Spectral multidimensional scaling

Yonathan Aflalo[a,1] and Ron Kimmel[b]

Departments of [a]Electrical Engineering and [b]Computer Science, Technion–Israel Institute of Technology, Haifa 32000, Israel

Edited* by Haim Brezis, Rutgers, The State University of New Jersey, Piscataway, NJ, and approved September 10, 2013 (received for review May 8, 2013)

An important tool in information analysis is dimensionality reduction. There are various approaches for large data simplification by scaling its dimensions down that play a significant role in recognition and classification tasks. The efficiency of dimension reduction tools is measured in terms of memory and computational complexity, which are usually a function of the number of the given data points. Sparse local operators that involve substantially less than quadratic complexity at one end, and faithful multiscale models with quadratic cost at the other end, make the design of dimension reduction procedure a delicate balance between modeling accuracy and efficiency. Here, we combine the benefits of both and propose a low-dimensional multiscale modeling of the data, at a modest computational cost. The idea is to project the classical multidimensional scaling problem into the data spectral domain extracted from its Laplace–Beltrami operator. There, embedding into a small dimensional Euclidean space is accomplished while optimizing for a small number of coefficients. We provide a theoretical support and demonstrate that working in the natural eigenspace of the data, one could reduce the process complexity while maintaining the model fidelity. As examples, we efficiently canonize nonrigid shapes by embedding their intrinsic metric into $\mathbb{R}^3$, a method often used for matching and classifying almost isometric articulated objects. Finally, we demonstrate the method by exposing the style in which handwritten digits appear in a large collection of images. We also visualize clustering of digits by treating images as feature points that we map to a plane.

flat embedding | distance maps | big data | diffusion geometry

Manifold learning refers to the process of mapping given data into a simple low-dimensional domain that reveals properties of the data. When the target space is Euclidean, the procedure is also known as flattening. The question of how to efficiently and reliably flatten given data is a challenge that occupies the minds of numerous researchers. Distance-preserving data-flattening procedures can be found in the fields of geometry processing, the mapmaker problem through exploration of biological surfaces (1–3), texture mapping in computer graphics (4, 5), nonrigid shape analysis (6), image (7–9) and video understanding (7, 10), and computational biometry (11), to name just a few. The flat embedding is usually a simplification process that aims to preserve, as much as possible, distances between data points in the original space, while being efficient to compute. One family of flattening techniques is multidimensional scaling (MDS), which attempts to map all pairwise distances between data points into small dimensional Euclidean domains. Review of MDS applications in psychophysics can be found in ref. 12, which includes the computational realization that human color perception is 2D.

A proper way to explore the geometry of given data points involves computing all pairwise distances. Then, a flattening procedure should attempt to keep the distance between all couples of corresponding points in the low dimensional Euclidean domain. Computing pairwise distances between points was addressed by extension of local distances on the data graph (1, 13), or by consistent numerical techniques for distance computation on surfaces (14). The complexity involved in storing all pairwise distances is quadratic in the number of data points, which is a limiting factor in large databases. Alternative models try to keep the size of the input as low as possible by limiting the input to distances of just nearby points.

One such example is locally linear embedding (15), which attempts to map data points into a flat domain where each feature coordinate is a linear combination of the coordinates of its neighbors. The minimization, in this case, is for keeping the combination similar in the given data and the target a flat domain. Because only local distances are analyzed, the pairwise distances matrix is sparse, with an effective size of $O(n)$, where n is the number of data points. Along the same line, the Hessian locally linear embedding (16) tries to better fit the local geometry of the data to the plane. Belkin and Niyogi (17) suggested embedding data points into the Laplace–Beltrami eigenspace for the purpose of data clustering. The idea is, yet again, to use distances of only nearby points by which a Laplace–Beltrami operator (LBO) is defined. The data can then be projected onto the first eigenvectors that correspond to the smallest eigenvalues of the LBO. The question of how to exploit the LBO decomposition to construct diffusion geometry was addressed by Coifman and coworkers (18, 19).

De Silva and Tenenbaum (20) recognized the computational difficulty of dealing with a full matrix of pairwise distances, and proposed working with a subset of landmarks, which is used for interpolation in the embedded space. Next, Bengio et al. (21) proposed to extend subsets by Nystrøm extrapolation, within a space that is an empirical Hilbert space of functions obtained through kernels that represent probability distributions; they did not incorporate the geometry of the original data. Finally, assuming n data points, Asano et al. (22) proposed reducing both memory and computational complexity by subdividing the problem into $O(\sqrt{n})$ subsets of $O(\sqrt{n})$ data points in each. That reduction may be feasible when distances between data points can be computed in constant time, which is seldom the case.

The computational complexity of multidimensional scaling was addressed by a multigrid approach in ref. 23, and vector extrapolation techniques in ref. 24. In both cases the acceleration, although effective, required all pairwise distances, an $O(n^2)$ input. An alternative approach to deal with the memory complexity was needed.

Significance

Scientific theories describe observations by equations with a small number of parameters or dimensions. Memory and computational efficiency of dimension reduction procedures is a function of the size of the observed data. Sparse local operators that involve almost linear complexity, and faithful multiscale models with quadratic cost, make the design of dimension reduction tools a delicate balance between modeling accuracy and efficiency. Here, we propose multiscale modeling of the data at a modest computational cost. We project the classical multidimensional scaling problem into the data spectral domain. Then, embedding into a low-dimensional space is efficiently accomplished. Theoretical support and empirical evidence demonstrate that working in the natural eigenspace of the data, one could reduce the complexity while maintaining model fidelity.

Author contributions: Y.A. and R.K. designed research and wrote the paper.

The authors declare no conflict of interest.

*This Direct Submission article had a prearranged editor.

Freely available online through the PNAS open access option.

[1]To whom correspondence should be addressed. E-mail: yaflalo@cs.technion.ac.il.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1308708110/-/DCSupplemental.

In ref. 25, geodesics that are suspected to distort the embedding due to topology effects were filtered out in an attempt to reduce distortions. Such filters eliminate some of the pairwise distances, yet not to the extent of substantial computational or memory savings, because the goal was mainly reduction of flattening distortions. In ref. 26, the eigenfunctions of the LBO on a surface were interpolated into the volume bounded by the surface. This procedure was designed to overcome the need to evaluate the LBO inside the volume, as proposed in ref. 27. Both models deal with either surface or volume LBO decomposition of objects in $\mathbb{R}^3$, and were not designed to flatten and simplify structures in big data.

Another dimensionality reduction method, intimately related to MDS, is the principal component analysis method (PCA). The PCA procedure projects the points to a low-dimensional space by minimizing the least-square fitting error. In ref. 28, that relation was investigated through kernel PCA, which is related to the problem we plan to explore.

Here, we use the fact that the gradient magnitude of distance functions on the manifold is equal to 1 almost everywhere. Interpolation of such smooth functions could be efficiently obtained by projecting the distances into the LBO eigenspace. In fact, translating the problem into its natural spectral domain enables us to reduce the complexity of the flattening procedure. The efficiency is established by considering a small fraction of the pairwise distances that are projected onto the LBO leading eigenfunctions.

Roughly speaking, the differential structure of the manifold is captured by the eigenbasis of the LBO, whereas its multiscale structure is encapsulated by sparse sampling of the pairwise distances matrix. We exemplify this idea by extracting the $\mathbb{R}^4$ structure of points in $\mathbb{R}^{10,000}$, canonizing surfaces to cope with nonrigid deformations, and mapping images of handwritten digits to the plane.

The proposed framework operates on abstract data in any dimension. In the first section, we prove the asymptotic behavior of using the leading eigenvectors of the LBO for representing smooth functions on a given manifold. Next, we demonstrate how to interpolate geodesic distances by projecting a fraction of the distances between data points into the LBO eigenspace. We next formulate the classical MDS problem in the LBO eigenspace. A relation to diffusion geometry (18, 19, 29) is established. Experimental results demonstrate the complexity savings.

Spectral Projection

Let us consider a Riemannian manifold, $\mathcal{M}$, equipped with a metric G. The manifold and its metric induce a LBO, often denoted by Δ_G, and here w.l.o.g. by Δ. The LBO is self-adjoint and defines a set of functions called eigenfunctions, denoted ϕ_i, such that $\Delta\phi_i = \lambda_i\phi_i$ and $\int_{\mathcal{M}}\phi_i(x)\phi_j(x)da(x) = \delta_{ij}$, where da is an area element. These functions have been popular in the field of shape analysis; they have been used to construct descriptors (30) and linearly relate between metric spaces (31). In particular, for problems involving functions defined on a triangulated surface, when the number of vertices of $\mathcal{M}$ is large, we can reduce the dimensionality of the functional space by considering smooth functions defined in a subspace spanned by just a couple of eigenfunctions of the associated LBO. To prove the efficiency of LBO eigenfunctions in representing smooth functions, consider a smooth function f with bounded gradient magnitude $\|\nabla_G f\|_G$. Define the representation residual function $r_n = f - \sum_{i=1}^{n}\langle f,\phi_i\rangle_G\phi_i$. It is easy to see that $\forall i, 1\le i\le n, \langle r_n,\phi_i\rangle_G = 0$, and $\forall i, 1\le i\le n, \langle\nabla_G r_n, \nabla_G\phi_i\rangle_G = \langle r_n, \Delta_G\phi_i\rangle_G = \lambda_i\langle r_n,\phi_i\rangle_G = 0$. Using these properties and assuming the eigenvalues are ordered in ascending order, $\lambda_1\le\lambda_2\ldots$, we readily have that $\|r_n\|_G^2 = \left\|\sum_{i=n+1}^{\infty}\langle r_n,\phi_i\rangle_G\phi_i\right\|_G^2 = \sum_{i=n+1}^{\infty}\langle r_n,\phi_i\rangle_G^2$ and $\|\nabla_G r_n\|_G^2 = \left\|\sum_{i=n+1}^{\infty}\langle r_n,\phi_i\rangle_G\nabla_G\phi_i\right\|_G^2 \ge \lambda_{n+1}\sum_{i=n+1}^{\infty}\langle r_n,\phi_i\rangle_G^2$. Then, $\frac{\|\nabla_G r_n\|_G^2}{\|r_n\|_G^2} \ge \lambda_{n+1}$. Moreover, as for $i\in\{1,..,n\}$, the inner product vanishes, $\langle\nabla_G r_n, \nabla_G\phi_i\rangle_G = 0$, we have $\|\nabla_G f\|_G^2 = \left\|\nabla_G r_n + \sum_{i=1}^{n}\langle f,\phi_i\rangle_G\nabla_G\phi_i\right\|_G^2 = \|\nabla_G r_n\|_G^2 + \sum_{i=1}^{n}\langle f,\phi_i\rangle_G^2\lambda_i$. It follows that $\|r_n\|_G^2 \le \frac{\|\nabla_G r_n\|_G^2}{\lambda_{n+1}} \le \frac{\|\nabla_G f\|_G^2}{\lambda_{n+1}}$.

Finally, for d dimensional manifolds, as shown in ref. 32, the spectra has a linear behavior in $n^{\frac{2}{d}}$, that is, $\lambda_n \approx C_1 n^{\frac{2}{d}}$ as $n\to\infty$. Moreover, the residual function r_n depends linearly on $\|\nabla_G f\|$, which is bounded by a constant. It follows that r_n converges asymptotically to zero at a rate of $O\left(n^{-\frac{2}{d}}\right)$. Moreover, this convergence depends on $\|\nabla_G f\|_G^2$, i.e., $\exists C_2$, such that

$$\forall f : \mathcal{M}\to\mathcal{M}\ \|r_n\|_G^2 \le \frac{C_2\|\nabla_G f\|_G^2}{n^{\frac{2}{d}}}. \qquad [1]$$

Spectral Interpolation

Let us consider a d dimensional manifold, $\mathcal{M}$, sampled at n points $\{V_i\}$, and $\mathcal{J}$ a subset of $\{1,2,\ldots,n\}$ such that $|\mathcal{J}| = m_s \le n$. Given a smooth function f defined on $V_{\mathcal{J}} = \{V_j, j\in\mathcal{J}\}$, our first question is how to interpolate f, that is, how to construct a continuous function $\tilde{f}$ such that $\tilde{f}(V_j) = f(V_j), \forall j\in\mathcal{J}$. One simple solution is linear interpolation. Next, assume the function $\tilde{f}$ is required to be as smooth as possible in L_2 sense. We measure the smoothness of a function by $E_{\text{smooth}}(f) = \int_{\mathcal{M}}\|\nabla f\|_2^2 da$. The problem of smooth interpolation could be rewritten as $\min_{h:\mathcal{M}\to\mathbb{R}} E_{\text{smooth}}(h)$ s.t. $\forall j\in\mathcal{J}, h(V_j) = f(V_j)$. We readily have $\int_{\mathcal{M}}\|\nabla h\|_2^2 da = \int_{\mathcal{M}}\langle\Delta h, h\rangle da$. Our interpolation problem could then be written as

$$\min_{h:\mathcal{M}\to\mathbb{R}}\int_{\mathcal{M}}\langle\Delta h, h\rangle da \quad \text{s.t.} \quad h(V_j) = f(V_j) \quad \forall j\in\mathcal{J}. \qquad [2]$$

Any discretization matrix of the LBO could be used for the eigenspace construction. Here, we adopt the LBO general $\mathbf{L} = \mathbf{A}^{-1}\mathbf{W}$ form, where $\mathbf{A}^{-1}$ is a diagonal matrix whose entries are inversely proportional to the metric infinitesimal volume elements. One example is the cotangent weights approximation (33) for triangulated surfaces; another example is the generalized discrete LBO suggested in ref. 17. In the first case, $\mathbf{W}$ is a matrix in which each entry is zero if the two vertices indicated by the two matrix indices do not share a triangle's edge, or the sum of the cotangents of the angles supported by the edge connecting the corresponding vertices. In the latter case, $\mathbf{W}$ is the graph Laplacian, where the graph is constructed by connecting nearest neighbors. Like any Laplacian matrix, the diagonal is the negative sum of the off-diagonal elements of the corresponding row. $\mathbf{A}$, in the triangulated surface case, is a diagonal matrix in which the $\mathbf{A}_{ii}$ th element is proportional to the area of the triangles about the vertex V_i. A similar normalization factor should apply in the more general case.

A discrete version of the smooth interpolation problem 2 can be rewritten as

$$\min_x x^T\mathbf{W}x \quad \text{s.t.} \quad \mathbf{B}x = f, \qquad [3]$$

where $\mathbf{B}$ is the projection matrix on the space spanned by the vectors $e_j, j\in\mathcal{J}$, where e_j is the j^{th} canonical basis vector, and f now represents the sampled vector $f(V_{\mathcal{J}})$.

To reduce the dimensionality of the problem, we introduce $\hat{f}$, the spectral projection of f to the set of eigenfunctions of the LBO $\{\phi_i\}_{i=1}^{m_e}$ as $\hat{f} = \sum_{i=1}^{m_e}\langle f,\phi_i\rangle\phi_i$, where $\Delta\phi_i = \lambda_i\phi_i$. Then, by denoting $\mathbf{\Phi}$ the matrix whose i^{th} column is ϕ_i, we have $\hat{f} = \mathbf{\Phi}\alpha$, where α is a vector such that $\alpha_i = \langle f,\phi_i\rangle$. Problem 3 can now be approximated by

$$\min_{\alpha\in\mathbb{R}^{m_e}} \alpha^T\mathbf{\Phi}^T\mathbf{W}\mathbf{\Phi}\alpha \quad \text{s.t.} \quad \mathbf{B}\mathbf{\Phi}\alpha = f. \qquad [4]$$

By definition, $\mathbf{\Phi}^T\mathbf{W}\mathbf{\Phi} = \mathbf{\Lambda}$, where $\mathbf{\Lambda}$ represents the diagonal matrix whose elements are the eigenvalues of $\mathbf{L}$. We can alternatively incorporate the constraint as a penalty in the target function, and rewrite our problem as $\min_{\alpha\in\mathbb{R}^{m_e}} \left(\alpha^T\mathbf{\Lambda}\alpha + \mu\|\mathbf{B}\mathbf{\Phi}\alpha - f\|^2\right) = \min_{\alpha\in\mathbb{R}^{m_e}} (\alpha^T(\mathbf{\Lambda} + \mu\mathbf{\Phi}^T\mathbf{B}^T\mathbf{B}\mathbf{\Phi})\alpha + 2\mu f^T\mathbf{B}\mathbf{\Phi}\alpha)$. The solution to this problem is given by

$$\alpha = 2\mu\left(\mathbf{\Lambda} + \mu\mathbf{\Phi}^T\mathbf{B}^T\mathbf{B}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^T\mathbf{B}^T f = \mathbf{M}f. \qquad [5]$$

Next, we use the above interpolation expressions to formulate the pairwise geodesics matrix in a compact yet accurate manner. The smooth interpolation, problem 2, for a pairwise distances function can be defined as follows. In this setting, let $\mathcal{I} = \mathcal{J}\times\mathcal{J}$ be the set of pairs of indices of data points, and $F(V_i, V_j)$ a value defined for each pair of points (or vertices in the surface example) (V_i, V_j), where $(i,j)\in\mathcal{I}$. We would like to interpolate the smooth function $D : \mathcal{M}\times\mathcal{M}\to\mathbb{R}$, whose values are given at $(V_i, V_j), (i,j)\in\mathcal{I}$, by $D(V_i, V_j) = F(V_i, V_j), \forall(i,j)\in\mathcal{I}$. For that goal, we first define a smooth-energy measure for such functions. We proceed as before by introducing

$$E_{\text{smooth}}(D) = \iint_{\mathcal{M}} \|\nabla_x D(x,y)\|^2 + \left\|\nabla_y D(x,y)\right\|^2 da(x)da(y).$$

The smooth interpolation problem can now be written as

$$\min_{D:\mathcal{M}\times\mathcal{M}\to\mathbb{R}} E_{\text{smooth}}(D) \quad \text{s.t.} \quad D(V_i, V_j) = F(V_i, V_j), \forall(i,j)\in\mathcal{I}. \qquad [6]$$

Given any matrix $\mathbf{D}$, such that $\mathbf{D}_{ij} = D(V_i, V_j)$, we have

$$\int_{\mathcal{M}} \left\|\nabla_x D(x, y_j)\right\|^2 da(x) = \int_{\mathcal{M}} \langle \Delta_x D(x, y_j), D(x, y_j)\rangle da(x) \approx \mathbf{D}_j^T\mathbf{W}\mathbf{D}_j = \text{trace}\left(\mathbf{W}\mathbf{D}_j\mathbf{D}_j^T\right),$$

where $\mathbf{D}_j^T$ represents the j^{th} column of $\mathbf{D}$. Then,

$$\iint_{\mathcal{M}} \|\nabla_x D(x,y)\|^2 da(x)da(y) \approx \sum_j \mathbf{A}_{jj}\ \text{trace}\left(\mathbf{W}\mathbf{D}_j\mathbf{D}_j^T\right) = \text{trace}\left(\mathbf{W}\sum_j \mathbf{D}_j\mathbf{D}_j^T\mathbf{A}_{jj}\right) = \text{trace}\left(\mathbf{W}\mathbf{D}\mathbf{A}\mathbf{D}^T\right).$$

A similar result applies to $\iint_{\mathcal{M}} \left\|\nabla_y D(x,y)\right\|^2 da(x)da(y)$. Then,

$$\iint_{\mathcal{M}} \|\nabla_x D(x,y)\|^2 da(x)da(y) \approx \text{trace}\left(\mathbf{D}^T\mathbf{W}\mathbf{D}\mathbf{A}\right)$$
$$\iint_{\mathcal{M}} \left\|\nabla_y D(x,y)\right\|^2 da(x)da(y) \approx \text{trace}\left(\mathbf{D}\mathbf{W}\mathbf{D}^T\mathbf{A}\right). \qquad [7]$$

The smooth energy can now be discretized for a matrix $\mathbf{D}$ by

$$E_{\text{smooth}}(\mathbf{D}) = \text{trace}\left(\mathbf{D}^T\mathbf{W}\mathbf{D}\mathbf{A}\right) + \text{trace}\left(\mathbf{D}\mathbf{W}\mathbf{D}^T\mathbf{A}\right). \qquad [8]$$

The spectral projection of D onto $\mathbf{\Phi}$, is given by

$$\tilde{D}(x,y) = \sum_i \langle D(\cdot, y), \phi_i\rangle\phi_i(x) = \sum_{i=1}^n\sum_{j=1}^n \underbrace{\left\langle \langle D(u,v), \phi_i(u)\rangle, \phi_j(v)\right\rangle}_{\boldsymbol{\alpha}_{ij}} \phi_j(y)\phi_i(x),$$

where $\boldsymbol{\alpha}_{ij} = \iint_{\mathcal{M}\times\mathcal{M}} D(x,y)\phi_i(x)\phi_j(y)da(x)da(y)$. In matrix notations, denoting by $\tilde{\mathbf{D}}$ the matrix such that $\tilde{\mathbf{D}}_{ij} = \tilde{D}(x_i, y_j)$, we can show that

$$\tilde{\mathbf{D}} = \mathbf{\Phi}\boldsymbol{\alpha}\mathbf{\Phi}^T. \qquad [9]$$

Combining Eqs. **8** and **9**, we can define the discrete smooth energy of the spectral projection of a function as

$$\begin{aligned}\tilde{E}_{\text{smooth}}(\tilde{\mathbf{D}}) &= \text{trace}\left(\mathbf{\Phi}\boldsymbol{\alpha}^T\mathbf{\Phi}^T\mathbf{W}\mathbf{\Phi}\boldsymbol{\alpha}\mathbf{\Phi}^T\mathbf{A}\right) + \text{trace}\left(\mathbf{\Phi}\boldsymbol{\alpha}\mathbf{\Phi}^T\mathbf{W}\mathbf{\Phi}\boldsymbol{\alpha}^T\mathbf{\Phi}^T\mathbf{A}\right)\\ &= \text{trace}\left(\mathbf{\Phi}\boldsymbol{\alpha}^T\mathbf{\Lambda}\boldsymbol{\alpha}\mathbf{\Phi}^T\mathbf{A}\right) + \text{trace}\left(\mathbf{\Phi}\boldsymbol{\alpha}\mathbf{\Lambda}\boldsymbol{\alpha}^T\mathbf{\Phi}^T\mathbf{A}\right)\\ &= \text{trace}\left(\boldsymbol{\alpha}^T\mathbf{\Lambda}\boldsymbol{\alpha}\mathbf{\Phi}^T\mathbf{A}\mathbf{\Phi}\right) + \text{trace}\left(\boldsymbol{\alpha}\mathbf{\Lambda}\boldsymbol{\alpha}^T\mathbf{\Phi}^T\mathbf{A}\mathbf{\Phi}\right)\\ &= \text{trace}\left(\boldsymbol{\alpha}^T\mathbf{\Lambda}\boldsymbol{\alpha}\right) + \text{trace}\left(\boldsymbol{\alpha}\mathbf{\Lambda}\boldsymbol{\alpha}^T\right). \qquad [10]\end{aligned}$$

Using the spectral smooth representation introduced in Eq. **9**, we can define the smooth spectral interpolation problem for a function from $\mathcal{M}\times\mathcal{M}$ to $\mathbb{R}$ as

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^{m_e\times m_e}} \text{trace}\left(\boldsymbol{\alpha}^T\mathbf{\Lambda}\boldsymbol{\alpha}\right) + \text{trace}\left(\boldsymbol{\alpha}\mathbf{\Lambda}\boldsymbol{\alpha}^T\right) \quad \text{s.t.} \quad \left(\mathbf{\Phi}\boldsymbol{\alpha}\mathbf{\Phi}^T\right)_{ij} = D(V_i, V_j), \quad \forall(i,j)\in\mathcal{I}. \qquad [11]$$

Expressing the constraint again as a penalty function, we end up with the following optimization problem

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^{m_e\times m_e}} \text{trace}\left(\boldsymbol{\alpha}^T\mathbf{\Lambda}\boldsymbol{\alpha}\right) + \text{trace}\left(\boldsymbol{\alpha}\mathbf{\Lambda}\boldsymbol{\alpha}^T\right) + \mu\sum_{(i,j)\in\mathcal{I}} \left\|\left(\mathbf{\Phi}\boldsymbol{\alpha}\mathbf{\Phi}^T\right)_{ij} - D(V_i, V_j)\right\|_F^2, \qquad [12]$$

where $\|\cdot\|_F$ represents the Frobenius norm, and m_e the number of eigenfunctions. Problem 12 is a minimization problem of a quadratic function of $\boldsymbol{\alpha}$. Then, representing $\boldsymbol{\alpha}$ as an $(m_e^2\times 1)$ vector α, the problem can be rewritten as a quadratic programming problem. We can find a matrix $\mathbf{M}$, similar to Eq. **5**, such that $\alpha = \mathbf{M}D$, where D is the row stack vector of the matrix $D(V_i, V_j)$.

Another, less accurate but more efficient, approach to obtain an approximation of the matrix $\boldsymbol{\alpha}$ is to compute

$$\boldsymbol{\alpha} = \mathbf{M}\mathbf{F}\mathbf{M}^T, \qquad [13]$$

where $\mathbf{F}$ is the matrix defined by $\mathbf{F}_{ij} = F(V_i, V_j)$ and $\mathbf{M}$ is the matrix introduced in Eq. **5**.

Notice that spectral projection is a natural choice for the spectral interpolation of distances because the eigenfunctions encode the manifold geometry, as do distance functions. Moreover, the eikonal equation, which models geodesic distance functions on the manifold, is defined by $\|\nabla_G D\| = 1$. Eq. **1**, in this case, provides us with a clear asymptotic convergence rate by spectral projection of the function D, because $\|\nabla_G D\|$ is a constant equal to 1.

At this point, we have all of the ingredients to present the spectral MDS. For simplicity, we limit our discussion to classical scaling.

Spectral MDS

MDS is a family of dimensionality reduction techniques that attempt to find a simple representation for a dataset given by the distances between every two data points. Given any metric space $\mathcal{M}$ equipped with a metric $D : \mathcal{M}\times\mathcal{M}\to\mathbb{R}$, and $\mathcal{V} = \{V_1, V_2, \ldots V_n\}$ a finite set of elements of $\mathcal{M}$, the multidimensional scaling of $\mathcal{V}$ in $\mathbb{R}^k$ involves finding a set of point $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$ in $\mathbb{R}^k$ whose pairwise Euclidean distances $d(X_i, X_j) = \left\|X_i - X_j\right\|^2$ are as close as possible to $D(V_i, V_j)$ for all (i,j). Such an embedding, for the MDS

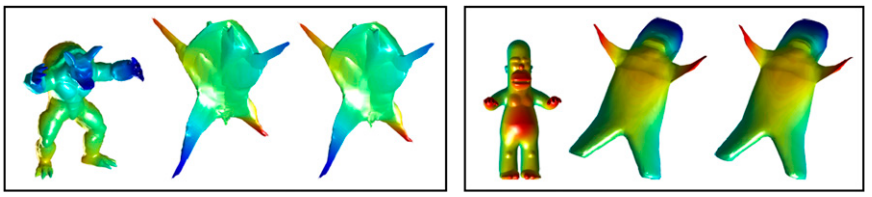

Fig. 1. Canonical forms of Armadillo (*Left*) and Homer (*Right*). Within each box is (from left to right) the shape, followed by canonical forms obtained by regular MDS, and spectral MDS.

member known as classical scaling, can be realized by the following minimization program $\min_{\mathbf{X}} \left\|\mathbf{X}^T\mathbf{X}-\frac{1}{2}\mathbf{JDJ}\right\|_F$, where $\mathbf{D}$ is a matrix defined by $\mathbf{D}_{ij}=D(V_i,V_j)^2$ and $\mathbf{J}=\mathbf{I}_n-\frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T$, or $\mathbf{J}_{ij}=\delta_{ij}-1/n$. Classical scaling (Eq. **12**) finds the first k singular vectors and corresponding singular values of the matrix $-\frac{1}{2}\mathbf{JDJ}$. The classical scaling solver requires the computation of a $(n\times n)$ matrix of distances, which is a challenging task when dealing with more than, say, thousands of data points. The quadratic size of the input data imposes severe time and space limitations. Several acceleration methods for MDS were tested over the years (20–24). However, to the best of our knowledge, the multiscale geometry of the data was never used for lowering the space and time complexities during the reduction process. Here, we show how spectral interpolation, in the case of classical scaling, is used to overcome these complexity limitations.

In the spectral MDS framework we first select a subset of m_s points, with indices $\mathcal{J}$ of the data. For efficient covering of the data manifold, this set can be selected using the farthest-point sampling strategy, which is known to be 2-optimal in the sense of covering. We then compute the geodesic distances between every two points $(V_i,V_j)\in\mathcal{M}\times\mathcal{M}, (i,j)\in\mathcal{I}=\mathcal{J}\times\mathcal{J}$.

Now, we are ready to choose any Laplacian operator that is constructed from the local relations between data points. The LBO takes into consideration the differential geometry of the data manifold. We compute Laplacian's first m_e eigenvectors that are arranged in an eigenbasis matrix denoted by $\mathbf{\Phi}$. Finally, we extract the spectral interpolation matrix $\boldsymbol{\alpha}$ from the computed geodesic distances and the eigenbasis $\mathbf{\Phi}$ using Eqs. **12** or **13**.

The interpolated matrix distance $\tilde{\mathbf{D}}$ between every two points of $\mathcal{M}$ can be computed by $\tilde{\mathbf{D}}=\mathbf{\Phi}\boldsymbol{\alpha}\mathbf{\Phi}^T$. It is important to note that there is no need to compute this matrix explicitly.

At this point let us justify our choice of representation. Denote by $D_y:\mathcal{M}\to\mathbb{R}^+$ the geodesic distance function from a source point y to the rest of the point in the manifold. Geodesic distance functions are characterized by the eikonal equation $\left\|\nabla_G D_y\right\|^2=1$. This property can be used in the bound provided by Eq. **1**. Specifically, $\left\|\nabla_G D_y^2\right\|^2=\left\|2D_y\nabla_G D_y\right\|^2\le 4\left\|D_y\right\|^2$. Plugging this relation to Eq. **1**, the error of the squared geodesic distance function projected to the spectral basis is given by $\frac{\|r_n\|_G^2}{\|D\|_G^2}\le\frac{4C_2}{n^{\frac{2}{d}}}$, where $\|D\|_G$ is the diameter of the manifold. We thereby obtained a bound on the relative projection error that depends only on Weyl's constant and the number of eigenvectors used in the reconstruction by projection.

Next, the spectral MDS solution is given by $\mathbf{X}\mathbf{X}^T=-\frac{1}{2}\tilde{\mathbf{D}}\mathbf{J}$. This decomposition can be approximated by two alternative methods. The first method considers the projection of $\mathbf{X}$ to the eigenbasis $\mathbf{\Phi}$, which is equivalent to representing $\mathbf{X}$ by $\tilde{\mathbf{X}}=\mathbf{\Phi}\boldsymbol{\beta}$. To that end, we have to solve $\mathbf{\Phi}\boldsymbol{\beta}\boldsymbol{\beta}^T\mathbf{\Phi}^T=-\frac{1}{2}\mathbf{J}\mathbf{\Phi}\boldsymbol{\alpha}\mathbf{\Phi}^T\mathbf{J}$. Because $\mathbf{\Phi}^T\mathbf{A}\mathbf{\Phi}=\mathbf{I}_{m_e}$, we have

$$\boldsymbol{\beta}\boldsymbol{\beta}^T=-\frac{1}{2}\underbrace{\mathbf{\Phi}^T\mathbf{A}\mathbf{J}\mathbf{\Phi}}_{\mathbf{H}}\boldsymbol{\alpha}\mathbf{\Phi}^T\mathbf{J}\mathbf{A}\mathbf{\Phi}=-\frac{1}{2}\mathbf{H}\boldsymbol{\alpha}\mathbf{H}^T, \qquad [14]$$

which leads to the singular value decomposition of the $(m_e\times m_e)$ matrix $-\frac{1}{2}\mathbf{H}\boldsymbol{\alpha}\mathbf{H}^T$.

In the second method, to overcome potential distortions caused by ignoring the high-frequency components, one may prefer to decompose the matrix $\mathbf{\Phi}\boldsymbol{\alpha}\mathbf{\Phi}^T$ using the algebraic trick presented in algorithm 1.

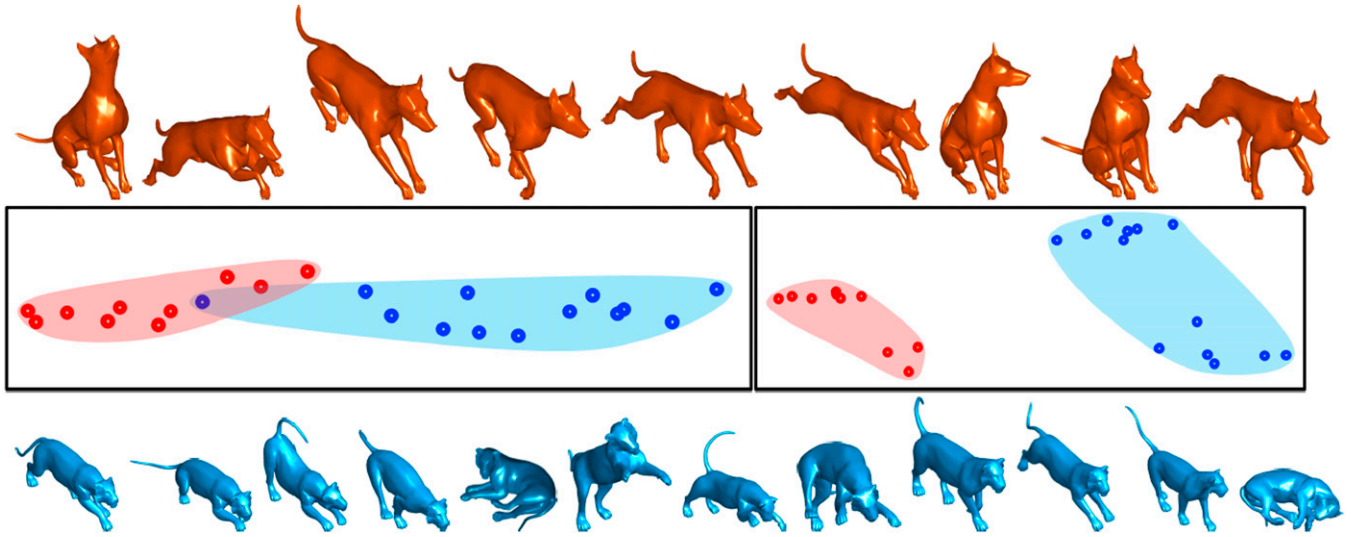

Fig. 2. Spectral MDS compared with MDS with the same number of sampled points (same complexity). Each point in the plane on the central frames represents a shape. The input to the MDS producing the position of the points was distances between the corresponding canonical forms (MDS applied to the result of MDS). Red points represent dogs, and blue points represent lionesses. The spectral MDS result (*Right*) provides better clustering and separation between classes compared with the regular MDS (*Left*).

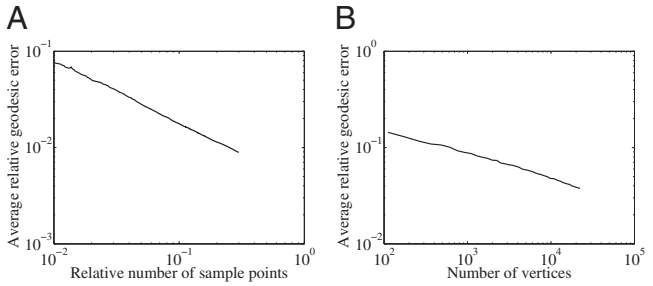

Fig. 3. (*A*) Relative geodesic distortion as a function of the number of sample points for interpolating **D**. (*B*) Geodesic error as a function of sample points.

Algorithm 1: Spectral MDS

Require: A data manifold $\mathcal{M}$ densely sampled at n points, number of subsampled points m_s, number of eigenvectors m_e.

Compute $\mathcal{J}$ a subset of m_s points sampled from $\mathcal{M}$.

Compute the matrix $\mathbf{D}$ of squared geodesic distances between every two points $(p_i, p_j), i \in \mathcal{J}, j \in \mathcal{J}$.

Compute the matrices $\mathbf{\Phi}, \mathbf{\Lambda}$ containing the first m_e^{th} eigenvectors and eigenvalues of the LBO of $\mathcal{M}$.

Compute the matrix $\boldsymbol{\alpha}$ according to **12** or **13**.

Compute the SVD decomposition of the $(n \times m_e)$ matrix $\mathbf{J\Phi}$, such that $\mathbf{J\Phi} = \mathbf{SUV}^T$, where $\mathbf{J} = \mathbf{I}_n - \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T$.

Compute the eigendecomposition of the $(m_e \times m_e)$ matrix $\mathbf{UV}^T\boldsymbol{\alpha}\mathbf{VU}$, such that $\mathbf{UV}^T\boldsymbol{\alpha}\mathbf{VU} = \mathbf{P\Gamma P}^T$.

Compute the matrix $\mathbf{Q} = \mathbf{SP}(\mathbf{\Gamma})^{\frac{1}{2}}$ such that $\mathbf{QQ}^T = \mathbf{J\Phi}\boldsymbol{\alpha}\mathbf{\Phi}^T\mathbf{J}$.

Return: the first k columns of the matrix $\mathbf{Q}$.

Relation to diffusion maps: An interesting flat-embedding option is known as diffusion maps (18, 29). Given a kernel function $K(\lambda)$, the diffusion distance between two points $(x,y) \in \mathcal{M}$ is defined as

$$D^2(x,y) = \sum_k (\phi_k(x) - \phi_k(y))^2 K(\lambda_k), \qquad [15]$$

where ϕ_k represents the k^{th} eigenfunction of $\Delta_{\mathcal{M}}$, and λ_k its associated eigenvalue. Note that $\mathbf{D}_{ij} = D^2(x_i, y_j)$. In a matrix form, Eq. **15** reads $\mathbf{D}_{ij} = \sum_{k=1}^{m_e} (\mathbf{\Phi}_{ik} - \mathbf{\Phi}_{jk})^2 \mathbf{K}_{kk}$, where $\mathbf{\Phi}$ represents the eigendecomposition matrix of the LBO and $\mathbf{K}$ is a diagonal matrix such that $\mathbf{K}_{kk} = K(\lambda_k)$.

Denoting by $\mathbf{\Psi}$, the matrix such that $\mathbf{\Psi}_{ij} = \mathbf{\Phi}_{ij}^2$, it follows that $\mathbf{D} = \mathbf{\Psi K}\mathbb{1}_{m_e}\mathbb{1}_n^T + (\mathbf{\Psi K}\mathbb{1}_{m_e}\mathbb{1}_n^T)^T - 2\mathbf{\Phi K\Phi}^T$. Next, let us apply classical scaling to $\mathbf{D}$, which by itself defines a flat domain. Because $\mathbb{1}_n^T\mathbf{J} = 0$, then $\mathbf{\Psi}K\mathbb{1}_{m_e}\mathbb{1}_n^T\mathbf{J} = 0$, and we readily have $\mathbf{JDJ} = -2\mathbf{J\Phi K\Phi}^T\mathbf{J} = \mathbf{J\Phi}(-2\mathbf{K})\mathbf{\Phi}^T\mathbf{J}$.

Applying MDS to diffusion distances turns out to be nothing but setting $\boldsymbol{\alpha} = -2\mathbf{K}$ in the proposed spectral MDS framework. The spectral MDS presented in Eq. **14** and algorithm 1 can be directly applied to diffusion distances without explicit computation of these distances. Moreover, using data-trained optimal diffusion kernels, as those introduced in ref. 34, we could obtain a discriminatively enhanced flat domain, in which robust and efficient classification between different classes is realized as part of the construction of the flat target space.

Numerical Experiments

The embedding of the intrinsic geometry of a shape into a Euclidean space is known as a canonical form. When the input to an MDS procedure is the set of geodesic distances between every two surface points, the output would be such a form. These structures, introduced in ref. 6, are invariant to isometric deformations of the shape, as shown in *SI Appendix*.

In our first example, we explore two shapes containing 5,000 vertices each for which we compute all pairwise geodesic distances. Then, we select a subset of just 50 vertices, using the farthest-point sampling strategy, and extract 50 eigenvectors of the corresponding LBO. Finally, we flatten the surfaces into their canonical forms using the spectral MDS procedure. A qualitative evaluation is presented in Fig. 1 demonstrating the similarity of the results of classical scaling on the full matrix, and the spectral MDS operating on just 1% of the data.

The relation of within classes and between classes for lionesses and dogs from the TOSCA database (35) is shown in Fig. 2. Spectral MDS allows us to consider more points and can thus serve as a better clustering tool.

Next, we compute the spectral interpolation of the matrix $\tilde{D}$ for sampled points from Armadillo's surface. We use the same number of eigenvectors as that of sample points, and plot the average geodesic error as a function of the points we use. The mean relative error is computed by $\text{mean}\left(\left|\tilde{D} - D\right| / \|D + \epsilon\|\right)$, and it is plotted as a function of the ratio between the number of sample points and the total number of vertices of the shape (Fig. 3, *Left*). Note that spectral interpolation of the geodesic distances works as predicted. In this example, 5% of the distances between the sample points provide us a distance map that is more than 97% accurate.

In the next experiment, we measured the influence of the number of sampled points on the accuracy of the reconstruction. We first generated several triangulations of a sphere, each at a different resolution reflected by the number of triangles. Next, for each such triangulation, we used spectral interpolation for calculating the geodesic distances using just $\sqrt{n}$ points, where n represents the number of vertices of the sphere at a given resolution. Fig. 3, *Right* shows the average geodesic error as a function of n. The average error is a decreasing function of the number of points, in the case where $\sqrt{n}$ points are used for the construction.

Next, we report on an attempt to extract an $\mathbb{R}^4$ manifold embedded in $\mathbb{R}^{10,000}$. We try to obtain the low-dimensional hyperplane structure from a given set of 100,000 points in $\mathbb{R}^{10,000}$. The spectral MDS could handle 10^5 points with very little effort, whereas regular MDS had difficulties already with 10^4 points. Being able to process a large number of samples is important when trying to deal with noisy data, as is often the case in big-data analysis. See MatLab code of this experiment in the *Supporting Information*.

Fig. 4 depicts the embedding of 5,851 images of handwritten digit 8 into a plane. The data are taken from the Modified National Institute of Standards and Technology (MNIST) database (36). We used a naive metric by which the distance between two images is equal to integration over the pixel-wise differences between them. Even in this trivial setting, the digits are arranged in the plane such that those on the right tilt to the right, and those on the left slant to the left. Fat digits appear at the bottom, and thin ones at the top.

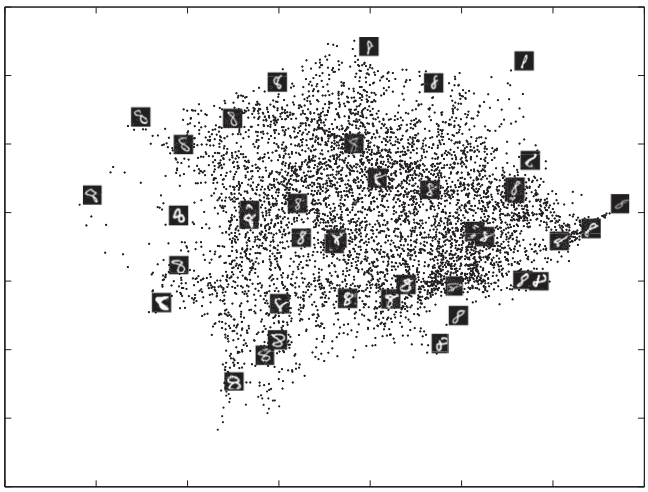

Fig. 4. Flat embedding of handwritten digit 8 from the MNIST database.

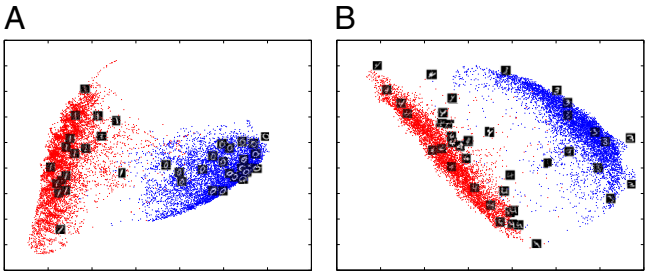

Fig. 5. Flat embedding of zeros (blue) vs. ones (red) (*A*), and fours (red) vs. threes (blue) (*B*).

Fig. 5 presents two cases of flat embedding of ones vs. zeros, and fours vs. threes, from the MNIST dataset. Here, the pairwise distance is the difference between the distance functions from the white regions in each image. While keeping the metric simple, the separation between the classes of digits is obvious. The elongated classes are the result of slanted digits that appear at different orientations. This observation could help in further study toward automatic character recognition.

Finally, we computed the regular MDS and the spectral MDS on different numbers of samples of randomly chosen images, referred to as points. Two hundred eigenvectors of the LBO were evaluated on an Intel i7 computer with 16 GB memory. The following table shows the time it took to compute MDS and spectral MDS embeddings.

No. of points	MDS run time, s	Spectral MDS run time, s
1,000	3	0.1
2,000	5.3	0.4
4,000	46	2.3
10,000	366	3.2

Conclusions

By working in the natural spectral domain of the data, we showed how to reduce the complexity of dimensionality reduction procedures. The proposed method exploits the local geometry of the data manifold to construct a low-dimensional spectral Euclidean space in which geodesic distance functions are compactly represented. Then, reduction of the data into a low-dimensional Euclidean space can be executed with low computational and space complexities. Working in the spectral domain allowed us to account for large distances while reducing the size of the data we process in an intrinsically consistent manner. The leading functions of the LBO eigenbasis capture the semidifferential structure of the data and thus provide an optimal reconstruction domain for the geodesic distances on the smooth data manifold. The proposed framework allowed us to substantially reduce the space and time involved in manifold learning techniques, which are important in the analysis of big data.

ACKNOWLEDGMENTS. This research was supported by Office of Naval Research Award N00014-12-1-0517 and Israel Science Foundation Grant 1031/12.

1. Schwartz EL, Shaw A, Wolfson E (1989) A numerical solution to the generalized mapmaker's problem: Flattening nonconvex polyhedral surfaces. *IEEE Trans PAMI* 11(9):1005–1008.
2. Drury HA, et al. (1996) Computerized mappings of the cerebral cortex: A multiresolution flattening method and a surface-based coordinate system. *J Cogn Neurosci* 8(1):1–28.
3. Dale AM, Fischl B, Sereno MI (1999) Cortical surface-based analysis. I. Segmentation and surface reconstruction. *Neuroimage* 9(2):179–194.
4. Zigelman G, Kimmel R, Kiryati N (2002) Texture mapping using surface flattening via multidimensional scaling. *IEEE Trans Vis Comp Graph* 8(2):198–207.
5. Grossmann R, Kiryati N, Kimmel R (2002) Computational surface flattening: A voxel-based approach. *IEEE Trans PAMI* 24(4):433–441.
6. Elad A, Kimmel R (2003) On bending invariant signatures for surfaces. *IEEE Trans PAMI* 25(10):1285–1295.
7. Schweitzer H (2001) Template matching approach to content based image indexing by low dimensional Euclidean embedding. *Computer Vision 2001. ICCV 2001: Proceedings of the Eighth International Conference on Computer Vision* (IEEE, New York), pp 566–571.
8. Rubner Y, Tomasi C (2001) Perceptual metrics for image database navigation. PhD dissertation (Stanford Univ, Stanford, CA).
9. Aharon M, Kimmel R (2006) Representation analysis and synthesis of lip images using dimensionality reduction. *Int J Comput Vis* 67(3):297–312.
10. Pless R (2003) Image spaces and video trajectories: Using Isomap to explore video sequences. *Computer Vision 2003: Proceedings of the Ninth IEEE International Conference on Computer Vision* (IEEE, New York), pp 1433–1440.
11. Bronstein AM, Bronstein MM, Kimmel R (2005) Three-dimensional face recognition. *Int J Comput Vis* 64(1):5–30.
12. Borg I, Groenen P (1997) Modern multidimensional scaling: Theory and applications. *J Educ Meas* 40(3):277–280.
13. Tenenbaum JB, de Silva V, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. *Science* 290(5500):2319–2323.
14. Kimmel R, Sethian JA (1998) Computing geodesic paths on manifolds. *Proc Natl Acad Sci USA* 95(15):8431–8435.
15. Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. *Science* 290(5500):2323–2326.
16. Donoho DL, Grimes C (2003) Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proc Natl Acad Sci USA* 100(10):5591–5596.
17. Belkin M, Niyogi P (2001) Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Information Processing Systems* (MIT Press, Cambridge, MA), pp 585–591.
18. Coifman RR, Lafon S (2006) Diffusion maps. *Appl Comput Harmon Anal* 21(1):5–30.
19. Coifman RR, et al. (2005) Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proc Natl Acad Sci USA* 102(21):7426–7431.
20. de Silva V, Tenenbaum JB (2003) Global versus local methods in nonlinear dimensionality reduction. *Advances in Neural Information Processing Systems* (MIT Press, Cambridge, MA), pp 705–712.
21. Bengio Y, Paiement JF, Vincent P (2003) Out-of-sample extensions for LLE, Isomap, MDS, eigenmaps, and spectral clustering. *Advances in Neural Information Processing Systems* (MIT Press, Cambridge, MA), pp 177–184.
22. Asano T, et al. (2009) A linear-space algorithm for distance preserving graph embedding. *Comput Geom Theory Appl* 42(4):289–304.
23. Bronstein MM, Bronstein AM, Kimmel R, Yavneh I (2006) Multigrid multidimensional scaling. *Numer Linear Algebra App* 13(2-3):149–171.
24. Rosman G, Bronstein MM, Bronstein AM, Sidi A, Kimmel R (2008) Fast multidimensional scaling using vector extrapolation. Technical report CIS-2008-01 (Technion, Haifa, Israel).
25. Rosman G, Bronstein MM, Bronstein AM, Kimmel R (2010) Nonlinear dimensionality reduction by topologically constrained isometric embedding. *Int J Comput Vis* 89(1):56–68.
26. Rustamov RM (2011) Interpolated eigenfunctions for volumetric shape processing. *Vis Comput* 27(11):951–961.
27. Raviv D, Bronstein MM, Bronstein AM, Kimmel R (2010) Volumetric heat kernel signatures. *Proc ACM Workshop 3D Object Retrieval* (ACM, New York), pp 39–44.
28. Williams CKI (2001) On a connection between kernel PCA and metric multidimensional scaling. *Advances in Neural Information Processing Systems* (MIT Press, Cambridge, MA), pp 675–681.
29. Bérard P, Besson G, Gallot S (1994) Embedding Riemannian manifolds by their heat kernel. *Geom Funct Anal* 4(4):373–398.
30. Sun J, Ovsjanikov M, Guibas L (2009) A concise and provably informative multi-scale signature based on heat diffusion. *Comp Graph Forum* 28(5):1383–1392.
31. Ovsjanikov M, Ben-Chen M, Solomon J, Butscher A, Guibas L (2012) Functional maps: A flexible representation of maps between shapes. *ACM Trans Graph* 3:30:1–30:11.
32. Weyl H (1950) Ramifications, old and new, of the eigenvalue problem. *Bull Am Math Soc* 56(2):115–139.
33. Pinkall U, Polthier K (1993) Computing discrete minimal surfaces and their conjugates. *Exper Math* 2(1):15–36.
34. Aflalo Y, Bronstein AM, Bronstein MM, Kimmel R (2012) Deformable shape retrieval by learning diffusion kernels. *Proc Scale Space Variational Meth Comp Vis* (Springer, Berlin), Vol 6667, pp 689–700.
35. Bronstein AM, Bronstein MM, Kimmel R (2008) *Numerical Geometry of Non-Rigid Shapes* (Springer, New York).
36. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proc IEEE* 86(11):2278–2324.