Multiscale Geometric Methods for Data Sets II: Geometric Multi-Resolution Analysis

ARTICLE *in* APPLIED AND COMPUTATIONAL HARMONIC ANALYSIS · MAY 2011

Impact Factor: 3 · DOI: 10.1016/j.acha.2011.08.001 · Source: arXiv

William Kenneth Allard
Duke University
12 PUBLICATIONS **427** CITATIONS

Mauro Maggioni
Duke University
68 PUBLICATIONS **1,574** CITATIONS

MULTISCALE GEOMETRIC METHODS FOR DATA SETS II: GEOMETRIC WAVELETS

WILLIAM K. ALLARD, GUANGLIANG CHEN, AND MAURO MAGGIONI

Abstract. Data sets are often modeled as point clouds in $\mathbb{R}^D$, for D large. It is often assumed that the data has some interesting low-dimensional structure, for example that of a d-dimensional manifold $\mathcal{M}$, with d much smaller than D. When $\mathcal{M}$ is simply a linear subspace, one may exploit this assumption for encoding efficiently the data by projecting onto a dictionary of d vectors in $\mathbb{R}^D$ (for example found by SVD), at a cost $(n + D)d$ for n data points. When $\mathcal{M}$ is nonlinear, there are no "explicit" constructions of dictionaries that achieve a similar efficiency: typically one uses either random dictionaries, or dictionaries obtained by black-box optimization. In this paper we construct data-dependent multi-scale dictionaries that aim at efficient encoding and manipulating of the data. Their construction is fast, and so are the algorithms that map data points to dictionary coefficients and vice versa. In addition, data points are guaranteed to have a sparse representation in terms of the dictionary. We think of dictionaries as the analogue of wavelets, but for approximating point clouds rather than functions.

1. Introduction

We construct Geometric Multi-Resolution Analysis for analyzing intrinsically low-dimensional point clouds in high-dimensional spaces, modeled as samples from a d-dimensional set $\mathcal{M}$ (in particular, a manifold) embedded in $\mathbb{R}^D$, in the regime $d \ll D$. This setting has been recognized as important in various applications, ranging from the analysis of sounds, images (RGB or hyperspectral, [23]), to gene arrays, EEG signals [15], and other types of manifold-valued data [54], and has been at the center of much investigation in the applied mathematics [20, 21, 26] and machine learning communities during the past several years. This has lead to a flurry of research on several problems, old and new, such as estimating the intrinsic dimensionality of point clouds [44, 46, 28, 10, 11, 13], parametrizing sampled manifolds [20, 37, 57, 3, 34, 35, 63, 40, 41], constructing dictionaries tuned to the data [1, 59] or for functions on the data [24, 9, 61, 47], and their applications to machine learning and function approximation [49, 48, 60, 25].

We focus on obtaining multi-scale representations in order to organize the data in a natural fashion, and obtain efficient data structures for data storage, transmission, manipulation, at different levels of precision that may be requested or needed for particular tasks. This work ties with much recent work in different directions:

Date: May 24, 2011.

Key words and phrases. Multiscale Analysis. Wavelets. Data Sets. Point Clouds. Frames. Sparse Approximation. Dictionary Learning.

The authors thank E. Monson for useful discussions.

GC was partially supported by ONR N00014-07-1-0625 and NSF CCF 0808847.

MM is grateful for partial support from DARPA, NSF, ONR, and the Sloan Foundation.

(a) Harmonic analysis and efficient representations of signals; (b) Data-adaptive signal representations in high dimensions and dictionary learning; (c) Hierarchical structures for organization of data sets; (d) Geometric analysis of low-dimensional sets in high-dimensional spaces.

Harmonic analysis and efficient representations of signals. Representations of classes of signals and data have been an important branch of research in multiple disciplines. In harmonic analysis, a linear infinite-dimensional function space $\mathcal{F}$ typical models the class of signals of interest, and linear representations in the form $f = \sum_i \alpha_i \phi_i$, for $f \in \mathcal{F}$ in terms of a dictionary of atoms $\Phi := \{\phi_i\} \subseteq \mathcal{F}$ are studied. Such dictionaries may be bases or frames, and are constructed so that the sequence of coefficients $\{\alpha_i\}_i$ has desirable properties, such as some form of sparsity, or a distribution highly concentrated at zero. Requiring sparsity of the representation is very natural from the viewpoints of statistics, signal processing, and interpretation of the representation. This, in part, motivated the construction of Fourier-like bases, wavelets, wedgelets, ridgelets, curvelets etc... [27, 12, 16], just to name a few. Several such dictionaries are proven to provide optimal representations (in a suitably defined sense) for certain classes of function spaces (e.g. some simple models for images) and/or for operators on such spaces. While orthogonal dictionaries were originally preferred (e.g. [29]), a trend developed towards over-complete dictionaries (e.g. frames [29, 18] and references therein) and libraries of dictionaries (e.g. wavelet and cosine packets [27], multiple dictionaries [43], fusion frames [14]), for which the set of coefficients $(\alpha_i)_i$ needed to represent a signal f is typically non-unique. Fast transforms, crucial in applications, have often been considered a fundamental hallmark of several of the transforms above, and was usually achieved through a multi-scale organization of the dictionaries.

Data-adaptive signal representation and dictionary learning. A more recent trend [16, 53, 1, 64, 51, 59], motivated by the desire to model classes of signals that are not well-modeled by the linear structure of function spaces, has been that of *constructing data-adapted dictionaries*: an algorithm is allowed to see samples from a class of signals $\mathcal{F}$ (not necessarily a linear function space), and constructs a dictionary $\Phi := \{\phi_i\}_i$ that optimizes some functional, such as the sparsity of the coefficients for signals in $\mathcal{F}$. The problem becomes being able to construct the dictionary Φ, typically highly over-complete, so that, given $f \in \mathcal{F}$, a rapid computation of the "best" (e.g. sparsest) coefficients $(\alpha_i)_i$ so that $f = \sum_i \alpha_i \phi_i$ is possible, and $(\alpha_i)_i$ is sparse. The problem of constructing Φ with the properties above, given a sample $\{f_n\}_n \subseteq \mathcal{F}$, is often called *dictionary learning*, and has been at the forefront of much recent research in harmonic analysis, approximation theory, imaging, vision, and machine learning: see [53, 1, 64, 51, 59] and references therein for constructions and applications.

There are several parameters in this problem: given training data from $\mathcal{F}$, one seeks Φ with I elements, such that every element in the training set may be represented, up to a certain precision ϵ, by at most m elements of the dictionary. The smaller I and m are, for a given ϵ, the better the dictionary.

Several current approaches may be summarized as follows [50]: consider a finite training set of signals $X_n = \{x_i\}_{i=1}^n \subset \mathbb{R}^D$, which we may represent by a $\mathbb{R}^{D \times n}$ matrix, and optimize the cost function

$$f_n(\Phi) = \frac{1}{n} \sum_{i=1}^{n} \ell(x_i, \Phi) \tag{1.1}$$

where $\Phi \in \mathbb{R}^{D\times I}$ is the dictionary, and ℓ a loss function, for example

$$\ell(x, \Phi) := \min_{\alpha\in\mathbb{R}^I} \frac{1}{2}||x - \Phi\alpha||^2_{\mathbb{R}^D} + \lambda||\alpha||_1 \tag{1.2}$$

where λ is a regularization parameter. This is basis pursuit [16] or lasso [62]. One typically adds constraints on the size of the columns of Φ, for example $||\phi_i||_{\mathbb{R}^D} \leq 1$ for all i, which we can write as $\Phi \in \mathcal{C}$ for some convex set $\mathcal{C}$. The overall problem may then be written as a matrix factorization problem with a sparsity penalty:

$$\min_{\Phi\in\mathcal{C},\alpha\in\mathbb{R}^{I\times n}} \frac{1}{2}||X_n - \Phi\alpha||^2_F + \lambda||\alpha||_{1,1} \, , \tag{1.3}$$

where $||\alpha||_{1,1} := \sum_{i_1,i_2} |\alpha_{i_1,i_2}|$. We refer the reader to [50] and references therein for techniques for attacking this optimization problem.

Constructions of such dictionaries (e.g. K-SVD [1], k-flats [59], optimization-based methods [50], Bayesian methods [64]) generally involve optimization or heuristic algorithms which are computationally intensive, do not shed light on the relationships between the dictionary size I, the sparsity of α, and the precision ϵ, and the resulting dictionary Φ is typically unstructured, and finding computationally, or analyzing mathematically, the sparse set of coefficients α may be challenging.

In this paper we construct data-dependent dictionaries based on a Geometric Multi-Resolution Analysis of the data. This approach is motivated by the intrinsically low-dimensional structure of many data sets, and is inspired by multi-scale geometric analysis techniques in geometric measure theory such as those in [38, 33], as well as by techniques in multi-scale approximation for functions in high-dimensions [8, 5, 6]. These dictionaries are structured in a multi-scale fashion (a structure that we call Geometric Multi-Resolution Analysis) and can be computed efficiently; the expansion of a data point on the dictionary elements is guaranteed to have a certain degree of sparsity m, and may be computed by a fast algorithm; the growth of the number of dictionary elements I as a function of ϵ is controlled depending on geometric properties of the data. We call the elements of these dictionaries *geometric wavelets*, since in some respects they generalize wavelets from vectors that analyze functions in linear spaces to affine vectors that analyze point clouds with possibly nonlinear geometry. The multi-scale analysis associated with geometric wavelets shares some similarities with that of standard wavelets (e.g. fast transforms, a version of two-scale relations, etc...), but is in fact quite different in many crucial respects. It is nonlinear, as it adapts to arbitrary nonlinear manifolds modeling the data space $\mathcal{F}$, albeit every scale-to-scale step is linear; translations or dilations do not play any role here, while they are often considered crucial in classical wavelet constructions. Geometric wavelets may allow the design of new algorithms for manipulating point clouds similar to those used for wavelets to manipulate functions.

The rest of the paper is organized as follows. In Sec. 2 we describe how to construct the geometric wavelets in a multi-scale fashion. We then present our algorithms in Sec. 3 and illustrate them on a few data sets, both synthetic and real-world, in Sec. 4. Sec. 5 introduces an orthogonal verison of the construction; more variations or optimizations of the construction are postponed to Sec. 6. The next two sections discuss how to represent and compress data efficiently (Sec. 7) and computational costs (Sec. 8). A naive attempt at modeling distributions is performed in Sec. 9. Finally, the paper is concluded in Sec. 10 by pointing out some future directions.

2. Construction of geometric wavelets

Let $(\mathcal{M}, \rho, \mu)$ be a metric measure space with μ a Borel probability measure and $\mathcal{M} \subseteq \mathbb{R}^D$. In the theoretical setting, $(\mathcal{M}, \rho, \mu)$ will be a smooth compact Riemannian manifold of dimensions d isometrically embedded in $\mathbb{R}^D$, endowed with the natural volume measure; in the numerical examples, $(\mathcal{M}, \rho, \mu)$ will be a finite discrete metric space with counting measure. We will be interested in the case when the "dimension" d of $\mathcal{M}$ is much smaller than the dimension of the ambient space $\mathbb{R}^D$. While d is typically unknown in practice, efficient (multi-scale, geometric) algorithms for its estimation are available (see [46], which also contains many references to previous work on this problem), under additional assumptions on the geometry of $\mathcal{M}$.

Our construction of the geometric wavelets consists of three steps:

1. A multi-scale geometric tree decomposition of $\mathcal{M}$ into subsets $\{C_{j,k}\}_{k\in\mathcal{K}_j, j\in\mathbb{Z}}$.
2. A d-dimensional affine approximation in each dyadic cell $C_{j,k}$, yielding a sequence of approximating piecewise linear sets $\{\mathcal{M}_j\}$, one for each scale j.
3. A construction of low-dimensional affine difference operators that efficiently encode the differences between $\mathcal{M}_j$ and $\mathcal{M}_{j+1}$.

This construction parallels, in a geometric setting, that of classical multi-scale wavelet analysis [29], and we therefore call it a Geometric Multi-Resolution Analysis (GMRA): the nonlinear space $\mathcal{M}$ replaces the classical function spaces, the piecewise affine approximation at each scale substitutes the linear projection on scaling function spaces, and the difference operators play the role of the classical linear wavelet projections. We show that when $\mathcal{M}$ is a smooth manifold, guarantees on the approximation rates of $\mathcal{M}$ by the $\mathcal{M}_j$ are easily derived (see Theorem 2.2 in Sec. 2.4).

We construct bases for the various affine operators involved, producing a hierarchically organized dictionary that is adapted to the data, which we expect to be useful in the applications discussed in the introduction.

2.1. Tree decomposition.

Let $B_r^{\mathcal{M}}(x)$ be the ρ-ball inside $\mathcal{M}$ of radius $r > 0$ centered at $x \in \mathcal{M}$. We start by a spatial multi-scale decomposition of the data set $\mathcal{M}$.

Definition 2.1. *A* **tree decomposition** *of a d-dimensional metric measure space $(\mathcal{M}, \rho, \mu)$ is a family of open sets in $\mathcal{M}$, $\{C_{j,k}\}_{k\in\mathcal{K}_j, j\in\mathbb{Z}}$, called* dyadic cells*, such that*

(i) *for every $j \in \mathbb{Z}$, $\mu(\mathcal{M} \setminus \cup_{k\in\mathcal{K}_j} C_{j,k}) = 0$;*
(ii) *for $j' \geq j$ and $k' \in \mathcal{K}_{j'}$, either $C_{j',k'} \subseteq C_{j,k}$ or $\mu(C_{j',k'} \cap C_{j,k}) = 0$;*
(iii) *for $j < j'$ and $k' \in \mathcal{K}_{j'}$, there exists a unique $k \in \mathcal{K}_j$ such that $C_{j',k'} \subseteq C_{j,k}$;*
(iv) *each $C_{j,k}$ contains a point $c_{j,k}$ such that $B^{\mathcal{M}}_{c_1\cdot 2^{-j}}(c_{j,k}) \subseteq C_{j,k} \subseteq B^{\mathcal{M}}_{2^{-j}}(c_{j,k})$, for a constant c_1 depending on intrinsic geometric properties of $\mathcal{M}$. In particular, we have $\mu(C_{j,k}) \sim 2^{-dj}$.*

The construction of such tree decompositions is possible on spaces of homogeneous type [17, 31, 32]. Let $\mathcal{T}$ be the tree structure associated to the decomposition above: for any $j \in \mathbb{Z}$ and $k \in \mathcal{K}_j$, we let children$(j, k) = \{k' \in \mathcal{K}_{j+1} : C_{j+1,k'} \subseteq C_{j,k}\}$. Note that $C_{j,k}$ is the disjoint union of its children $C_{j+1,k'}$, $k' \in$ children(j, k), due to (ii). We assume that $\mu(\mathcal{M}) \sim 1$ such that there is only one cell at the root of the tree with scale $\log_{2^d} \mu(\mathcal{M}) = 0$ (thus we will only consider $j \geq 0$). For every $x \in \mathcal{M}$, with abuse of notation we use (j, x) to represent the unique $(j, k(x)), k(x) \in \mathcal{K}_j$ such that $x \in C_{j,k(x)}$. The family of dyadic cells $\{C_{j,k}\}_{k\in\mathcal{K}_j}$ at scale j generates a σ-algebra $\mathcal{F}_j$. Functions measurable with respect to this σ-algebra are piecewise constant on each cell.

In this paper we will construct dyadic cells on i.i.d. μ-distributed samples $\{x_i\}_{i=1}^n$ from $\mathcal{M}$ according to the following variation of the construction of diffusion maps [20, 22]: we connect each x_i to its k-nearest neighbors (default value is $k = 50$), with weights $W_{ij} = K(x_i, x_j) = e^{-||x_i-x_j||^2/\epsilon_i\epsilon_j}$, where ϵ_i is the distance between x_i and its $k/2$-nearest neighbor, to obtain a weighted graph on the samples x_i (this construction is used and motivated in [55]). We then make use of METIS [42] to produce the multi-scale partitions $\{C_{j,k}\}$ and the dyadic tree $\mathcal{T}$ above. In a future publication we will discuss how to use a variation of cover trees [4], which has guarantees in terms of both the quality of the decomposition and computational costs, and has the additional advantage of being easily updatable with new samples.

We may also construct the cells $C_{j,k}$ by intersecting Euclidean dyadic cubes in $\mathbb{R}^D$ with $\mathcal{M}$: if $\mathcal{M}$ is sufficiently regular and so is its embedding in $\mathbb{R}^D$ (e.g. $\mathcal{M}$ a smooth compact isometrically embedded manifold, or a dense set of samples, distributed according to volume measure, from it), then the properties in Definition 2.1 are satisfied for j large enough. In this case, a careful numerical implementation is needed in order to not be penalized by the ambient dimensionality (e.g. [7] and references therein).

2.2. Multiscale singular value decompositions and geometric scaling functions.

The tools we build upon are classical in multi-scale geometric measure theory [39, 30, 32], especially in its intersection with harmonic analysis, and it is also related to adaptive approximation in high dimensions, see for example [8, 5, 6] and references therein. An introduction to the use of such ideas for the estimation of intrinsic dimension of point clouds is in [46] and references therein (see [44, 45] for previous short accounts).

We will associate several gadgets to each dyadic cell $C_{j,k}$, starting with some geometric objects: the mean

$$c_{j,k} := \mathbb{E}_\mu[x|x \in C_{j,x}] = \frac{1}{\mu(C_{j,k})}\int_{C_{j,k}} x\,d\mu(x) \;\in \mathbb{R}^D \tag{2.1}$$

and the covariance operator restricted to $C_{j,k}$

$$\mathrm{cov}_{j,k} = \mathbb{E}_\mu[(x - c_{j,k})(x - c_{j,k})^*|x \in C_{j,k}] \in \mathbb{R}^{D\times D}\,. \tag{2.2}$$

Here and in what follows points in $\mathbb{R}^D$ are identified with D-dimensional column vectors. For a prescribed $d_{j,k}$ (e.g. $d_{j,k} = d$), let the rank-$d_{j,k}$ Singular Value Decomposition (SVD) [36] of $\mathrm{cov}_{j,k}$ be

$$\mathrm{cov}_{j,k} \approx \Phi_{j,k}\Sigma_{j,k}\Phi_{j,k}^*, \tag{2.3}$$

where $\Phi_{j,k}$ is an orthonormal $D \times d_{j,k}$ matrix and Σ is a diagonal $d_{j,k} \times d_{j,k}$ matrix. We let

$$\mathbb{V}_{j,k} := V_{j,k} + c_{j,k} \quad , \qquad V_{j,k} = \langle\Phi_{j,k}\rangle\,, \tag{2.4}$$

where $\langle A\rangle$ denotes the span of the columns of A, so that $\mathbb{V}_{j,k}$ is the affine subspace of dimension $d_{j,k}$ parallel to $V_{j,k}$ and passing through $c_{j,k}$. It is an approximate

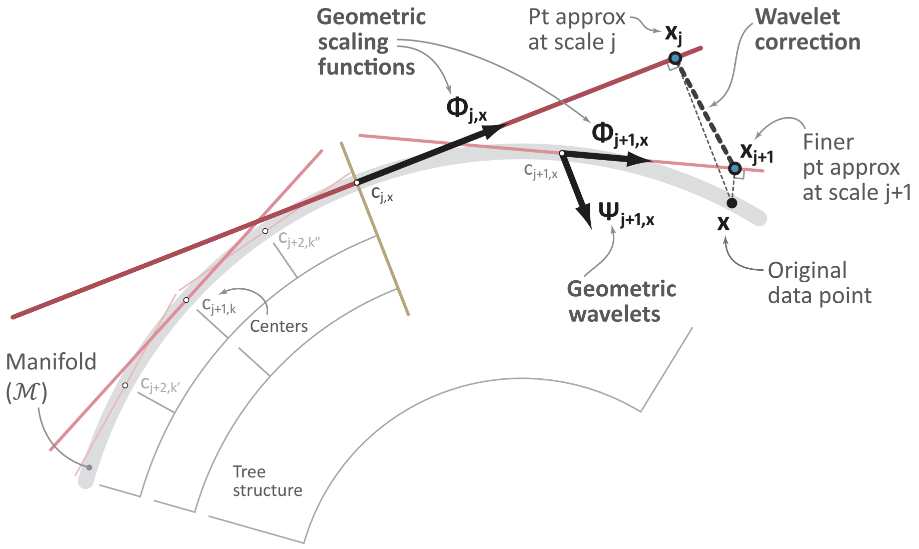

FIGURE 1. An illustration of the geometric wavelet decomposition. The centers $c_{j,x}$'s are represented as lying on $\mathcal{M}$ while in fact they are only close to $\mathcal{M}$, and the corresponding planes $\mathbb{V}_{j,x}$ are represented as tangent planes, albeit they are only an approximation to them. Art by E. Monson.

tangent space to $\mathcal{M}$ at location $c_{j,k}$ and scale 2^{-j}; and in fact it provides the best $d_{j,k}$-dimensional planar approximation to $\mathcal{M}$ in the least square sense:

$$\mathbb{V}_{j,k} = \underset{\Pi}{\operatorname{argmin}} \int_{C_{j,k}} ||x - \mathbb{P}_{\Pi}(x)||^2 \, d\mu(x)\,. \tag{2.5}$$

where Π is taken on the set of all affine $d_{j,k}$-planes, and $\mathbb{P}_{\Pi}$ is the orthogonal projection onto the affine plane Π. We think of $\{\Phi_{j,k}\}_{k\in\mathcal{K}_j}$ as the geometric analogue of a family of scaling functions at scale j, and therefore call *geometric scaling functions*. Let $\mathbb{P}_{j,k}$ be the associated affine projection

$$\mathbb{P}_{j,k}(x) := \mathbb{P}_{\mathbb{V}_{j,k}}(x) = \Phi_{j,k}\Phi^*_{j,k}(x - c_{j,k}) + c_{j,k} \qquad , \quad x \in C_{j,k}\,. \tag{2.6}$$

Then $\mathbb{P}_{j,k}(C_{j,k})$ is the projection of $C_{j,k}$ onto its local linear approximation, at least for $2^{-j} \lesssim \operatorname{reach}(\mathcal{M})$. We let

$$\mathcal{M}_j := \{\mathbb{P}_{j,k}(C_{j,k})\}_{k\in\mathcal{K}_j} \tag{2.7}$$

be a coarse approximation of $\mathcal{M}$ at scale j, the geometric analogue to what the projection of a function onto a scaling function subspace is in wavelet theory. Under general conditions, $\mathcal{M}_j \to \mathcal{M}$ in the Hausdorff distance, as $J \to +\infty$. It is natural to define the nonlinear projection of $\mathcal{M}$ onto $\mathcal{M}_j$ by

$$x_j \equiv P_{\mathcal{M}_j}(x) := \mathbb{P}_{j,k}(x) \qquad , \qquad x \in C_{j,k} \tag{2.8}$$

2.3. **Geometric wavelets.** We would like to efficiently encode the difference needed to go from $\mathcal{M}_j$ to $\mathcal{M}_{j+1}$, for $j \geq 0$. Fix $x \in \mathcal{M}$: the difference $x_{j+1} - x_j$ is a high-dimensional vector in $\mathbb{R}^D$, in general not contained in $\mathcal{M}_{j+1}$. However it may be decomposed into a sum of vectors in certain well-chosen low-dimensional spaces, which are shared across multiple points, in a multi-scale fashion. Recall that we use the notation (j,x) to denote the unique pair (j,k), with $k \in \mathcal{K}_j$, such that $x \in C_{j,k}$. We proceed as follows: for $j \leq J-1$ we let

$$\begin{aligned} Q_{\mathcal{M}_{j+1}}(x) &:= x_{j+1} - x_j \\ &= x_{j+1} - \mathbb{P}_{j,x}(x_{j+1}) + \mathbb{P}_{j,x}(x_{j+1}) - \mathbb{P}_{j,x}(x) \\ &= (I - P_{j,x})(x_{j+1} - c_{j,x}) + P_{j,x}(x_{j+1} - x) \\ &= (I - P_{j,x})(\underbrace{x_{j+1} - c_{j+1,x}}_{\in V_{j+1,x}} + c_{j+1,x} - c_{j,x}) - P_{j,x}(x - x_{j+1}). \qquad (2.9) \end{aligned}$$

Define the wavelet subspace

$$W_{j+1,x} := (I - P_{j,x})\, V_{j+1,x}. \tag{2.10}$$

It is clear that $\dim W_{j+1,x} \leq \dim V_{j+1,x} = d_{j+1,x}$. Let $Q_{j+1,x}$ be the orthogonal projection onto $W_{j+1,x}$. We define several quantities below:

$$t_{j+1,x} := c_{j+1,x} - c_{j,x}; \tag{2.11}$$

$$w_{j+1,x} := (I - P_{j,x})\, t_{j+1,x}; \tag{2.12}$$

$$\mathbb{Q}_{j+1,x}(x) := Q_{j+1,x}(x - c_{j+1,x}) + w_{j+1,x}. \tag{2.13}$$

Then we may rewrite (2.9) as

$$\begin{aligned} Q_{\mathcal{M}_{j+1}}(x) &= \underbrace{Q_{j+1,x}(x_{j+1} - c_{j+1,x})}_{\in W_{j+1,x}} + w_{j+1,x} - P_{j,x}\left(x - x_J + \sum_{l=j+1}^{J-1}(x_{l+1} - x_l)\right) \\ &= \mathbb{Q}_{j+1,x}(x_{j+1}) - P_{j,x}\sum_{l=j+1}^{J-1}(x_{l+1} - x_l) - P_{j,x}(x - x_J) \\ &= \mathbb{Q}_{j+1,x}(x_{j+1}) - P_{j,x}\sum_{l=j+1}^{J-1} Q_{\mathcal{M}_{l+1}}(x) - P_{j,x}(x - x_J). \qquad (2.14) \end{aligned}$$

Here $J \geq j+1$ is the index of the finest scale (and the last term vanishes as $J \to +\infty$, under general conditions). We let $\Psi_{j+1,x}$ be an orthonormal basis for $W_{j+1,x}$, that we will call a *geometric wavelet basis*. In terms of the geometric scaling functions and wavelets, the above may be written as

$$\begin{aligned} x_{j+1} - x_j = {} & \Psi_{j+1,x}\Psi_{j+1,x}^*\,(x_{j+1} - c_{j+1,x}) + w_{j+1,x} - \Phi_{j,x}\Phi_{j,x}^*\sum_{l=j+1}^{J-1} Q_{\mathcal{M}_{l+1}}(x) \\ & - \Phi_{j,x}\Phi_{j,x}^*\,(x - x_J)\,. \qquad (2.15) \end{aligned}$$

This equation splits the difference $x_{j+1} - x_j$ into a component in $W_{j+1,x}$, a second component that only depends on the cell $(j+1,x)$ (but not on the point x itself), accounting for the translation of centers and lying in the orthogonal complement of $V_{j,x}$ but not necessarily in $W_{j+1,x}$, and a sum of terms which are projections on $V_{j,x}$ of differences in the same form $x_{l+1} - x_l$, but at finer scales. By construction we have the two-scale equation

$$P_{\mathcal{M}_{j+1}}(x) = P_{\mathcal{M}_j}(x) + Q_{\mathcal{M}_{j+1}}(x)\,, \quad x \in \mathcal{M} \tag{2.16}$$

which can be iterated across scales, leading to a multi-scale decomposition along low-dimensional subspaces, with efficient encoding and algorithms. We think of

$P_{j,k}$ as being attached to the node (j,k) of $\mathcal{T}$, and the $Q_{j+1,k'}$ as being attached to the edge connecting the node $(j+1,k')$ to its parent.

We say that the set of multi-scale piecewise affine operators $\{P_{\mathcal{M}_j}\}$ and $\{Q_{\mathcal{M}_{j+1}}\}$ form a **Geometric Multi-Resolution Analysis**, or GMRA for short.

2.4. **Approximation for manifolds.** We analyze the error of approximation to a d-dimensional manifold in $\mathbb{R}^D$ by using geometric wavelets representation. Our analysis gives a full explanation of the examples in Sec. 4.1.

Theorem 2.2. *Let $(\mathcal{M},\rho,\mu)$ be a compact $\mathcal{C}^{1+\alpha}$ Riemannian manifold of dimension d isometrically embedded in $\mathbb{R}^D$, with $\alpha\in(0,1]$, and μ absolutely continuous with respect to the volume measure on $\mathcal{M}$. Let $\{P_{\mathcal{M}_j}, Q_{\mathcal{M}_{j+1}}\}$ be a GMRA for $(\mathcal{M},\rho,\mu)$. For any $x\in\mathcal{M}$, there exists a scale $j_0=j_0(x)$ such that for any $j\geq j_0$ and any $p>0$, if we let $d\mu_{j,x}:=\mu(C_{j,x})^{-1}d\mu$,*

$$\left|\left|\left|\left|z-P_{\mathcal{M}_j}(z)\right|\right|_{\mathbb{R}^D}\right|\right|_{L^p(C_{j,x},d\mu_{j,x}(z))}=\left|\left|\left|\left|z-P_{\mathcal{M}_{j_0}}(z)-\sum_{l=j_0}^{j-1}Q_{\mathcal{M}_{l+1}}(z)\right|\right|_{\mathbb{R}^D}\right|\right|_{L^p(C_{j,x},d\mu_{j,x}(z))}$$
$$\leq ||\kappa||_{L^\infty(C_{j,x})}\,2^{-(1+\alpha)j}+o(2^{-(1+\alpha)j})\,. \tag{2.17}$$

If $\alpha<1$, $\kappa(x)$ depends on the $\mathcal{C}^{1+\alpha}$ norm of a coordinate chart from $T_x(\mathcal{M})$ to $C_{j,x}\subseteq\mathcal{M}$ and on $\left\|\frac{d\mu}{d\mathrm{vol}}\right\|_{L^\infty(C_{j,x})}$.

If $\alpha=1$,

$$\kappa(x)=\left\|\frac{d\mu}{d\mathrm{vol}}\right\|_{L^\infty(C_{j,x})}\min(\kappa_1(x),\kappa_2(x))\,, \tag{2.18}$$

with

$$\kappa_1(x):=\frac{1}{2}\max_{i\in\{1,\dots,D-d\}}||H_i(x)||; \tag{2.19}$$

$$\kappa_2^2(x):=\max_{w\in\mathbb{S}^{D-d}}\frac{d(d+1)}{4(d+2)(d+4)}\left[\left\|\sum_{l=1}^{D-d}w_lH_l(x)\right\|_F^2-\frac{1}{d+2}\left(\sum_{l=1}^{D-d}w_l\mathrm{Tr}(H_l(x))\right)^2\right], \tag{2.20}$$

and the $D-d$ matrices $H_l(x)$ are the d-dimensional Hessians of $\mathcal{M}$ at x.

Observe that κ_2 can be smaller than κ_1 (by a constant factor) or larger (by factors depending on d^2), depending on the spectral properties and commutativity relations between the Hessians H_l. κ_2^2 may be unexpectedly small, in the sense that it may scale as $d^{-2}r^4$ as a function of d and r, as observed in [46].

The proof is postponed to the Appendix.

2.5. **Non-manifold data and measures of approximation error.** When constructing a GMRA for point-cloud data not sampled from manifolds, we may choose the dimension $d_{j,k}$ of the local linear approximating plane $\mathbb{V}_{j,k}$ by a criterion based on local approximation errors. Note that this affects neither the construction of geometric scaling functions, nor that of the wavelet subspaces and bases.

A simple measure for absolute error of approximation at scale j is:

$$\begin{aligned}\mathcal{E}_j^2 &= \int_{\mathcal{M}} ||P_{\mathcal{M}_j}(x) - x||^2_{\mathbb{R}^D}\, d\mu(x) = \sum_{k\in\mathcal{K}_j} \int_{C_{j,k}} ||P_{j,k}(x) - x||^2_{\mathbb{R}^D}\, d\mu|_{C_{j,k}}(x)\\ &= \sum_{k\in\mathcal{K}_j} \mu(C_{j,k}) \frac{1}{\mu(C_{j,k})} \int_{C_{j,k}} ||P_{j,k}(x) - x||^2_{\mathbb{R}^D}\, d\mu|_{C_{j,k}}(x)\\ &= \sum_{k\in\mathcal{K}_j} \mu(C_{j,k}) \sum_{l\ge d_{j,k}+1} \lambda_l^2(\mathrm{cov}_{j,k}). \qquad (2.21)\end{aligned}$$

We can therefore control $\mathcal{E}_j$ by choosing $d_{j,k}$ based on the spectrum of $\mathrm{cov}_{j,k}$. If we perform relative thresholding of $\mathrm{cov}_{j,k}$, i.e. choose the smallest $d_{j,k}$ for which

$$\sum_{l\ge d_{j,k}+1} \lambda_l^2(\mathrm{cov}_{j,k}) \le \epsilon_j \sum_{l\ge 1} \lambda_l^2(\mathrm{cov}_{j,k}), \qquad (2.22)$$

for some choice of ϵ_j (e.g. $\epsilon_j = (c\theta^j) \vee \epsilon$ for some $\theta \in (0,1)$ and $\epsilon > 0$), then we may upper bound the above as follows:

$$\mathcal{E}_j^2 \le \sum_{k\in\mathcal{K}_J} \mu(C_{j,k})\epsilon_j ||C_{j,k}||_F^2 \le \epsilon_j |||\mathcal{M}|||_F\,, \qquad (2.23)$$

where for a partitioned matrix $A = [A_1, A_2, \dots, A_r]$ and discrete probability measure μ on $\{1, \dots, r\}$ we define

$$|||A|||_F^2 := \sum_{i=1}^{r} \mu(\{i\}) ||A_i||_F^2. \qquad (2.24)$$

If we perform absolute thresholding of $\mathrm{cov}_{j,k}$, i.e. choose the smallest $d_{j,k}$ for which $\sum_{l\ge d_{j,k}+1} \lambda_l^2(\mathrm{cov}_{j,k}) \le \epsilon_j$, then we have the rough bound

$$\mathcal{E}_j^2 \le \sum_{k\in\mathcal{K}_j} \mu(C_{j,k})\epsilon_j \le \epsilon_j \cdot \mu(\mathcal{M}). \qquad (2.25)$$

Of course, in the case of a d-dimensional $\mathcal{C}^2$ manifold $\mathcal{M}$ with volume measure, if we choose $d_{j,k} = d$, by Theorem 2.2 we have

$$\mathcal{E}_j \lesssim \sum_{k\in\mathcal{K}_j} \mu(C_{j,k}) ||\kappa||_\infty 2^{-2j} = \mu(\mathcal{M}) ||\kappa||_\infty 2^{-2j}. \qquad (2.26)$$

3. Algorithms

We present in this section algorithms implementing the construction of the GMRA and the corresponding Geometric Wavelet Transform (GWT).

3.1. Construction of Geometric Multi-Resolution Analysis.

The first step in the construction of the geometric wavelets is to perform a geometric nested partition of the data set, forming a tree structure. For this end, one may consider various methods listed below:

(I). Use of METIS [42]: a multiscale variation of iterative spectral partitioning. We construct a weighted graph as done for the construction of diffusion maps [20, 22]: we add an edge between each data point and its k nearest neighbors, and assign to any such edge between x_i and x_j the weight $e^{-||x_i-x_j||^2/\sigma}$. Here k and σ are parameters whose selection we do not discuss here (but see [55] for a discussion in the context of molecular dynamics

```
GMRA = GeometricMultiResolutionAnalysis $(X_n, \tau_0, \epsilon)$

// **Input:**
// $X_n$: a set of $n$ samples from $\mathcal{M}$
// $\tau_0$: some method for choosing local dimensions
// $\epsilon$: precision
// **Output:**
// A tree $\mathcal{T}$ of dyadic cells $\{C_{j,k}\}$, their local means $\{c_{j,k}\}$ and bases $\{\Phi_{j,k}\}$, together with a family of geometric wavelets $\{\Psi_{j,k}\}, \{w_{j,k}\}$

Construct the dyadic cells $C_{j,k}$ with centers $\{c_{j,k}\}$ and form a tree $\mathcal{T}$.
$J \leftarrow$ finest scale with the $\epsilon$-approximation property.
Let $\mathrm{cov}_{J,k} = |C_{J,k}|^{-1} \sum_{x \in C_{J,k}} (x - c_{J,k})(x - c_{J,k})^*$, for $k \in \mathcal{K}_J$, and compute $\mathrm{SVD}(\mathrm{cov}_{J,k}) = \Phi_{J,k} \Sigma_{J,k} \Phi_{J,k}$ (where the dimension of $\Phi_{J,k}$ is determined by $\tau_0$).
for $j = J - 1$ down to $0$
    for $k \in \mathcal{K}_j$
        Compute $\mathrm{cov}_{j,k}$ and $\Phi_{j,k}$ as above.
        For each $k' \in \mathrm{children}(j,k)$, construct the wavelet bases $\Psi_{j+1,k'}$ and translations $w_{j+1,k'}$.
    end
end
For convenience, set $\Psi_{0,k} := \Phi_{0,k}$ and $w_{0,k} := c_{0,k}$ for $k \in \mathcal{K}_0$.
```

FIGURE 2. Pseudo-code for the construction of geometric wavelets

data). In practice, we choose k between 10 and 50, and choose σ adaptively at each point x_i as the distance between x_i and its $\lfloor k/2 \rfloor$ nearest neighbor.

(II). Use of cover trees [4].

(III). Use of iterated PCA: at scale 1, compute the top d principal components of data, and partition the data based on the sign of the $(d+1)$-st singular vector. Repeat on each of the two partitions.

(IV). Iterated k-means: at scale 1 partition the data based on k-means clustering, then iterate on each of the elements of the partition.

Each construction has pros and cons, in terms of performance and guarantees. For (I) we refer the reader to [42], for (II) to [4] (which also discussed several other constructions), for (III) and (IV) to [58]. Only (II) provides the needed properties for the cells $C_{j,k}$. However constructed, we denote by $\{C_{j,k}\}$ the family of resulting dyadic cells, and let $\mathcal{T}$ be the associated tree structure, as in Definition 2.1.

In Fig. 2 we display pseudo-code for the geometric multi-resolution analysis of a data set X_n given a precision $\epsilon > 0$ and a method τ_0 for choosing local dimensions (e.g., using thresholds or a fixed dimension). The code first constructs a family of multi-scale dyadic cells (with local centers $c_{j,k}$ and bases $\Phi_{j,k}$), and then computes the geometric wavelets $\Psi_{j,k}$ and translations $w_{j,k}$ at all scales. In practice, we use METIS [42] to construct a dyadic (not 2^d-adic) tree $\mathcal{T}$ and the associated cells $C_{j,k}$.

3.2. **The Fast Geometric Wavelet Transform and its Inverse.** For simplicity of presentation, we shall assume $x = x_J$; otherwise, we may first project x onto the local linear approximation of the cell $C_{J,x}$ and use x_J instead of x from now on. That is, we will define $x_{j;J} = P_{\mathcal{M}_j}(x_J)$, for all $j < J$, and encode the differences

```
$\{q_{j,x}\}$ =FGWT(GMRA, $x$)

// **Input:** GMRA structure, $x \in \mathcal{M}$
// **Output:** A sequence $\{q_{j,x}\}$ of wavelet coefficients

$p_{J,x} = \Phi^*_{J,x}(x - c_{J,x})$
**for** $j = J$ **down to** $1$
        $q_{j,x} = (\Psi^*_{j,x}\Phi_{j,x})\, p_{j,x}$
        $p_{j-1,x} = (\Phi^*_{j-1,x}\Phi_{J,x})\, p_{J,x} + \Phi^*_{j-1,x}(c_{J,x} - c_{j-1,x})$
**end**
$q_{0,x} = p_{0,x}$ (for convenience)
```

FIGURE 3. Pseudo-code for the Forward Geometric Wavelet Transform

```
$\hat{x}$ =IGWT(GMRA,$\{q_{j,x}\}$)

// **Input:** GMRA structure, wavelet coefficients $\{q_{j,x}\}$
// **Output:** Approximation $\hat{x}$ at scale $J$

$Q_{J,x} = \Psi_{J,x} q_{J,x} + w_{J,x}$
**for** $j = J-1$ **down to** $1$
        $Q_j(x) = \Psi_{j,x} q_{j,x} + w_{j,x} + \Phi_{j-1,x}\Phi^*_{j-1,x} \; \sum_{\ell > j} Q_\ell(x)$
**end**
$\hat{x} = \Psi_{0,x} q_{0,x} + w_{0,x} + \sum_{j>0} Q_j(x)$
```

FIGURE 4. Pseudo-code for the Inverse Geometric Wavelet Transform

$x_{j+1;J} - x_{j;J}$ using the geometric wavelets. Note also that $\|x_{j;J} - x_j\| \le \|x - x_J\|$ at all scales.

The geometric scaling and wavelet coefficients $\{p_{j,x}\}, \{q_{j+1,x}\}$, for $j \ge 0$, of a point $x \in \mathcal{M}$ are chosen to satisfy the equations

$$P_{\mathcal{M}_j}(x) = \Phi_{j,x} p_{j,x} + c_{j,x}; \tag{3.1}$$

$$Q_{\mathcal{M}_{j+1}}(x) = \Psi_{j+1,x} q_{j+1,x} + w_{j+1,x} - P_{j,x} \sum_{l=j+1}^{J-1} Q_{\mathcal{M}_{l+1}}(x). \tag{3.2}$$

The computation of the coefficients, from fine to coarse, is simple and fast: since we assume $x = x_J$, we have

$$\begin{aligned} p_{j,x} &= \Phi^*_{j,x}(x_J - c_{j,x}) = \Phi^*_{j,x}(\Phi_{J,x} p_{J,x} + c_{J,x} - c_{j,x}) \\ &= \left(\Phi^*_{j,x}\Phi_{J,x}\right) p_{J,x} + \Phi^*_{j,x}(c_{J,x} - c_{j,x}). \end{aligned} \tag{3.3}$$

Moreover the wavelet coefficients $q_{j+1,x}$ (defined in (3.2)) are obtained from (2.15):

$$q_{j+1,x} = \Psi^*_{j+1,x}(x_{j+1} - c_{j+1,x}) = \left(\Psi^*_{j+1,x}\Phi_{j+1,x}\right) p_{j+1,x}. \tag{3.4}$$

Note that $\Phi^*_{j,x}\Phi_{J,x}$ and $\Psi^*_{j+1,x}\Phi_{j+1,x}$ are both small matrices (at most $d_{j,x} \times d_{j,x}$), and are the only matrices we need to compute and store (once for all, and only up to a specified precision) in order to compute all the wavelet coefficients $q_{j+1,x}$ and the scaling coefficients $p_{j,x}$, given $p_{J,x}$ at the finest scale.

In Figs. 3 and 4 we display pseudo-codes for the computation of the Forward and Inverse Geometric Wavelet Transforms (F/IGWT). The input to FGWT is a GMRA object, as returned by `GeometricMultiResolutionAnalysis`, and a point

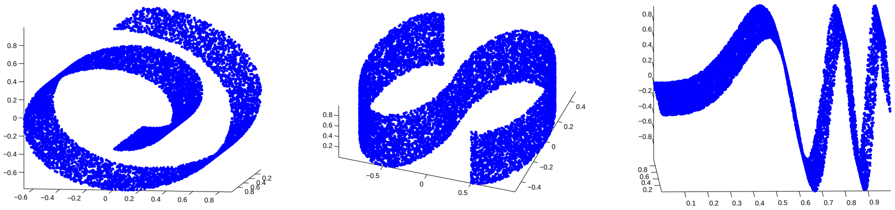

FIGURE 5. Toy data sets for geometric wavelets transform.

$x \in \mathcal{M}$. Its output is the wavelet coefficients of the point x at all scales, which are then used by IGWT for reconstruction of the point at all scales.

For any $x \in \mathcal{M}_J$, the set of coefficients

$$q_x = (q_{J,x}; q_{J-1,x}; \dots; q_{1,x}; p_{0,x}) \tag{3.5}$$

is called the discrete *geometric wavelet transform* (GWT) of x. Letting $d^w_{j,x} = \operatorname{rank}(\Psi_{j+1,x})$, the length of the transform is $d + \sum_{j>0} d^w_{j,x}$, which is bounded by $(J+1)d$ in the case of samples from a d-dimensional manifold (due to $d^w_{j,x} \le d$).

Remark 3.1. Note that for the variation of the GMRA without adding tangential corrections (see Sec. 6.2), the algorithms above (as well as those in Sec. 5) can be simplified. First, in Fig. 2 we will not need to store the local bases functions $\{\Phi_{j,k}\}$. Second, the steps in Figs. 3 and 4 can be modified not to involve $\{\Phi_{j,k}\}$, similarly as in Figs. 17 and 18 of next section.

4. EXAMPLES

We conduct numerical experiments in this section to demonstrate the performance of the algorithm (i.e., Figs. 2, 3, 4).

4.1. **Low-dimensional smooth manifolds.** To illustrate the construction presented so far, we consider simple synthetic datasets: a *SwissRoll*, an *S-Manifold* and an *Oscillating2DWave*, all two-dimensional manifolds but embedded in $\mathbb{R}^{50}$ (see Fig. 5). We apply the algorithm to construct the GMRA and obtain the forward geometric wavelet transform of the sampled data (10000 points, without noise) in Fig. 6. We use the manifold dimension $d_{j,k} = d = 2$ at each node of the tree when constructing scaling functions, and choose the smallest finest scale for achieving an absolute precision .001 in each case. We compute the average magnitude of the wavelet coefficients at each scale and plot it as a function of scale in Fig. 6. The reconstructed manifolds obtained by the inverse geometric wavelets transform (at selected scales) are shown in Fig. 7, together with a plot of relative approximation errors,

$$\mathcal{E}^{\text{rel}}_{j,2} = \frac{1}{\sqrt{\operatorname{Var}(X_n)}} \sqrt{\frac{1}{n} \sum_{x \in X_n} \left(\frac{||x - P_{j,x}(x)||}{||x||} \right)^2}, \tag{4.1}$$

where X_n is the training data of n samples. Both the approximation error and the magnitude of the wavelet coefficients decrease quadratically with respect to scale as expected.

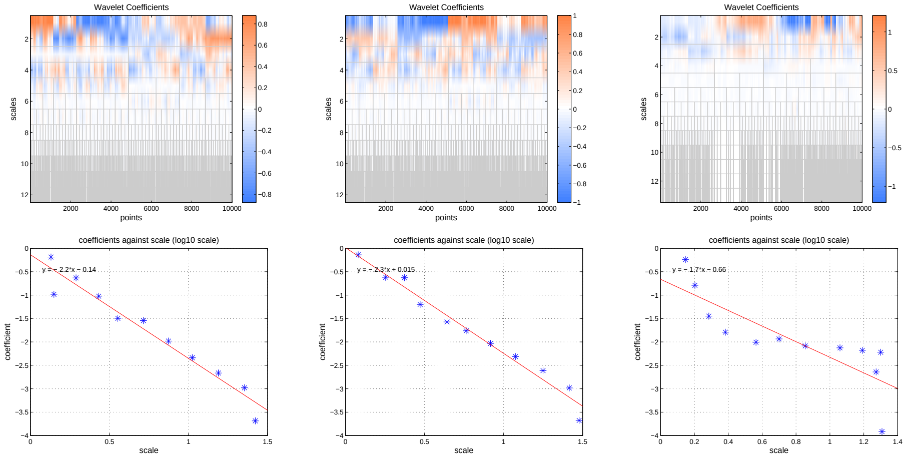

Figure 6. Top row: Wavelet coefficients obtained by the algorithm for the three data sets in Fig. 5. The horizontal axis indexes the points (arranged according to the tree), and the vertical axis multi-indexes the wavelet coefficients, from coarse (top) to fine (bottom) scales: the block of entries $(x, j), x \in C_{j,k}$ displays $\log_{10} |q_{j,x}|$, where $q_{j,x}$ is the vector of geometric wavelet coefficients of x at scale j (see Sec. 3). In particular, each row indexes multiple wavelet elements, one for each $k \in \mathcal{K}_j$. Bottom row: magnitude of wavelet coefficients decreasing quadratically as a function of scale.

We threshold the wavelet coefficients to study the compressibility of the wavelet coefficients and the rate of change of the approximation errors (using compressed wavelet coefficients). For this end, we use a smaller precision 10^{-5} so that the algorithm can examine a larger interval of thresholds. We first threshold the wavelet coefficients of the *Oscillating2DWave* data at the level .01 and plot in Fig. 8 the reduced matrix of wavelet coefficients and the corresponding best reconstruction of the manifold (i.e., at the finest scale). Next, we threshold the wavelet coefficients of all three data sets at different levels (from 10^{-5} to 1) and plot in Fig. 9 the compression and error curves.

4.2. **Real data.**

4.2.1. *MNIST Handwritten Digits.* We first consider the MNIST data set of images of handwritten digits[1], each of size 28 × 28. We use the digits 0 and 1, and randomly sample for each digit 3000 images from the database. Fig. 10 displays a small subset of the sample images of the two digits, as well as all 6000 sample images projected into top three PCA dimensions. We apply the algorithm to construct the geometric wavelets and show the wavelet coefficients and the reconstruction errors at all scales in Fig. 11. We select local dimensions for scaling functions by keeping 50% and 95% of the variance, respectively, at the nonleaf and leaf nodes. We observe that the magnitudes of the coefficients stops decaying after a certain scale. This indicates that the data is not on a smooth manifold. We expect optimization of the tree and

[1] Available at `http://yann.lecun.com/exdb/mnist/`.

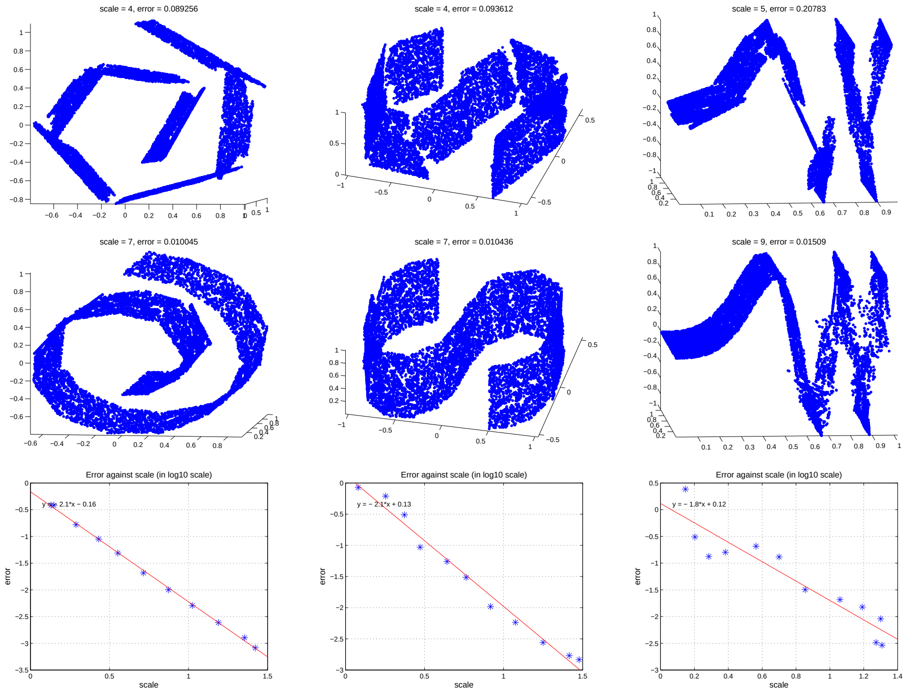

FIGURE 7. Top and Middle: Reconstructions by the algorithm of the three toy data sets in Fig. 5 at two selected scales. Bottom: Reconstruction errors as a function of scale.

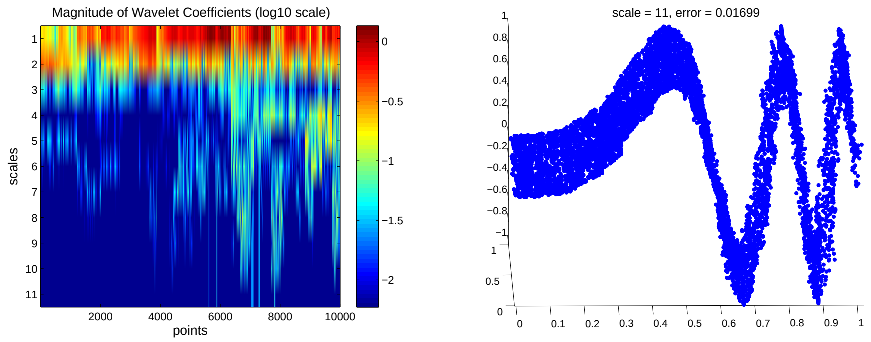

FIGURE 8. We threshold the wavelet coefficients of the *Oscillating2DWave* data at the level of .01 and prune the dyadic tree accordingly. The figure, from left to right, respectively shows the reduced matrix of wavelet coefficients (only their magnitudes), and the corresponding best approximation of the manifold.

of the wavelet dimensions in future work to lead to a more efficient representation in this case.

We then fix a data point (or equivalently an image), for each digit, and show in Fig. 12 its reconstructed coordinates at all scales and the corresponding dictionary

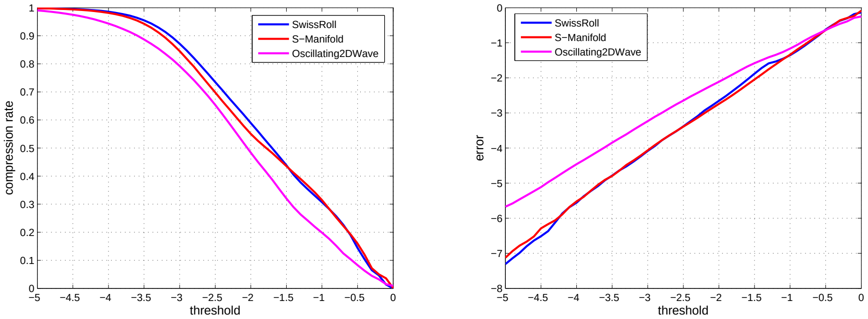

FIGURE 9. Left: the compression ratio of the matrix of the wavelet coefficients shown in Fig. 6. Right: the corresponding approximation errors. The linearity is consistent with Theorem 2.2, and essentially says that thresholding level δ generates approximation errors of order at most $O(\delta)$.

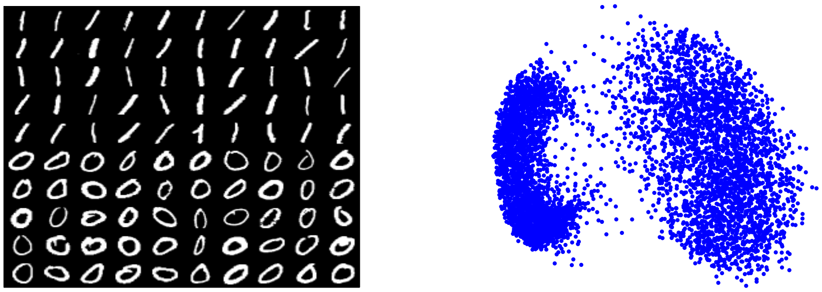

FIGURE 10. Some examples of the MNIST digits 1 and 0 (left) and 6000 sample images shown in top three PCA dimensions (right)

elements (all of which are also images). We see that at every scale we have a handwritten digit, which is an approximation to the fixed image, and those digits are refined successively to approximate the original data point. The elements of the dictionary quickly fix the orientation and the thickness, and then they add other distinguishing features of the image being approximated.

4.2.2. *Human Face Images.* We consider the cropped face images in both the Yale Face Database B[2] and the Extended Yale Face Database B[3], which are available for 38 human subjects each seen in frontal pose and under 64 illumination conditions. (Note that the original images have large background variations, sometimes even for one fixed human subject, so we decide not to use them and solely focus on the faces.) Among these 2432 images, 18 of them are corrupted, which we discard. Fig. 13 displays a random subset of the 2414 face images. Since the images have large size (192 × 168), to reduce computational complexity we first project the

[2] `http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html`

[3] `http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html`

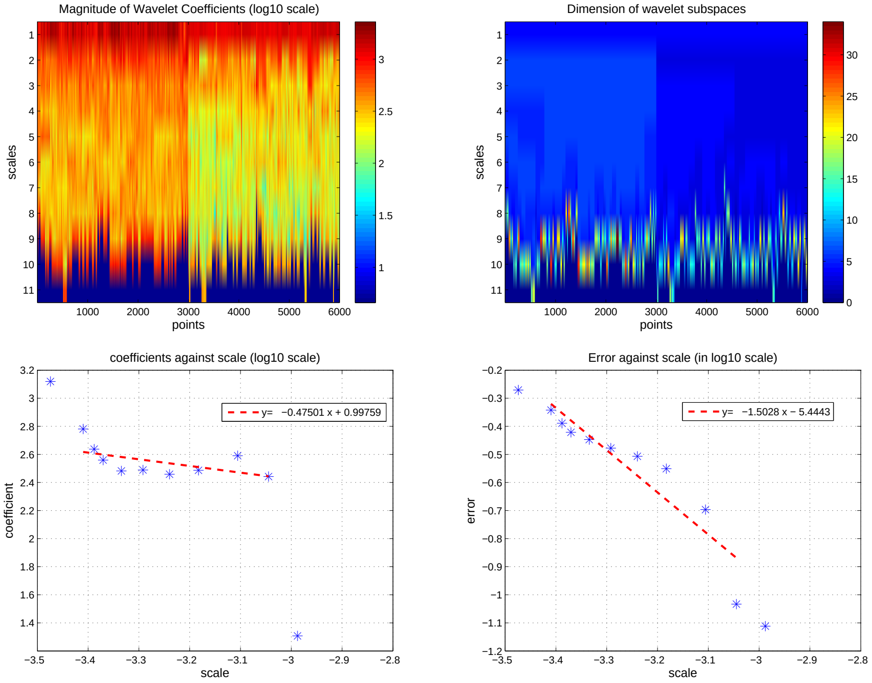

FIGURE 11. Top left: geometric wavelet representation of the MNIST digits 1 and 0. As usual, the vertical axis multi-indexes the wavelet coefficients, from coarse (top) to fine (bottom) scales: the block of entries at $(x, j), x \in C_{j,k}$ is $\log_{10} |q_{j,x}|$, where $q_{j,x}$ is the vector of geometric wavelet coefficients of x at scale j (see Sec. 3). In particular, each row indexes multiple wavelet elements, one for each $k \in \mathcal{K}_j$. Top right: dimensions of the wavelet subspaces (with the same convention as in the previous plot). Bottom: magnitude of coefficients (left) and reconstruction error (right) as functions of scale. The red lines are fitted omitting the first and last points (in each plot) in order to more closely approximate the linear part of the curve.

images into the first 500 dimensions by SVD, keeping about 99.5% variance. We apply the algorithm to the compressed data to construct the geometric wavelets and show the wavelet coefficients, dimensions and reconstruction errors at all scales in Fig. 14. Again, we have kept 50% and 95% of the variance, respectively, at the nonleaf and leaf nodes when constructing scaling functions. Note that both the magnitudes of the wavelet coefficients and the approximation errors have similar patterns with those for the MNIST digits (see Fig. 11), indicating again a lack of manifold structure in this data set. We also fix an image and show in Fig. 15 its reconstructed coordinates at all scales and the corresponding wavelet bases (all of which are also images).

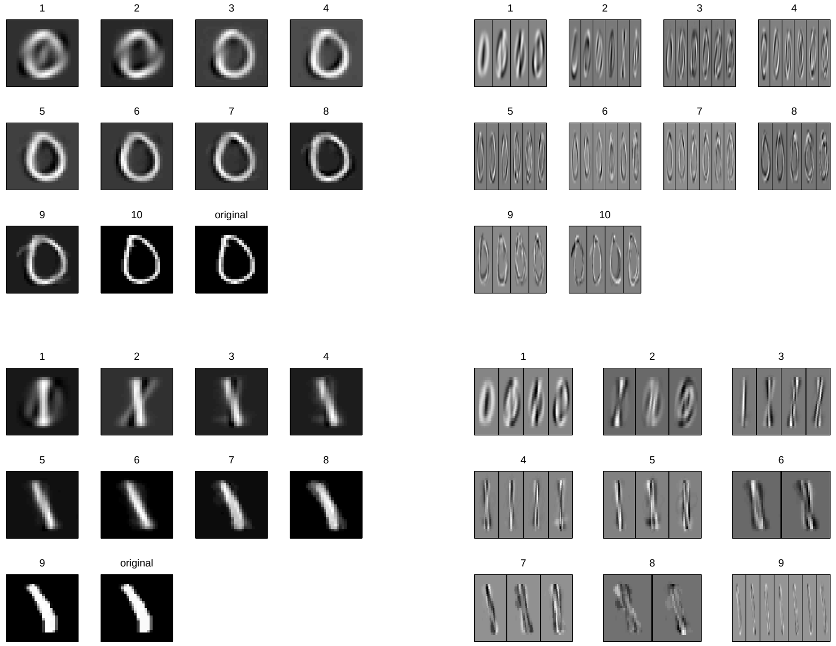

Figure 12. Left column: in each figure we plot coarse-to-fine geometric wavelet approximations of the original data point (represented in the last image). Right column: elements of the wavelet dictionary (ordered from coarsest to finest scales) used in the expansion on the left.

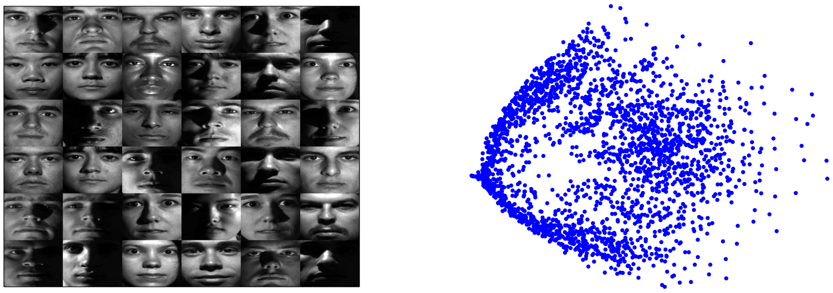

Figure 13. Left: A random subset of the 2414 face images (38 human subjects in frontal pose under 64 illumination conditions); Right: the entire data set shown in top three PCA dimensions.

5. Orthogonal Geometric Multi-Resolution Analysis

Neither the vectors $Q_{\mathcal{M}_{j+1}}(x)$, nor any of the terms that comprise them, are in general orthogonal across scales. On the one hand, this is natural since $\mathcal{M}$ is

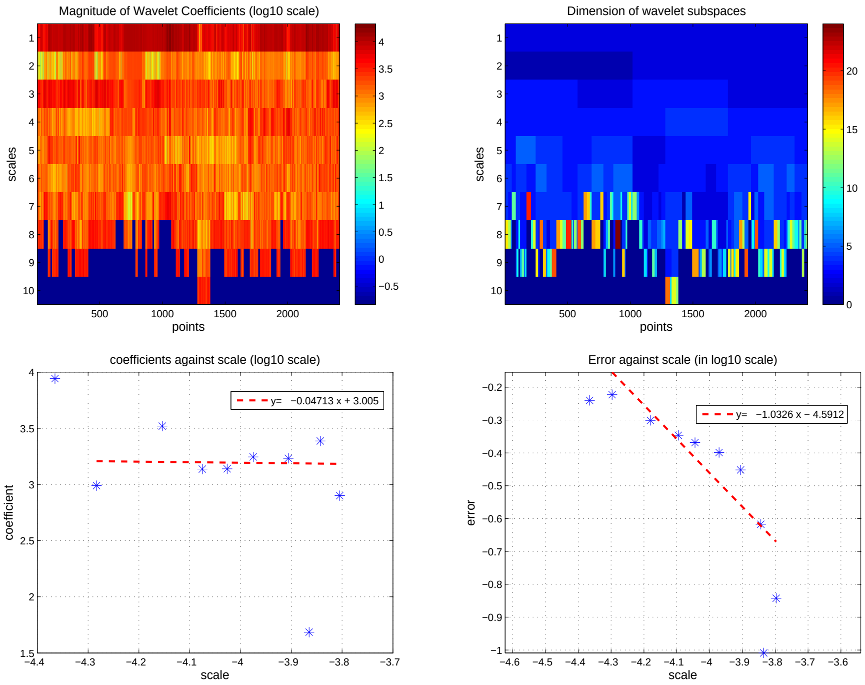

FIGURE 14. Top left: magnitudes of the wavelet coefficients of the cropped faces (2414 images) arranged in a tree. Top right: dimensions of the wavelet subspaces. Bottom: magnitude of coefficients (left) and reconstruction error (right) as functions of scale. The red lines are fitted omitting the first and last points (in each plot) in order to more closely approximate the linear part of the curve.

nonlinear, and the lack of orthogonality here is a consequence of that. On the other hand, the $Q_{\mathcal{M}_{j+1}}(x)$ may be almost parallel across scales or, for example, the subspaces $W_{j+1,x}$ may share directions across scales. If that was the case, we could more efficiently encode the dictionary by not encoding shared directions twice. A different construction of geometric wavelets achieves this. We describe this modification with a coarse-to-fine algorithm, which seems most natural. We start at scales 0 and 1, letting

$$S_{0,x} = V_{0,x} \quad , \quad S_{1,x} = S_{0,x} \oplus W_{1,x} \quad , \quad U_{1,x} = W_{1,x}, \tag{5.1}$$

and for $j \geq 1$,

$$U_{j+1,x} = P_{S_{j,x}^{\perp}}(W_{j+1,x}) \quad , \quad S_{j+1,x} = S_{j,x} \oplus U_{j+1,x} \tag{5.2}$$

Observe that the sequence of subspaces $S_{j,x}$ is increasing: $S_{0,x} \subseteq S_{1,x} \subseteq \cdots \subseteq S_{j,x} \subseteq \ldots$ and the subspace $U_{j+1,x}$ is exactly the orthogonal complement of $S_{j,x}$ into $S_{j+1,x}$. This is a situation analogous to that of classical wavelet theory. Also, we may write

$$W_{j+1,x} = U_{j+1,x} \oplus P_{S_{j,x}}(W_{j+1,x}) \tag{5.3}$$

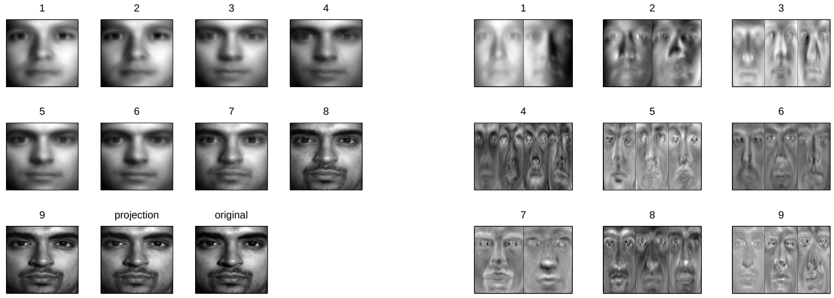

FIGURE 15. Left: in images 1-9 we plot coarse-to-fine geometric wavelet approximations of the projection and the original data point (represented in the last two images). Right: elements of the wavelet dictionary (ordered from coarse to fine in 1-9) used in the expansion on the left.

where the direct sum is orthogonal. At each scale j we do not need to construct a new wavelet basis for each $W_{j+1,x}$, but we only need to construct a new basis for $U_{j+1,x}$, and express $Q_{j+1,x}(x)$ in terms of this new basis, and the wavelet and scaling function bases constructed at the previous scales. This reduces the cost of encoding the wavelet dictionary as soon as $\dim(U_{j+1,x}) < \dim(W_{j+1,x})$ which, as we shall see, may occur in both artificial and real world examples. From a geometrical perspective, this roughly corresponds to the normal space to $\mathcal{M}$ at a point not varying much at fine scales.

Finally, we note that we can define new projections of a point x into these subspaces $S_{j,x}$:

$$s_{j,x} = P_{S_{j,x}}(x - c_{j,x}) + c_{j,x}. \tag{5.4}$$

Note that since $V_{j,x} \subseteq S_{j,x}$, $s_{j,x}$ is a better approximation than x_j to x at scale j (in the least squares sense). Also,

$$s_{j+1,x} - s_{j,x} = U_{j+1,x}U^*_{j+1,x}(x - c_{j+1,x}) + (I - P_{S_{j,x}})(c_{j+1,x} - c_{j,x}). \tag{5.5}$$

We display in Figs. 16, 17, 18 pseudo-codes for the orthogonal GMRA and the corresponding forward and inverse transforms. The reader may want to compare with the corresponding routines for the regular GMRA construction, displayed in Figs. 2, 3, 4. Note that as the name suggests, the wavelet bases $\Psi_{j,k}$ along any path down the tree are mutually orthogonal. Moreover, the local scaling function at each node of such a path is effectively the union of the wavelet bases of the node itself and its ancestors. Therefore, the Orthogonal GMRA tree will have small height if the data set has a globally low dimensional structure, i.e., there is small number of normal directions in which the manifold curves.

Example: A connection to Fourier analysis. Consider band-limited functions of band B:

$$BF_B = \{f : \text{supp.}\, \hat{f} \subseteq [-B\pi, B\pi]\}\,. \tag{5.6}$$

Classes of smooth functions (e.g. $W^{k,2}$) are essentially characterized by their L^2-energy in dyadic spectral bands of the form $[-2^{j+1}\pi, -2^j\pi] \cup [2^j\pi, 2^{j+1}\pi]$, i.e. by

```
OrthoGMRA = OrthogonalGMRA ($X_n, \tau_0, \epsilon$)

// **Input:**
// $X_n$: a set of $n$ samples from $\mathcal{M}$
// $\tau_0$: some method for choosing local dimensions
// $\epsilon$: precision
// **Output:**
// A tree $\mathcal{T}$ of dyadic cells $\{C_{j,k}\}$ with their local means $\{c_{j,k}\}$, and a family of orthogonal geometric wavelets $\{U_{j,k}\}$, and corresponding translations $\{w_{j,k}\}$

Construct the cells $C_{j,k}$, and form a dyadic tree $\mathcal{T}$ with local centers $c_{j,k}$.
Let $\mathrm{cov}_{0,k} = |C_{0,k}|^{-1} \sum_{x \in C_{0,k}} (x - c_{0,k})(x - c_{0,k})^*$, for $k \in \mathcal{K}_0$, and compute $\mathrm{SVD}(\mathrm{cov}_{0,k}) = \Phi_{0,k}\Sigma_{0,k}\Phi_{0,k}$ (where the dimension of $\Phi_{0,k}$ is determined by $\tau_0$).
Set $j = 0$ and $\Psi_{0,k} := \Phi_{0,k}, w_{0,k} := c_{0,k}$
Let $J$ be the maximum scale of the tree
**while** $j < J$
        **for** $k \in \mathcal{K}_j$
        Let $\Phi_{j,k}^{(cum)} = [\Psi_{\ell,k''}]_{0 \le \ell \le j}$ be the union of all wavelet bases of the cell $C_{j,k}$ and its ancestors. If the subspace spanned by $\Phi_{j,k}^{(cum)}$ can approximate the cell within the given precision $\epsilon$, then remove all the offspring of $C_{j,k}$ from the tree. Otherwise, do the following.
            Compute $\mathrm{cov}_{j+1,k'}$ and $\Phi_{j+1,k'}$, for all $k' \in \mathrm{children}(j,k)$, as above
            For each $k' \in \mathrm{children}(j,k)$, construct the wavelet bases $U_{j+1,k'}$ as the complement of $\Phi_{j+1,k'}$ in $\Phi_{j,k}^{(cum)}$. The translation $w_{j+1,k'}$ is the projection of $c_{j+1,k'} - c_{j,k}$ into the space orthogonal to that spanned by the $\Phi_{j,k}^{(cum)}$.
        **end**
        $j = j + 1$
**end**
```

FIGURE 16. Pseudo-code for the construction of an Orthogonal Geometric Multi-Resolution Analysis.

```
$\{q_{j,x}\}$ =orthoFGWT(orthoGMRA, $x$)

// **Input:** orthoGMRA structure, $x \in \mathcal{M}$
// **Output:** A sequence $\{q_{j,x}\}$ of wavelet coefficients

$r = x$
**for** $j = J$ **down to** $0$
        $q_{j,x} = U_{j,x}^*(r - c_{j,x})$
        $r = r - (U_{j,x} \cdot q_{j,x} + w_{j,x})$
**end**
```

FIGURE 17. Pseudo-code for the Forward Orthogonal Geometric Wavelet Transform

the L^2-size of their projection onto $BF_{2^{j+1}} \ominus BF_{2^j}$ (some care is needed in that smooth frequency cutoff, but this issue is not relevant for our purposes here). We

```
$\hat{x}$ =orthoIGWT(orthoGMRA,$\{q_{j,x}\}$)

// Input: orthoGMRA structure, wavelet coefficients $\{q_{j,x}\}$
// Output: Approximation $\hat{x}$ at scale $J$

$\hat{x} = 0$
for $j = 0$ to $J$
        $\hat{x} = \hat{x} + U_{j,x} q_{j,x} + w_{j,x}$
end
```

Figure 18. Pseudo-code for the Inverse Orthogonal Geometric Wavelet Transform

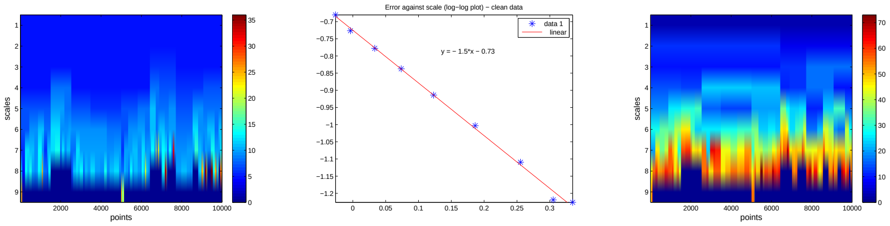

Figure 19. We construct an Orthogonal Geometric Multi-Resolution Analysis (see Sec. 5) on a random sample of 10000 band-limited functions. Left: dimension of the GMRA wavelet subspaces. Center: approximation error as a function of scale. Right: dominant frequency in each GMRA subspace, showing that frequencies are sorted from low (top, coarse GMRA scales) to high (bottom, fine GMRA scales). This implies that the geometric scaling function subspaces roughly corresponds to a Littlewood-Paley decomposition, and the GWT of a function f corresponds to a rough standard wavelet transform.

generate random smooth (band-limited!) functions as follows:

$$f(x) = \sum_{j=0}^{J} a_j(\omega) \cos(jx) \tag{5.7}$$

with a_j random Gaussian (or bounded) with mean $2^{-\lfloor \frac{j}{J} \rfloor \alpha}$ and standard deviation $2^{-\lfloor \frac{j}{J} \rfloor \alpha} \cdot \frac{1}{5}$. The GMRA associated with a random sample of this family of functions takes advantage of the multi-scale nature of the data and organizes this family of functions in a Littlewood-Paley type of decomposition: the scaling function subspace at scale j roughly corresponds to $BF_{2^{j+1}} \ominus BF_{2^j}$, and the GMRA of a point is essentially a block Fourier transform, where coefficients in the same dyadic band are grouped together. Observe that the cost of the GMRA of a point f is comparable to the cost of the Fast Fourier Transform.

6. Variations, greedy algorithms, and optimizations

We discuss several techniques for reducing the encoding cost of the geometric wavelet dictionary and/or speeding up the decay of the geometric wavelet coefficients.

6.1. Splitting of the wavelet subspaces.

Fix a cell $C_{j,k}$. For any $k' \in \text{children}(j,k)$, we may reduce the cost of encoding the subspace $W_{j+1,k'}$ by splitting it into a part that depends only on (j,k) and another on $(j+1,k')$:

$$W^{\cap}_{j,k} := \cap_{k' \in \text{children}(j,k)} W_{j+1,k'} \tag{6.1}$$

and $W^{\perp}_{j+1,k'}$ be the orthogonal complement of $W^{\cap}_{j,k}$ in $W_{j+1,k'}$. We may choose orthonormal bases $\Psi^{\cap}_{j,k}$ and $\Psi^{\perp}_{j+1,k'}$ for $W^{\cap}_{j,k}$ and $W^{\perp}_{j+1,k'}$ respectively, and let $Q^{\cap}_{j,k}$, $Q^{\perp}_{j+1,k'}$ be the associated orthogonal projections. For the data in $C_{j+1,k'}$, we have therefore constructed the geometric wavelet basis

$$\Psi_{j+1,k'} = [\Psi^{\cap}_{j,k} | \Psi^{\perp}_{j+1,k'}]\,, \tag{6.2}$$

together with orthogonal splitting of the projector

$$Q_{j+1,k'} = Q^{\cap}_{j,k} + Q^{\perp}_{j+1,k'}\,, \tag{6.3}$$

where the first term in the right-hand side only depends on the parent (j,k), and the children-dependent information necessary to go from coarse to fine is encoded in the second term. This is particularly useful when $\dim\left(W^{\cap}_{j,k}\right)$ is large relative to $\dim\left(W_{j+1,k'}\right)$.

6.2. A fine-to-coarse strategy with no tangential corrections.

In this variation, instead of the sequence of approximations $x_j = \mathbb{P}_{V_{j,x}}(x)$ to a point $x \in \mathcal{M}$, we will use the sequence $\tilde{x}_j = \mathbb{P}_{V_{j,x}}(\tilde{x}_{j+1})$, for $j < J$, and $\tilde{x}_J := x_J$. The collection $\widetilde{\mathcal{M}}_j$ of $\tilde{x}_j$ for all $x \in \mathcal{M}$ is a coarse approximation to the manifold $\mathcal{M}$ at scale j. This roughly corresponds to considering only the first term in (2.14), disregarding the tangential corrections. The advantage of this strategy is that the tangent planes and the corresponding dictionary of geometric scaling functions do not need to be encoded. The disadvantage is that the point $\tilde{x}_j$ does not have the same clear-cut interpretation as x_j has, as it is not anymore the orthogonal projection of x onto the best (in the least square sense) plane approximating $C_{j,x}$. Moreover, $\tilde{x}_j$ really depends on J: if one starts the transform at a different finest scale, the sequence changes. Notwithstanding this, if we choose J so that $||x_J - x|| < \epsilon$, for some precision $\epsilon > 0$, then this sequence does provide an efficient multi-scale encoding of x_J (and thus of x up to precision ϵ).

The claims above become clear as we derive the equations for the transform:

$$\begin{aligned} Q_{\widetilde{\mathcal{M}}_j}(\tilde{x}_{j+1}) &:= \tilde{x}_{j+1} - \tilde{x}_j = \tilde{x}_{j+1} - P_{j,x}(\tilde{x}_{j+1} - c_{j,x}) - c_{j,x} \\ &= \left(I - \Phi_{j,x}\Phi^*_{j,x}\right)\left((\tilde{x}_{j+1} - c_{j+1,x}) + (c_{j+1,x} - c_{j,x})\right). \end{aligned} \tag{6.4}$$

Noting that $\tilde{x}_{j+1} - c_{j+1,x} \in \langle \Phi_{j+1,x} \rangle$, we obtain

$$Q_{\widetilde{\mathcal{M}}_j}(\tilde{x}_{j+1}) = \Psi_{j+1,x}\Psi^*_{j+1,x}\left(\tilde{x}_{j+1} - c_{j+1,x}\right) + w_{j+1,x}, \tag{6.5}$$

where $\Psi_{j+1,x}, w_{j+1,x}$ are the same as in (2.15). By definition we still have the multi-scale equation

$$\tilde{x}_{j+1} = \tilde{x}_j + Q_{\widetilde{\mathcal{M}}_{j+1}}(\tilde{x}_{j+1}) \tag{6.6}$$

for $\{\tilde{x}_j\}$ defined as above.

6.3. **Out-of-sample extension.** In many applications it will be important to extend the geometric wavelet expansion to points that were not sampled, and/or to points that do not lie exactly on $\mathcal{M}$. For example, $\mathcal{M}$ may be composed of data points satisfying a model, but noise or outliers in the data may not lie on $\mathcal{M}$.

Fix $x \in \mathbb{R}^D$, and let J be the finest scale in the tree. Let $c_{J,x}$ be a closest point to x in the net $\{c_{J,k}\}_{k\in\mathcal{K}_J}$; such a point is unique if x is close enough to $\mathcal{M}$. For $j \leq J$, we will let (j,x) be the index of the (unique) cell at scale j that contains $c_{J,x}$. With this definition, we may calculate a geometric wavelet expansion of the point $P_{J,x}(x)$. However, $e_J(x) := x - P_{J,x}(x)$ is large if x is far from $\mathcal{M}$. We may encode this difference by greedily projecting it onto the family of linear subspaces $W_{J,x},\ldots,W_{1,x}$ and $V_{0,x}$, i.e. by computing

$$\begin{aligned}
Q_{\mathcal{M}^\perp,J}(x) &:= Q_{J,x}(e_J(x)),\\
Q_{\mathcal{M}^\perp,J-1}(x) &:= Q_{J-1,x}(e_J(x) - Q_{\mathcal{M}^\perp,J}(x)),\\
&\,\cdots\cdots\cdot\\
Q_{\mathcal{M}^\perp,0}(x) &:= P_{0,x}(e_J(x) - Q_{\mathcal{M}^\perp,J}(x) - \cdots - Q_{\mathcal{M}^\perp,1}(x)).
\end{aligned} \tag{6.7}$$

These projections encode, greedily along the multi-scale "normal" subspaces $\{Q_{j,x}\}$.

The computational complexity of this operation is comparable to that of computing two sets of wavelet coefficients, plus that of computing the nearest neighbor of x among the centers $\{c_{J,k}\}_{k\in\mathcal{K}_J}$ at the finest scale. By precomputing a tree for fast nearest neighbor computations, this essentially requires $O(\log(|\mathcal{K}_J|))$ operations. Also, observe that $|\mathcal{K}_J|$ in general does not depend on the number of points n, but on the precision in the approximation specified in the tree construction.

6.4. **Spin-cycling: multiple random partitions and trees.** Instead of one multi-scale partition and one associated tree, in various situations it may be advantageous to construct multiple multi-scale partitions and corresponding trees. This is because a single partition introduces somewhat arbitrary cuts and possible related artifacts in the approximation of $\mathcal{M}$, and in the construction of the geometric wavelets in general. Generating multiple partitions or families of approximations is a common technique in signal processing. For example, in [19] it is shown that denoising by averaging the result of thresholding on multiple shifted copies of the Haar system is as optimal (in a suitable asymptotic, minimax sense) as performing the same algorithm on a single system of smoother wavelets (and in that paper the technique was called *spin-cycling*). In the study of approximation of metric spaces by trees [2], it is well understood that using a suitable weighted average of metrics of suitably constructed trees is much more powerful than using a single tree (this may be seen already when trying to find tree metrics approximating the Euclidean metric on an interval).

In our context, it is very natural to consider a family of trees and the associated geometric wavelets, and then perform operations on either the union of such geometric wavelet systems (which would be a generalization of sorts of tight frames, in a geometric context), or perform operations on each system independently and then average. In particular, the construction of trees via cover trees [4] is very easily randomized, while still guaranteeing that each instance of such trees is well-balanced and well-suited for our purposes. We leave a detailed investigation to a future publication.

7. Data representation and compression

A generic point cloud with n points in $\mathbb{R}^D$ can trivially be stored in space Dn. If the point cloud lies, up to, say, a least-squares error (relative or absolute) ϵ in a linear subspace of dimension $d_\epsilon \ll D$, we could encode n points in space

$$\underbrace{Dd_\epsilon}_{\substack{\text{cost of}\\ \text{encoding basis}}} + \underbrace{nd_\epsilon}_{\substack{\text{cost of encoding}\\ n \text{ points}}} = d_\epsilon(D+n), \tag{7.1}$$

which is clearly much less than nD. In particular, if the d-dimensional point cloud lies is a d-dimensional subspace, then $d_\epsilon = d$.

Let us compute the cost of encoding with a geometric multi-resolution analysis a manifold $\mathcal{M}$ of dimension d sampled at n points, and fix a precision $\epsilon > 0$. We are interested in the case $n \to +\infty$. The representation we use is, as in (2.17):

$$x \sim x_J = P_{\mathcal{M}_0}(x) + \sum_{j=1}^{J} Q_{\mathcal{M}_j}(x), \tag{7.2}$$

where we choose the smallest J such that $||x - x_J|| < \epsilon$, i.e. $J = \log_2 \epsilon^{-\frac{1}{2}}$ because of Theorem 2.2. However, d_ϵ as defined above with global SVD may be as large as D in this context, even for $d = 1$.

Since $\mathcal{M}$ is nonlinear, we expect the cost of encoding a point cloud sampled from $\mathcal{M}$ to be larger than that of encoding a d-dimensional flat $\mathcal{M}$; however the geometric wavelet encoding is not much more expensive, having a cost:

$$\underbrace{dD + 2\epsilon^{-\frac{d}{2}}(d^\perp + 2^{-d}d^\cap + 2)D}_{\substack{\text{cost of}\\ \text{encoding basis}}} + \underbrace{nd(1 + \log_2 \epsilon^{-\frac{1}{2}})}_{\substack{\text{cost of encoding}\\ n \text{ points}}} \tag{7.3}$$

In Sec. 7.2 we compare this cost with that in 7.1 on several data sets. To see that the cost of the geometric wavelet encoding is as promised, we start by counting the geometric wavelet coefficients used in the multi-scale representation. Recall that $d^w_{j,x} = \operatorname{rank}(\Psi_{j,x})$ is the number of wavelet coefficients at scale j for the given point x. Clearly, $d^w_{j,k} \le d$. Then, the geometric wavelet transform of all points takes space at most

$$nd + \sum_{j=1}^{J} \sum_{x} d^w_{j,x} \le nd + ndJ \le nd(1 + \log_2 \epsilon^{-\frac{1}{2}}), \tag{7.4}$$

independently of D. The dependency on n, d is near optimal, and this shows that data points have a sparse, or rather, compressible, representation in terms of geometric wavelets. Next we compute the cost of the geometric wavelet dictionary, which contains the geometric wavelet bases $\Psi_{j,k}$, translations $w_{j,k}$, and cell centers $c_{j,k}$. If we add the tangential correction term as in (2.15), then we should also include the geometric scaling functions $\Phi_{j,k}$ in the cost. Let us assume for now that we do not need the geometric scaling functions. Define

$$d^\cap_{j,k} := \operatorname{rank}(\Psi^\cap_{j,k}) \quad , \quad d^\perp_{j+1,k'} := \operatorname{rank}(\Psi^\perp_{j+1,k'}) \tag{7.5}$$

and assume that $d^{\cap}_{j,k} \le d^{\cap}$, $d^{\perp}_{j+1,k'} \le d^{\perp}$ for fixed constants $d^{\cap}, d^{\perp} \le d$. The cost of encoding the wavelet bases $\{\Psi_{j,k}\}_{k\in\mathcal{K}_j, 0\le j\le J}$ is at most

$$\underbrace{dD}_{\text{cost of }\Psi_{0,k}} + \sum_{j=0}^{J-1} \underbrace{2^{dj}}_{\substack{\#\text{ cells}\\ \text{at scale } j}} \underbrace{d^{\cap} D}_{\substack{\text{cost of}\\ \Psi^{\cap}_{j,k}}} + \underbrace{2^{d(j+1)}}_{\substack{\#\text{ cells}\\ \text{at scale } j+1}} \underbrace{d^{\perp} D}_{\substack{\text{cost of}\\ \Psi^{\perp}_{j+1,k'}}}$$

$$= dD + \frac{2^{dJ}-1}{2^d-1}(d^{\cap}D + 2^d d^{\perp} D) \le dD + 2\epsilon^{-\frac{d}{2}}(d^{\perp} + 2^{-d}d^{\cap})D. \tag{7.6}$$

The cost of encoding $w_{j,k}, c_{j,k}$ is $2\sum_{j=0}^{J} 2^{dj} D \le 2\cdot 2^{dJ+1}\cdot D = 4D\epsilon^{-\frac{d}{2}}$. Therefore, the overall cost of the dictionary is

$$dD + 2\epsilon^{-\frac{d}{2}}(d^{\perp} + 2^{-d}d^{\cap} + 2)D. \tag{7.7}$$

In the case that we also need to encode the geometric scaling functions $\Phi_{j,k}$, we need an extra cost of $\sum_{j=0}^{J} 2^{dj} dD \le 2\epsilon^{-\frac{d}{2}} dD$.

7.1. **Pruning of the geometric wavelets tree.** In this section we discuss how to prune the geometric wavelets tree with the goal of minimizing the total cost for *ϵ-encoding* a given data set, i.e., encoding the data within the given precision $\epsilon > 0$. Since we are not interested in the intermediate approximations, we will adopt the GMRA version without adding the tangential corrections (see Sec. 6.2) and thus there is no need to encode the scaling functions. The encoding cost includes both the cost of the dictionary, defined for simplicity as the number of dictionary elements $\{\Psi_{j,k}, w_{j,k}, c_{j,k}\}$ multiplied by the ambient dimension D, and the cost of the coefficients, defined for simplicity to be the number of nonzero coefficients required to reconstruct the data up to precision ϵ.

7.1.1. *Discussion.* We fix an arbitrary nonleaf node $C_{j,k}$ of the partition tree $\mathcal{T}$ and discuss how to ϵ-encode the local data in $C_{j,k}$ in order to achieve minimal encoding cost. We assume that the data in the children nodes $C_{j+1,k'}, k' \in \text{children}(j,k)$, has been optimally ϵ-encoded by some methods, with scaling functions $\Phi_{j+1,k'}$ of dimensions $d_{j+1,k'}$ and corresponding encoding costs $\varphi_{j+1,k'}$. For example, when $C_{j+1,k'}$ is a leaf node, it can be optimally ϵ-encoded by using a local PCA plane of minimal dimension $d^{\epsilon}_{j+1,k'}$, with the corresponding encoding cost

$$\varphi_{j+1,k'} = n_{j+1,k'}\cdot d^{\epsilon}_{j+1,k'} + D\cdot d^{\epsilon}_{j+1,k'} + D, \tag{7.8}$$

where $n_{j+1,k'}$ is the size of this node.

We consider the following ways of ϵ-encoding the data in $C_{j,k}$:

(I) using the existing methods for the children $C_{j+1,k'}$ to encode the data in $C_{j,k}$ separately;

(II) using only the parent node and approximating the local data by a PCA plane of minimal dimension $d^{\epsilon}_{j,k}$ (with basis $\Phi^{\epsilon}_{j,k}$);

(III) using a multi-scale structure to encode the data in the node $C_{j,k}$, with the top $d^{w}_{j,k}$ PCA directions $\Phi^{w}_{j,k}$ being the scaling function at the parent node and $d^{w}_{j+1,k'}$ dimensional wavelets encoding differences between $\Phi_{j+1,k'}$ and $\Phi^{w}_{j,k}$. Here, $0 \le d^{w}_{j,k} \le d^{\epsilon}_{j,k}$.

We refer to the above methods as *children-only* encoding, *parent-only* encoding and *wavelet* encoding, respectively. We make the following comments. First, method (I) leads to the sparsest coefficients for each point, while method (II) produces the smallest dictionary. Second, in method (III), it is possible to use other combinations of the PCA directions as the scaling function for the parent, but we will not consider those in this paper. Lastly, the children-only and parent-only encoding methods can be thought of corresponding to special cases of the wavelet encoding method, i.e., when $d^w_{j,k} = 0$ and $d^w_{j,k} = d^\epsilon_{j,k}$, respectively.

We compare the encoding costs of the three methods above. Suppose there are $n_{j,k}$ points in the node $C_{j,k}$ and $n_{j+1,k'}$ points in each $C_{j+1,k'}$, so that $n_{j,k} = \sum_{k'} n_{j+1,k'}$. When we encode the data in $C_{j,k}$ with a $d^\epsilon_{j,k}$ dimensional plane, we need space

$$n_{j,k} \cdot d^\epsilon_{j,k} + D \cdot d^\epsilon_{j,k} + D. \tag{7.9}$$

If we use the children nodes to encode the data in $C_{j,k}$, the cost is

$$\sum_{k'} \varphi_{j+1,k'}. \tag{7.10}$$

The encoding cost of the wavelet encoding method has a more complex formula, and is obtained as follows. Suppose that we put at the parent node a $d^w_{j,k}$ dimensional scaling function consisting of the top $d^w_{j,k}$ principal vectors, where $0 \le d^w_{j,k} \le d^\epsilon_{j,k}$, and that $\Psi_{j+1,k'}$ are the corresponding wavelet bases for the children nodes. Let $d^\cap_{j,k} \ge 0$ be the dimension of the intersection of the wavelet functions, and write $d^w_{j+1,k'} = d^\cap_{j,k} + d^\perp_{j+1,k'}$. Note that the intersection only needs to be stored once for all children. Then the overall encoding cost is

$$\begin{aligned}
\varphi^w_{j,k} = {} & \underbrace{\sum_{k'} \varphi_{j+1,k'} - d_{j+1,k'}(n_{j+1,k'} + D)}_{\textit{children excluding the scaling functions and coefficients}} + \underbrace{n_{j,k} \cdot d^w_{j,k} + D \cdot d^w_{j,k} + D}_{\textit{the parent}} \\
& + \underbrace{n_{j,k} \cdot d^\cap_{j,k} + D \cdot d^\cap_{j,k}}_{\textit{intersection of children wavelets}} + \underbrace{\sum_{k'} n_{j+1,k'} \cdot d^\perp_{j+1,k'} + D \cdot d^\perp_{j+1,k'} + D}_{\textit{children-specific wavelets}} \\
= {} & \underbrace{\sum_{k'} \varphi_{j+1,k'} - (d_{j+1,k'} - d^\perp_{j+1,k'}) \cdot (n_{j+1,k'} + D)}_{\textit{new cost for children}} + \underbrace{(n_{j,k} + D) \cdot (d^w_{j,k} + d^\cap_{j,k})}_{\textit{parent and children intersection}} \\
& + \underbrace{D + \sum_{k'} D}_{\textit{parent center and wavelet translations}} .
\end{aligned} \tag{7.11}$$

Once the encoding costs in (7.9), (7.10) and (7.11) (for all $0 \le d^w_{j,k} \le d^\epsilon_{j,k}$) are all computed, we pick the method with the smallest cost for encoding the data in $C_{j,k}$, and also update $\Phi_{j,k}, \varphi_{j,k}$ correspondingly. We propose in the next section a pruning algorithm for practical realization of the above ideas.

7.1.2. *A pruning algorithm.* The algorithm requires as input a data set X_n and a precision parameter $\epsilon > 0$, and outputs a forest with orthonormal matrices $\{\Phi_{j,k}\}$ and $\{\Psi_{j,k}\}$ attached to the nodes and an associated cost function $\varphi_{j,k}$ defined on every node of the forest quantifying the cost of optimally ϵ-encoding the data in that node.

Our strategy is bottom-up. That is, we start at the leaf nodes and ϵ-encode them by using local PCA planes of minimal dimensions, and let $\{\Phi_{j,k}\}$ and $\{\varphi_{j,k}\}$ be their bases and corresponding encoding costs. We then proceed to their parents and determine the optimal way of encoding them using (7.9), (7.10) and (7.11). If the parent-only encoding achieves the minimal encoding cost, then we remove all the offspring of this node from the tree, including the children. If the children-only is the best, then we separate out the children subtrees from the tree and form new trees (we also remove the parent from the original tree and discard it). Note that these new trees are already optimized, thus we will not need to examine them again. If the wavelet encoding with some $\Phi^{w}_{j,k}$ (and corresponding wavelet bases $\Psi_{j+1,k'}$) does the best, then we update $\Phi_{j,k} := [\Phi^{w}_{j,k}\ \Phi^{\cap}_{j,k}]$ and $\varphi_{j,k}$ accordingly and let $\Phi_{j+1,k'}$ store the complement of $\Phi^{\cap}_{j,k}$ in $\Phi_{j+1,k'}$. We repeat the above steps for higher ancestors until we reach the root of the tree. We summarize these steps in Fig. 20 below.

```
PrunGMRA = PruningGMRA ($X_n, \epsilon$)

// Input:
// $X_n$: a set of $n$ samples from $\mathcal{M}$
// $\epsilon$: precision
// Output:
// A forest $\mathcal{F}$ of dyadic cells $\{C_{j,k}\}$ with their local means $\{c_{j,k}\}$ and PCA bases
$\{\Phi_{j,k}\}$, and a family of geometric wavelets $\{\Psi_{j,k}\}, \{w_{j,k}\}$, as well as encoding costs
$\{\varphi_{j,k}\}$, associated to the nodes

Construct the dyadic cells $C_{j,k}$, and form a tree $\mathcal{T}$ with local centers $c_{j,k}$.
For every leaf node in the tree $\mathcal{T}$, compute the minimal dimension $d^{\epsilon}_{j,k}$ and corre-
sponding basis $\Phi_{j,k}$ and encoding costs $\varphi_{j,k}$ for achieving precision $\epsilon$
for $j = J-1 : -1 : 1$
        • Find all the nonleaf nodes of the tree $\mathcal{T}$ at scale $j$
        • For each of the nodes $(j,k), k \in \mathcal{K}_j$,
            (1) Compute the encoding costs of the three methods, i.e., parent-only,
                children-only, and wavelet, using equations (7.9), (7.10) and (7.11).
            (2) Update $\varphi_{j,k}$ with the minimum cost.
                if parent-only is the best,
                delete all the offspring of the node from $\mathcal{T}$, and let $\Phi_{j,k} = \Phi^{\epsilon}_{j,k}$
                elseif children-only is the best,
                separate out the children subtrees from $\mathcal{T}$ and form new trees, and also
                remove and discard the parent node
                else
                update $\Phi_{j,k} := [\Phi^{w}_{j,k}\Phi^{\cap}_{j,k}]$ and $\varphi_{j,k}$ accordingly and let $\Phi_{j+1,k'}$ store the
                complement of $\Phi^{\cap}_{j,k}$ in $\Phi_{j+1,k'}$.
                end
end
```

FIGURE 20. Pseudo-code for the construction of the Pruning Geometric Wavelets

7.2. **Comparison with SVD.** In this section we compare our algorithm with Singular Value Decomposition (SVD) in terms of encoding cost for various precisions.

We may think of the SVD, being a global analysis, as providing a sort of Fourier geometric analysis of the data, to be contrasted with our GMRA, a multi-scale wavelet analysis. We use the two real data sets above, together with a new data set, the Science News, which comprises about 1100 text documents, modeled as vectors in 1000 dimensions, whose i-th entry is the frequency of the i-th word in a dictionary (see [25] for detailed information about this data set). For GMRA, we now consider three different versions: (1) the regular GMRA, but with the optimization stratigies discussed in Secs. 6.1 and 6.2 (2) the orthogonal GMRA (in Sec. 5) and (3) the pruning GMRA (in Sec. 7.1). For each version of the GMRA, we threshold the wavelet coefficients to study the rates of change of the approximation errors and encoding costs. We present three different costs: one for encoding the wavelet coefficients, one for the dictionary, and one for both (see Fig. 21).

We compare these curves with those of SVD, which is applied in two ways: first, we compute the SVD costs and errors using all possible PCA dimensions; second, we gradually threshold the full SVD coefficients and correspondingly compress the dictionary (i.e., discard those multiplying identically zero coefficients). The curves are superposed in the same plots (see the black curves in Fig. 21).

8. Computational considerations

The computational cost may be split as follows.

Construction of proximity graph: we find the k nearest neighbors of each of the n points. Using fast nearest neighbor codes (e.g. cover trees [4] and references therein) the cost is $O_{d,D}(n \log n)$, with the constant being exponential in d, the intrinsic dimension of $\mathcal{M}$, and linear in D, the ambient dimension. The cost of computing the weights for the graph is $O(knD)$.

Graph partitioning: we use METIS [42] to create a dyadic partition, with cost $O(kn \log n)$. We may (albeit in practice we do not) compress the METIS tree into a 2^d-adic tree; however, this will not change the computational complexity below.

Computation of the $\Phi_{j,k}$'s: At scale j each cell $C_{j,k}$ of the partition has a number of points $n_{j,k} = O(2^{-jd}n)$, and there are $|\mathcal{K}_j| = O(2^{jd})$ such $C_{j,k}$'s. The cost of computing the rank-d SVD in each $C_{j,k}$ is $O(n_{j,k}Dd)$, by using the algorithms of [56]. Summing over $j = 0, 1, \dots, J$ with $J \sim \log_{2^d} n$ we obtain a total cost $O(Dn \log n)$. At this point we have constructed all the $\Phi_{j,k}$'s. Observe that instead of $J \sim \log_{2^d} n$ we may stop at the coarsest scale at which a predetermined precision ϵ is reached (e.g. $J \sim \log_2 \frac{1}{\sqrt{\epsilon}}$ for a smooth manifold). In this case, the cost of this part of the algorithm only depends on ϵ and is independent of n. A similar but more complex strategy that we do not discuss here could be used also for the first two steps.

Computation of the $\Psi_{j,k}$'s: For each cell $C_{j,k}$, where $j < J$, the wavelet bases $\Psi_{j+1,k'}$, $k' \in \text{children}(j,k)$ are obtained by computing the partial SVD of a $d \times 2^d d$ matrix of rank at most d, which takes $O(D \cdot 2^d d \cdot d)$. Summing this up over all $j < J$, we get a total cost of $O(nDd^2)$.

Overall, the algorithm costs

$$O(nD(\log(n) + d^2)) + O_{d,D}(n \log n)\,. \tag{8.1}$$

The cost of performing the FGWT of a point (or its inverse) is the sum of the costs of finding the closest leaf node, projecting onto the corresponding geometric scaling

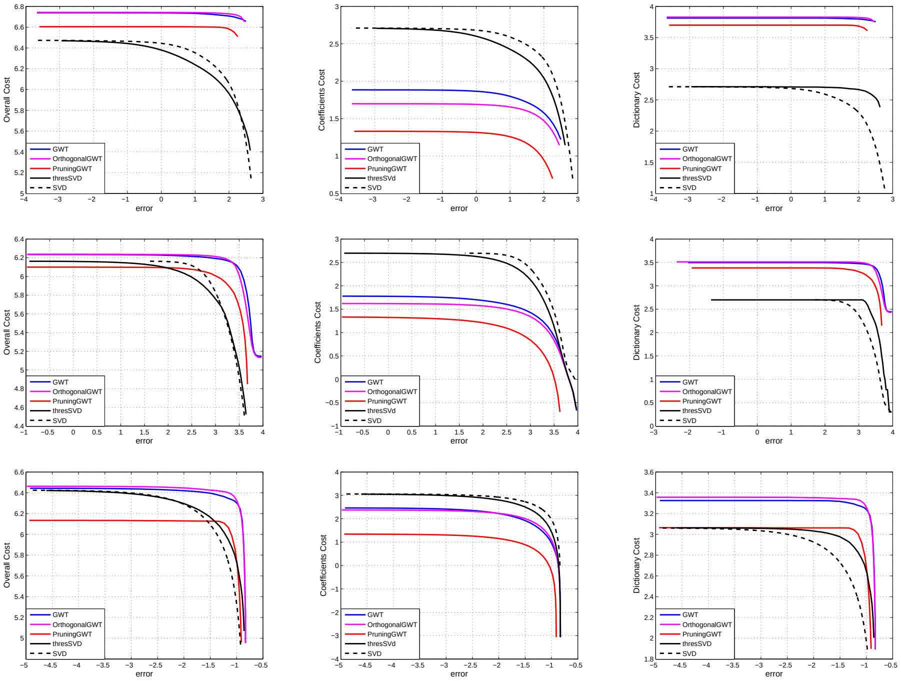

Figure 21. Cost-error curves for different kinds of encoding costs (left to right columns: overall, coefficients, dictionary) obtained on the three real data sets (top to bottom rows: MNIST digits, Yale Faces, and Science News) by the GMRA and SVD algorithms (represented by different curves in different colors). We see that all GMRA versions outperform SVD and its thresholding version in terms of coefficient costs (middle column), but take more space to store the dictionary (right column). This makes sense from the sparse coding perspective. Overall, the pruning GMRA algorithm does the best, while the other two GMRA versions have very close performance with both versions of SVD (see left column).

function plane, and then computing the multi-scale coefficients:

$$\underbrace{O_d(D\log n)}_{\substack{\text{cost of finding}\\ \text{nearest } c_{J,k}}} + \underbrace{dD}_{\substack{\text{cost of}\\ \text{projecting on } \Phi_{J,x}}} + \underbrace{O(d^2\log\epsilon^{-\frac{1}{2}})}_{\substack{\text{cost of multi-scale}\\ \text{transform}}}\ , \tag{8.2}$$

with the O_d in the first term subsuming an exponential dependence on d. The cost of the IGWT is similar, but without the first term.

We report some results in practical performance in Fig. 22.

9. A naïve attempt at modeling distributions

We present a simple example of how our techniques may be used to model measures supported on low-dimensional sets which are well-approximated by the multi-scale planes we constructed; results from more extensive investigations will be reported in an upcoming publication.

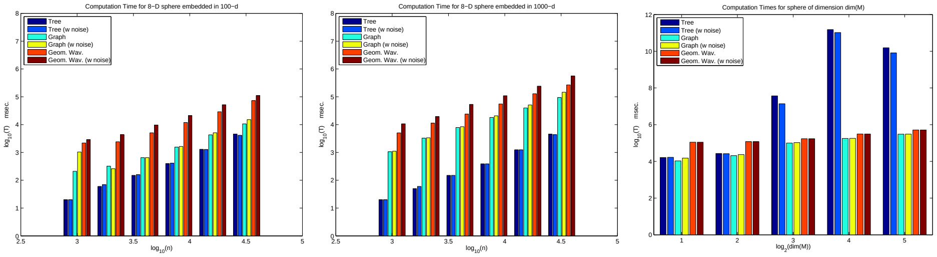

FIGURE 22. Timing experiments for the construction of geometric wavelets. We record separately the time to construct the nearest neighbor graph ('Graph'), the multi-scale partitions ('Tree'), and the geometric wavelets ('Geom. Wav.'). Left: time in *miliseconds* (on the vertical axis, in $\log_{10}$ scale) vs. n (on the horizontal axis, also $\log_{10}$ scale) for $\mathbb{S}^d(n, D, \sigma)$, for $n = 1000, 2000, 4000, 8000, 16000, 32000$, $d = 8$, $D = 100$, and $\sigma = 0, \frac{0.5}{\sqrt{D}}$. All the computational times grow linearly in n, with the noise increasing the computational time of each sub-computation. Center: same as left, but with $D = 1000$. A comparison with the experiment on the left shows that the increased ambient dimensionality does not cause, in this instance, almost any increase in the noiseless case, and in the noisy case the increase is a meager factor of 10, which is exactly the cost handling vectors which are 10 times larger in distance computations, with no curse of ambient dimensionality. Right: computation times as a function of intrinsic dimension: we vary $d = 2, 4, 8, 16, 32$ (in $\log_{10}$ scale on the horizontal axis)), and notice a mild increase in computation time, but with higher variances in the times for the computation of the multi-scale partitions.

We sample n training points from a point cloud $\mathcal{M}$ and, for a fixed scale j, we consider the coarse approximation $\mathcal{M}_j$ (defined in (2.7)), and on each local linear approximating plane $V_{j,k}$ we use the training set to construct a multi-factor Gaussian model on $C_{j,k}$: let $\pi_{j,k}$ be the estimated distribution. We also estimate from the training data the probability $\pi_j(k)$ that a given point in $\mathcal{M}$ belongs to $C_{j,k}$ (recall that j is fixed, so this is a probability distribution over the $|\mathcal{K}_j|$ labels of the planes at scale j). We may then generate new data points by drawing a $k \in \mathcal{K}_j$ according to π_j, and then drawing a point in $V_{j,k}$ from the distribution $\pi_{j,k}$: this defines a probability distribution supported on $\mathcal{M}_j$, that we denote by $p_{\mathcal{M}_j}$.

In this way we may generate new data points which are consistent with both the geometry of the approximating planes $V_{j,k}$ and with the distribution of the data on each such plane. In Fig. 23 we display the result of such modeling on a simple manifold. In Fig. 24 we construct $p_{\mathcal{M}_j}$ by training on 2000 handwritten 7's from the MNIST database, and on the same training set we train two other algorithms: the first one is based on projecting the data on the first a_j principal components, where a_j is chosen so that the cost of encoding the projection and the projected data is the same as the cost of encoding the GMRA up to scale j and the GMRA of the data, and then running the same multi-factor Gaussian model used above

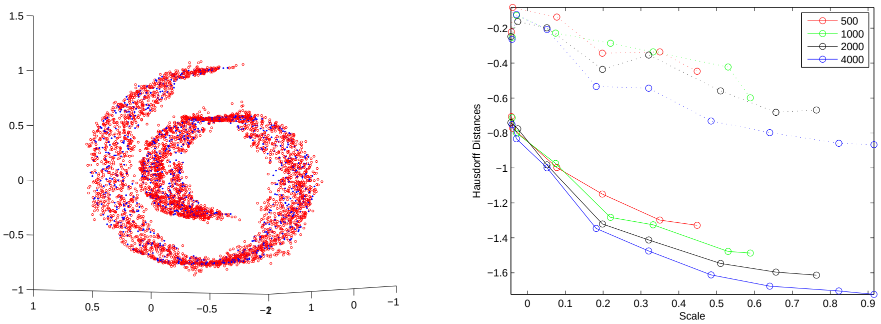

FIGURE 23. We generate a family of multi-scale models $\{p_i\}_{i=1}^4$, from $500, 1000, 2000, 4000$ (corresponding to $i = 1, \ldots, 4$) training samples from the swiss-roll manifold. Left: the blue points are 1000 training points, the red points are 4000 points generated according to p_2 at the finest scale $j = 6$. Right: for each $i = 1, \ldots, 4$ and each scale j, we generate from p_i at scale j a point cloud of 4000 samples, and measure its Hausdorff distance (dotted lines) and "Hausdorff median distance" (continuous lines) from a randomly generated point cloud with 4000 points from the true distribution on the swiss roll. The x-axis is the scale j of the model used, and colors map the size of the training set. The construction of these models and the generation of the points clouds takes a few seconds on a standard desktop.

for generating $\pi_{j,k}$. This leads to a probability distribution we denote by p_{SVD_j}. Finally, we compare with the recently-introduced Multi-Factor Analyzer (MFA) Bayesian models from [64]. In order to test the quality of these models, we consider the following two measures. The first measure is simply the Hausdorff distance between 2000 randomly chosen samples according to each model and the training set: this is measuring how close the generated samples are to the training set. The second measure quantifies if the model captures the variability of the true data, and is computed by generating multiple point clouds of 2000 points for a fixed model, and looking at the pairwise Hausdorff distances between such point clouds, called the within-model Hausdorff distance variability.

The bias-variance tradeoff in the models $p_{\mathcal{M}_j}$ is the following: as j increases the planes better model the geometry of the data (under our usual assumptions), so that the bias of the model (and the approximation error) decreases as j increases; on the other hand the sampling requirements for correctly estimating the density of $C_{j,k}$ projected on $V_{j,k}$ increases with j as less and less training points fall in $C_{j,k}$. A pruning greedy algorithm that selects, in each region of the data, the correct scale for obtaining the correct bias-variance tradeoff, depending on the samples and the geometry of the data, similar in spirit to the what has been studied in the case of multi-scale approximation of functions, will be presented in a forthcoming publication.

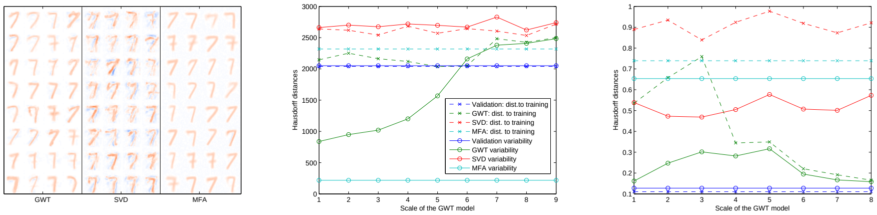

Figure 24. A training set of 2000 digits 7 from the MNIST data set are used to train probability models with GMRA ($p_{\mathcal{M}_j}$, one for each scale j in the GMRA of the training set), SVD (p_{SVD_j}, one for each GMRA scale, see text), and MFA p_{MFA}. Left: 32 digits drawn from $p_{\mathcal{M}_5}$, p_{SVD_5} and p_{MFA}: the quality of $p_{\mathcal{M}_5}$ and p_{MFA} is qualitatively better than that of p_{SVD_5}; moreover $p_{\mathcal{M}_5}$ seem to capture more variability than p_{MFA}. Center: plots of the Hausdorff distance to training set and in-model Hausdorff distance variability. We see that both $p_{\mathcal{M}_j}$ and p_{MFA} have similar distance to the training set, while p_{SVD_j}, being a model in the ambient space, generates points farther from the distribution. Looking at the plots of the in-model Hausdorff distance variability, we see that such measure increases for $p_{\mathcal{M}_j}$ as a function of j (reflecting the increasing expression power of the model), while the same measure for p_{MFA} is very small, implying that MFA fails to capture the variability of the distribution, and simply generates an almost fixed set of points (in fact, local averages of points in the training set), well-scattered along the training set. Timings: construction of GMRA and model construction for all scales for GMRA took approximately 1 min, for SVD 0.3 min, for MFA about 15 hrs. Right: a similar experiment with a training set of 2000 points from a swissroll shaped manifold with no noise: the finest scale GMRA-based models perform best (in terms of both approximation and variability, the SVD-based models are once again unable to take advantage of the low-intrinsic dimension, and MFA-based models fail as well, to succeed they seem to require tuning the parameters far from the defaults, as well as a much larger training set. Timings: construction of GMRA and model construction for all scales for GMRA took approximately 4 sec, for SVD 0.5 sec, for MFA about 4 hrs.

10. Future work

We consider this work as a first “bare bone” construction, which may be refined in a variety of ways and opens the way to many generalizations and applications. For example:

- **User interface**. We are currently developing a user interface for interacting with the geometric wavelet representation of data sets [52].
- **Higher order approximations**. One can extend the construction presented here to piecewise quadratic, or even higher order, approximators,

in order to achieve better approximation rates when the underlying set is smoother than $\mathcal{C}^2$.

- **Better encoding strategies for the geometric wavelet tree**. The techniques discussed in this paper are not expected to be optimal, and better tree pruning/tuning constructions may be devised. In particular, to optimize the encoding cost of a data set, the geometric wavelet tree should be pruned and slightly modified to use a near-minimal number of dictionary elements to achieve a given approximation precision ϵ.
- **Sparsifying dictionary**. While the approximation only depends on the subspaces $\langle\Phi_{j,k}\rangle$, the sparsity of the representation of the data points will in general depend on the choice of $\Phi_{j,k}$ and $\Psi_{j,k}$, and such choice may be optimized ("locally" in space and in dimension) by existing algorithms, thereby retaining both the approximation guarantees and the advantages of running these black-box algorithms only on small number of samples and in a low-dimensional subspace.
- **Probabilistic construction**. One may cast the whole construction in a probabilistic setting, where subspaces are enriched with distributions on those subspaces, thereby allowing geometric wavelets to generate rich families of probabilistic models.

11. Appendix

Proof of Theorem 2.2. The first equality follows by recursively applying the two-scale equation (2.16), so we only need to prove the upper bound. We start with the case $p = +\infty$. By compactness, for every $x \in \mathcal{M}$ and for j_0 large enough and $j \geq j_0$, there is a unique point $z_{j,x} \in \mathcal{M}$ closest to $c_{j,x}$, and $C_{j,x}$ is the graph of a $\mathcal{C}^{1+\alpha}$ function $f := f_{j,x} : P_{T_{z_{j,x}}}(C_{j,x}) \to C_{j,x}$, where $T_{z_{j,x}}(\mathcal{M})$ is the plane tangent to $\mathcal{M}$ at $z_{j,x}$. Note that this is true whether we construct dyadic cells $C_{j,x}$ with respect to the manifold metric ρ, or by intersecting Euclidean dyadic cubes with $\mathcal{M}$. The following calculations are in the spirit of those in [46]. Since all the quantities involved are invariant under rotations and translations, up to a change of coordinates we may assume that $f(z_{j,x}) = 0$, $T_{z_{j,x}} = \langle x_1, \ldots, x_d\rangle$. Assume $\alpha = 1$, i.e. the manifold is $\mathcal{C}^2$. In the coordinates above the function $f =: (f_1, \ldots, f_{D-d})$ above may be written

$$f_i(w) = \frac{1}{2}(w - z_{j,x})^T H_i f|_{z_{j,x}}(w - z_{j,x}) + o(||w - z_{j,x}||^2)\,,$$

where H_i is the $d \times d$ Hessian of the i-th coordinate f_i of f. The calculations in [46] show that $\mathbb{V}_{j,x}$ is parallel to $T_{z_{j,x}}$, and differs from it by a translation along the normal space $N_{c_{j,x}}$, since $\mathbb{V}_{j,x}$ passes through $c_{j,x}$ while $T_{z_{jx}}$ passes through $z_{j,x}$.

Therefore we have

$$
\begin{aligned}
&\big\| ||z - P_{\mathcal{M}_j}(z)||_{\mathbb{R}^D} \big\|_{L^\infty(C_{j,x})} = \sup_{z \in C_{j,x}} ||z - \mathbb{P}_{j,x}(z)||_{\mathbb{R}^D} \\
&= \sup_{z \in C_{j,x}} ||z - P_{T_{z_{j,x}}}(z - c_{j,x}) - c_{j,x}||_{\mathbb{R}^D} \\
&\leq \sup_{z \in C_{j,x}} ||(z - z_{j,x}) - P_{T_{z_{j,x}}}(z - z_{j,x})||_{\mathbb{R}^D} + ||z_{j,x} - c_{j,x}||_{\mathbb{R}^D} \\
&\leq \sup_{w \in P_{T_{z_{j,x}}}(C_{j,x})} \left\| \frac{1}{2}(w - z_{j,x})^* H_i f|_{z_{j,x}}(w - z_{j,x}) + o(||w - z_{j,x}||^2) \right\|_{\mathbb{R}^D} \\
&\qquad + ||z_{j,x} - c_{j,x}||_{\mathbb{R}^D} \\
&\leq 2\kappa 2^{-2j} + o(2^{-2j}) \,,
\end{aligned}
$$

where $\kappa = \frac{1}{2}\max_{i \in \{1,\dots,D-d\}} ||H_i||$ is a measure of extrinsic curvature, and where we used that $c_{j,x}$ is in the convex hull of $C_{j,x}$. A similar calculation applies to the case where $f_i \in \mathcal{C}^{1+\alpha}$, where $O(||w - z_{j,x}||^{1+\alpha})$ replaces the second order terms, and κ is replaced by $||\nabla f_i||_{\mathcal{C}^\alpha}$.

We now derive an $L^2(C_{j,x}, \mu_{j,x})$ estimate:

$$
\begin{aligned}
&\big\| ||z - P_{\mathcal{M}_j}(z)||_{\mathbb{R}^D} \big\|^2_{L^2(C_{j,x}, d\mu_{j,x}(z))} \\
&= \frac{1}{\mu(C_{j,x})} \int_{C_{j,x}} ||z - \mathbb{P}_{j,x}(z)||^2_{\mathbb{R}^D} \, d\mu(z) \\
&= \min_{\Pi:\text{ an affine } d-\text{plane}} \frac{1}{\mu(C_{j,x})} \int_{C_{j,x}} \|z - P_\Pi(z)\|^2 \frac{d\mu(z)}{d\text{vol}(z)} d\text{vol}(z) \\
&\leq \left\| \frac{d\mu}{d\text{vol}} \right\|_{L^\infty(C_{j,x})} \min_{A \text{ of rank } d} ||\text{cov}_{j,x} - A||^2_F \\
&= \left\| \frac{d\mu}{d\text{vol}} \right\|_{L^\infty(C_{j,x})} \sum_{l=d+1}^{D} \lambda_l(\text{cov}_{j,x}) \\
&\leq \left\| \frac{d\mu}{d\text{vol}} \right\|_{L^\infty(C_{j,x})} \frac{d(d+1)}{2} \lambda_{d+1}(\text{cov}_{j,x}) 2^{-4j} + o(2^{-4j}) \\
&\leq \left\| \frac{d\mu}{d\text{vol}} \right\|_{L^\infty(C_{j,x})} \max_{w \in \mathbb{S}^{D-d}} \frac{d(d+1)}{4(d+2)(d+4)} \left[\left\| \sum_{l=1}^{D-d} w_l H_l \right\|^2_F - \frac{1}{d+2} \left(\sum_{l=1}^{D-d} w_l \text{Tr}(H_l) \right)^2 \right] 2^{-4j} \\
&\qquad + o(2^{-4j}) \,,
\end{aligned}
$$

where the first inequality follows from the fact that there are no more than $d(d+1)/2$ curvature directions, and the second inequality follows from the bounds in [46], and could be strengthened in obvious ways if some decay of $\lambda_l(\text{cov}_{j,k})$ for $l = d+1, \dots, d(d+1)/2$ is assumed. The estimate in (2.17) follows. □

The measure of curvature multiplying 2^{-4j} is the last bound appeared in [46] and may be as large as $O((D-d)\kappa^2)$, but also quite small depending on the eigenvalues of the Hessians H_l.

References

[1] M. Aharon, M. Elad, and A. Bruckstein, *K-SVD: Design of dictionaries for sparse representation*, PROCEEDINGS OF SPARS 05', 2005, pp. 9–12.

[2] Yair Bartal, *Probabilistic approximation of metric spaces and its algorithmic applications*, In 37th Annual Symposium on Foundations of Computer Science, 1996, pp. 184–193.

[3] M. Belkin and P. Niyogi, *Using manifold structure for partially labelled classification*, Advances in NIPS **15** (2003).

[4] A. Beygelzimer, S. Kakade, and J. Langford, *Cover trees for nearest neighbor*, ICML, 2006, pp. 97–104.

[5] Peter Binev, Albert Cohen, Wolfgang Dahmen, Ronald Devore, and Vladimir Temlyakov, *Universal algorithms for learning theory part i: Piecewise constant functions*, Journ. Mach. Learn. **6** (2005), 1297–1321.

[6] ______, *Universal algorithms for learning theory part ii: Piecewise polynomials*, Constr. Approx. **26** (2007), no. 2, 127–152.

[7] Peter Binev, Wolfgang Dahmen, and Philipp Lamby, *Fast high-dimensional approximation with sparse occupancy trees*, Journal of Computational and Applied Mathematics **235** (2011), no. 8, 2063 – 2076.

[8] Peter Binev and Ronald Devore, *Fast computation in adaptive tree approximation*, Numer. Math (2004), 193–217.

[9] J.C. Bremer, R.R. Coifman, M. Maggioni, and A.D. Szlam, *Diffusion wavelet packets*, Appl. Comp. Harm. Anal. **21** (2006), no. 1, 95–112, (Tech. Rep. YALE/DCS/TR-1304, 2004).

[10] F. Camastra and A. Vinciarelli, *Intrinsic dimension estimation of data: An approach based on grassberger-procaccia's algorithm*, Neural Processing Letters **14** (2001), no. 1, 27–34.

[11] ______, *Estimating the intrinsic dimension of data with a fractal-based method*, IEEE P.A.M.I. **24** (2002), no. 10, 1404–10.

[12] E. Candès and D. L Donoho, *Curvelets: A surprisingly effective nonadaptive representation of objects with edges*, Curves and Surfaces (L. L. Schumaker et al., ed.), Vanderbilt University Press, Nashville, TN, 1999.

[13] Wenbo Cao and Robert Haralick, *Nonlinear manifold clustering by dimensionality*, ICPR **1** (2006), 920–924.

[14] P. G. Casazza and G. Kutyniok, *Contemp. math.*, Wavelets, Frames and Operator Theory, vol. 345, ch. Frames of Subspaces, pp. 87–113, Amer. Math. Soc., Providence, RI, 2004.

[15] E. Causevic, R.R. Coifman, R. Isenhart, A. Jacquin, E.R. John, M. Maggioni, L.S. Prichep, and F.J. Warner, *QEEG-based classification with wavelet packets and microstate features for triage applications in the ER*, vol. 3, ICASSP Proc., May 2006, 10.1109/ICASSP.2006.1660859.

[16] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders, *Atomic decomposition by basis pursuit*, SIAM Journal on Scientific Computing **20** (1998), no. 1, 33–61.

[17] Michael Christ, *A $T(b)$ theorem with remarks on analytic capacity and the Cauchy integral*, Colloq. Math. **60/61** (1990), no. 2, 601–628. MR MR1096400 (92k:42020)

[18] Ole Christensen, *An introduction to frames and Riesz bases*, Applied and Numerical Harmonic Analysis, Birkhäuser Boston Inc., Boston, MA, 2003. MR MR1946982 (2003k:42001)

[19] R. R. Coifman and D.L. Donoho, *Translation-invariant de-noising*, Springer-Verlag, 1995, pp. 125–150.

[20] 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: Diffusion maps*, PNAS **102** (2005), no. 21, 7426–7431.

[21] ______, *Geometric diffusions as a tool for harmonic analysis and structure definition of data: Multiscale methods*, PNAS **102** (2005), no. 21, 7432–7438.

[22] R.R. Coifman and S. Lafon, *Diffusion maps*, Appl. Comp. Harm. Anal. **21** (2006), no. 1, 5–30.

[23] R.R. Coifman, S. Lafon, M. Maggioni, Y. Keller, A.D. Szlam, F.J. Warner, and S.W. Zucker, *Geometries of sensor outputs, inference, and information processing*, Proc. SPIE (John C. Zolper; Eds. Intelligent Integrated Microsystems; Ravindra A. Athale, ed.), vol. 6232, May 2006, p. 623209.

[24] R.R. Coifman and M. Maggioni, *Diffusion wavelets*, Appl. Comp. Harm. Anal. **21** (2006), no. 1, 53–94, (Tech. Rep. YALE/DCS/TR-1303, Yale Univ., Sep. 2004).

[25] ______, *Multiscale data analysis with diffusion wavelets*, Proc. SIAM Bioinf. Workshop, Minneapolis (2007).

[26] ______, *Geometry analysis and signal processing on digital data, emergent structures, and knowledge building*, SIAM News, November 2008.

[27] R.R Coifman, Yves Meyer, S Quake, and Mladen V Wickerhauser, *Signal processing and compression with wavelet packets*, Progress in wavelet analysis and applications (Toulouse, 1992), Frontières, Gif, 1993, pp. 77–93. MR 95d:94004

[28] J. Costa and A.O. Hero, *Learning intrinsic dimension and intrinsic entropy of high dimensional datasets*, Proc. of EUSIPCO (Vienna), 2004.

[29] I Daubechies, *Ten lectures on wavelets*, Society for Industrial and Applied Mathematics, 1992.

[30] G. David and S. Semmes, *Uniform rectifiability and quasiminimizing sets of arbitrary codimension*, AMS.

[31] Guy David, *Wavelets and singular integrals on curves and surfaces*, Lecture Notes in Mathematics, vol. 1465, Springer-Verlag, Berlin, 1991. MR 92k:42021

[32] Guy David, *Wavelets and singular integrals on curves and surfaces*, Springer-Verlag, 1991.

[33] Guy David and Stephen Semmes, *Analysis of and on uniformly rectifiable sets*, Mathematical Surveys and Monographs, vol. 38, American Mathematical Society, Providence, RI, 1993. MR 94i:28003

[34] D. L. Donoho and C. Grimes, *When does isomap recover natural parameterization of families of articulated images?*, Tech. Report 2002-27, Department of Statistics, Stanford University, August 2002.

[35] D. L Donoho and Carrie Grimes, *Hessian eigenmaps: new locally linear embedding techniques for high-dimensional data*, Proc. Nat. Acad. Sciences (2003), 5591–5596.

[36] GH Golub and CF Van Loan, *Matrix computations*, Johns Hopkins University Press, 1989.

[37] V. De Silva J. B. Tenenbaum and J. C. Langford, *A global geometric framework for nonlinear dimensionality reduction*, Science **290** (2000), no. 5500, 2319–2323.

[38] Peter W. Jones, *Rectifiable sets and the traveling salesman problem*, Invent. Math. **102** (1990), no. 1, 1–15. MR 91i:26016

[39] ———, *The traveling salesman problem and harmonic analysis*, Publ. Mat. **35** (1991), no. 1, 259–267, Conference on Mathematical Analysis (El Escorial, 1989). MR MR1103619 (92c:42013)

[40] P.W. Jones, M. Maggioni, and R. Schul, *Manifold parametrizations by eigenfunctions of the Laplacian and heat kernels*, Proc. Nat. Acad. Sci. **105** (2008), no. 6, 1803–1808.

[41] ———, *Universal local manifold parametrizations via heat kernels and eigenfunctions of the Laplacian*, Ann. Acad. Scient. Fen. **35** (2010), 1–44, http://arxiv.org/abs/0709.1975.

[42] G. Karypis and V. Kumar, *A fast and high quality multilevel scheme for partitioning irregular graphs*, SIAM Journal on Scientific Computing **20** (1999), no. 1, 359–392.

[43] J. l. Starck, M. Elad, and D.L. Donoho, *Image decomposition via the combination of sparse representations and a variational approach*, IEEE Transactions on Image Processing **14** (2004), 1570–1582.

[44] A.V. Little, Y.-M. Jung, and M. Maggioni, *Multiscale estimation of intrinsic dimensionality of data sets*, Proc. A.A.A.I., 2009.

[45] A.V. Little, J. Lee, Y.-M. Jung, and M. Maggioni, *Estimation of intrinsic dimensionality of samples from noisy low-dimensional manifolds in high dimensions with multiscale SVD*, Proc. S.S.P., 2009.

[46] A.V. Little, M. Maggioni, and L. Rosasco, *Multiscale geometric methods for data sets I: Estimation of intrinsic dimension*, in preparation (2010).

[47] M. Maggioni, J.C. Bremer Jr., R.R. Coifman, and A.D. Szlam, *Biorthogonal diffusion wavelets for multiscale representations on manifolds and graphs*, vol. 5914, SPIE, 2005, p. 59141M.

[48] M. Maggioni and S. Mahadevan, *Fast direct policy evaluation using multiscale analysis of markov diffusion processes*, ICML 2006, 2006, pp. 601–608.

[49] S. Mahadevan and M. Maggioni, *Proto-value functions: A spectral framework for solving markov decision processes*, JMLR **8** (2007), 2169–2231.

[50] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, *Online learning for matrix factorization and sparse coding*, Journ. Mach. Learn. Res. **11** (2010), 19–60.

[51] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro, *Online dictionary learning for sparse coding*, ICML, 2009, p. 87.

[52] E.E. Monson, G. Chen, R. Brady, and M. Maggioni, *Data representation and exploration with geometric wavelets*, Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium, Dec 2010, pp. 243–244.

[53] B. A. Olshausen and D. J. Field, *Sparse coding with an overcomplete basis set: A strategy employed by V1?*, Vision Research (1997), no. 37, 3311–3325.

[54] Inam Ur Rahman, Iddo Drori, Victoria C. Stodden, and David L. Donoho, *Multiscale representations for manifold-valued data*, SIAM J. Multiscale Model. Simul **4** (2005), 1201–1232.

[55] M. A. Rohrdanz, W. Zheng, M. Maggioni, and C. Clementi, *Determination of reaction coordinates via locally scaled diffusion map*, submitted (2010).

[56] V Rokhlin, A Szlam, and M Tygert, *A randomized algorithm for principal component analysis*, SIAM Jour. Mat. Anal. Appl. **31** (2009), no. 3, 1100–1124.

[57] ST Roweis and LK Saul, *Nonlinear dimensionality reduction by locally linear embedding*, Science **290** (2000), 2323–2326.

[58] A. Szlam, *Asymptotic regularity of subdivisions of euclidean domains by iterated PCA and iterated 2-means*, Appl. Comp. Harm. Anal. **27** (2009), no. 3, 342–350.

[59] A. Szlam and G. Sapiro, *Discriminative k-metrics*, Proceedings of the 26th Annual International Conference on Machine Learning, 2009, pp. 1009–1016.

[60] A.D. Szlam, M. Maggioni, and R.R. Coifman, *Regularization on graphs with function-adapted diffusion processes*, Jour. Mach. Learn. Res. (2008), no. 9, 1711–1739, (YALE/DCS/TR1365, Yale Univ, July 2006).

[61] A.D. Szlam, M. Maggioni, R.R. Coifman, and J.C. Bremer Jr., *Diffusion-driven multiscale analysis on manifolds and graphs: top-down and bottom-up constructions*, vol. 5914-1, SPIE, 2005, p. 59141D.

[62] R Tibshirani, *Regression shrinkage and selection via the lasso*, J. Royal. Statist. Soc B. **58** (1996), no. 1, 267–288.

[63] Zhenyue Zhang and Hongyuan Zha, *Principal manifolds and nonlinear dimension reduction via local tangent space alignment*, SIAM Journal of Scientific Computing **26** (2002), 313–338.

[64] M. Zhou, H. Chen, J. Paisley, L. Ren, G. Sapiro, and L. Carin, *Non-parametric Bayesian dictionary learning for sparse image representations*, Neural and Information Processing Systems (NIPS), 2009.

Mathematics Department, Duke University, P.O. Box 90320, Durham, NC 27708, U.S.A.
E-mail address: `wka@math.duke.edu`

Mathematics Department, Duke University, P.O. Box 90320, Durham, NC 27708, U.S.A.
E-mail address: `glchen@math.duke.edu`

Mathematics and Computer Science Departments, Duke University, P.O. Box 90320, Durham, NC 27708, U.S.A.
E-mail address: `mauro@math.duke.edu (corresponding author)`