Diffusion Wavelet Packets

James C Bremer, Ronald R Coifman, Mauro Maggioni, Arthur D Szlam

Program in Applied Mathematics
Department of Mathematics
Yale University
New Haven,CT,06510
U.S.A.

Abstract

Diffusion wavelets can be constructed on manifolds, graphs and allow an efficient multiscale representation of powers of a diffusion operator acting on their domain. While diffusion wavelets are expected to perform rather well in general, in many applications it is necessary to have more versatility in choosing a basis in which to analyze, denoise, compress and manipulate a signal. Wavelet packets have proven very successful in many applications, ranging from image denoising, 2 and 3 dimensional compression of data (e.g. images, seismic data, hyperspectral data) and in discrimination tasks as well. Till now these tools for signal processing have been available only in Euclidean settings, and in low dimensions. Building upon the recent construction of diffusion wavelets, we show how to construct diffusion wavelet packets, generalizing the classical construction of wavelet packets, and allowing the same algorithms existing in the Euclidean setting to be lifted to rather general geometric and anisotropic settings, in higher dimension, on manifolds, graphs and even more general spaces. We show fast algorithms exists for computations involving these objects, discuss some applications and examples.

Key words: Wavelet Packets, Diffusion Wavelets, Local Discriminant Bases, Heat Diffusion, Laplace-Beltrami Operator, Diffusion Semigroups, Spectral Graph Theory.

Email addresses: james.bremer@yale.edu (James C Bremer), coifman@math.yale.edu (Ronald R Coifman), mauro.maggioni@yale.edu (Mauro Maggioni), arthur.szlam@yale.edu (Arthur D Szlam).
URL: www.math.yale.edu/~mmm82 (Mauro Maggioni).

1 Introduction

This is a companion paper to *Diffusion Wavelets* [1] (see also [2–4]). Diffusion wavelets arise from a multiresolution structure induced by a diffusion semigroup acting on some space, such a manifold, a graph, a space of homogeneous type, etc... This general setting includes for examples graphs with associated Laplacian, and continuous or sampled manifolds with the associated Laplace-Beltrami diffusion. Even in $\mathbb{R}^n$ they lead to the construction of new wavelet bases with interesting properties. For large classes of operators arising in applications, there exist fast algorithms to construct diffusion wavelets, compute the wavelet transform, and compute various functions of the operator (notably the associated Green's function) in compressed form.

Diffusion wavelets allow to lift to these general settings algorithms of multiscale signal processing that were before available only with wavelets in $\mathbb{R}^n$, and only with respect to simple geometric dilation operators.

Notwithstanding the level of generality and tunability of this construction, it is well-known that in several applications wavelet bases are just not flexible enough to do the job. A library of bases, whose elements are well-localized in space and frequency, offer far greater flexibility, which is needed in several applications, such as compression, denoising, discrimination. The classical wavelet packets of [5] constitute a library of bases, that can be built fast and organized hierarchically in a dyadic tree of (orthogonal) subspaces. The best basis algorithm of Coifman and Wickerhauser [5,6] is a flexible tool for selecting a basis best tailored for a given task, among which the ones above.

In this paper we present a construction of diffusion wavelet packets, which generalize the classical wavelet packets, and enrich the diffusion scaling function and wavelet bases of [1]. We show that, in many situations arising in applications, fast algorithms exist for the construction of the packets, for the analysis and synthesis of a function on the packets, and present examples suggesting that the classical best basis algorithms generalize and perform very satisfactorily the aforementioned tasks.

Material related to this paper, such as examples, and Matlab scripts for their generatation, will be made available on the web page at `http://www.math.yale.edu/~mmm82/diffusionwavelets.html`.

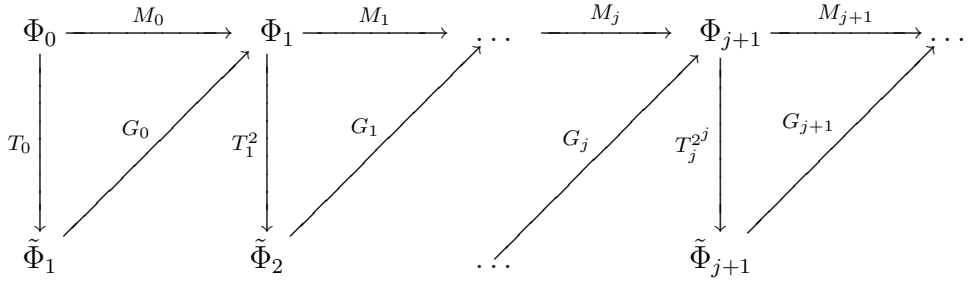

Fig. 1. Diagram for downsampling, orthogonalization and operator compression. (All triangles are commutative by construction)

2 Diffusion Wavelets

We refer the reader to [1] for the construction of diffusion wavelets and for the notation which will be used throughout this paper.

Here we simply recall some basic notation, for ease of the reader.

We restrict ourselves, for reasons of space, to the finite-dimensional, purely discrete setting. A generalization to the continuous case is presented in [1].

The following definition is from [7]:

Definition 1 *Let $\{T^t\}_{t\in[0,+\infty)}$ be a family of operators on a measure space (X,μ), each mapping $\mathcal{L}^2(X,\mu)$ into itself. Suppose this family is a semigroup, i.e. $T^0 = I$ and $T^{t_1+t_2} = T^{t_1}T^{t_2}$ for any $t_1, t_2 \in [0,+\infty)$, and $\lim_{t\to 0^+} T^t f = f$ in $\mathcal{L}^2(X,\mu)$ for any $f \in \mathcal{L}^2(X,\mu)$.*

Such a semigroup is called a symmetric diffusion semigroup if it satisfies the following:

(i) $||T^t||_p \le 1$, for every $1 \le p \le +\infty$ (contraction property).
(ii) Each T^t is compact and self-adjoint. (symmetry property).
(iii) T^t is positive: $Tf \ge 0$ for every $f \ge 0$ (a.e.) in $\mathcal{L}^2(X)$ (positivity property).
(iv) The semigroup has a generator $-\Delta$, so that

$$T^t = e^{-t\Delta}, \tag{2.1}$$

In the applications we have in mind, (X,μ) is a manifold, or a graph, and in applications to the study and decomposition of singular integrals it can be a space of homogeneous type. The operator T, on a manifold, could be the generator of Laplace-Beltrami diffusion semigroup, or, on a graph, $I - L$ where L is a graph Laplacian. See [1] for examples and discussion of the above definition.

It is easy to see that the spectrum σ_T of T is necessarily contained in $[0,1]$.

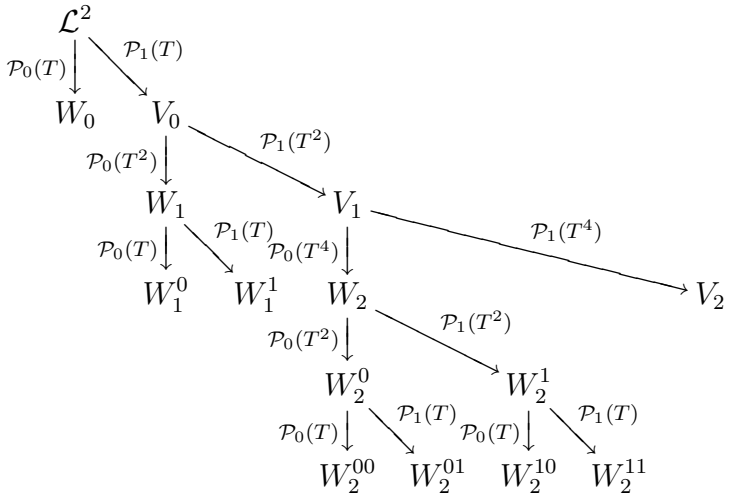

Fig. 2. Diagram for wavelet packet construction

The construction of diffusion wavelets is based on the further assumption that the spectrum is concentrated around 0. This means that high powers of T have low numerical rank, and hence can be compressed by projecting on an appropriate subspace.

The idea is then to interpret T as a dilation operator and apply dyadic powers T^{2^j} to pass from scale to scale, and as this is done an orthonormal basis of scaling functions for the numerical range of T^{2^j} is constructed. The dyadic power of T dilates the basis elements, while the orthonormalization step of the resulting functions downsamples them to an orthonormal basis of the range of that power, which constitutes the next scaling function space. Wavelets are obtained by constructing a local orthonormal basis for the orthogonal complement of each scaling space and the previous one. This is summarized in the diagram 1, and the details can be found in [1]. If the size of an eigenvalue is interpreted as inverse of frequency, diffusion scaling function spaces are "low-frequency" approximation spaces, while diffusion wavelet subspaces are "frequency-band" subspaces at different scales. The construction of diffusion wavelet packets is based upon further subdividing these wavelet subspaces, or the associated "frequency-bands".

3 Diffusion Wavelet Packets

As it happens in the classical Euclidean setting, the wavelet subspaces W_j can be split into a orthogonal sums of smaller subspaces. In this way a large family of orthonormal bases, given by the orthogonal direct sum of the orthonormal bases in the smaller subspaces, can be generated. One can then search through these bases fast, because of their dyadic tree structure, and search for the basis

best suited for a specific task [5].

In the classical setting the splitting of a wavelet subspace can be done using the same low-pass and high-pass filters used to split wavelets and scaling functions spaces. This yields a very symmetric and elegant structure. However, it has been observed that in some applications the using different filters at different splitting stages yields better results (this are sometimes called nonstationary wavelet packets, see [8] and references therein).

The construction we are going to present is not as symmetric as the one in the classical, stationary case, since the translation at different scales are not linked by simple dilation and downsampling operations, invariantly across scales. To introduce the diffusion wavelet packets, we revisit the construction of diffusion wavelets in order to generalize low-pass and high-pass filters to our setting. With this generalization it will be apparent how the classical construction can be carried through.

We take the point of view of thinking of low-pass filters as projections onto the numerical range of a (dyadic) power of T, and high-pass filters as projections onto the numerical kernel of the same (dyadic) power of T. In the classical case, T can be thought of as a (usually smooth) multiplier $\hat{T}$ on the Fourier transform side, with, say, $\hat{T}(0) = 1$, $\frac{d}{d\xi}\hat{T}|_{\xi=0} = 0$, and, say, monotonically decreasing. It is then clear that the classical multiresolution scaling function ("approximation") subspaces V_j can be made correspond to the (numerical) range of $T(2^j \cdot)$, while the multiresolution wavelet ("detail") subspaces correspond to the (numerical) kernels of the same power of T.

We proceed analogously in our setting. For any fixed $\epsilon > 0$, we define the $\epsilon-$numerical range of an operator A as

$$\mathrm{ran}_\epsilon(A) = \overline{< Av \in \mathcal{L}^2 : ||Av||_2 \geq \epsilon ||v||_2 >}.$$

The definition of $\epsilon-$numerical kernel is analogous (reverse the inequality!). If A has singular value decomposition

$$A = U \Sigma V^T,$$

where U and V are unitary, and Σ is diagonal, with positive diagonal entries $\sigma_1^2 \geq \sigma_2^2 \geq ... \geq 0$ in non-increasing order, we can write

$$Af = \sum_i \sigma_i^2 < f, v_i > u_i \ ,$$

where v_i are the columns of V and u_i the rows of U (these are $\mathcal{L}^2$ functions). Then one can define

$$A_\epsilon f = \sum_{\sigma_i^2 > \epsilon} \sigma_i^2 < f, v_i > u_i$$

and we have

$$\mathrm{ran}_\epsilon(A) = \mathrm{ran}(A_\epsilon) \ .$$

From now on we assume that $\epsilon > 0$ has been fixed, and all the ranges we will consider will be ϵ-numerical ranges, and be denoted simply by ran.

For an operator T, we have the chain of subspaces

$$\mathcal{L}^2 \supseteq \mathrm{ran}(T) \supseteq \mathrm{ran}(T^{1+2}) \supseteq \cdots \supseteq \mathrm{ran}(T^{1+2+\cdots+2^j}) \supseteq \ldots$$

which is naturally a multiresolution analysis. We let $V_0 = \mathcal{L}^2$ and, for $j \geq 1$,

$$V_j = \mathrm{ran}(T^{2^j-1}) = \ker((T^*)^{2^j-1})^\perp \ ,$$

so that the chain above reads

$$\mathcal{L}^2 = V_0 \supseteq V_1 \supseteq \cdots \supseteq V_j \supseteq \ldots \ .$$

This notation is unfortunately not consistent with the most standard one, but it seems so appropriate to our setting that we will adopt it nevertheless.

One then defines, for $j \geq 0$, W_j as the orthogonal complement of V_{j+1} in V_j, so that

$$V_{j+1} = V_j \oplus^\perp W_j \ .$$

In [2] it is shown how orthonormal bases of scaling functions for each subspace V_j can be constructed, and it is shown that in the finite dimensional case, for large classes of operators of interest in applications, an algorithm exists for constructing them fast, in time $n \log^2(n)$ where n is the cardinality of the space. The wavelet packets are an extension of this construction, where the spaces W_j are split further into smaller subspaces. Observe that we have, for $j \geq 1$,

$$W_j = \mathrm{ran}(T^{2^j-1}) \setminus \mathrm{ran}(T^{2^{j+1}-1}) = \ker((T^*)^{2^j-1})^\perp \setminus \ker((T^*)^{2^{j+1}-1})^\perp \ , \quad (3.1)$$

and $W_1 = \mathcal{L}^2 \setminus \mathrm{ran}(T)$. For a fixed $j \geq 1$, T^l, for any $l \geq 0$ is a multiplier on W_j, in the sense that $T^l(W_j) \subseteq W_j$, by (3.1). Moreover, if $l \geq 2^j$, $T^l(W_j) = 0$. From the point of view of spectral theory, we think of W_j as the Littlewood-Paley space corresponding to "frequencies" $\lambda \in [2^j, 2^{j+1})$, except that the frequencies in the general framework are given by the inverse of the eigenvalues of T, and by the spectral theorem the operator $\hat{T}$ acts in the spectral domain by multiplication of functions on its spectrum by λ.

This leaves us with 2^j powers of T, namely $T^{2^j-1}, T^{2^j-2}, \ldots, T, T^0$ acting on W_j and leaving W_j invariant. Each of them has a (numerical) range and (numerical) kernel in W_j that gives a natural splitting of W_j. In fact, different powers can be used at different splitting stages to induce a dyadic hierarchical refinement of W_j as described below. Since we will need a flexible notation

for the operation of projecting onto the (numerical) range or (numerical) kernel (which correspond, as noted above) to low- and high-pass filtering, we introduce the following:

Notation 1 *For an integer $l \geq 0$ we denote its binary representation by $\mathbf{b}(l)$. The k-th binary digit, starting from the most significative one (so for example $k = 0$ indicates the leftmost digit), will be denoted by $\epsilon_k(l) \in \{0, 1\}$.*

Notation 2 *For an operator A and $\epsilon \in \{0, 1\}$, we let*

$$\mathcal{P}_\epsilon(A) = \begin{cases} P_{\ker(A)} & , \epsilon = 0 \\ P_{\mathrm{ran}(A)} & , \epsilon = 1 \end{cases} \tag{3.2}$$

where P_V, with V close subspace, denotes the orthogonal projection onto V.

We can now define a partition of W_j into smaller subspaces as follows. For $j \geq 1$ and $s = 0, \ldots, 2^j - 1$, we let

$$P_{W_j^s} = \mathcal{P}_{\epsilon_j(s)}(T^{2^0}) \cdots \mathcal{P}_{\epsilon_1(s)}(T^{2^{j-2}}) \mathcal{P}_{\epsilon_0(s)}(T^{2^{j-1}}) \tag{3.3}$$

where W_j^s is *defined* as the range of the projection above.

There is an obvious correspondence between the binary expansion of s and the wavelet packet subspace W_j^s, and a multiscale family of dyadic subdivisions of the spectrum of T into nested partitions, in bijection with the branches of the wavelet packet tree.

Remark 2 *When the construction of the wavelet packet tree is carried till a leaf subspace has dimension 1, then this 1-dimensional subspace is spanned by an eigenfunction of T. In the numerical implementation, this is true to accuracy, mainly dependending on the spectral gaps.*

4 Construction of the wavelet packets

The actual algorithm implementing the mathematical construction described in the previous section presents several delicate aspects, most of which are common to the construction of diffusion wavelets and are discussed in [1].

The orthogonalization and "downsampling" step is again the crucial one. In the implementation of the algorithm that we used to create all the examples in this paper we used the algorithm called "modified Gram-Schmidt with mixed $\mathcal{L}^2 - \mathcal{L}^1$ pivoting" which was introduced in [1]. In the construction of diffusion wavelet packets there are of course many more orthogonalization steps, one for each splitting of the wavelet packet subspaces.

Suppose we need to split a diffusion wavelet packet subspace W_j^l into the ϵ-kernel and range of T^r. In practice one can try several approaches rather than compute precisely $\ker_\epsilon(T^r)$. In many examples presented in this paper we chose for example to split W_j^l into the η-kernel and range of T^r, where η was chosen so that these two subspaces would have the same dimension. This choice of splitting creates more symmetric packet trees, and in practice seems to be a good heuristic.

5 Examples

We consider here two examples of construction of diffusion wavelet packets.

The first example is one-dimensional, and focuses on an highly anisotropic diffusion operator. We wanted to demonstrate how it affects the structure of the wavelet packets and (below) the representation and compression of functions.

The second example is the construction of diffusion wavelet packets on the sphere, with respect to the canonical Laplace-Beltrami diffusion. We are considering the sphere, here and in the examples that follow, only out of simplicity, and ease of visualization. It is clear that our construction can be carried out on any manifold, for example with respect to the diffusion induced by Laplace-Beltrami operator, in a completely analogous way. In particular, the algorithms used had no knowledge whatsoever about the sphere or its geometry, being given uniquely the matrix representing the operator.

5.1 Example: anisotropic diffusion on a circle

This example is considered in [1], from the diffusion wavelets perspective. We start with 256 points equispaced on a circle, and consider a highly non-uniform impedance on the circle (see Figure 3). We then consider the operator T which is the discretization of the non-homogeneous heat equation on this circle, with this conductance. The scaling functions, wavelets and wavelet packets exhibit an very non-homogeneous scaling which is dependent on the conductance around the location of their support (see Figures 4 and 5).

5.2 Example: Laplace-Beltrami diffusion on a sphere

In this example we consider 2000 points uniformly randomly distributed on the sphere, and the operator T is obtained through the Laplace-Beltrami nor-

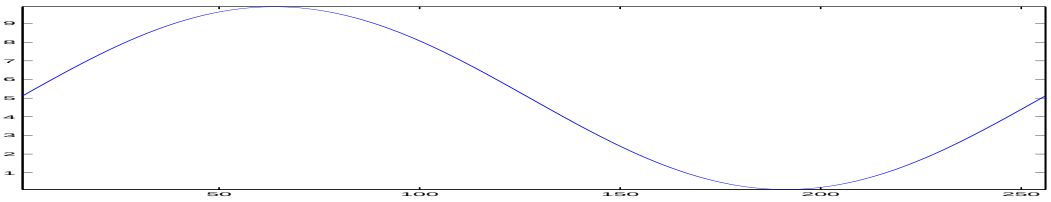

Fig. 3. Impedance of the anisotropic diffusion operator T: large on one part of the circle and almost 0 on another.

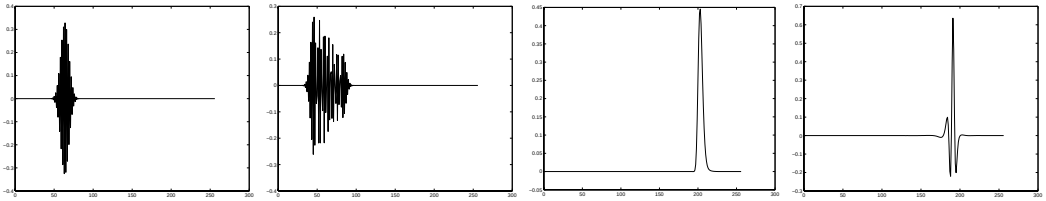

Fig. 4. Some level 7 scaling functions on the anisotropic circle. Notice that the functions exhibit high frequency behavior in high impedance regions and low frequency behavior in low impedance regions.

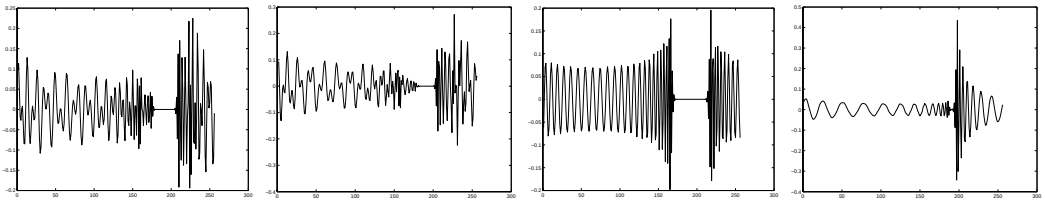

Fig. 5. Wavelet and wavelet packet functions on the anisotropic circle sampled at 256 points.

malization suggested in [3,4]. In Figure 6 we represent some diffusion wavelets and wavelet packets on the sphere.

6 Best Basis algorithms

A (diffusion) wavelet packet decomposition produces a large number of bases of $\mathcal{L}^2(X)$, each adapted to representing functions with different space and frequency localization properties. We would like to efficiently choose from this library of bases the basis which is "most suitable" for a particular task, for example for efficiently representing a particular function, or maybe for efficiently discriminate between two different classes of functions.

The Best Basis Algorithm of Coifman and Wickerhauser, introduced in [5], is a dynamic programming algorithm for choosing from a library the basis which minimizes an additive information cost function. Assuming the cost function has been well chosen for the particular application (be it compression or denoising or discrimination), the resulting basis is expected to perform well in the task at hand. For example if the goal is compression, the selected

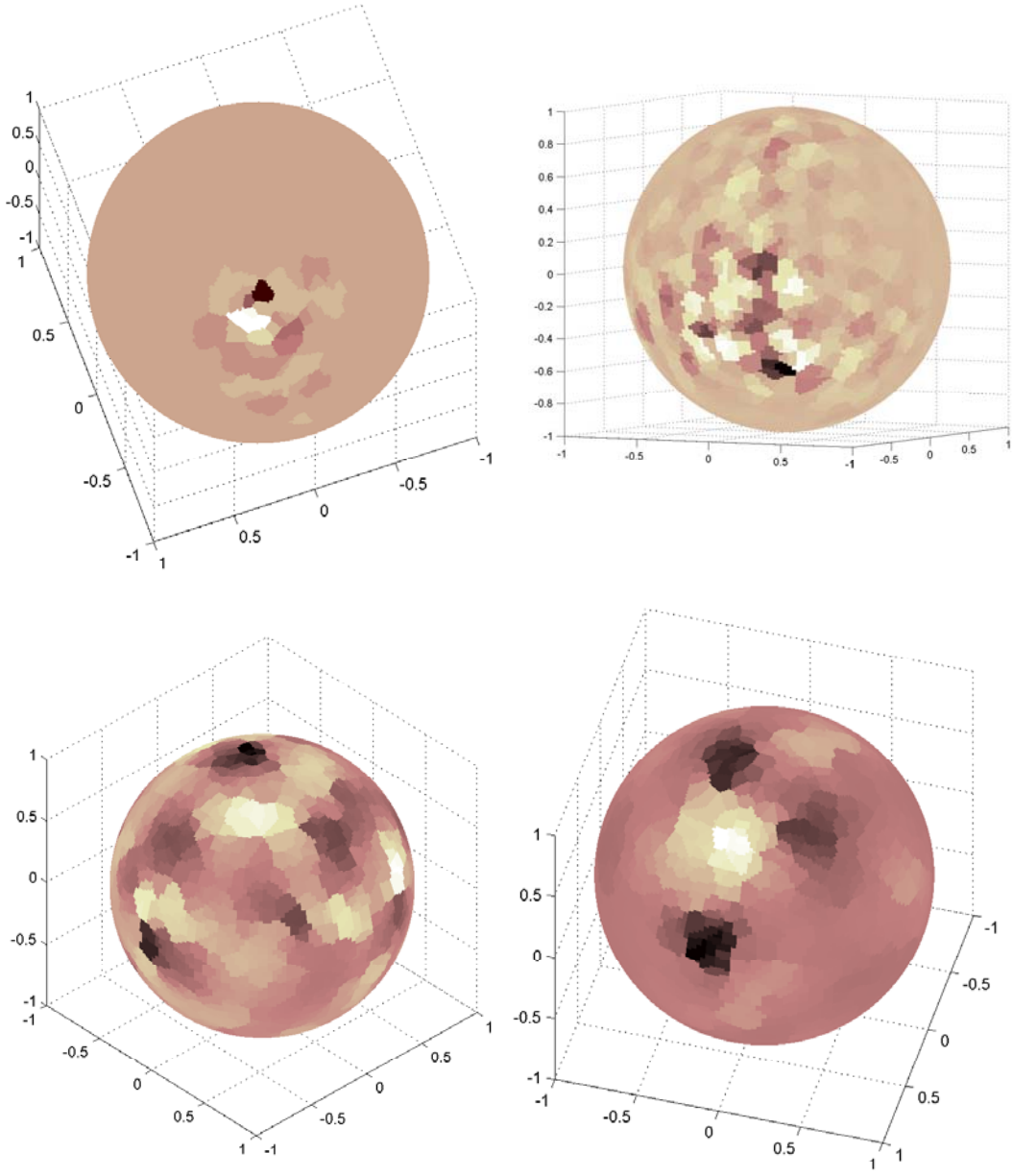

Fig. 6. Some diffusion wavelets and wavelet packets on the sphere, sampled randomly uniformly at 2000 points.

basis will be well adapted to representing the function; i.e. it will consist of waveforms which closely match the function.

We will now give a brief overview of the best basis algorithm as it applies to diffusion wavelet packets and discuss some of its applications, including compression and denoising.

The diffusion wavelet packet decomposition computes bases for a sequence V_0, V_1, W_1, V_2, W_2, V_3, W_3, W_2^0, W_2^1, ... of subspaces of $L^2(X)$ which can be arranged in a tree where each space is the direct sum of its children, as represented in Figure 3.

There are then many different ways to obtain a basis for V_0. For instance, the union of the bases from V_2, W_1, and W_0 forms one; as does the union of the bases V_3, W_2^0, W_2^1, W_1, and W_0. In general, any list of nodes such that the list does not contain an ancestor and a direct descendant, and so that every leaf is the descendant of a node in the list, produces an orthonormal basis.

Fix a function $f \in L^2(X)$ and let τ be a real-valued additive cost function; that is, let τ be a function defined on sequences $\{a_i\}$ such that $\tau(\emptyset) = 0$ and

$$\tau(\{a_i\}) = \sum_i \tau(a_i).$$

Given a basis $\mathfrak{B}$ of $L^2(X)$, let $f_{\mathfrak{B}}$ denote the sequence of coefficients of f with respect to $\mathfrak{B}$. The goal is to choose from among the library of bases given by wavelet packet decompositions the basis $\mathfrak{B}$ which minimizes $\tau(f_{\mathfrak{B}})$.

6.2 Dynamic programming

Dynamic programming algorithms are generally useful for solving optimization problems. The approach is to solve a problem by recursively combining the solution of smaller subproblems. The critical requirement for dynamic programming to be successful is that the optimal solution at a particular stage is obtained by combining the optimal solutions of its subproblems. But this is precisely the structure exhibited by the Best Basis Algorithm. Consider any non-leaf node N (i.e. node with children) in the tree. Denote its children by N_0 and N_1 and assume that we have solved the best basis problem for its children. That is, assume we have already chosen from among the many different wavelet packet bases for N_0 and N_1 those which minimize the cost function. Because the cost function is additive, the basis for N which minimizes the cost function must be either the original basis for N or the union of the bases for the subspaces N_0 and N_1.

It is now clear how to proceed. We compute the cost of the basis for each node. Then, beginning at the bottom of the tree, we compare the cost of each parent node to its children. If the cost of the children is lower, we replace the parent's basis with the union of its children's bases. When we have reached the top node, we will have found the best basis.

The best basis algorithm allows a fast search through a large library of bases for the 'best' basis with respect to some measure of information density, or concentration of energy of the coefficients. If some of the basis elements at some node are similar to the function to be compressed, then just a few coefficients suffice to capture a lot of the energy of the function using those basis elements. On $\mathbb{R}$, the usual Wavelet Packet construction (and, in fact, if the diffusion operator is a convolution with a smooth local function, the Diffusion Wavelet Packet construction) assigns locations in time-frequency to each node. If the function to be compressed consists of various elements, each with some location in time-frequency, the Best Basis algorithm supplied with a Wavelet Packet tree is good at finding and approximating these elements. If we let the spectrum of the diffusion operator stand in for frequency, the same is true for Diffusion Wavelet Packet construction. Thus, we should be able to well compress these sorts of functions with this algorithm. By changing the diffusion operator, one can change the notion of 'frequency' to suit the functions that need to be compressed (see for example the rather extreme example of compression of the anistropic diffusion on the circle below).

We note that if the data set is close to a Riemannian manifold, and the diffusion operator is an approximation of the normal heat operator, than smooth functions are well approximated by diffusion wavelet packets. This can be seen in several ways, we proceed as follows. First of all we consider the continuous case only, the discrete one follows from standard approximation arguments.

Suppose X be a compact Riemannian manifold with no boundary, and let $\{\xi_1, \xi_2, \ldots\}$ be the orthonormal basis of smooth$\mathcal{C}^\infty$ eigenfunctions of the Laplacian Δ on X with corresponding eigenvalues $\{\lambda_1, \lambda_2, \ldots\}$, in increasing order. It is well known that $\lambda_1 = 0$ and λ_k increases monotonically to $+\infty$, the eigenvalues have no accumulation point besides $+\infty$ and the eigenspace corresponding to each eigenvalue is finite dimensional.

The Sobolev space $\mathcal{H}^s(X)$, $s \in \mathbb{R}$ is defined as

$$\mathcal{H}^s = \{u \in \mathcal{D}'(X) : \sum_{k=1}^{+\infty} \lambda_k^s |a_k|^2 < +\infty\},$$

with norm

$$||u||_{\mathcal{H}^s} = \left(\sum_{k=0}^{+\infty} (1+\lambda_k)^s |a_k|^2\right)^{\frac{1}{2}}.$$

Here $\mathcal{D}'(X)$ is the set of all distributions on X acting on $\mathcal{C}^\infty$ via the usual pairing.

The Sobolev embedding Theorem applies in this context, implying that $\mathcal{H}^s \subset$

$\mathcal{C}^l(X)$ if $s > l + \frac{\dim X}{2}$. Moreover the usual duality relation $(\mathcal{H}^s)^* = \mathcal{H}_{-s}$ holds.

Consider the heat diffusion semigroup $T^t = e^{-t\Delta}$ and the associated diffusion wavelet packets. Because of the regularizing properties of T, these packets are then very smooth functions.

But they have another, more important property: each diffusion wavelet packet is orthogonal to all the eigenfunctions of the Laplace operator outside the "frequency band" on which it is supported. This follows immediately from our construction, and it is tautological since being supported in a "frequency band" in this case means exactly being in the subspace spanned by the eigenfunctions corresponding to eigenvalues in a certain band or, equivalently, being orthogonal to all the eigenfunctions outside that band.

Then the following rough estimates are routine: suppose the wavelet packet ψ is orthogonal to $\{\xi_k\}_{k \geq K}$ and $f \in \mathcal{H}^s$. We have

$$
\begin{aligned}
|<f,\psi>| &= |<\sum_{k=0}^{+\infty} <f,\xi_k>\xi_k, \psi>| \\
&\leq \sum_{k=K+1}^{+\infty} |a_k||<\xi_k,\psi>| \\
&\leq \sum_{k=K+1}^{+\infty} |a_k|(1+\lambda_k)^{\frac{s}{2}}(1+\lambda_k)^{-\frac{s}{2}} \\
&\leq \left(\sum_{k=K+1}^{+\infty} |a_k|^2(1+\lambda_k)^s\right)^{\frac{1}{2}} \left(\sum_{k=K+1}^{+\infty} (1+\lambda_k)^{-s}\right)^{\frac{1}{2}} \\
&\leq ||u||_{\mathcal{H}^s} \left(\sum_{k=K+1}^{+\infty} (1+\lambda_k)^{-s}\right)^{\frac{1}{2}}
\end{aligned}
$$

which is decreasing to 0 in K with rate dependent on the speed of convergence of $\left(\sum_{k=0}^{+\infty}(1+\lambda_k)^{-s}\right)^{\frac{1}{2}}$. This rate of convergence improves for increasing $s > 0$, as expected, depending on the growth rate of the eigenvalues $\{\lambda_k\}_k$).

Hence diffusion wavelet packets approximate well smooth (in the Sobolev sense) functions. In fact, they do so locally, since the support of most wavelet packets is small, which is a main difference from the eigenfunction expansion. The rate of convergence of a diffusion wavelet (packet) expansion will depend on the local smoothness of the functions, while the rate of convergence of the Laplace eigenfunction expansion is affected by the global smoothness. This is of course completely analogous to what happens even in one dimension on the line, where we global Fourier series correspond to (are!) Laplace eigenfunctions and diffusion wavelets correspond to classical wavelets.

We also want to remark that our diffusion does not depend only on geometrical properties of the space, since in fact the diffusion operator T can have anisotropies on its own, not dictated by the geometry of the space (e.g. curvature, like the Laplace-Beltrami operator). If we define smoothness spaces as spaces of functions well-approximated by diffusion wavelet packets associated to T, these functions spaces will depend on T and on its anisotropies. We will see a simple but yet remarkable example of the anisotropy of the spaces well-approximated by diffusion wavelet packets in the compression example on the anisotropic circle presented below. The study of these spaces is most naturally carried out in the biorthogonal setting, which we are currently investigating [9].

6.4 Best basis for denoising

If we know that a class of functions is well compressed by wavelet packets, then by thresholding the coefficients, we expect to be able to denoise functions from the class (assuming, of course, that the 'noise' is not well compressed by wavelet packets). Efficient and asymptotically optimal denoising algorithms for denoising have been studied by Donoho and Johnstone [10].

6.5 Local discriminant bases

The best basis algorithm can also be used to solve classification problems in function spaces on a data set or a manifold, using the Local Discriminant Basis algorithm of [11–13]. Instead of using entropy (or some other measure of description efficiency) to choose nodes, we use an additive cost function that measures how well a particular basis node discriminates between the classes. More specifically, we assume we have training functions f_{nm}, grouped into M classes C_m. For each basis function w, we find

$$\Gamma_{C_m}(w) = \sum_{n=1}^{|C_m|} |< w, f_{nm} >|^2 / \sum_{n=1}^{|C_m|} |f_{nm}|^2 ,$$

and then apply a discriminant measure D to the sequence $\Gamma_{C_m}(w)$, obtaining

$$\Gamma(w) = \sum_{i=1}^{M-1} \sum_{j=i+1}^{M} D(\Gamma_{C_i}(w) - \Gamma_{C_j}(w))$$

as the cost of a basis function w. The discriminant measure is chosen to reward large differences in correlations between classes; some popular choices include

$|x-y|^p$, or symmetric cross entropy,

$$D(x,y) = x \log x/y + y \log y/x \,.$$

After we have calculated the costs of the individual w's,to calculate the cost of a node, we sum the costs of the basis elements in that node. Then we run the "worst" basis algorithm, finding the basis with the highest cost. Finally, the most discriminating few basis vectors are passed into a regression scheme, hopefully allowing a large reduction in the dimensionality of the problem. Note that each step in this process is fast once we have the coefficients for the training data (with constants depending on the linearly on the number of training functions and quadratically in the number of classes) and that finding the coefficients for the training data is fast because of the properties of the diffusion wavelet transform.

This algorithm works when we are able to encode information about the geometry of the dataset in the diffusion operator, and our function classes respect that information. For example, if the data were a discretization of a manifold in some Euclidean space (or if it lay close to a manifold), we could take the heat kernel of the manifold as our operator, calculated from the points as in [4,3]. If the classes of functions to be distinguished are differentiated by the location of their energy, or otherwise differentiated by the geometry of the manifold, then the algorithm will find and utilize this information. We can expect even better success if the functions are also differentiated by their behavior in 'frequency', i.e the function classes are localized in different regions of the spectrum of the Laplacian. Note that the only part of the algorithm operating on the original points of the dataset is the nearest neighbor search in the construction of the diffusion kernel. This has the important consequence of desensitizing the algorithm to the ambient space of the data. It also suggests that some classification problems that are not presented with an obvious geometric structure might be given one by finding an appropriate diffusion.

7 Examples

7.1 Compression on the Anisotropic Circle

We choose 256 equally spaced points on the circle and construct an anisotropic diffusion operator T with impedance as shown in Figure 3, see also [1]. We expect that the wavelet packet bases generated using T will be well adapted to representing high frequency details where the impedance of T is large and slowly varying features where it is low.

To test this thesis we construct a function F with these properties. We find the

coefficients of F and its reflection in their best diffusion wavelet packet bases using the l^1 norm as a cost function. Since the reflection has the opposite structure, that is, its high frequency component is aligned with the low impedance and its low frequency component is aligned with high impedance, we expect the representation of F to be more efficient than that of its reflection.

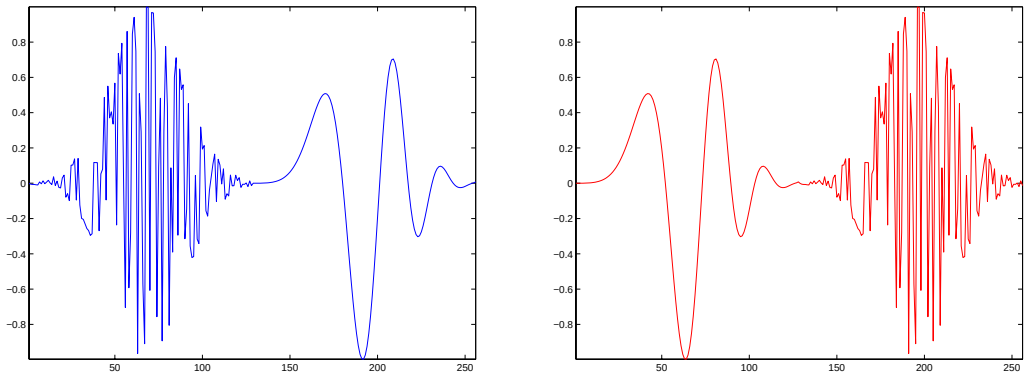

Fig. 7. The function F (on the left) and its reflection.

This is indeed what happens; Figure 8 compares the magnitude of the resulting coefficients of F and its reflection.

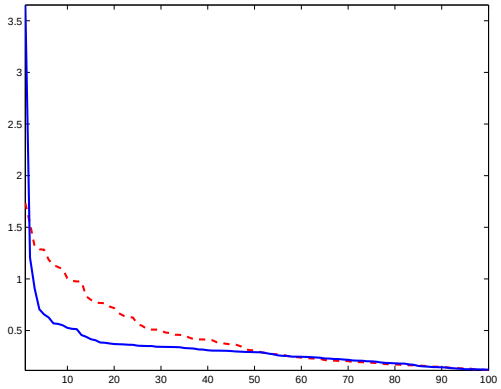

Fig. 8. Comparison of the magnitude first 50 coefficients of F and its reflection in their best wavelet packet bases.

7.2 Compression on the Sphere, I

Our data set consists of 2000 random points uniformly distributed on the unit sphere and our diffusion is the operator T with kernel $K(x, y) = \chi_{|x-y|<\epsilon}$. We construct a function F which is the sum of a trigonometric polynomial in the spherical coordinates ϕ and θ, and a sharp ridge. Figure 9 shows two different views of the function F.

We find the best diffusion wavelet packet basis using the l^1 norms of coefficients as a cost function. The coefficients of F in this basis and its coefficients in the

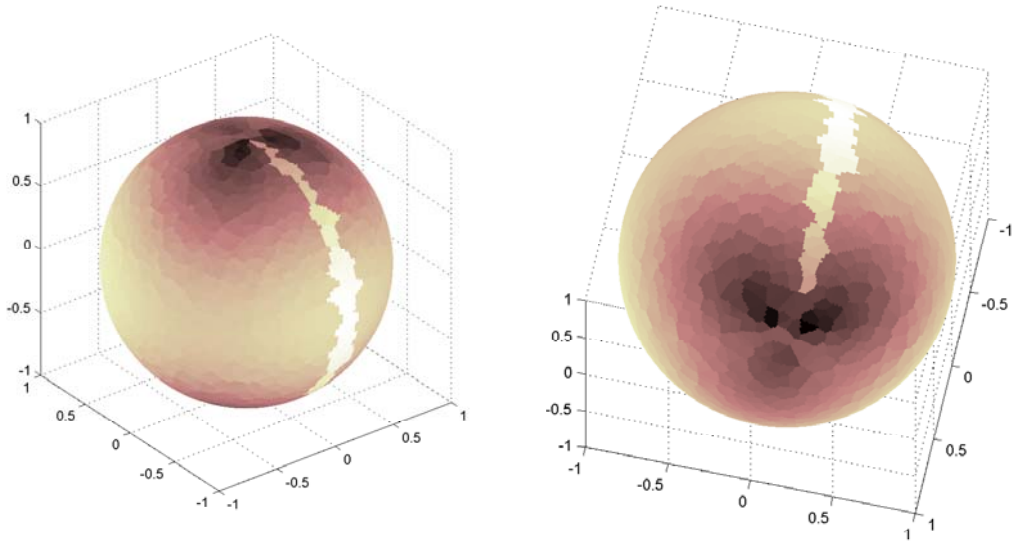

Fig. 9. Two different views of the function F on the sphere.

delta basis are compared in Figure 10. The total l^1 norm of the coefficients in the delta basis is more than 2621 while the l^1 norm of the coefficients in the best basis is approximately 216.

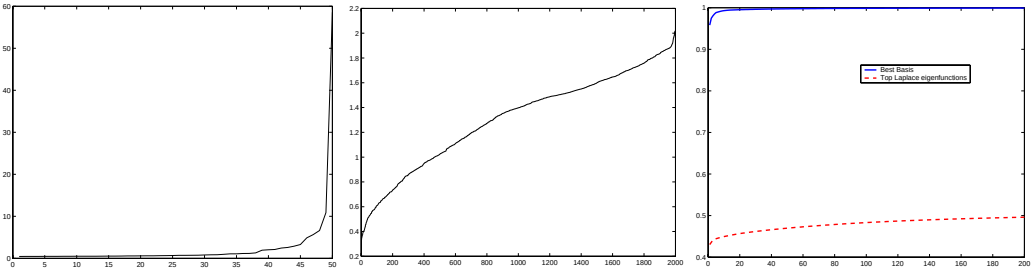

Fig. 10. Left to right: 50 top coefficients of F in its best diffusion wavelet basis, distribution coefficients$F$ in the delta basis, first 200 coefficients of F in the best basis and in the basis of eigenfunctions.

The first 20 coefficients of F in the best basis are sufficient to reconstruct 99% of its power. The reconstruction of F from the first 50 coefficients of the best basis is shown in Figure 11. Notice that both the smoothly varying and the sharp features of F are reconstructed (albeit imperfectly).

Since the function F has sharp discontinuities, we expect the best wavelet packet basis to perform better than the basis of eigenfunctions of the Laplace-Beltrami operator Δ_B. In Figure 11, we see a comparison of the coefficients in the wavelet packet best basis and eigenfunction basis. As expected, we recover the power of F much faster from the best basis.

The eigenfunctions we used for comparison are computed in the following way: we consider 10000 points uniformaly distributed on the sphere, and we consider the diffusion operator Δ obtaining by approximating the continuous Laplace-Beltrami operator Δ_B as suggested in [3,4]. We compute the eigenfunctions

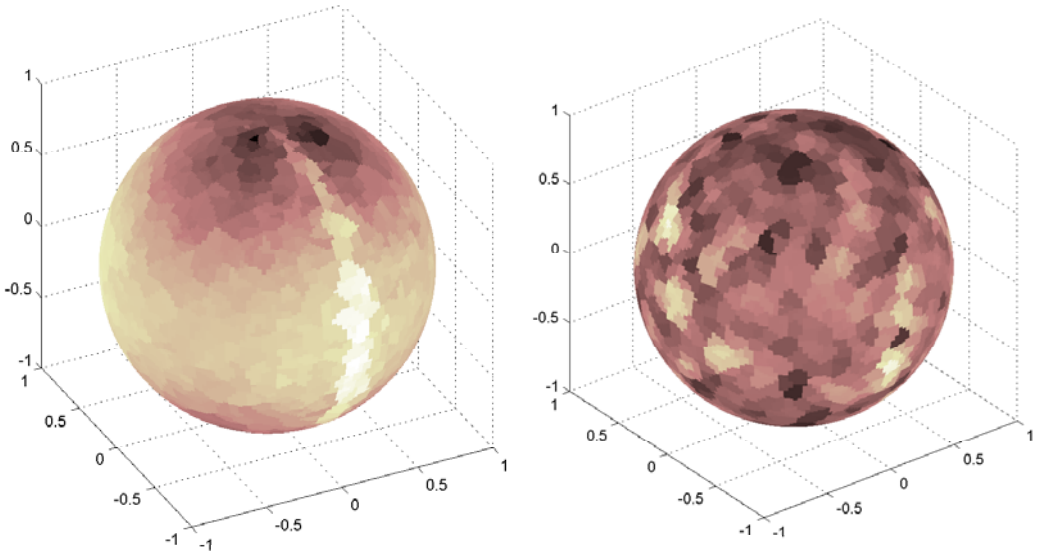

Fig. 11. Left: reconstruction of the function F with top 50 best basis packets. Right: reconstruction with top 200 eigenfunctions of the Beltrami Laplacian operator.

of this operator, and then interpolate them to the 2000 points considered in the example above. The reason is that the operator $\tilde{\Delta}$ we obtain if we try to approximate to Δ_B using only 2000 points is quite far from the true Δ_B (see the estimates in [3]). As a result, the eigenfunctions of $\tilde{\Delta}$ are terrible approximations of the true eigenfunctions of Δ_B, in particular of course for high-frequency eigenfunctions, which are important in our problem because of their role in approximating discontinuities. In fact, the high-frequency eigenfunctions of $\tilde{\Delta}$ are very well-localized (!), instead of being global spherical prolates, and do a much better job than the eigenfunctions of Δ_B.

7.3 Compression on the Sphere, II

Our dataset and operator are as before. This time we construct a function by adding a "polar cap" to a reasonably smooth ripple. Two views of F are shown in Figure 12.

Once again we use the l^1 norm as a cost function. The coefficients of F in this basis and its coefficients in the delta basis are compared in Figure 13. Once again the representation of F in the best basis is quite efficient: the l^1 norm of the best basis coefficients is 246.0366 as compared with 2366 for the delta basis.

We will now compare the best wavelet packet basis representation of F to its representation via eigenfunctions of Δ. In Figure 13 we see the power of F captured by the first n coefficients in each representation. The first 200 coefficients of F in the best basis capture more than 90 percent of its power

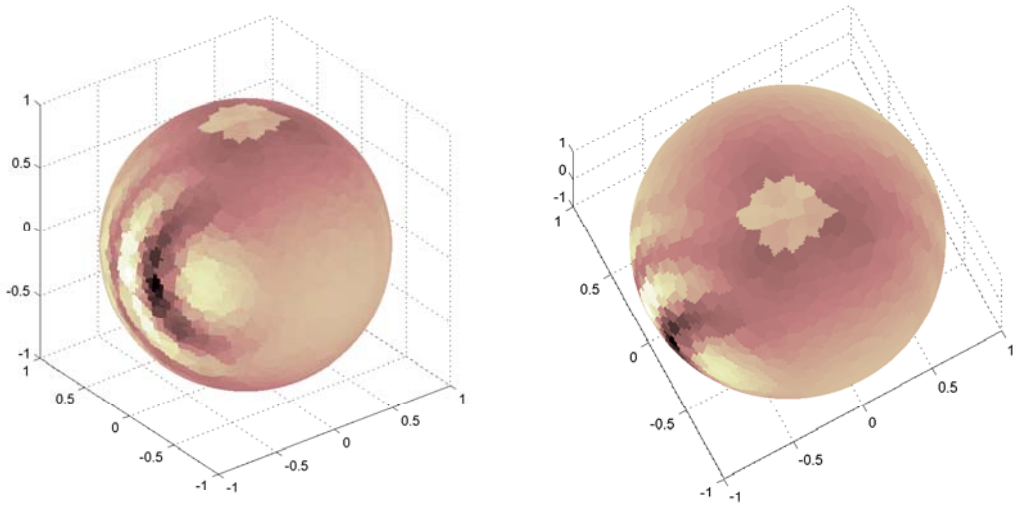

Fig. 12. Two different views of the function F on the sphere.

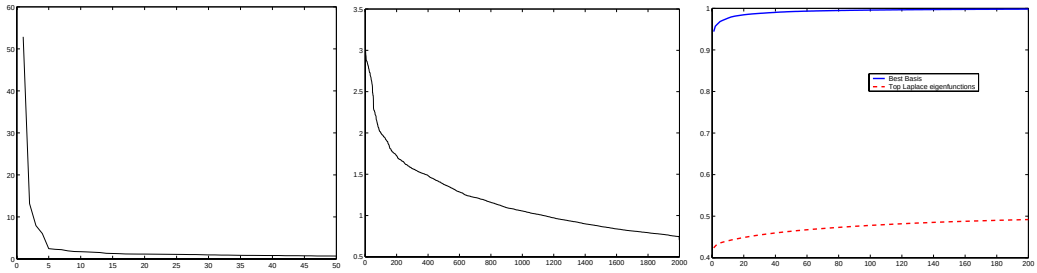

Fig. 13. Left to right: 50 top coefficients of F in its best diffusion wavelet basis, distribution coefficients$F$ in the delta basis, first 200 coefficients of F in the best basis and in the basis of eigenfunctions.

while the first 200 coefficients of F in the basis of eigenfunctions of Δ captures less than half of $F's$ power. The reconstruction of F from the first 200 coefficients of the best basis is shown in Figure 14. Since the eigenfunctions are not localized in space, the reconstruction of F from them relies on delicate cancellations. This requires a large number of coefficients to even begin to reconstruct the major features of the function F. The result of reconstructing F from 200 eigenbasis coefficients is seen in Figure 14.

7.4 De-noising on the Sphere

Now we test the best wavelet packet de-noising technique of Donoho and Johnstone [10] on the 2000 point sphere. We begin with a function G which is a periodic wave on the sphere; two views of G are given in Figure 15. We add $.25\eta$ to the function G, where η is Gaussian white noise of mean 0 and variance 1: we represent this function in Figure 16.

The best diffusion wavelet basis is found using the cost function described in

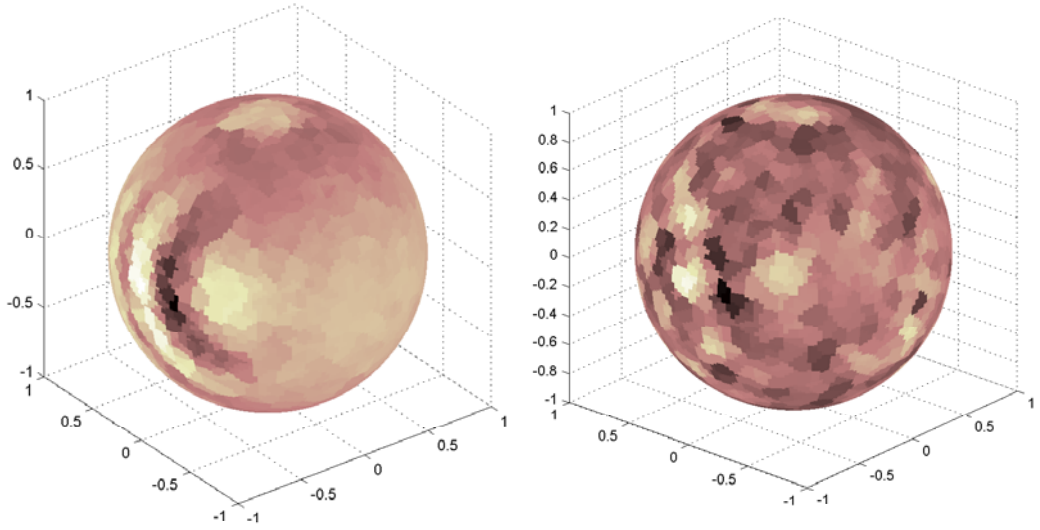

Fig. 14. Left: reconstruction of the function F from 200 best basis diffusion wavelet packet coefficients. Right: reconstruction from top 200 eigenfunctions.

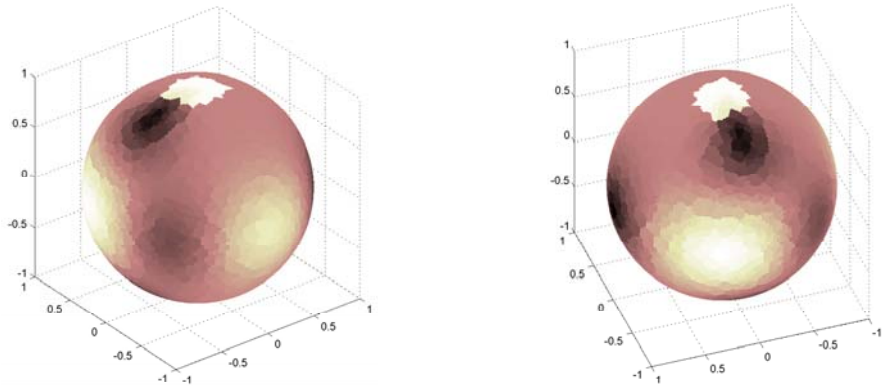

Fig. 15. Two views of G.

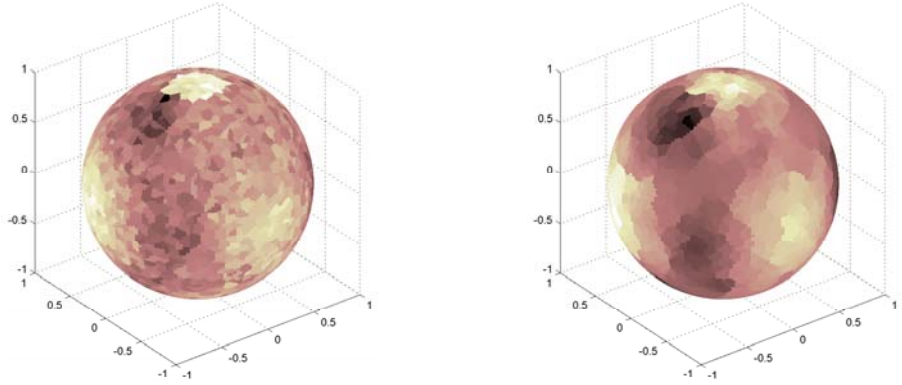

Fig. 16. Left: G with noise; right: G denoised

[5,6]. The first 40 coefficients of G in the best basis are used to reconstruct the function (see Figure 16). The Signal-to-Noise Ratio of the function G with added noise is 6.49, the Signal-to-Noise Ration of the denoised function is 10.8.

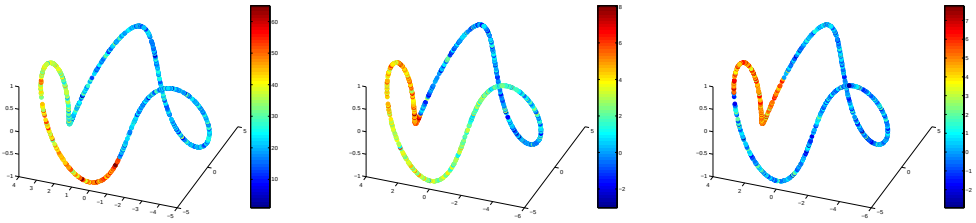

Fig. 17. Left to right, a realization of a function from class 1, 2, 3.

7.5 Local Discriminant Bases on a Space Curve

To illustrate the LDB algorithm using diffusion wavelets, we adapt an example from Naoki Saito's thesis [11]. For our data set, we choose 1024 random sample points on the curve

$$x(t) = (4 + \sin 6\pi t)(\cos 2\pi t)$$
$$y(t) = (4 + \sin 6\pi t)(\sin 2\pi t)$$
$$z(t) = \cos 6\pi t$$

with a uniform distribution. We will attempt to distinguish between three classes of synthetic functions on this data set. Let s be arclength, the length of the curve be L, a and c be uniform random variables with ranges $[L/8, L/4]$ and $[L/4, 3L/4]$ respectively, let $b = a+c$, and η be a standard normal variable. Then class 1 functions are of the form $(6 + \eta)\chi_{a,b}(s)$, class 2 functions are of the form $(6 + \eta)\chi_{a,b}(s - a)/(b - a)$, and class 3 functions are of the form $(6 + \eta)\chi_{a,b}(b - s)/(b - a)$. To all three classes of functions, we add standard normal white Gaussian noise. Note that though we define the functions by the natural parameter of the curve, we do not give the algorithm the functions in that parameter; rather, the points are given in the order they were sampled. The algorithm has to discover the parameter.

A wavelet packet tree is generated from Gaussian kernel normalized as in [3,4] to approximate the Laplace-Beltrami operator on the curve (which is standard one dimensional diffusion with respect to arclength), the 'worst' basis is chosen using symmetric cross entropy as a discriminant measure, and we select the 30 costliest basis vectors and run a CART (Classification And Regression Tree) to do the actual classification. A CART tested on the set of 999 functions in the original 1024 coordinates has a test error of .137. Using the full LDB coordinates brings the test error down to .112. The test error using only the 30 top LDB coordinates is .059. In this example, an inspection of the classification tree and chosen basis functions shows that the algorithm works because the basis functions respect the geometry of the data set, and the function classes are determined by where on the data set they have more energy.

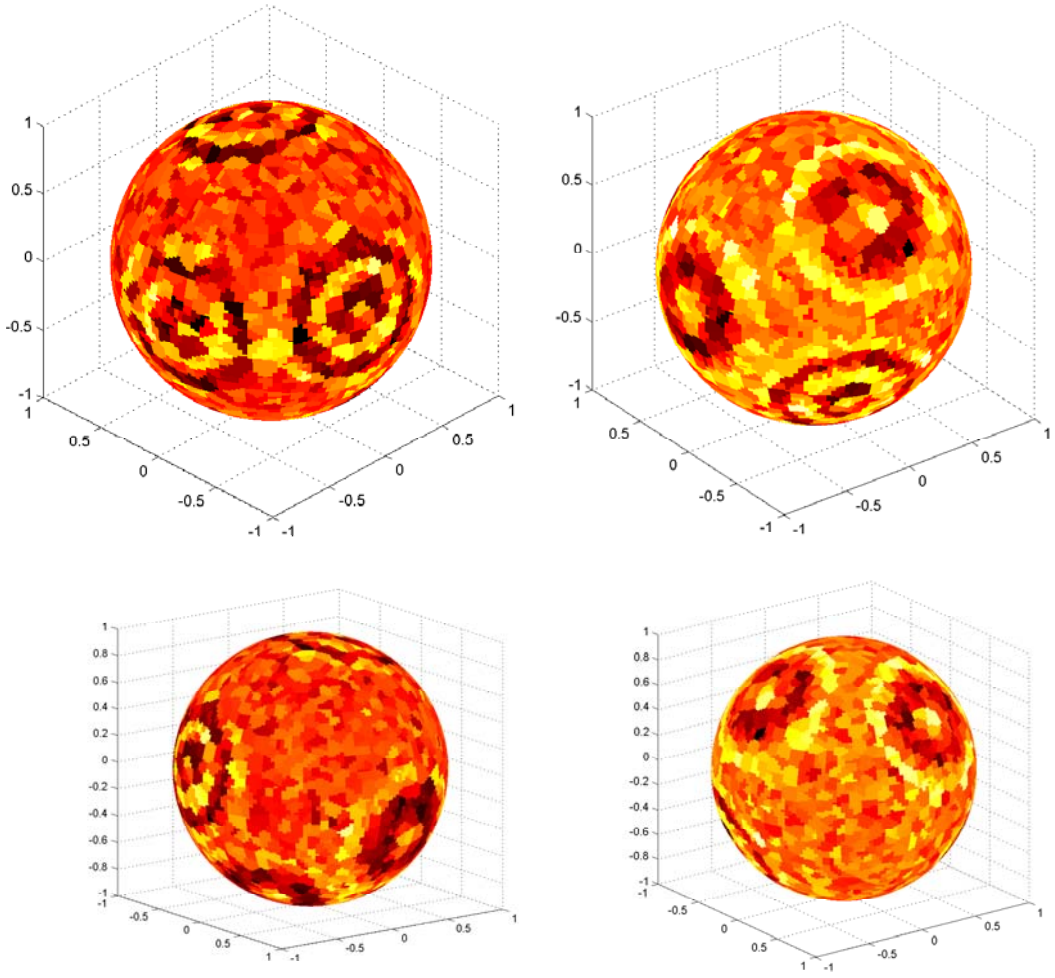

Fig. 18. Left to right, a realization of a function from class 1 and 2, top and bottom are two views of the same realization, from two antipodal points of view.

7.6 Local Discriminant Bases on the Sphere, I

We now give two examples using LDB to distinguish between classes of synthetic functions on the uniformly sampled 2000 point sphere from above. As above, we use the normalized χ operator to generate the packet tree.

For the first example, we fix a point v, and then let w be the random perturbation of v given by $w = (v + .1\eta)/||v + .1\eta||$, where η is a standard normal Gaussian in $\mathbb{R}^3$. We center a ripple of the form

$$f(x) = \chi_D \cos(C \cos^{-1} < w, x >)$$

about w, and depending on the class, C is chosen so there are one or two complete radial oscillations. Now we place 5 more ripples of random type (one or two oscillations) in random non-overlapping locations around the sphere. Finally, we add .2 white Gaussian noise. See Figure 18.

A CART run on the delta basis of the original data set has a test error of .175 with 300 training functions and 1000 test functions. If we use the symmetric cross entropy to find the top 20 LDB coordinates, we can reduce the test error to .035. Note for comparison that a CART using the first 300 eigenfunctions of our χ operator as coordinates has a .31 test error. Observe that the eigenfunctions of χ are quite different from the eigenfunctions of the continuous Laplace-Beltrami operator Δ_B on the sphere. In particular, the high-frequency eigenfunctions of χ are nicely localized and oscillating, and in general are much better suited to these discrimination tasks than good approximations to the eigenfunctions of Δ_B. This remark applies to the following example as well.

7.7 Local Discriminant Bases on the Sphere, II

For our second example, we place three equally spaced points along a random (but fixed) equator of the circle. As above, we then perturb each point by adding $.1\eta$ and renormalizing, where η is a standard normal Gaussian. Now we center a patch of texture of the form

$$f(x) = \chi_D \sin(20 \cos^{-1} < v, x >)$$

at each perturbed point, where v is either in the plane of the equator or perpendicular to it, depending on the class of the function. For class 1 functions, the textures on all three patches are the same, with equal probability that the ridges on the patches are either parallel or perpendicular to the chosen equator. For class 2 functions, after choosing the orientation of the texture patches, we randomly select one patch, and orient it the other way. We add some truncated Gaussian bumps at random locations near the poles as decoys, and as usual, we add .2 white Gaussian noise. See Figure 19.

At this point, a CART run on the original data with 300 training functions has a test error (with 1000 test functions) of .481, that is, it essentially saw no difference between the classes. Building a packet tree with the normalized χ diffusion from above, and picking the top 40 LDB vectors using the l^2 discriminant measure yields a test error rate of .125. In this example using the first 300 eigenfunctions of the χ operator as coordinates does almost as well, leading to a test error of .181.

8 Appendix: Computational Complexity

In the construction of diffusion wavelet packets, as in the construction of diffusion wavelets ([1]), one of the key steps is the orthogonalization (or down-

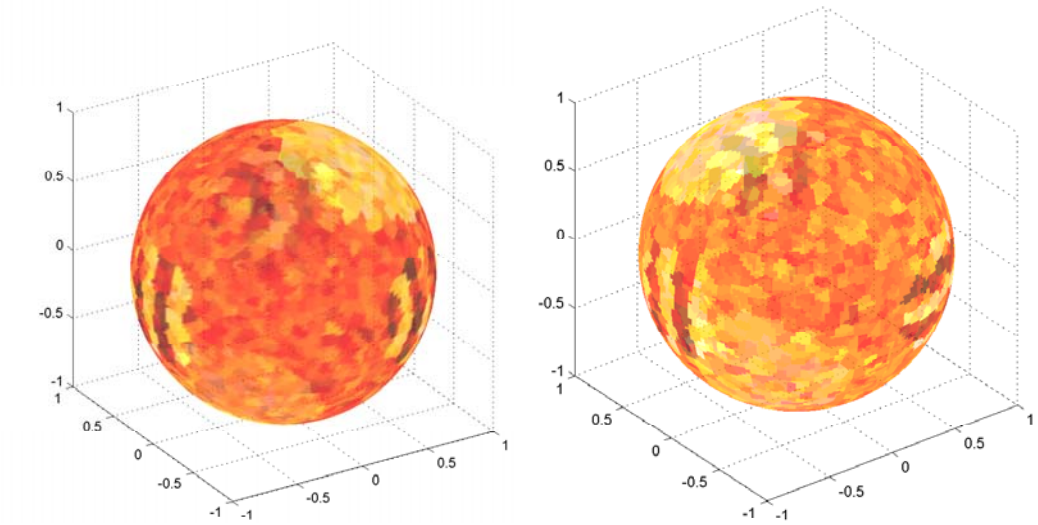

Fig. 19. Left to right, a realization of a function from class 1 and 2 respectively. Note that the third smooth texture patch is on the back side of the sphere, and can be viewed in semitransparency. The other two smooth patches are decoys in random non-overlapping positions.

sampling) step. In the paper we used the orthogonalization algorithm called modified Gram-Schmidt with mixed $\mathcal{L}^2 - \mathcal{L}^1$ pivoting in [1].

In this appendix we show that the computational complexity for computing the whole diffusion wavelet packet tree is, under certain conditions satisfied in several applications, of order $\mathcal{O}(n \log^2 n)$, which is essentially the cost of one (the most expensive) orthogonalization step. We refer the reader again to [1] for a discussion of which conditions guarantee that fast algorithms for the computation of diffusion wavelets exist: those conditions are the same we need for diffusion wavelet packets.

First, some notation: W_j^s, where s is a binary number or empty, will denote the wavelet space with splitting defined by s. $|s|$ is the length of s, zero if s is empty. The children of W_j^s are W_j^{s0} and W_j^{s1}. The parent of W_j^s is W_j^{s-1}, and j determines the level of first split from the backbone. So W_1 (i.e. s is empty) is the space spanned by $I - T_0(\Phi_0)$, and W_2 is the space spanned by $I - T_1^2$. The basis for V_j is Φ_j, and for W_j^s is $\Psi_{j,s}$, and $T_{j,s}$ specifies in which basis the operator is written. Finally, $|W_j^s|$ is the cost of computing everything at the j, s step, and $|\mathring{W}_j^s|$ is the cost of computing everything for all W_j^s's progeny (but not W_j^s), and $|\overline{W_j^s}|$ is the cost of everything including progeny.

We proceed as follows. Start the wavelet algorithm as usual. However, at V_2, instead of just using T_2^2 to compute T_2^4, we will also calculate the matrix $T_2 = \Phi_2 T_1 \Phi_2^t$. At the jth step, assume that $T_{k-1}^{2^{k-1}}, T_{j-1}^{2^{j-2}}, \ldots, T_{j-1}$, where $k = \min(j-1, \log n - j + 1)$, have all been calculated (note the subscript of T specifies the basis on which the matrix acts). Now, instead of just calculating

$T_j^{2^{j-1}}$, and $T_j^{2^j}$, we calculate all of the powers 2^k of T in the basis Φ_j such that $k \leq \min(j, \log n - j)$.

So at this point we will start to assume all of our matrix multiplications and all calls to the orthonormalization algorithm are at worst $O(n \log^2 n)$. With these assumptions, and the assumption that each V_j is half the size of the previous one, we can count the operations on the "backbone" of the packet tree. At each V_j, we have to compute at most $C(j+2)$ (and actually just $C(\min(j, \log n - j) + 2)$ matrix multiplications or calls to the orthonormalization algorithm to get all the necessary operators (the $+2$ is for orthonormalization and for applying the operator, the C because pushing each operator forward takes two sparse matrix multiplications). Each one takes at most $|\Phi_{j-1}| \log^2 |\Phi_{j-1}|$ operations. So in total, since the series $\sum \frac{j}{2^{2j}}$ is summable, $O(n \log^2 n)$ to do the backbone.

Now we should calculate the cost of the wavelet branches of the tree.

We use the same philosophy as above for generating the wavelet spaces- before proceeding to find the bases of the children of a space, we record all operators that will be used by that space's children. note that this keeps us from having to walk all the way back up the tree, and also, that since the basis transformations should be sparse, at least one matrix in all the multiplications should be sparse. So the first claim is that $|\mathring{W}_j^s| \leq C2(j+2)(j-|s|-1)|\Psi_j^s|^2$. This can be seen by induction on $j-|s|-1$(i.e. we induct up the tree). If $j-|s|-1=1$, since we only use positive powers of the operator, there is only one split possible. For simplicity, to estimate the operation count, we will calculate all j powers of the operator at each node of the jth branch (instead of $j-|s|-1$ powers.

We need to apply $T_{j,s}$, and $I - T_{j,s}$ to $\Psi_{j,s}$, and orthonormalize the resulting sets of vectors. This costs $2C(j+2)|\Psi_{j,s}|^2$ operations, as desired.

Now assume the statement is true for $j-|s|-1=k$. We need to calculate $|\mathring{W}_j^s|$, where $j-|s|-1=k+1$.

As above, $|W_j^{s0}|$ and $|W_j^{s1}|$ are each $C(j+2)|\Psi_j^s|^2$. Also, since Ψ_j^{s0} and Ψ_j^{s1} split W_j^s, $|\Psi_j^{s0}| + |\Psi_j^{s1}| = |\Psi_j^s|$. Set $|\Psi_j^{s0}| = \alpha$. Then

$$|\mathring{W}_j^s| = |W_j^{s0}| + |W_j^{s1}| + |\mathring{W}_j^{s0}| + |\mathring{W}_j^{s1}|$$

$$= C(j+2)|\Psi_j^s|^2 + C(j+2)(j-|s|-2)\alpha^2|\Psi_j^s|^2 + C(j+2)(j-|s|-2)(1-\alpha)^2|\Psi_j^s|^2$$

$$\leq 2C(j+2)|\Psi_j^s|^2 + 2C(j+2)(j-|s|-2)|\Psi_j^s|^2 = 2C(j+2)(j-|s|-1)|\Psi_j^s|^2 .$$

Now to get from V_{j-1} to W_j takes $C(j+2)|\Phi_{j_1}|^2$ operations, and $|\Psi_j| = n/2^j$. At this point I have used the assumption that each application of M halves

the space- one might worry about the size of the wavelet spaces being to large if it does more than half, but then the wavelet spaces could be bounded by the V one higher on the backbone. Then the cost $|\overline{W}_j| \leq \frac{2C(j+2)jn^2}{2^{2j}}$. The series $\sum \frac{j^2}{2^{2j}}$ is summable, so the entire algorithm is $O(n \log^2 n)$.

9 Acknowledgements

We would like to thank Fred Warner for helpful comments during the preparation of the manuscript.

References

[1] R. Coifman, M. Maggioni, Diffusion wavelets, Applied Computational Harmonic Analysis, submitted.

[2] R. Coifman, M. Maggioni, Multiresolution analysis associated to diffusion semigroups: construction and fast algorithms, Tech. rep., Department of Computer Science, Yale University (2004).

[3] S. Lafon, Diffusion maps and geometric harmonics, Ph.D. thesis, Yale University (2004).

[4] R. Coifman, S. Lafon, A. Lee, M. Maggioni, F. Warner, S. Zucker, Geometric diffusions as a tool for harmonic analysis and structure definition of data, Proceedings of National Academy of Sciences.

[5] R. Coifman, M. Wickerhauser, Entropy-based algorithms for best basis selection, IEEE Trans. Info. Theory.

[6] R. R. Coifman, Y. Meyer, S. Quake, M. V. Wickerhauser, Signal processing and compression with wavelet packets, in: Progress in wavelet analysis and applications (Toulouse, 1992), Frontières, Gif, 1993, pp. 77–93.

[7] E. Stein, Topics in Harmonic Analysis related to the Littlewood-Paley theory, Princeton University Press, 1970.

[8] M. Nielsen, Highly nonstationary wavelet packets, Appl. Comput. Harmon. Anal.

[9] R. Coifman, M. Maggioni, Biorthogonal diffusion wavelets and markov multiresolution chains, in preparation.

[10] D. Donoho, I. Johnstone, Ideal denoising in an orthonormal basis chosen from a library of bases, Tech. rep., Stanford University (1994).

[11] N. Saito, Local feature extraction and its applications using a library of bases, Ph.D. thesis, Yale Mathematics Department (1994).

[12] R. Coifman, N. Saito, F. Geshwind, F. Warner, Discriminant feature extraction using empirical probability density estimation and a local basis library, Patter Recognition.

[13] R. Coifman, N. Saito, Constructions of local orthonormal bases for classification and regression, C. R. Acad. Sci. Paris 319 Série I (1994) 191–196.