CONNECTING SPECTRAL AND SPRING METHODS FOR MANIFOLD LEARNING

Shannon M. Hughes, Peter J. Ramadge

Department of Electrical Engineering, Princeton University, Princeton, NJ 08544

ABSTRACT

Diffusion Maps (DiffMaps) has recently provided a general framework that unites many other spectral manifold learning algorithms, including Laplacian Eigenmaps, and it has become one of the most successful and popular frameworks for manifold learning to date. However, Diffusion Maps still often creates unnecessary distortions, and its performance varies widely in response to parameter value changes. In this paper, we draw a previously unnoticed connection between DiffMaps and spring-motivated methods. We show that DiffMaps has a physical interpretation: it finds the arrangement of high-dimensional objects in low-dimensional space that minimizes the elastic energy of a particular spring network. Within this interpretation, we recognize the root cause of a variety of problems that are commonly observed in the Diffusion Maps output, including sensitivity to user-specified parameters, sensitivity to sampling density, and distortion of boundaries. We then show how to exploit the connection between Diffusion Map and spring criteria to create a method that can be efficiently applied post hoc to alleviate these commonly observed deficiencies in the Diffusion Maps output.

***Index Terms*—** multidimensional signal processing, unsupervised learning

1. INTRODUCTION

A theme often seen in signal processing is that our ability to process data effectively is highly dependent on our ability to characterize its underlying structure. For example, representing real world images in terms of wavelets has led to superior methods for compression, denoising, etc. New datasets, e.g. from genomics, neuroscience, etc., often involve acquiring a very large volume of data (i.e. a high-dimensional vector) as we vary a small number of variables in the experiment. If there is a continuous mapping relating these high-dimensional data points and underlying variables, we can effectively model our data points as lying on a low-dimensional manifold in high-dimensional space. In this paper, we examine methods that try to discover this type of underlying structure in datasets as a precursor to other types of processing.

To set up some notation, given points $x_i \in \mathbb{R}^N, i = 1, \ldots, M$, N large, that are randomly sampled from a d-dimensional ($d \ll N$) manifold $\mathcal{M} \subset \mathbb{R}^N$, the manifold learning problem is to find an embedding $f : \mathcal{M} \to \mathbb{R}^d$ that best preserves the structure of $\mathcal{M}$. Ideally, we would like $y_i = f(x_i) \in \mathbb{R}^d$ to reflect an underlying parameterization of (e.g. a chart on) the manifold $\mathcal{M}$. In practice, however, one seeks embedded points $y_i, i = 1, \ldots, M$, that optimize some objective function $J(x_1, \ldots, x_M; y_1, \ldots, y_M)$. Successful choices for J will be discussed later.

Over the past several years, a large number of algorithms for tackling this problem have emerged. These include ISOMAP [1], Locally Linear Embedding (LLE) [2], Laplacian Eigenmaps (LapEigs) [3], Hessian Eigenmaps [4], Local Tangent Space Alignment (LTSA) [5], Maximum Variance Unfolding (MVU) [6], Kernel PCA [7], and Diffusion Maps [8]. The last and most recent of these has provided a more general framework that unites many of the preceding algorithms including ISOMAP, LLE, LapEigs, and LTSA. Diffusion Maps solves a relatively quick eigendecomposition problem in order to obtain its solution and generally produces a good embedding (under favorable parameters). It, and its subcases, have thus become some of the most popular manifold learning algorithms to date.

However, while quite successful, the DiffMaps algorithm still lacks several desirable properties for manifold learning. We shall address the following two issues.

1: Unnecessary distortions, particularly of linear subspaces. DiffMaps often introduces unnecessary distortions into its embeddings. This is seen most clearly in the case where the data points $\{x_i\}_{i=1}^M$ lie in a linear subspace of $\mathbb{R}^N$ of dimension less than or equal to d. Here, it is clearly possible to preserve all inter-point relationships perfectly by simple linear projection; indeed, PCA would produce such an embedding. However, DiffMaps generally fails to do so. (See e.g. Fig. 1.)

2: Robustness to variation in user-specified parameters. DiffMaps often produces dramatically different results for different values of user-specified parameters (see e.g. the examples in [3, 8]). Moreover, there is currently no commonly accepted method for choosing appropriate parameters, with individual researchers favoring particular heuristics. As a result, while DiffMaps is able to produce good results under very careful selection of parameters, it is often hit-or-miss in practice, depending largely on the skill, experience, or patience of the individual researcher. In order to be more robust and widely usable, a manifold learning algorithm must produce good results for a much wider range of input parameters or, ideally, eliminate the need for user-specified parameters entirely.

As part of a program to address these issues, we relate a different framework to DiffMaps in Section 2. In Section 2.1, we will show how this alternate approach can be used to explain some of the observed problems with DiffMaps. We then confirm this intuition through side-by-side comparison of the DiffMaps criterion with a closely related one. Finally, we show how these two criteria, one easy to optimize but problematic in output, the other difficult to optimize but producing better results, can cross-inform each other: in particular, we show how to efficiently tweak the output of DiffMaps posthoc to alleviate undesirable effects.

2. SPRING INTERPRETATION OF DIFFUSION MAPS

We start by reviewing the operation of DiffMaps. DiffMaps, and its previously mentioned subcases, use as embedding the eigenfunctions of an operator $A : L^2(\mathcal{M}) \to L^2(\mathcal{M})$ defined by $(Af)(x) = \int_{\mathcal{M}} k(x, y) f(y) d\mu(y)$, where the kernel k varies according to the algorithm. In particular, the heat kernel

$$k_{heat}(x, y) = e^{-\frac{\|x-y\|^2}{t}} \tag{1}$$

(with user-specified parameter t) is commonly used. Equivalently, they select $f : \mathcal{M} \to \mathbb{R}^d$ to optimize the functional $J(f) = \|A(f)\|^2_{L^2(\mathcal{M})}$ subject to the constraint that $<f^k, f^l> = \delta_{ij}$, where f^k is the k-th component of the function f and δ_{ij} is 1 if $i = j$, 0 otherwise. On the set of samples $\{x_i\}_{i=1}^M$, this is choosing $y_i \in \mathbb{R}^d$, $i = 1, \ldots, M$, that minimize the objective

$$J(x_1, \ldots, x_M; y_1, \ldots, y_M) = \sum_{i,j} W_{ij} (\|y_i - y_j\|)^2 \tag{2}$$

with $W_{ij} = k(x_i, x_j)$, subject to the constraint that the vectors $Y^k, k = 1, \ldots, d$, formed from the k-th component of each y_i, satisfy, for a suitable inner product, $<Y^k, Y^l> = \delta_{kl}$.

Intuitively, for $k(a,b)$ monotonically decreasing with $\|a-b\|$, this objective creates an embedding that preserves the manifold structure by strongly penalizing large distances between y_i and y_j if x_i and x_j are close on the manifold. The result is an embedding in which y_i and y_j are close if x_i and x_j are. The constraint is necessary to prevent degenerate solutions such as collapsing all points to a single location (a global minimizer of the objective function), or collapsing the points into a lower dimensional space, which also lowers the objective value. The constraint thus ensures that the points "spread out" in all d dimensions.

As an alternative interpretation, consider a stretched ideal spring with one endpoint fixed at y_0 and one free endpoint y. This spring exerts a restoring force $F(y) = -k(y-y_0)(\|y-y_0\|-r)$ where k is the spring constant and r its natural length. One can determine the potential energy in such a stretched spring by calculating the work done to stretch it. This is an integral of force exerted over distance and results in the expression $U(y) = \frac{1}{2}k(\|y-y_0\|-r)^2$. Now suppose we have $\{y_i\}_{i=1}^M$ such that each pair y_i and y_j is connected by a spring with constant k_{ij} and rest length r_{ij}. Then, the potential energy of this spring network is

$$U = C\sum_{i,j} k_{ij}(\|y_i - y_j\| - r_{ij})^2 \tag{3}$$

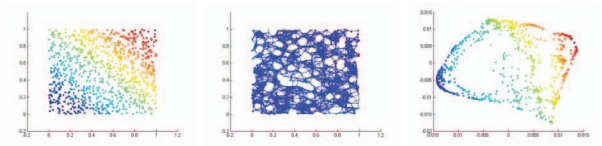
Comparing this with the expression for the DiffMaps objective (2), we see that DiffMaps is minimizing the potential energy of a network of springs. In this network, points y_i and y_j are connected by a spring with spring constant $k_{ij} = W_{ij} = k(x_i, x_j)$ with W_{ij} the DiffMaps weight, and rest length $r_{ij} = 0$, $\forall i,j$. Thus, for typical k, stronger springs connect points that were originally close in the high-dimensional space and weaker springs connect points that were originally far apart.

Since $F = \nabla U$, an arrangement of the points in which elastic energy is minimized is also one in which the net force on each point is zero, i.e. it is an equilibrium point of the system. Thus, we could physically interpret DiffMaps as attaching a spring between each pair of points in the high-dimensional space, using stronger springs for closer points and weaker springs for farther points (to reflect the greater importance we wish to assign to preserving local distances), then forcibly compressing the entire network of springs and points into a lower-dimensional space and allowing the points to settle according to the forces placed upon them by the springs. Since the springs have zero rest lengths, without additional constraint all points will settle to a common point. However, the DiffMaps constraint prevents this, forcing the points to spread out in all directions.

2.1. Spring Intuition View of Diffusion Maps' Problems

The spring interpretation gives insight into a number of observed problems typically found in DiffMaps embeddings. For example, consider a two-dimensional embedding of randomly sampled points from the unit square in $\mathbb{R}^2$. An ideal embedding should leave the points where they are. However, if we attach zero-rest-length springs to pairs of points that are close (as in Fig. 1) and allow the points to equilibrate according to the forces placed upon them, the resulting embedding clumps points in areas of greater spring density (i.e. of slightly greater random sampling density) while opening holes in areas of lesser spring density [8, 9]. See Fig. 1. Hence, we already see an explanation for distortions of linear subspaces. This matches the previous observation that the DiffMaps result, in the absence of an explicit sampling density correction, magnifies random variations in sampling density. The effectiveness of the sampling density correction scheme proposed in [8] will be examined empirically later.

Moreover, the result is highly sensitive to the parameter t of the kernel k_{heat}. For example, a slightly lower value of t might allow more spring connections across areas of lower sampling density, preventing holes in the embedding, while a slightly higher value might produce even more holes. More generally, relying entirely on the relative strengths of the springs to preserve the geometry of the points means that small variations in this delicate balance of spring weights can produce dramatically different results. Finally, at boundaries and corners, points only have neighbors in some directions and thus experience unbalanced forces. Hence, points along the boundaries are compressed and corners rounded.



(a) (b) (c)

Fig. 1. Demonstration of undesirable effects in the embedding produced by zero-rest-length springs. (a) Random samples of original unit square in $\mathbb{R}^2$. (b) Original unit square with inter-point springs drawn. (c) Diffusion maps embedding according to these springs.

A common thread of the above observations is that the zero rest length assumption is unnatural: we always pay a penalty in the DiffMaps criterion when we move points farther apart, regardless of how far apart they were in the original high-dimensional space. In contrast, we never pay a direct penalty for compressing distances between points. This criterion thus provides an incentive to unnecessarily distort distances.

3. EFFECT OF REST LENGTHS

Given the previous observations that the problems of Diffusion Maps are due to the zero rest length assumption, we might wonder what would occur if we instead took r_{ij} to be the original distance between x_i and x_j, resulting in the more natural objective function:

$$J_{\text{spring}}(y_1,\ldots,y_M) = \sum_{i,j} W_{ij}\left(\|y_i - y_j\| - \|x_i - x_j\|\right)^2$$

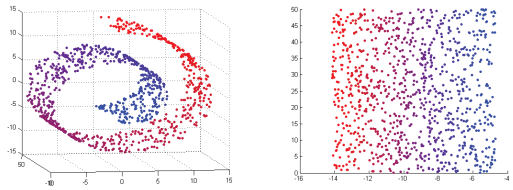
Interestingly, this criterion appears previously in the manifold learning literature, first in the early work of Chalmers and colleagues [10] who used a spring framework, and an iterative minimization approach based on computing forces and accelerations on individual points, to tackle the "graph layout" problem. It then appears in the thesis of Lisha Chen [11], who noted that this objective has an interpretation as a sort of localized multidimensional scaling.

In order to evaluate the impact of rest length, we directly compare the result of DiffMaps(zero rest lengths) with that of the spring criterion (same criterion with nonzero rest lenghts), both theoretically and experimentally. Details of the experimental optimization procedure will follow in Section 4. The results confirm our intuition that the zero rest lengths are to blame for the problems of DiffMaps.

3.1. Experimental Comparisons

We show embedding results for both criteria on two different datasets. First, as an example that can be easily visualized and for which we have clear ground truth, we consider the 2-dimensional Swiss roll in $\mathbb{R}^3$. We have randomly sampled 1,000 points from a uniform distribution on this manifold. Then we compared the embedding achieved by DiffMaps using the heat kernel, the "best" choice of user-specified t (found by exhaustive search), and the correction for non-uniform sampling density described in [8], against that achieved using the spring criterion with the same weights W_{ij}.

Fig. 2 shows the embeddings. The generation embedding in Fig. 2(b) represents ground truth. Fig. 3 gives a comparison of local



(a) Swiss roll in $\mathbb{R}^3$ (b) Generation embedding

(c) Diffusion maps embedding (d) Spring embedding

Fig. 2. Diffusion maps and spring embeddings of the Swiss roll

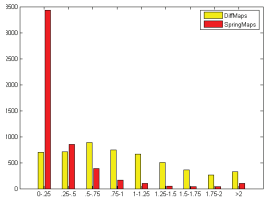
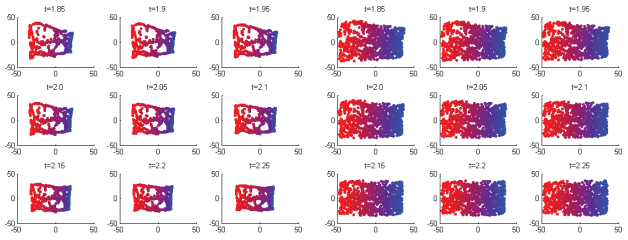
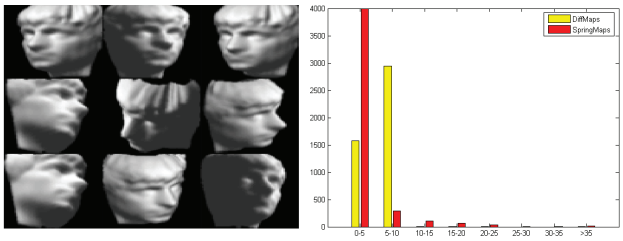


Fig. 3. Distance distortion histogram for Swiss roll embeddings. Each bar gives the number of pairs i, j for which $|\|y_i - y_j\| - \|x_i - x_j\||$ falls within the corresponding x-axis bin. (out of 5,186 pairs with $W_{ij} > 0.01$ and $i \neq j$.) For spring criterion, these distances fall overwhelmingly into lower distortion ranges.



(a) (b)

Fig. 4. Comparison of Parameter Sensitivity. Embeddings of Diffusion Maps (a) and spring criterion (b) for various t around the "best" value of 2.2. We observe less variation in the spring results.



(a) (b)

Fig. 5. (a) Nine sample face images from the test dataset, illustrating the three degrees of freedom of the dataset: (1) lighting angle, (2) horizontal and (3) vertical angle of pose. (b) Histogram of embedding distance distortions. Bars show the number of pairwise distance distortions within each bin. $(4,534$ pairs with $W_{ij} > 0.05)$.

distances distortion in each embedding and Fig. 4 illustrates the parameter sensitivity of the two algorithms. We analyze these results below.

As a more complex example, we examine a standard dataset [1], consisting of 698 64×64 images of a synthetic face. The faces have two pose and one angle of illumination degrees of freedom. Sample face images are shown in Fig. 5(a). Since this manifold is 3-dimensional in 4,096-dimensional space, visual comparison with ground truth is impossible. The distortion bar chart in Fig. 5(b) summarizes the performance gained using both the original DiffMaps and spring criteria.

3.2. Observations from Comparison of these Criteria

Preservation of Local Inter-point Distances. Under correct choice of nonzero rest lengths as $\|x_i - x_j\|$, preservation of *all* local interpoint distances is guaranteed whenever possible, regardless of parameter choice. If an embedding exists such that $\|y_i - y_j\| = \|x_i - x_j\|$ for all i, j with $W_{ij} > 0$ (i.e., all local pairwise distances are perfectly preserved), then $J_{\text{spring}} = 0$ for this embedding, and it is a global minimizer of J_{spring}, regardless of the particular values taken by the nonzero W_{ij}. However, if $\|y_i - y_j\| \neq \|x_i - x_j\|$ for any i, j with $W_{ij} > 0$, then $J_{\text{spring}} > 0$ for this embedding. A global minimizer of J_{spring} thus preserves all local pairwise distances (i.e. all that are assigned nonzero weight) if such a local-distance-preserving embedding exists. As noted previously, the zero rest length DiffMaps criterion rarely preserves all these distances.

Need for Additional Constraints. Since the DiffMaps criterion is optimized by collapsing all the points onto a single location (and improved by collapsing the points onto a lower-dimensional subspace), DiffMaps is forced to impose the additional constraint on the y_is to keep this from happening. Unfortunately, this constraint alone is enough to introduce unnecessary distortions, since points randomly sampled from a uniform distribution on a subset of a linear hyperplane will satisfy the DiffMaps constraint with probability zero. Also, it prevents the algorithm from discovering that the data can be effectively embedded in a space of dimension less than d. By contrast, J_{spring} suffers when the points are collapsed onto each other, so no such inconvenient constraints are needed.

Parameter Sensitivity. We noted above that a perfect embedding of the points, if it exists, will minimize J_{spring} regardless of the values of the weights W_{ij}, or, more explicitly, regardless of the parameter t's value. More generally, by incorporating a direct incentive to preserve local distances rather than indirectly counting on a delicate balance of the weights W_{ij} to do all the work, J_{spring} becomes more robust to small changes in the parameters that determine the weights. See Fig. 4.

Distortion of Boundaries. DiffMaps embeddings often show distortion of boundaries (especially corners) of manifolds. As noted previously, the spring interpretation predicts this. By contrast, in Section 3.1, we see that the criterion J_{spring} produces little distortion of boundaries and corners.

Robustness to Variations in Sampling Density. As observed, DiffMaps typically suffers in performance when the x_i have been sampled unevenly from the manifold and tends to magnify these variations. Such deviations from uniform density occur almost surely even in the case of random sampling from a uniform distribution on the manifold. As a posthoc fix, DiffMaps sometimes applies a correction to the weights W_{ij}, normalizing them by an estimate of the local sampling density [8]. In Section 3.1, we see it is more effective to incorporate the desired inter-point distances into the criterion directly (J_{spring}), rather than counting on the weights to indirectly influence them (J_{diff}).

Ease of Optimization While using nonzero rest lengths has a number of clear advantages, it also results in nonconvexity of the criterion. To see how this arises, examine the simple example of two points x_1, x_2, originally a distance $d_0 = \|x_1 - x_2\|$ apart, being embedded into $\mathbb{R}^1$. Fix y_1 at the origin and consider how $J_{\text{diff}}(y_1, y_2)$ and $J_{\text{spring}}(y_1, y_2)$ vary with y_2. J_{diff} is a simple quadratic and therefore convex, but J_{spring} is non-convex around the origin because the objective must necessarily increase as y_2 approaches y_1 in order to penalize solutions for which $\|y_1 - y_2\| < d_0$. Hence, non-convexity is a necessary effect of nonzero rest lengths.

4. COMBINING STRENGTHS OF THE TWO CRITERIA

We conclude from the observations above that nonzero rest lengths produce superior results with respect to a variety of criteria at the expense of non-convexity. However, given the connection between the easily computed DiffMaps solution and the more effective, but nonconvex spring criterion, one might ask if this connection can be used to produce a more advantageous method.

Defining the matrix inner product $<\cdot,\cdot>_W$ as $<A, B>_W = \sum_{ij} W_{ij} A_{ij} B_{ij}$, with $\|\cdot\|_W$ the corresponding norm, we note that

$$\begin{aligned} \min J_{\text{spring}} &= \min \|D(X) - D(Y)\|_W^2 \\ &= \min \|D(Y)\|_W^2 - 2<D(X), D(Y)>_W \\ &= \min J_{\text{diff}}(y_1, \ldots, y_M) - 2<D(X), D(Y)>_W \end{aligned}$$

where $D(X)_{ij} = \|x_i - x_j\|$, $D(Y)_{ij} = \|y_i - y_j\|$.

We notice then that if the new inner product term is large for the DiffMaps embedding, i.e. if the pairwise distances in the DiffMaps solution correlate well with those in the original high-dimensional space, then the DiffMaps solution will already be close to the spring criterion's global minimizer. Thus, we can exploit this connection between the globally optimizable DiffMaps criterion and the spring one, by employing gradient descent from the DiffMaps solution to quickly optimize the spring one. We note that this gradient is:

$$\frac{\partial}{\partial (y_k)_l} J_{\text{spring}} = 4 \sum_{j \neq k} W_{kj} \left(\frac{\|y_k - y_j\| - \|x_k - x_j\|}{\|y_k - y_j\|} \right) ((y_k)_l - (y_j)_l)$$

Indeed, this gradient descent strategy successfully alleviates problems in the DiffMaps output as demonstrated by the spring results obtained this way in Sec. 3.1.

4.1. Preventing Folding in the Embedding

As a final note, we will discuss one additional quality that we would like in an embedding that has not received much attention thus far. Consider a manifold formed by folding a sheet of paper along its midline and then slightly opening the fold to form the shape of a partially open file folder. The fold line naturally separates the manifold into two halves. The local pairwise distances between points on the manifold fall into two categories: (1) those lying entirely within one planar half of the manifold and (2) a smaller set of local pairwise distances that span the fold connecting the two planar sections of the manifold. Now, consider two candidate embeddings of this manifold into the two-dimensional plane: (a) we fold the manifold bringing the two halves together, and (b) we unfold the manifold and lay it flat. Both candidate embeddings preserve the local distances in category (1) perfectly. However, for distances in category (2): embedding (a) slightly compresses the local distances whereas embedding (b) greatly expands them. From the perspective of preserving local distances, embedding (a) is the better embedding, although we might in fact much prefer embedding (b), e.g. if our goal is an embedding that reflects the underlying parameterization of the manifold. DiffMaps is subject to similar concerns, although it is worth noting that the DiffMaps constraint creates an incentive to spread the points out, possibly at the cost of preserving local distances.

We can address the situation in which we would prefer embedding (b) by noting that a fold necessarily implies moving points that are far apart in the high-dimensional space closer together in the embedding. However, since geodesic distance is always at least as great as Euclidean distance, one should not have to compress distances in order to recover an embedding that reflects the underlying parameterization of the manifold. In this sense, compressing distances is less desirable than expanding them. This suggests that we impose an asymmetrical penalty on the distortion of distance, penalizing distance compression more than expansion. We incorporate"one-way" springs: springs that exert force only when compressed. The result is the same objective and gradient expressions but with the old weight W_{ij} replaced with the new weight $\mathbb{W}_{ij} = W_{ij} + V_{ij} I(\|x_i - x_j\| - \|y_i - y_j\|)$. $I(x)$ is the indicator function (1 when $x > 0$, 0 otherwise).

5. CONCLUSIONS

We have explored an alternate, more physical, interpretation of the DiffMaps criterion, which gives additional insight into a number of observed problems with DiffMaps embeddings. Through this interpretation, we have been able to trace these deficiencies back to the zero rest length assumption and have demonstrated that fixing this alleviates the observed problems: perfectly preserving all local inter-point distances exactly when possible to do so and eliminating unnecessary and awkward constraints on the solution. However, fixing these problems comes at the cost of a nonconvex criterion; we have demonstrated that this a natural effect of penalizing distance compression. We have then shown how we can efficiently find minima of this alternate criterion by exploiting its relationship with the original DiffMaps solution and presented experimental results which show that minimizing the alternate criterion using DiffMaps for initialization, produces results that better preserve local distances, are more robust to variations in sampling density, better preserve boundaries, and are less sensitive to a user's choice of parameters.

6. REFERENCES

[1] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 260, pp. 2319–23, 2000.

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

[3] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[4] D. L. Donoho and C. Grimes, "Hessian eigenmaps: New locally linear embedding techniques for high-dimensional data," *PNAS*, vol. 100, pp. 5591–6, 2003.

[5] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *SIAM Jour. on Sci. Computing*, vol. 26, no. 1, pp. 313–38, 2005.

[6] K. Q. Weinberger and L. K. Saul, "Unsupervised learning of image manifolds by semidefinite programming," *Int. J. Comput. Vision*, vol. 70, no. 1, pp. 77–90, 2006.

[7] B. Scholkopf, A. Smola, and K. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1996.

[8] R. R. Coifman and S. Lafon, "Diffusion maps," *Applied and Computational Harmonic Analysis*, vol. 21, pp. 5–30, 2006.

[9] S. Lafon, *Diffusion Maps and Geometric Harmonics*, PhD Thesis Yale University, 2004.

[10] A. Morrison, G. Ross, and M. Chalmers, "Fast multidimensional scaling through sampling, springs and interpolation," *Information Visualization*, vol. 2, no. 1, pp. 68–77, 2003.

[11] L. Chen, *Local Multidimensional Scaling for Nonlinear Dimension Reduction, Graph Layout, and Proximity Analysis*, PhD Thesis, University of Pennsylvania, 2006.