

# Topologically Constrained Isometric Embedding

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

Department of Computer Science,  
Technion – Israel Institute of Technology,  
Haifa 32000, Israel

Email: {rosman|bron|mbron|ron}@cs.technion.ac.il

**Abstract**— We present a new algorithm for nonlinear dimensionality reduction that consistently uses global information, which 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. 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) is a general name for algorithms that explain a given data set of high dimensionality, in terms of a small number of variables or coordinates. Such methods are used in a variety of 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 attempt to find a map from the data to the coordinate system of given dimensionality that represent the given data by minimizing some error measure. Unlike classical dimensionality reduction methods such as principal component analysis (PCA) [7], the map is nonlinear. The data are usually assumed to arise from a high-dimensional manifold  $\mathcal{M}$ , embedded into a Euclidean space  $\mathbb{R}^n$ . The metric structure of the manifold is represented by the geodesic distances  $\delta : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}$ . In the discrete setting, the data are represented as a graph whose vertices  $\mathbf{z}_1, \dots, \mathbf{z}_N$  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. NLDR algorithms usually approximate *local* (short) distances on the data manifold by the Euclidean distances in the embedding space,  $\delta_{ij} = \|\mathbf{z}_i - \mathbf{z}_j\|_2$ , for  $i, j$  such that  $a_{ij} = 1$ .

The goal is to find a set of coordinates  $\mathbf{x}_1, \dots, \mathbf{x}_N$  in a low-dimensional space  $\mathbb{R}^m$  ( $m \ll n$ ) 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] tries to express each point as a linear combination of its neighbors. The deviation of each point from this linear combination is summed over 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 lowest  $d + 1$  eigenvectors, and using eigenvectors  $2..d + 1$  as the coordinates of the data points.

*Diffusion maps* have been recently proposed as an extension of Laplacian eigenmaps, able to handle non-uniform sampling of the manifold [1].

The *Hessian eigenmaps* algorithm [10], uses a different error measure, claiming coordinate functions should be locally linear, and therefore have a small as possible quadratic component. The algorithm therefore seeks functions which minimize the Hessian, summed over the manifold. This algorithm expresses the sum of the quadratic components of each coordinate function, approximated at each point, as a function to minimize, again resulting in an Eigenvalue problem, whose first  $d + 1$  minimizing vectors give us the desired coordinate vectors.

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, with the preservation of local distances explicitly imposed as constraints. Solving the resulting SDP problem, however, has a high computational cost, despite attempts to lower it [12].

*Isomap*, unlike the local methods, tries to preserve a *global* invariant – the geodesic distances on the data manifold. Geodesics are approximated by assuming short Euclidean distances to be equal to short geodesic ones, and propagating distances from each point using the proximity graph. A multidimensional scaling (MDS) algorithm is then used to find a set of coordinates whose Euclidean distances in a way . While the approximation of the geodesics may be interrupted by noise, if the manifold is sampled densely, the approximated distances remain robust even when the amplitude of the noise is large relative to the assumed neighborhood size. This allows the algorithm to reconstruct the intrinsic representation of the manifold, even when strong additive noise causes locally acting methods to fail.

Donoho and Grimes have pointed out [13] that isomap may give distorted results in the case where the data is intrinsically non-convex. The reason for this is that isomap assumes that the shortest path connecting each two points is a line in intrinsic coordinates. This is only correct if this path is included in the manifold, which means by definition of convexity that the manifold is intrinsically convex. This assumption comes

into play when finding the correct Euclidean representation. The least squares MDS algorithm minimizes the Stress error function[14],

$$\mathbf{X} = \underset{\mathbf{X}}{\operatorname{argmin}} \sum_{i < j} w_{ij} (d_{ij}(\mathbf{X}) - \delta_{ij})^2.$$

Here  $\mathbf{X} = (x_{ij})$  is an  $N \times m$  matrix whose rows are the coordinate vectors in the new, low dimensional, Euclidean space.  $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$  is the Euclidean distance between points  $\mathbf{x}_i$  and  $\mathbf{x}_j$  in the new coordinates in  $\mathbb{R}^m$ . The non-negative weight  $w_{ij}$  is usually taken to be 1 for all point pairs.

Note that the minimal path length between two points  $i$  and  $j$  in a sub-region of Euclidean space is equal to  $d_{ij}$  if and only if the line connecting points  $i$  and  $j$  is included in the region. It is this assumption that distorts the results of the Isomap algorithm on non-convex manifolds. We suggest to avoid this problem by avoiding the imitation of problematic geodesic distances. An algorithm for detecting and avoiding these is described in Section II. Its correctness is shown in Section III. Results on synthetic data, as well as an application for eye tracking, are shown in Section IV.

## II. TOPOLOGICALLY CONSTRAINED ISOMETRIC EMBEDDING

Assuming the manifold is intrinsically flat, we suggest a way to detect these geodesics. These geodesics' influence can then be avoided by setting the weight associated with their end points to zero. We claim the correct geodesics under the Euclidean manifold assumption are those that do not touch the boundary of the manifold, as we discuss in Section III. We therefore proceed as follows:

- Detect the boundary points of the manifold
- Detect geodesics which pass through boundary points
- Minimize a weighted Stress with the weights of boundary touching geodesics set to zero, and the rest of the weights set to one.

### A. Detecting Boundary Points

Detection of boundary points of sampled manifolds and surfaces is a well developed field and several algorithms have been developed over the years (see for example [15], [16]) and can be used. We have tested for detection of the boundary points two methods, based on the fact that for a  $d$ -manifold  $\mathcal{M}$  with boundary, a homeomorphism taking each point's neighborhood into a Euclidean subspace of dimension  $d$  exists only for interior points, whereas boundary points are only homeomorphic to partial regions of such a subspace. Both methods look at the  $K$  nearest neighbors of each point, in coordinates reconstructed from the distances using classical multidimensional scaling (MDS).

The first method, given the local coordinates, attempts to detect the boundary by assuming the point, along with two of its neighbors form part of a curve along the boundary, and try to determine whether or not it is a part of the boundary. The algorithm goes as follows:

- 1 Consider the set of  $K$  nearest neighbors of point  $i$ . Apply MDS to compute a local representation. Let  $\{\mathbf{z}_i\}_{i=1}^K$  be the  $d$ -dimensional local coordinates obtained by this representation.
- 2 For each pair,  $j, k$  of neighbors of  $i$ , such that the angle  $\angle \mathbf{z}_j \mathbf{z}_i \mathbf{z}_k$  is close enough to  $\pi$ , we mark the pair as valid. Let  $\overline{\mathbf{z}_j \mathbf{z}_k}$  and  $(\overline{\mathbf{z}_j \mathbf{z}_k})^\perp$  denote the vector connecting the points  $\mathbf{z}_j$  and  $\mathbf{z}_k$  and its rotation by  $90^\circ$  clockwise, respectively. If we find a point  $\mathbf{z}_m$  such that  $\frac{\langle \overline{\mathbf{z}_m \mathbf{z}_i}, (\overline{\mathbf{z}_j \mathbf{z}_k})^\perp \rangle}{\|\overline{\mathbf{z}_m \mathbf{z}_i}\| \|(\overline{\mathbf{z}_j \mathbf{z}_k})^\perp\|}$  is close to 1, we label the pair  $j, k$  as satisfied.
- 3 If the fraction of satisfied valid pairs out of all valid pairs is smaller than some threshold, we consider the point  $i$  to be a boundary point.

The second method, more suitable for manifolds of higher intrinsic dimensionality, tries to look at the direction normal to the boundary. Moving in the normal direction, the density of sampling points should drop to zero. We can check along each direction denoted by the difference vector between the point  $i$  and one of its neighbors,  $j$ . Assuming close to uniform density of the points, one such neighboring point  $j$  should produce a vector pointing close to the normal direction.

The algorithm, for each point,  $i$ , is as follows:

- 1 Consider the set of  $K$  nearest neighbors of point  $i$ . Apply MDS to compute a local representation. Let  $\{\mathbf{z}_i\}_{i=1}^K$  be the  $d$ -dimensional local coordinates obtained by this representation.
- 2 For each point  $j \in \{1..K\}$  consider the direction  $\overline{\mathbf{z}_i \mathbf{z}_j}$ . This direction is normal to a hyperplane set at  $\mathbf{z}_i$ . Check if the ratio of the number of points on each side of the hyperplane is larger than a certain threshold.
- 3 If the fraction of large ratio directions is above a certain threshold, mark point  $i$  as a boundary point.

After detecting the boundary points, we determine which geodesics pass through boundary points. Note that when propagating the distances from a point on the proximity graph, a dynamic programming algorithm is often used. The detection of geodesics which pass close to the boundary can be incorporated into such dynamic programming algorithms. We demonstrate this for the Dijkstra ([17], see also [18]) algorithm, but it can be done for other dynamic programming weighted shortest paths algorithms such as those by Bellman and Ford, or Floyd.

For example, the Dijkstra algorithm is modified as follows: While extending the shortest path from source  $i$  to point  $j$ , towards point  $j$ 's neighbors: For each neighbor  $k$  of point  $j$ , check if the shortest path from  $i$  to point  $k$  is longer than the route through  $j$ . If so, update the path length to the shorter path, as is done in the Dijkstra algorithm, but in addition, mark the newly updated path as a geodesic to be removed from the

MDS, provided that,

- i The path from  $i$  to  $j$  is a path marked to be removed, or
- ii  $j$  is a boundary point, and the path from  $i$  to  $k$  through  $j$  travels through more than one point.

The restriction on the addition of points is meant to prevent paths that end in a boundary point, but do not touch the boundary and continue, from being removed.

Geodesics marked for removal have their point pair given weight  $w_{ij} = 0$  in the weighted Stress function. All other point pairs are given a weight of one.

We note that this removal of geodesics should allow non-convex manifolds to be analyzed correctly, as is shown in Section III, but this is assuming the manifold is intrinsically flat. In the more general case, this is not sufficient. A possible sufficient rule would be to remove geodesics connecting points  $i$  and  $j$  such that  $d(\mathbf{x}_i, \mathbf{x}_j) < d(\mathbf{x}_j, \partial\mathcal{M}) + d(\mathbf{x}_i, \partial\mathcal{M})$ , where  $\partial\mathcal{M}$  is the boundary of the manifold to be analyzed. This rule, however, is too restrictive and may be improved upon in certain cases.

### B. Minimizing the Stress Function

When minimizing the Stress function one has to consider both the speed of convergence and the prospects of converging to a local minimum.

The least squares Stress function is non-convex and as such, its convergence may converge to local minima, far from the global minimum. In order to get a good convergence, we use as an initial solution the solution obtained by classical scaling. Classical scaling tries to minimize the Strain function [14],

$$\mathbf{X} = \underset{\mathbf{X}}{\operatorname{argmin}} \|\mathbf{J}^T (\mathbf{D}(\mathbf{X}) - \Delta) \mathbf{J}\|_{\mathbb{F}}^2.$$

The classical scaling solution has been used before [19] as initialization for the fully weighted Stress functional. Although this does not guarantee convergence, in practice, for intrinsically flat examples we've experimented with, the algorithm converged to the global minimum.

We then use the SMACOF algorithm [14] to find the optimal solution for the weighted Stress function. The SMACOF algorithm can be shown to be equivalent to a gradient descent step with a constant step size. Unlike constant step gradient descent for nonlinear functions, SMACOF iterations are proven to converge (see [14]). To further accelerate the convergence, we have used a multiscale framework, and for convergence at each scale we have used vector extrapolation methods.

Multiscale methods have been used before for accelerating the SMACOF algorithm[20]. When minimizing the weighted Stress function, it is important to correctly perform the restriction to a coarser grid. We use the farthest point sampling algorithm to select the representatives of each grid, but add more points from the boundaries, to allow correct interpolation of the fine grid using the coarse grid elements. We solve the problem at each grid, starting from the coarsest level, and use the coarse grid solution to interpolate the starting position of the finer grid elements before allowing it to converge. Each element grid is linearly interpolated using the coordinates of

its nearest coarse grid neighbors. The interpolation weights are determined using a least squares fitting problem with regularization made to ensure all nearest neighbors are used if possible. At each grid, after interpolating the coordinates of the elements, the SMACOF algorithm is used to minimize the Stress function, before moving to the next grid. In order to accelerate the convergence, vector extrapolation methods are used on the iteration vectors.

Vector extrapolation methods, such as Minimal Polynomial Extrapolation and Reduced Rank Extrapolation take a series of iterate vectors derived by some iteration scheme, and extrapolate the limit solution of this series. While these algorithms were derived assuming a linear iterative scheme, in practice they work well enough for nonlinear equations, such as those found in computational fluid dynamics[21]. The methods we have experimented with are the Minimal Polynomial Extrapolation (MPE) algorithm, and the Reduced Rank Extrapolation method (RRE) algorithm. An overview of these algorithms is given here, as these algorithms were previously described in details [22], [23], [24]. In describing these algorithms we use the following notation: Let  $\{\mathbf{x}_i\}_{i=0}$  be a series of iteration vectors resulting from an iterative process. We denote  $\mathbf{u}^{(j)} = \mathbf{x}^{(j+1)} - \mathbf{x}^{(j)}$ , and  $\mathbf{v}^{(j)} = \mathbf{u}^{(j+1)} - \mathbf{u}^{(j)}$  in the following description of these methods. The matrices  $\mathbf{U}$  and  $\mathbf{V}$  are matrices whose columns are the vectors  $\{\mathbf{u}^{(i)}\}$  and  $\{\mathbf{v}^{(i)}\}$  respectively, for a certain sequence of iterations.

1) *Minimal Polynomial Extrapolation (MPE)*: This method computes the minimal polynomial  $P(\lambda)$  of  $\mathbf{A}$  such that  $P(\mathbf{A})\mathbf{u}_0 = 0$ . As shown by [24], the coefficients of this polynomial can be computed from a linear equation in  $\{\mathbf{u}^{(j)}\}$ . The resulting polynomial can then be used to extrapolate the solution to the iterative process. In practice, the solution of the extrapolation is defined as

$$\mathbf{s}_{n,k} = \sum_{j=0}^k \gamma_j \mathbf{x}_{n+j}$$

Where the coefficients  $\gamma_j$  are defined as

$$\gamma_j = \frac{c_j}{\sum_{i=0}^k c_i}$$

Where the vector  $\mathbf{c}$  is obtained by solving

$$\mathbf{U}_{n,k} \mathbf{c} = -\mathbf{u}_{n+k}$$

2) *Reduced Rank Extrapolation method (RRE)*: Again, looking at a linear iteration scheme, since  $\mathbf{v}^{(j)} = (\mathbf{A} - \mathbf{I})\mathbf{u}_j^{(j)}$ , assuming we have  $N$  iteration vectors, and if the matrix  $\mathbf{V}$  is of full rank, we could write

$$(\mathbf{I} - \mathbf{A})^{-1} = -\mathbf{U}\mathbf{V}^{-1}.$$

The Reduced Rank Extrapolation method instead solves the equation system

$$\begin{aligned} \mathbf{s} &= \mathbf{x}^{(0)} + \mathbf{U}\xi \\ 0 &= \mathbf{u}^{(0)} + \mathbf{V}\xi. \end{aligned}$$

but does so with smaller matrices  $\mathbf{U}, \mathbf{V}$ , of size  $N \times k$ .

We have used in our algorithm the RRE method, although in practice, for the SMACOF algorithm, both the MPE and the RRE algorithm gave comparable results, giving us a speed-up of at least times 3. A comparison of the convergence with and without vector extrapolation and multiscale methods is shown in Figure 1. The Stress values shown are taken from the convergence on the problem shown in Figure 3, with 1200 points.

### III. CORRECTNESS OF THE ALGORITHM

The correctness of the Isomap algorithm has been shown for case of convex manifolds by Bernstein et. al [25]. More specifically, they prove the graph distance converges to the geodesic distance. They assume, however, that these distances should be line distances in the Euclidean subspace Isomap maps. This is true provided this region is convex. If the region is non-convex, the shortest paths connecting points may have to curve around the boundary of the region. We therefore show we detect (and remove) all paths for which the line distance should be different from the geodesic distance.

We assume the manifold we wish to map is a smooth, compact, connected, manifold  $\mathcal{M}'$ , isometric to a non-convex continuous subset  $\mathcal{M}$  of  $\mathbb{R}^d$ . The set  $\mathcal{M}$  is a subset of a strongly geodesically convex set,  $\mathcal{C}$  (for example, we can choose  $\mathcal{C} = \text{conv}(\mathcal{M})$ , the convex hull of  $\mathcal{M}$ ). Since geodesics in  $\mathcal{C}$  that stay in the interior of  $\mathcal{M}$  are the same in both manifolds, a shortest path between the points in  $\mathcal{M}$  is also a unique geodesic. These geodesics are approximated correctly according to [25].

The geodesics in  $\mathcal{C}$  that pass outside of  $\mathcal{M}$  are removed from the set of distances processed by the MDS procedure in the proposed algorithm. This is shown by the following proposition:

*Proposition 1:* Let  $\mathcal{C}'$  be a compact manifold isometric to a convex region in a Euclidean space and  $\mathcal{M}'$  a submanifold of  $\mathcal{C}'$ . For any geodesic  $g_{\mathcal{C}'}(p, q)$  in  $\mathcal{C}'$  which is not restricted to  $\mathcal{M}'$ , the shortest path between  $p, q$  in  $\mathcal{M}'$  will touch the boundary of  $\mathcal{M}'$ , and is thus ignored by the MDS procedure in the proposed topologically constrained isometric embedding algorithm.

*Proof:* Consider the shortest path from  $p$  to  $q$ , where  $p, q \in \text{int}(\mathcal{M})$ , and  $\text{int}(\mathcal{M})$  is the interior of  $\mathcal{M}$ . We consider the case where this geodesic in  $\mathcal{C}$  passes partially outside of  $\mathcal{M}$ . We also assume that this geodesic enters and exits only once the region of  $\mathcal{C} \setminus \mathcal{M}$ . If a geodesic enters and leaves  $\mathcal{C} \setminus \mathcal{M}$  more than once, we can restrict the analysis to a subsection of the geodesic.

Assume the shortest path connecting  $p, q$  in  $\mathcal{M}$  does not touch the boundary of  $\mathcal{M}$ . By the choice of  $p$  and  $q$ , the shortest path restricted to  $\mathcal{M}$  between  $p$  and  $q$  is not a straight line (the line connecting  $p$  and  $q$  is not included in  $\mathcal{M}$ ). As long as it does not touch the boundary, we can shorten the length of the geodesic, by replacing sections of it with straight segments. That is, we replace a curved section between points  $p'$  and  $q'$  by a straight line, as shown in Figure ??.

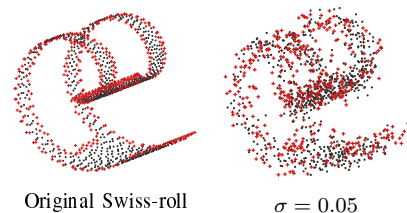


Fig. 2. On the left: Swiss hole surface without noise. On the right: A Swiss hole contaminated by Gaussian noise with  $\sigma = 0.015$  and  $\sigma = 0.05$ , and the spiral surface. Detected boundary points are shown in red.

The length of the curve becomes smaller and it is bounded from below by the length  $d(p, q)$ . We can only achieve this length by passing through  $\mathcal{C} \setminus \mathcal{M}$ , using a straight line – this geodesic is unique since  $\mathcal{C}'$  is strongly geodesically convex. Thus, we can always find a shorter curve as long as our geodesic does not touch a boundary point, in contradiction to the definition of a geodesic. Therefore the shortest path has to touch the boundary. Since the isometry to  $\mathcal{M}'$  is bijective, the shortest path must touch the boundary in the embedded space as well. Therefore, we detect it when calculating the shortest path and remove it from the minimization by setting to zero the appropriate weight. ■

### IV. EXAMPLES AND APPLICATIONS

We have used the algorithm on several synthetic examples, as well as in image analysis tasks. For example, we have taken a Swiss roll surface with a rectangular hole cut into it. We've checked the reconstruction of the intrinsic coordinates obtained using our algorithm, as well as using several other, locally acting, algorithms, after subjecting the surface coordinates to various levels of Gaussian additive noise. The original surfaces, embedded in three dimensional space are shown in Figure 2. The resulting mapping created by several algorithms as well as ours is shown for the noiseless case and a case with strong noise in Figures 3,4. This example demonstrates the robustness of the algorithm with respect to noise under the assumption of Euclidean isometry.

Although in the case of real-life data analysis the data may not be isometric to a Euclidean manifold under a reasonable choice of distances between data points, in practice our algorithm can be applied to image manifolds analysis. Specifically, we tried to map the gaze direction of a person from a sequence of grayscale images. Assuming the facial pose and expressions do not change dramatically, images of the area of the eyes form a manifold which can be parameterized by the location of the pupil on the eye ball. Similar to previous image manifold experiments [26], we have used Euclidean distances between the row-stacked image vectors as the extrinsic distance measure. In order to minimize the effect of head movement, a simple block matching was used, and the same facial region was compared. We then used our algorithm to produce a mapping of the image manifold. The resulting two dimensional plot of the intrinsic coordinates of

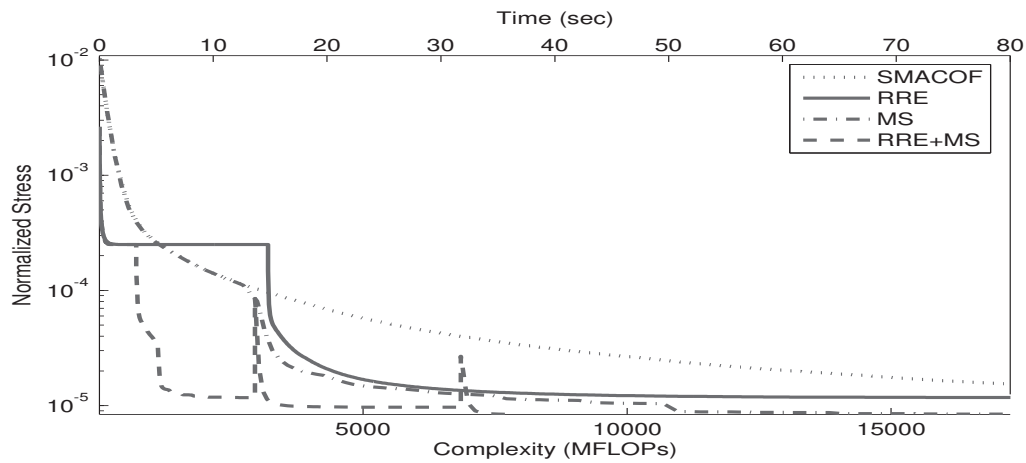


Fig. 1. Convergence (in terms of Stress value) of basic SMACOF (dotted), SMACOF with RRE acceleration (solid), SMACOF with multiscale (dash-dotted) and SMACOF with both RRE and multiscale (dashed), in terms of CPU time and MFLOPs. CPU time is approximated.

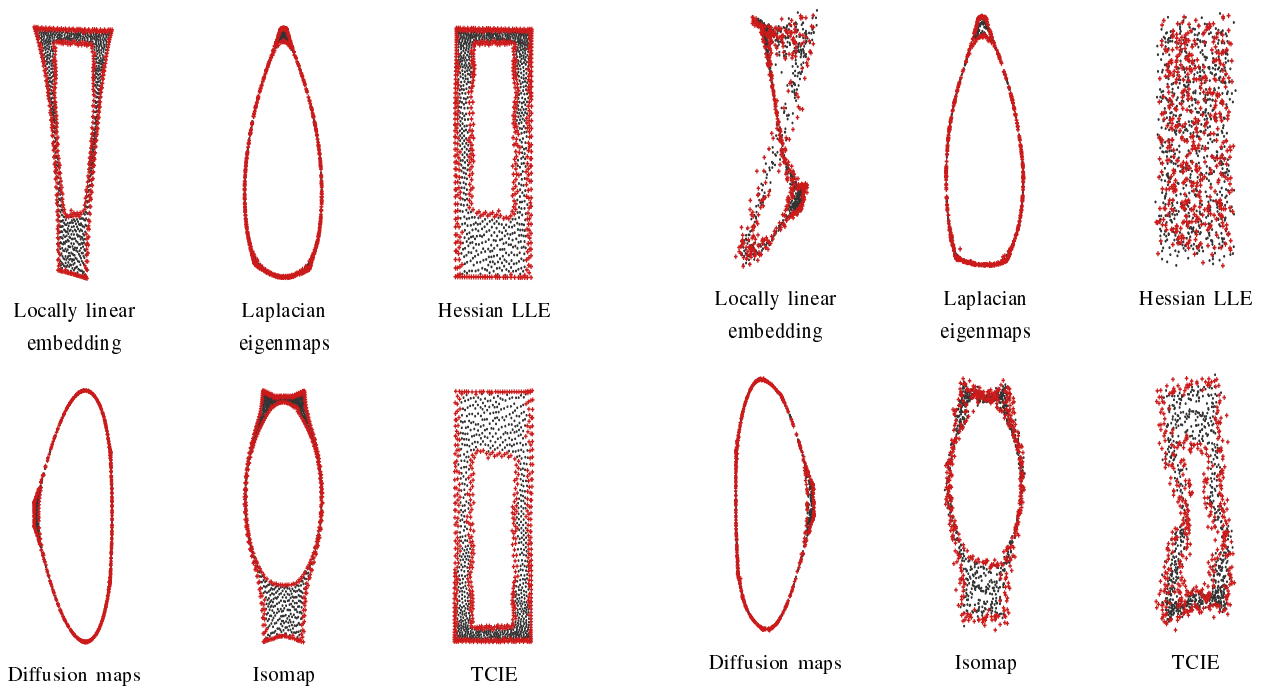


Fig. 3. Embedding of the swiss roll (without noise), as produced by LLE, Laplacian eigenmaps, Hessian LLE, diffusion maps, Isomap, and our algorithm. Detected boundary points are shown as red pluses.

Fig. 4. Embedding of a 2D manifold contaminated by 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.

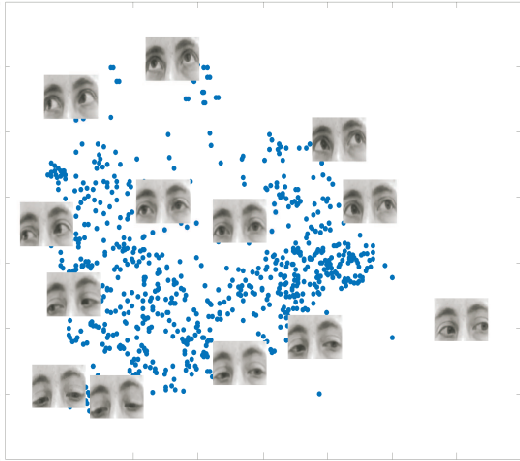


Fig. 5. The intrinsic coordinates of the image manifold of the eyes area with different gaze directions, as mapped by our algorithm.

the data points, as well as sampled images is shown in Figure 5.

## V. CONCLUSION

We have demonstrated a new algorithm for nonlinear dimensionality reduction. We have shown its robustness to noise on synthetic examples in comparison to local methods, and demonstrated its usefulness on real image data. Future work may include the use of multigrid methods (see for example [20]), as well as improvements to allow handling of larger data sets. In addition, the boundary detection algorithms currently used may be replaced with or compared to other algorithms. More specifically, with respect to image manifolds analysis, we would like to use different distance measures between images [3], as well as trying to weaken the flat isometry assumption, which is rarely accurate real-life data.

## ACKNOWLEDGMENT

The authors would like to thank Prof. Avram Sidi for his advice regarding vector extrapolation algorithms. The authors would also like to thank the respective owners of the various nonlinear dimensionality reduction algorithms shown 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. [Online]. Available: citeseer.ist.psu.edu/belkin01laplacian.html
- [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] C. Grimes and D. L. Donoho, "Image manifolds which are isometric to Euclidean space," Department of Statistics, Stanford University, Stanford, CA 94305-4065, Tech. Rep. 2002-27, 2002.
- [14] I. Borg and P. Groenen, *Modern multidimensional scaling: Theory and applications*. New York: Springer Verlag, 1997.
- [15] 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.
- [16] 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.
- [17] E. W. Dijkstra, "A note on two problems in connection with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.
- [18] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. MIT Press and McGraw-Hill, 1990.
- [19] 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.
- [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, "Extrapolating 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] 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.
- [26] 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.