\mathbf{z}_i, \mathbf{z}_j)$ and $d_{ij}(\mathbf{X}) = \|\mathbf{x}_i - \mathbf{x}_j\|_2$ is the Euclidean distance between points $\mathbf{x}_i$ and $\mathbf{x}_j$ in $\mathbb{R}^m$.

The underlying assumption of Isomap is that $\mathcal{M}$ is *isometric* to $\mathcal{C} \subset \mathbb{R}^m$ with the *induced metric* $d_{\mathcal{C}}$, that is, $\delta(\mathbf{z}_i, \mathbf{z}_j) = d_{\mathbb{R}^m}(\mathbf{x}_i, \mathbf{x}_j)$ for all $i, j = 1, ..., N$. If $\mathcal{C}$ is convex, the restricted metric $d_{\mathbb{R}^m}|_{\mathcal{C}}$ coincides with the induced metric $d_{\mathcal{C}}$ and Isomap succeeds recovering the parametrization of $\mathcal{M}$. Otherwise, $\mathcal{C}$ has no longer Euclidean geometry and MDS cannot be used.

The assumption of convexity of $\mathcal{C}$ appears to be too restrictive, as many data manifolds have complicated topology which violates this assumption. Donoho and Grimes [15] showed examples of data in which is $\mathcal{C}$ is non convex, and pointed out that Isomap fails in such cases. Here, we suggest a solution based on removing pairs of points inconsistent with the convexity assumption. Our approach, hereinafter referred to as the *topologically constrained isometric embedding* (TCIE), allows handling data manifolds of arbitrary topology. An algorithm for detecting and removing the inconsistent distances is described in Section II. Numerical implementation details appear in Section III. Results on synthetic and real-life data are shown in Section IV.

## II. Topologically Constrained Isometric Embedding

1 Compute the $N \times N$ matrix of *geodesic distances* $\mathbf{\Delta} = (\delta_{ij})$.
2 Detect the boundary points $\partial\mathcal{M}$ of the data manifold.
3 Detect a subset of *consistent distances* according to the following criterion,

$$\bar{P}_1 = \{(i,j) : c(\mathbf{z}_i, \mathbf{z}_j) \cap \partial\mathcal{M} = \emptyset\}, \quad (1)$$

or

$$\bar{P}_2 = \{(i,j) : \delta(\mathbf{z}_i, \mathbf{z}_j) \le \delta(\mathbf{z}_j, \partial\mathcal{M}) + \delta(\mathbf{z}_i, \partial\mathcal{M})\}, \quad (2)$$

where $\delta(\mathbf{z}, \partial\mathcal{M}) = \inf_{\mathbf{z}'\in\partial\mathcal{M}} \delta(\mathbf{z}, \mathbf{z}')$ denotes the distance from $\mathbf{z}$ to the boundary.
4 Minimize the *weighted stress*,

$$\mathbf{X}^* = \underset{\mathbf{X}\in\mathbb{R}^{N\times m}}{\operatorname{argmin}} \sum_{i<j} w_{ij}\left(d_{ij}(\mathbf{X}) - \delta_{ij}\right)^2,$$

with $w_{ij} = 1$ for all $(i,j) \in P$ and zero otherwise. The obtained points $\mathbf{x}_1^*, ..., \mathbf{x}_N^*$ are the desired representation of $\mathcal{M}$ in $\mathbb{R}^m$.

Omitting steps 2 and 3 and setting $w_{ij} = 1$, we obtain the Isomap as a particular case of the TCIE algorithm. Bernstein *et al.* [16] proved that the graph distances converge to the true geodesic distances, i.e., that the discretization is consistent.

The Isomap algorithm assumes that the parametrization $\mathcal{C}$ of $\mathcal{M}$ is a convex subset of $\mathbb{R}^m$, and relies on the isometry assumption to find the map from $\mathcal{M}$ to the metric space $(\mathcal{C}, d_{\mathcal{C}})$ by means of MDS (the stress in the solution will be zero). MDS can be used because $d_{\mathcal{C}} = d_{\mathbb{R}^m}|_{\mathcal{C}}$ due to the convexity assumption. In the case when $\mathcal{C}$ is non-convex, this is not necessarily true, as there may exist pairs of points for which $d_{\mathcal{C}} \neq d_{\mathbb{R}^m}|_{\mathcal{C}}$. We call such pairs *inconsistent*. An example of such a pair is shown in Figure 1. We denote the set of all consistent pairs by

$$P = \{(i,j) : d_{\mathcal{C}}(\mathbf{x}_i, \mathbf{x}_j) = d_{\mathbb{R}^m}|_{\mathcal{C}}(\mathbf{x}_i, \mathbf{x}_j)\} \subseteq I \times I.$$

In the TCIE algorithm, steps 2 and 3 are used to find a subset $\bar{P} \subseteq P$ of pairs of points that will be consistently used in the MDS problem, using criteria (1) and (2), first proposed in [17] for matching of partially-missing shapes. In the following propositions justifying the two criteria, we rely upon the following properties of isometries: (i) an isometry is a smooth map, copying boundaries to boundaries and interiors to interiors; and (ii) the distance to boundary is preserved under isometry, i.e., $\delta(\mathbf{z}_i, \partial\mathcal{M}) = d_{\mathcal{C}}(\mathbf{x}_i, \partial\mathcal{C})$. Both propositions tacitly assume the continuous case.

Let $\mathcal{M}$ be a compact manifold with boundary $\partial\mathcal{M}$, isometrically parameterized on $(\mathcal{C} \subset \mathbb{R}^m, d_{\mathcal{C}})$. Then,

*Proposition 1:* $\bar{P}_1 = \{(i,j) : c(\mathbf{z}_i, \mathbf{z}_j) \cap \partial\mathcal{M} = \emptyset\} \subseteq P$.

*Proof:* Let $(i,j) \in \bar{P}_1$. To prove the proposition, it is sufficient to show that the pair of points $(i,j)$ is consistent, i.e., $(i,j) \in P$. Let $c_{\mathcal{M}}(\mathbf{z}_1, \mathbf{z}_2)$ be the geodesic connecting $\mathbf{z}_i$ and $\mathbf{z}_j$ in $\mathcal{M}$, and let $c_{\mathcal{C}}(\mathbf{x}_1, \mathbf{x}_2)$ be its image under $\varphi^{-1}$ in $\mathcal{C}$. Since $c(\mathbf{z}_i, \mathbf{z}_j) \cap \partial\mathcal{M} = \emptyset$ and due to property (i), $c_{\mathcal{C}}(\mathbf{x}_i, \mathbf{x}_j) \subset \mathrm{int}(\mathcal{C})$.

Assume that $(i,j)$ is inconsistent. This implies that $d_{\mathcal{C}}(\mathbf{x}_i, \mathbf{x}_j) > d_{\mathbb{R}^m}(\mathbf{x}_i, \mathbf{x}_j)$, i.e., that the geodesic $c_{\mathcal{C}}(\mathbf{x}_i, \mathbf{x}_j)$ is not a straight line. Therefore, there exits a point $x \in c_{\mathcal{C}}(\mathbf{x}_i, \mathbf{x}_j)$, in whose proximity $c_{\mathcal{C}}(\mathbf{x}_i, \mathbf{x}_j)$ is not a straight line. Since $c_{\mathcal{C}}(\mathbf{x}_i, \mathbf{x}_j) \subset \mathrm{int}(\mathcal{C})$, there exists a ball $B_\epsilon(\mathbf{x})$ with the Euclidean metric $d_{\mathbb{R}^m}$ around $\mathbf{x}$ of radius $\epsilon > 0$. Let us take two points on the segment of the geodesic within the ball, $\mathbf{x}', \mathbf{x}'' \in c_{\mathcal{C}}(\mathbf{x}_i, \mathbf{x}_j) \cap B_\epsilon(\mathbf{x})$. The geodesic $c_{\mathcal{C}}(\mathbf{x}', \mathbf{x}'')$ coincides with the segment of $c_{\mathcal{C}}(\mathbf{x}_i, \mathbf{x}_j)$ between $\mathbf{x}', \mathbf{x}''$. Yet, this segment is not a straight line, therefore we can shorten the geodesic by replacing this segment with $c_{\mathbb{R}^m}(\mathbf{x}', \mathbf{x}'')$, in contradiction to the fact that $c_{\mathcal{C}}(\mathbf{x}_1, \mathbf{x}_2)$ is a geodesic. Therefore, $(i,j) \in P$. ■

*Proposition 2:* $\bar{P}_2 = \{(i,j) : \delta(\mathbf{z}_i, \mathbf{z}_j) \le \delta(\mathbf{z}_j, \partial\mathcal{M}) + \delta(\mathbf{z}_i, \partial\mathcal{M})\} \subseteq P$.

*Proof:* Let $(i,j) \in \bar{P}_2$. We have to show that $(i,j) \in P$. According to properties (i) and (ii) shown above we have

$$\begin{aligned} d_{\mathcal{C}}(\mathbf{x}_i, \mathbf{x}_j) &= \delta(\mathbf{z}_i, \mathbf{z}_j) \\ &\le \delta(\mathbf{z}_j, \partial\mathcal{M}) + \delta(\mathbf{z}_i, \partial\mathcal{M}) \\ &= d_{\mathcal{C}}(\mathbf{x}_i, \partial\mathcal{C}) + d_{\mathcal{C}}(\mathbf{x}_j, \partial\mathcal{C}). \end{aligned} \quad (3)$$

Assume that $(i,j)$ is inconsistent. This implies that the geodesic connecting $\mathbf{x}_i$ and $\mathbf{x}_j$ in $\mathbb{R}^m$ is not entirely contained in $\mathcal{C}$ (we assume without loss of generality that the geodesic

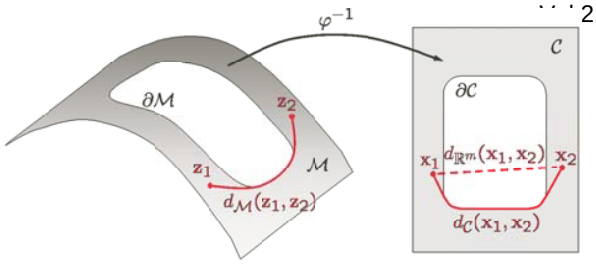


Fig. 1. Example of two inconsistent points $\mathbf{z}_1, \mathbf{z}_2 \in \mathcal{M}$, and the geodesic connecting them. Also shown are the two images of these points under the isometry $\varphi^{-1}$, a geodesic connecting them in $\mathcal{C}$, and the line connecting them in $\mathbb{R}^m$.

originating from $\mathbf{x}_i$ to $\mathbf{x}_j$ intersects the boundary $\partial\mathcal{C}$ at points $\mathbf{x}'$ and $\mathbf{x}''$). Consequently,

$$\begin{aligned} d_{\mathcal{C}}(\mathbf{x}_i, \mathbf{x}_j) &> d_{\mathbb{R}^m|\mathcal{C}}(\mathbf{x}_i, \mathbf{x}_j) \\ &\geq d_{\mathcal{C}}(\mathbf{x}_i, \mathbf{x}') + d_{\mathcal{C}}(\mathbf{x}_j, \mathbf{x}'') \\ &\geq d_{\mathcal{C}}(\mathbf{x}_i, \partial\mathcal{C}) + d_{\mathcal{C}}(\mathbf{x}_j, \partial\mathcal{C}), \end{aligned} \tag{4}$$

which contradicts inequality (3). Therefore, $(i, j) \in P$.

■

We note this proof holds even if $\mathcal{C}$ is a subset of a generic metric space. The metric $d_{\mathcal{C}}$ would have to be replaced with the metric induced from that space, but the Euclidean MDS procedure would not be able to give us the correct mapping. This would require the use of non-Euclidean embedding, e.g. as in [17].

## III. Numerical Solution of the TCIE Problem

### A. Detection of boundary points

Detection of boundary points on discrete manifolds has been studied extensively (see for example [18], [19]). We compared two boundary detection methods, based on studying the properties of the coordinates of nearest neighbors of each point, reconstructed from local distances using classical MDS. The first method assumes the point and its two opposite neighbors are a part of a curve along the boundary. It then tries to find points that are placed outside of this boundary on both sides of it, violating the conjecture. The algorithm goes as follows:

**for** $i = 1, ..., N$ **do**
  Find the set $\mathcal{N}(i)$ of the $K$ nearest neighbors of the point $i$.
  Apply MDS to the $K \times K$ matrix $\mathbf{\Delta}_K = (\delta_{kl \in \mathcal{N}(i)})$ and obtain a set of local coordinates $\mathbf{x}'_1, ..., \mathbf{x}'_K \in \mathbb{R}^m$.
  **for** $j, k \in \mathcal{N}(i)$ *such that* $\frac{\langle \mathbf{x}'_j - \mathbf{x}'_i, \mathbf{x}'_k - \mathbf{x}'_i \rangle}{\|\mathbf{x}'_j - \mathbf{x}'_i\| \cdot \|\mathbf{x}'_k - \mathbf{x}'_i\|} \approx -1$ **do**
    Mark the pair $(j, k)$ as `valid`.
    **if** $|\mathbf{x}' : \frac{\langle \mathbf{x}' - \mathbf{x}'_i, \mathbf{v}_l \rangle}{\|\mathbf{x}' - \mathbf{x}'_i\|} \approx 1| \geq \tau_a |\mathcal{N}(i)|$ *for all* $l = 1, ..., m-1$ **then**
      Label the pair $(j, k)$ as `satisfied`. (here $\mathbf{v}_l$ denotes the $l$th vector of an orthonormal basis of the subspace of $\mathbb{R}^m$ orthogonal to $\mathbf{x}'_j - \mathbf{x}'_k$).
    **end**
  **end**
  **if** *the ratio of* `satisfied` *to* `valid` *pairs is smaller than threshold* $\tau_b$ **then**
    Label point $i$ as `boundary`.
  **end**
**end**

The second method tries to explore the direction of the normal to the boundary. Moving along the normal direction, the density of sampling points should drop to zero. We can check along each direction from the point $i$ to one of its neighbors $j$. Assuming approximately uniform density of the points, one such neighboring point $j$ should produce a vector pointing close to the normal direction. This method is more suitable for manifolds of higher intrinsic dimension.

**for** $i = 1, ..., N$ **do**
  Find the set $\mathcal{N}(i)$ of the $K$ nearest neighbors of the point $i$.
  Apply MDS to the $K \times K$ matrix $\mathbf{\Delta}_K = (\delta_{kl \in \mathcal{N}(i)})$ and obtain a set of local coordinates $\mathbf{x}'_1, ..., \mathbf{x}'_K \in \mathbb{R}^m$.
  **for** $j = 1, ..., K$ **do**
    **if** $\frac{|\{\mathbf{x} \in \mathbb{R}^m : \langle \mathbf{x}'_i - \mathbf{x}'_j, \mathbf{x} - \mathbf{x}'_i \rangle > 0\}|}{|\{\mathbf{x} \in \mathbb{R}^m : \langle \mathbf{x}'_i - \mathbf{x}'_j, \mathbf{x} - \mathbf{x}'_i \rangle \leq 0\}|} \leq \tau_a$ **then**
      mark $j$ as `candidate`.
    **end**
  **end**
**end**
**if** *the number of* `candidate` *points is larger than* $\tau_b$ **then**
  Label point $i$ as `boundary`.
**end**

Once the boundary points are detected, the subset of consistent distances $\bar{P}$ is found according to criterion (2) and the matrix of weights $\mathbf{W}$.

### B. SMACOF algorithm

The minimization of the weighted stress is carried out using an iterative optimization algorithm with the SMACOF iteration

[14],

$$\mathbf{X}^{(k+1)} = \mathbf{V}^{\dagger}\mathbf{B}(\mathbf{X}^{(k)})\mathbf{X}^{(k)},$$

where $\mathbf{V}^{\dagger}$ denotes matrix pseudoinverse,

$$v_{ij} = \begin{cases} -w_{ij} & i \neq j \\ -\sum_{k\neq i} v_{ik} & i = j, \end{cases}$$

and $\mathbf{B}(\mathbf{X})$ is an $N \times N$ matrix dependent of $\mathbf{X}$ with elements,

$$b_{ij}(\mathbf{X}) = \begin{cases} -d_S(s_i, s_j)d_{ij}^{-1}(\mathbf{X}) & i \neq j \text{ and } d_{ij}(\mathbf{X}) \neq 0 \\ 0 & i \neq j \text{ and } d_{ij}(\mathbf{X}) = 0 \\ -\sum_{k\neq i} b_{ik} & i = j. \end{cases}$$

The SMACOF iteration produces a monotonous non-increasing sequence of stress values, and can be shown to be equivalent to a scaled steepest descent iteration with constant step size [20].

### C. Vector extrapolation

To speed up the convergence of the SMACOF iterations, we employ *vector extrapolation*. These methods use a sequence of solutions at subsequent iterations of the optimization algorithm and extrapolate the limit solution of the sequence. While these algorithms were derived assuming a linear iterative scheme, in practice, they work well also for nonlinear schemes, like some processes in computational fluid dynamics [21]. For further details, we refer the reader to [22], [23], [24].

The main idea of vector extrapolation is, given a sequence of solutions $\mathbf{X}^{(k)}$ from iterations $k = 0, 1, ...$, to approximate the limit $\lim_{k\to\infty} \mathbf{X}^{(k)}$, which must coincide with the optimal solution $\mathbf{X}^*$. The extrapolation $\hat{\mathbf{X}}$ is constructed as an affine combination of previous iterates,

$$\hat{\mathbf{X}} = \sum_{j=0}^{K} \gamma_j \mathbf{X}^{(k+j)}; \quad \sum_{j=0}^{K} \gamma_j = 1.$$

The coefficients $\gamma_j$ are determined in different ways. In the *reduced rank extrapolation* (RRE) method, $\gamma_j$ are obtained by the solution of the minimization problem,

$$\min_{\gamma_0,..,\gamma_K} \|\sum_{j=0}^{K} \gamma_j \Delta\mathbf{X}^{(k+j)}\|, \quad \text{s.t.} \sum_{j=0}^{K} \gamma_j = 1,$$

where $\Delta\mathbf{X}^{(k)} = \mathbf{X}^{(k+1)} - \mathbf{X}^{(k)}$. In the *minimal polynomial extrapolation* (MPE) method,

$$\gamma_j = \frac{c_j}{\sum_{i=0}^{K} c_i}, \quad j = 0, 1, ..., K,$$

where $c_i$ arise from the solution of the minimization problem,

$$\min_{c_0,..,c_{K-1}} \|\sum_{j=0}^{K} c_j \Delta\mathbf{X}^{(k+j)}\|, \quad c_K = 1,$$

which in turn can be formulated as a linear system [24].

### D. Multiresolution optimization

Another way to accelerate the solution of the MDS problem is using *multiresolution* (MR) methods [20]. The main idea is subsequently approximating the solution by solving the MDS problem at different resolution levels. At each level, we work with a *grid* consisting of points with indices $\Omega_L \subset \Omega_{L-1} \subset ... \subset \Omega_0 = \{1, ..., N\}$, such that $|\Omega_l| = N_l$. At the $l$th level, the data is represented as an $N_l \times N_l$ matrix $\mathbf{\Delta}_l$, obtained by extracting the rows and columns of $\mathbf{\Delta}_0 = \mathbf{\Delta}$, corresponding to the indices $\Omega_l$. The solution $X_l^*$ of the MDS problem on the $l$th level is transferred to the next level $l-1$ using an *interpolation operator* $P_l^{l-1}$, which can be represented as an $N_{l-1} \times N_l$ matrix.

1 Construct the hierarchy of grids $\Omega_0, ..., \Omega_L$ and interpolation operators $P_1^0, ..., P_L^{L-1}$.
2 Start with some initial $X_L^{(0)}$ at the coarsest grid, and $l = L$.
3 **while** $l \geq 0$ **do**
4 Solve the $l$th level MDS problem

$$\mathbf{X}_l^* = \underset{\mathbf{X}_l \in \mathbb{R}^{N_l \times m}}{\operatorname{argmin}} \sum_{i,j \in \Omega_l} w_{ij}(d_{ij}(\mathbf{X}_l) - \delta_{ij})^2$$

using SMACOF iterations initialized with $X_l^{(0)}$.
5 Interpolate the solution to the next resolution level, $X_{l-1}^{(0)} = P_l^{l-1}(X_l^*)$
6 $l \longleftarrow l - 1$
7 **end**

We use a modification of the *farthest point sampling* (FPS) [25] strategy to construct the grids, in which we add more points from the boundaries, to allow correct interpolation of the fine grid using the coarse grid elements. We use linear interpolation with weights determined using a least squares fitting problem with regularization made to ensure all available nearest neighbors are used.

The multiresolution scheme can be combined with vector extrapolation by employing MPE or RRE methods at each resolution level. In our experiments we used the RRE method, although in practice, for the SMACOF algorithm, both the MPE and the RRE algorithms gave comparable results, giving us a three-fold speedup. A comparison of the convergence with and without vector extrapolation and multiresolution methods is shown in Figure 2. The stress values shown are taken from the problem shown in Figure 4.

### E. Initialization

Since the stress function is non-convex, convex optimization method may converge to local minima. In order to avoid local convergence, we initialized the LSMDS problem by classical scaling result [26]. Although such an initialization does not guarantee global convergence in theory, in practice, we converge to the global minimum.

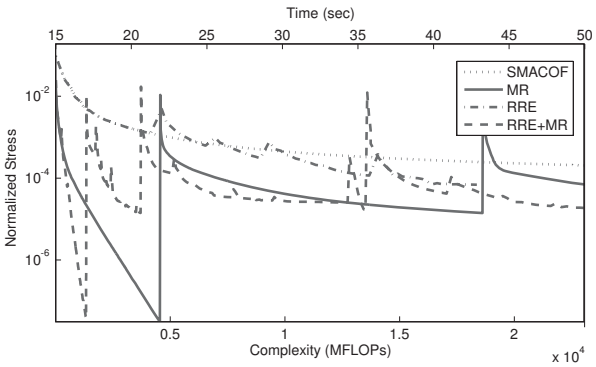


Fig. 2. Convergence (in terms of stress value) of basic SMACOF (dotted), SMACOF with RRE acceleration (dash-dotted), SMACOF with multiscale (solid) and SMACOF with both RRE and multiscale (dashed), in terms of CPU time and MFLOPS. CPU time is approximated. Convergence was stopped at the same relative change of stress value.

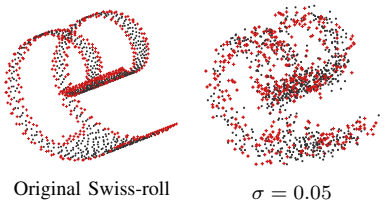


Fig. 3. Left: Swiss hole surface without noise. Right: A Swiss hole contaminated with additive Gaussian noise with $\sigma = 0.015$ and $\sigma = 0.05$, and the spiral surface. The detected boundary points are shown in red.

## IV. Examples and Applications

We applied the proposed algorithm on several synthetic examples, as well as image analysis problems. In the first experiment, we worked with the Swiss roll surface with a rectangular hole ("Swiss hole") sampled at 1200 points. The data was contaminated by Gaussian noise of different variance (Figure 3). We analyzed this manifold using our algorithm and compared the result to other local methods (see results in Figures 4,5). The TCIE algorithm demonstrates a very good robustness to noise.

In the second experiment, we generated a set of images of two discs, one stationary and the other moving. A small amount of additive white Gaussian noise was added to the images (Figure 6). The data manifold can be parameterized according to the location of the center of the moving disc, as shown in Figure 6 (see [15]). The results are shown in Figure 7. While Isomap and our algorithm manage to recover a meaningful parametrization, other methods produce results significantly different from the "ideal" parametrization. Compared to Isomap, the TCIE algorithm tends to less distort and expand the hole.

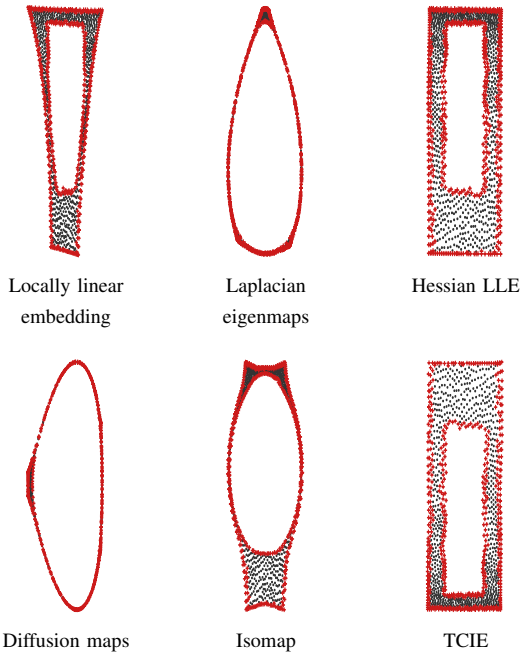


Fig. 4. Left to right top to bottom: Embedding of the Swiss roll (without noise), produced by LLE, Laplacian eigenmaps, Hessian LLE, diffusion maps, Isomap, and our algorithm. Detected boundary points are shown as red pluses.

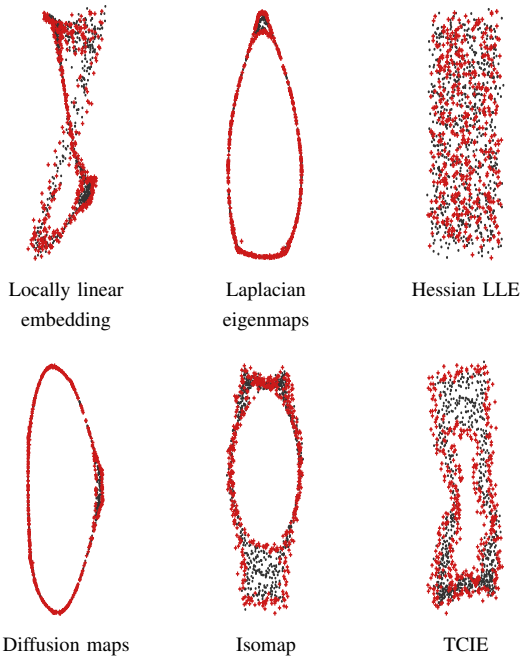


Fig. 5. Left to right top to bottom: Embedding of a 2D manifold contaminated by additive Gaussian noise with $\sigma = 0.05$, as produced by LLE, Laplacian eigenmaps, Hessian LLE, diffusion maps, Isomap, and our algorithm. Detected boundary points are shown as red pluses.

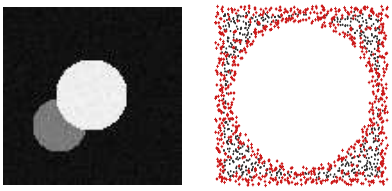

Fig. 6. Left: An example of the discs images. Coordinates of darker disc's center parameterize for the image manifold. Right: The parameterization manifold. The detected boundary points are shown in red.

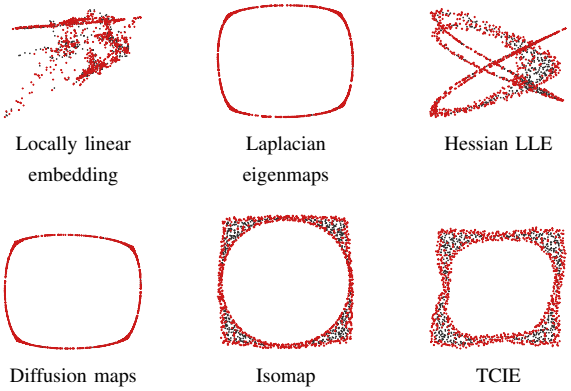


Fig. 7. Left to right top to bottom: Analysis of the disc images produced by LLE, Laplacian eigenmaps, Hessian LLE, diffusion maps, Isomap, and our algorithm. Detected boundary points are shown as red pluses.

Although in practical cases the data manifold is not necessarily isometrically parameterizable in a low-dimensional Euclidean space, our algorithm appears to a good approximation to be able to produce meaningful results in image analysis applications. Figure 8 demonstrates the recovery of gaze direction of a person from a sequence of gray-scale images. Assuming facial pose and expressions do not change significantly, images of the area of the eyes form a manifold approximately parameterized by the direction of the gaze. Similar to previous image manifold experiments [27], we use Euclidean distances between the row-stacked images as the distance measure. In order to reduce the effect of head movement, simple block matching was used.

## V. Conclusion

We introduced a new method for nonlinear dimensionality reduction. Experiments on synthetic and real-life examples show that our approach compares favorably to other state-of-the-art manifold learning methods, especially in better ability to handle data manifolds with complicated topology and significant amounts of noise. In our future work we plan exploring the use multigrid methods [20], as well as other numerical improvements that would allow handling of large data sets. Boundary detection plays a major role in validation of pairs of points handled by the flattening method. We intend to further explore alternatives for this part of our algorithm. Finally, we plan to test different distance measures between images [3],

Fig. 8. The intrinsic coordinates of the image manifold of the eyes area with different gaze directions, as mapped by our algorithm.

and evaluate their performances to other applications.

## Acknowledgment

We would like to thank Prof. Avram Sidi for his advice regarding vector extrapolation algorithms. We would also like to thank the respective owners of the code of various nonlinear dimensionality reduction methods demonstrated in this publication.

## References

[1] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, "Geometric diffusions as a tool for harmonic analysis and structure definition of data," *Proceedings of the National Academy of Sciences*, vol. 102, no. 21, pp. 7426–7431, May 2005.

[2] R. Pless, "Using Isomap to explore video sequences," in *Proceedings of the 9th International Conference on Computer Vision*, Nice, France, October 2003, pp. 1433–1440.

[3] M. Aharon and R. Kimmel, "Representation analysis and synthesis of lip images using dimensionality reduction," *International Journal of Computer Vision*, vol. 67, no. 3, pp. 297–312, 2006.

[4] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," 2002. [Online]. Available: citeseer.ist.psu.edu/article/belkin02laplacian.html

[5] R. M. Diaz and A. Q. Arencibia, Eds., *Coloring of DT-MRI FIber Traces using Laplacian Eigenmaps*. Las Palmas de Gran Canaria, Spain: Springer Verlag, February 24–28 2003.

[6] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Three-dimensional face recognition," *International Journal of Computer Vision (IJCV)*, vol. 64, no. 1, pp. 5–30, August 2005.

[7] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification and Scene Analysis*, 2nd ed. Wiley-Interscience, 2000.

[8] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[9] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in Neural Information Processing Systems*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds., vol. 14. Cambridge, MA: MIT Press, 2002, pp. 585–591.

[10] C. Grimes and D. L. Donoho, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," *Proceedings of the National Academy of Sciences*, vol. 100, no. 10, pp. 5591–5596, May 2003.

[11] K. Q. Weinberger and L. K. Saul, "Unsupervised learning of image manifolds by semidefinite programming," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2. Washington D.C.: IEEE Computer Society, 2004, pp. 988–995.

[12] K. Q. Weinberger, B. D. Packer, and L. K. Saul, "Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization," in *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, Barbados, January 2005.
[13] E. L. Schwartz, A. Shaw, and E. Wolfson, "A numerical solution to the generalized mapmaker's problem: Flattening nonconvex polyhedral surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 1005–1008, November 1989.
[14] I. Borg and P. Groenen, *Modern multidimensional scaling: Theory and applications*. New York: Springer Verlag, 1997.
[15] C. Grimes and D. L. Donoho, "When does isomap recover the natural parameterization of families of articulates images?" Department of Statistics, Stanford University, Stanford, CA 94305-4065, Tech. Rep. 2002-27, 2002.
[16] M. Bernstein, V. de Silva, J. C. Langford, and J. B. Tenenbaum, "Graph approximations to geodesics on embedded manifolds," Stanford University," Technical Report, January 2001.
[17] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching," *Proceedings of the National Academy of Sciences*, vol. 103, no. 5, pp. 1168–1172, January 2006.
[18] G. Guy and G. Medioni, "Inference of surfaces, 3D curves, and junctions from sparse, noisy, 3D data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 11, pp. 1265–1277, November 1997.
[19] T. E. Boult and J. R. Kender, "Visual surface reconstruction using sparse depth data," in *Computer Vision and Pattern Recognition*, vol. 86, 1986, pp. 68–76.
[20] M. M. Bronstein, A. M. Bronstein, and R. Kimmel, "Multigrid multidimensional scaling," *Numerical Linear Algebra with Applications (NLAA)*, vol. 13, no. 2-3, pp. 149–171, March-April 2006.
[21] A. Sidi, "Efficient implementation of minimal polynomial and reduced rank extrapolation methods," *J. Comput. Appl. Math.*, vol. 36, no. 3, pp. 305–337, 1991.
[22] S. Cabay and L. Jackson, "Polynomial extrapolation method for finding limits and antilimits of vector sequences," *SIAM Journal on Numerical Analysis*, vol. 13, no. 5, pp. 734–752, October 1976.
[23] R. P. Eddy, "Extrapolationg to the limit of a vector sequence," in *Information Linkage between Applied Mathematics and Industry*, P. C. Wang, Ed. Academic Press, 1979, pp. 387–396.
[24] D. A. Smith, W. F. Ford, and A. Sidi, "Extrapolation methods for vector sequences," *SIAM Review*, vol. 29, no. 2, pp. 199–233, June 1987.
[25] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Zeevi, "The farthest point strategy for progressive image sampling," *IEEE Transactions on Image Processing*, vol. 6, no. 9, pp. 1305–1315, September 1997. [Online]. Available: citeseer.ist.psu.edu/eldar97farthest.html
[26] A. Kearsley, R. Tapia, and M. W. Trosset, "The solution of the metric stress and sstress problems in multidimensional scaling using newton's method," *Computational Statistics*, vol. 13, no. 3, pp. 369–396, 1998.
[27] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, December 2000.