

RANDOMIZED WAVELETS ON ARBITRARY
DOMAINS AND APPLICATIONS TO
FUNCTIONAL MRI ANALYSIS

S. GÖRKEM ÖZKAYA

A DISSERTATION
PRESENTED TO THE FACULTY
OF PRINCETON UNIVERSITY
IN CANDIDACY FOR THE DEGREE
OF DOCTOR OF PHILOSOPHY

RECOMMENDED FOR ACCEPTANCE
BY THE PROGRAM IN
APPLIED AND COMPUTATIONAL MATHEMATICS

ADVISER: INGRID DAUBECHIES

SEPTEMBER 2012

© Copyright by S. Görkem Özkaya, 2012.

All Rights Reserved

Abstract

Wavelets have been shown to be effective bases for many classes of natural signals and images. Standard wavelet bases have the entire vector space \mathbb{R}^n as their natural domain. It is fairly straightforward to adapt these to rectangular subdomains, and there also exist constructions for domains with more complex boundaries. However those methods are ineffective when we deal with domains that are very arbitrary and convoluted. A particular example of interest is the human cortex, which is the part of the human brain where all the cognitive activity takes place. In this thesis, we use the lifting scheme to design wavelets on arbitrary volumes, and in particular on volumes having the structure of the human cortex. These wavelets have an element of randomness in their construction, which allows us to repeat the analysis with many different realizations of the wavelet bases and averaging the results, a method that improves the power of the analysis. Next, we apply this type of wavelet transforms to the statistical analysis to fMRI data, and we show that it enables us to achieve greater spatial localization than other, more standard techniques.

Acknowledgements

I am grateful to my advisor Ingrid Daubechies for the guidance, support, and freedom that she provided to me during my graduate studies. I have enormous respect and admiration for her academic personality, her philosophy of applied mathematics, and her human side. I have been extremely fortunate to be her student.

I am thankful to Dimitri Van De Ville for having me as a visiting graduate student at EPFL during the summer of 2010, and for his continuous mentorship from then on. He has been a source of inspiration for me with his passion in science and positive attitude.

I would like to thank the faculty at Princeton, from whom I learned a lot. I would like to thank Phil Holmes and Amit Singer for participating in my dissertation committee, and Robert Calderbank, Weinan E, Peter Ramadge and Sergio Verdu for participating in my preliminary and general exams.

I would like to thank Rayan Saab for giving his time generously for my questions and for his feedback during the writing of this dissertation.

I would like to thank the professors who taught me during my undergraduate and Masters degrees at Bogazici University. In particular, I would like to thank Alp Eden for all his support and encouragement while supervising my Masters thesis, and to him, Aysin Ertuzun, Nilgun Isik, Kadri Ozcaldiran and Ayse Soysal for their support as I applied to graduate school.

I would like to thank Howard Bergman, Christina Lipsky, Valerie Marino, Rebecca Louie and Audrey Mainzer for their help in administrative matters throughout my graduate studies.

I would like to thank my friends Mihai Cucuringu, Aytug Goksu, Deniz Gunduz, Hakan Kaya, Taner Kilincoglu, Menachem Lazar, Onur Ozyesil, Gungor Polatkan, Jesus Puente, Fethi Ramazanoglu, Serdar Selamet, Soner Sevinc, Sergey Voronin, and all my fellow graduate students for the friendly environment during the graduate

school years.

I would like to thank my mother, my father, my brother, my parents-in-law and all of my relatives in Turkey. Their unconditional love and support throughout the years has been the motivation that kept me working.

Finally, I would like to thank my wife Aslıhan Demirkaya for her patience, love, support and for always standing beside me throughout this dissertation.

Contents

Abstract	iii
Acknowledgements	iv
1 Introduction	1
2 Wavelets and the lifting scheme	8
2.1 Wavelet bases and wavelet transforms	8
2.1.1 First generation wavelets	9
2.1.2 Examples	10
2.2 Multiresolution analyses and fast wavelet transforms	10
2.2.1 Multiresolution analysis	12
2.2.2 Refinement equation, finite filters and set of partitionings	13
2.2.3 Synthesizing the scaling functions with the cascade algorithm	15
2.2.4 Wavelets	16
2.2.5 Sets of biorthogonal filters	17
2.2.6 Fast wavelet transforms	19
2.3 The lifting scheme	20
2.3.1 Operator notation	20
2.3.2 Fast wavelet transform and biorthogonality with the operator notation	23
2.3.3 Obtaining a new biorthogonal set of filters from old ones	24

2.3.4	Heuristics for a good discrete wavelet transform	25
2.3.5	Tailoring a wavelet transform in successive lifting steps	27
2.4	Appendix: Orthogonal bases, Riesz bases and frames	31
2.4.1	Riesz bases and orthogonal bases	31
2.4.2	Biorthogonal bases	31
2.4.3	Frames	32
3	Randomized domain-adapted wavelets	33
3.1	The structures needed for the construction	33
3.1.1	The domain	34
3.1.2	Embedded grids and the random merging algorithm	36
3.2	The wavelet transform	38
3.2.1	The finest available resolution	38
3.2.2	The lifting steps	42
3.2.3	Unbalanced Haar wavelets	45
3.2.4	Average interpolating wavelets	47
3.3	Numerical experiments	50
3.3.1	The potential to reduce noise by sparse representation	50
3.3.2	Improvement due to averaging	52
3.3.3	The effect of the second prediction	55
3.3.4	Translation-invariant processing	57
4	Application to wavelet-based statistical analysis of fMRI data	60
4.1	Statistical testing of fMRI data	61
4.1.1	One sample t-test	61
4.1.2	The general linear model	62
4.1.3	Multiple comparison problem	68
4.2	Spatial filtering for improving statistical power	69

4.2.1	Statistical parametric mapping and wavelets	69
4.2.2	Gaussian versus wavelet filtering	70
4.2.3	Classical wavelet-based analysis	70
4.2.4	WSPM: Joint spatio-wavelet statistical analysis	73
4.2.5	Anatomically adapted wavelets	74
4.3	Experimental results	76
4.3.1	Simulated data	76
4.3.2	Real data	76
5	Summary and future work	81

Chapter 1

Introduction

This thesis constructs and applies algorithms to carry out a wavelet analysis for data located in three-dimensional structures that are essentially “fat surfaces”, i.e., are ϵ -neighborhoods of possibly convoluted two-dimensional surfaces. This new approach was developed for the analysis of fMRI data.

This last paragraph contains several terms that we would like to explain already here, before going into the technical details of subsequent chapters.

Wavelets

A *wavelet basis* is a collection of functions $(\psi_{j,m})$, which typically have two indices j and m , denoting *scale* and *position*, respectively. In the classic setting, elements of a wavelet basis are translated and dilated versions of a prototype called the “mother wavelet” function. In other words, given a mother wavelet $\psi(x)$, the basis element $\psi_{j,m}$ is given by

$$\psi_{j,m}(x) = C_j \psi(2^j(x - m)),$$

where $j \in \mathbb{Z}$, $m \in 2^{-j}\mathbb{Z}$ and C_j is the normalizing constant that ensures $\|\psi_{j,m}\| = 1$.

It is generally expected that the mother wavelet has some degree of smoothness, and is either compactly supported, or has most of its energy contained in a compact

domain.

With a good wavelet basis, only a small fraction of the coefficients in the wavelet expansion

$$f = \sum_{j,m} \gamma_{j,m} \psi_{j,m},$$

suffices to approximate the function f whenever it belongs to a signal class of interest, and a local change in the signal only affects a small number of corresponding coefficients. These properties make wavelets optimal bases in compression, denoising and estimation applications [9, 7, 28].

Wavelets in higher dimensions

One dimensional wavelet designs can be translated into higher dimensional Euclidean spaces by means of tensor products, and these are called *separable wavelet bases*. There are other means of obtaining wavelets directly in higher dimensions, typically leading to *nonseparable wavelet bases*. The Fourier transform was used as an indispensable tool in most of the constructions in the earlier years of the development of wavelet theory: most of the classical wavelet bases for \mathbb{R}^n , whether separable or nonseparable, depend on the Fourier transform in their designs.

The lifting scheme and the second generation wavelets

Standard wavelet bases mentioned above have the entire vector space \mathbb{R}^n as their natural domain. It is fairly straightforward to adapt these to rectangular subdomains, and there also exist constructions for domains with more complex boundaries. However, those methods are ineffective when we deal with domains that are very arbitrary and convoluted. A particular example of interest is the cortex of the human brain, which is the place where all the cognitive activity takes place. For such arbitrary subsets of \mathbb{R}^n , which do not possess a mathematical group structure, tools like the Fourier transform are not available to be utilized in the design. Fortunately however, very

flexible frameworks such as the the lifting scheme [30, 28] are available to construct wavelet bases on these domains directly in the spatial domain. The wavelets obtained by such methods are no longer the translated and dilated copies of a prototype, and they are called the *second generation wavelets*. The second generation wavelets possess many of the appealing features of the first generation wavelet transforms, such as leading to sparse and localized representations, and the availability of fast transform algorithms.

Some prior applications of second generation wavelets

One of the first constructions of biorthogonal wavelets on triangulated surfaces was by Launsbery et al. [20]. This scheme was later improved by means of the lifting scheme, which was published by Sweldens [21, 30, 28].

Lifting-based wavelets on nonstandard grids have been used in numerous areas; an example is the construction in [26], of wavelet bases on the sphere, using an adaptive subdivision scheme. Spherical wavelets have applications in astronomical and geophysical data analysis.

Other examples of lifting-based wavelets have applications in computer graphics, where they can be used to efficiently represent meshes. In [19], they are used in compression of three dimensional meshes. In [37], an application in texture synthesis over arbitrary manifolds is given.

In [2, 6, 36, 16] wavelets of this type are used for numerical solutions of partial differential equations.

An application in computational digital photography is given in [11]; in this paper a wavelet basis that is customized to a given image is constructed by the lifting scheme, which has wavelets and scaling functions that avoid edges, as much as possible. These wavelets are then successfully applied to problems of dynamic range compression, edge preserving smoothing, detail enhancement and image colorization.

A neuroimaging application is given in [38], where it is applied to capturing shape variations of the cortex and in tracking cortical folding changes in newborns.

For a review of the theory of second generation wavelets and a survey of other applications, we refer to [18].

Functional brain imaging

Functional brain imaging refers to acquiring time series of three-dimensional images of the human brain, which measure some correlate of the neural activity. This is different from anatomical (or structural) imaging in which the measured quantity is chosen to obtain as much contrast as possible to differentiate between different tissue types.

Functional magnetic resonance imaging (fMRI) has become the most widely used functional brain imaging technology in neuroscience research. It has led to an explosion in papers aiming to understand the organization of the human brain. Earlier fMRI studies mainly concentrated on determining the regions where the active component peaked, and were therefore not overly concerned about losing lower-amplitude information before applying spatial low pass filtering to the data to improve the statistical power [22]. However, later studies have demonstrated that fMRI signal contains much additional detail information. For instance, the work by J. Haxby and his collaborators [17] showed, in an experiment measuring discrimination between different stimuli, that enough information was present, even after removal of the main activation area for each stimulus, to allow for successful stimulus classification. This motivates the use of tools like wavelet analysis to detect fine details that might be less apparent and that could be lost to thresholding in the spatial domain. However, the standard wavelet transforms are mostly designed to have very special domains, like a rectangle in two dimensions or a cube in three dimensions. The activity of interest in the brain, on the other hand, occurs on the cortex, which is an intricately convoluted

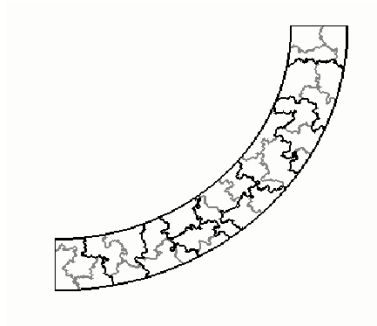
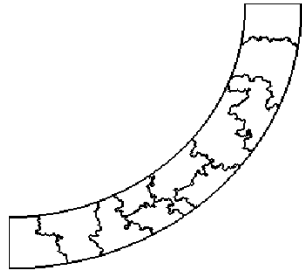
set in three dimensions. Using standard wavelets without any modification will have the obvious drawback of mixing signal from the cortex (gray matter) with the signal coming from the off-cortex regions. There would be an artificial edge at the boundary of the cortex, which would inflate the detail coefficients in the wavelet transform, reducing its ability to form a sparse representation of the data.

Domain-adapted wavelets

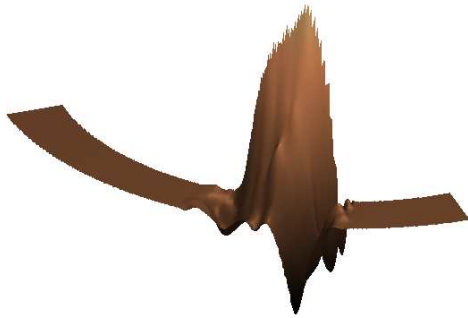
In this thesis, mainly motivated by the functional brain imaging problem, we use the lifting scheme to design wavelets on arbitrary two and three dimensional domains. One of the key steps in the construction of the wavelets is to define a nested family of partitions on the domain. This includes parent, child, neighbor and sibling relations. Our algorithm of defining these structures has an element of randomness in it, which results in getting multiple sets of wavelet bases, each coming from a different realization of the partitioning. This in turn allows us to repeat the analysis with many different realizations of the wavelet bases and averaging the results, a method that improves the power of the analysis. An example of a nonstandard domain, a random partitioning on it and an example wavelet and scaling function (to be explained in the next chapter) is displayed in Figure 1.1.

Wavelet-based statistical parametric mapping

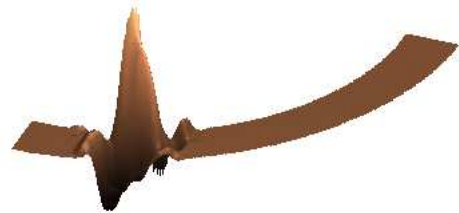
One of the basic objectives in an fMRI study is to detect whether a given location point (voxel) is responding to the given stimulus or not. The question can be answered by means of statistical tests that are applied to the corresponding time series. However, these statistical tests always have the chance of a positive result when there was no response (false positive), and a negative result when there actually was some response (false negative). In an fMRI image, there is a large number of voxels, and the statistical test needs to be performed for each of these voxels. This means that



(a)



(b)



(c)

Figure 1.1: Illustration of partitioning and adapted wavelets on an annular shaped domain. (a) Sample random partitioning of two levels. (b) Scaling function. (c) Wavelet

there is a large number of decisions to make about the absence or presence of response. Due to the noisy nature of the fMRI data, some spatial filtering is necessary in order to have control over the total number of false positives, while not losing the sensitivity of the test altogether. Statistical parametric mapping (SPM) [12], which is one of the most widely used methods, uses a Fourier-based low pass filtering as a means of reducing noise. However this has the drawback of destroying fine-scale spatial details. An alternative is a wavelet-based framework is due to Van De Ville et al. [33, 32, 31, 35], which includes theoretical bounds on the false positive rate, when the null hypothesis is true in the region of interest. This framework has no restrictions on the type of wavelet basis to be used, so it is suitable to use the domain adapted wavelets that are constructed so as to be spatially adapted to the cortex of the human brain.

Organization of the thesis

The organization of the thesis will be as follows. In Chapter 2 we will study the preliminaries about multiresolution analysis, wavelet transforms, and the lifting scheme. Chapter 3 will be about the methodology of construction of the adapted wavelets and will also give some numerical experiments. In Chapter 4 we will give results about the application of this type of wavelets in statistical analysis of fMRI data, in the wavelet-based statistical parametric mapping framework. Chapter 5 contains a summary and conclusion.

Chapter 2

Wavelets and the lifting scheme

In this chapter, we review the basics of wavelet theory, as well as of the lifting scheme, a method that can be used to design wavelets on irregular domains, without relying on tools from Fourier analysis. It is due to Wim Sweldens [28, 30].

Our presentation and notation will mostly follow [28]. A generalized language for wavelets will be used, which also covers the classical (first generation) wavelets as a special case.

2.1 Wavelet bases and wavelet transforms

In a general setting, a wavelet basis is a collection of functions

$$\{\psi_{j,m}(x) : j \in \mathcal{J}, m \in \mathcal{M}(j)\},$$

where $\mathcal{J} \subset \mathbb{Z}$ is the index set for scale (frequency), and for each j , $\mathcal{M}(j)$ is the index set for location. The collection enables us to expand a given function f as

$$f = \sum_{j \in \mathcal{J}, m \in \mathcal{M}(j)} \gamma_{j,m} \psi_{j,m},$$

which induces the representation of the function f with a sequence of numbers $(\gamma_{j,m})$

$$f \mapsto (\gamma_{j,m}),$$

and this mapping is called a *wavelet transform*.

In a good wavelet transform, only a small fraction of the coefficients suffices to approximate the function f whenever it belongs to a signal class of interest, and a local change in the signal only affects a small number of corresponding coefficients. These make wavelets optimal bases in compression, denoising and estimation applications [28, 9, 21].

2.1.1 First generation wavelets

In the classic setting, a wavelet basis is obtained by translating and dilating (up to a normalizing constant) a single template function, which is called the *mother wavelet*. These type of wavelets are referred as the *first generation wavelets*. For the one-dimensional case, the domain is the infinite real line \mathbb{R} , the scale index set is $\mathcal{J} = \mathbb{N}$, and for $j \in \mathcal{J}$, the location index set is $\mathcal{M}(j) = 2^{-j}\mathbb{Z}$. Given a mother wavelet $\psi(x)$, the basis element $\psi_{j,m}$ is defined to be

$$\psi_{j,m}(x) = C_j \psi(2^j(x - m)),$$

where C_j is the normalizing constant that ensures $\|\psi_{j,m}\| = 1$.

In order to obtain a desirable wavelet basis, the mother wavelet should either be compactly supported, or have most of its energy be contained in a compact domain.

2.1.2 Examples

- The Haar basis is the earliest known example of a wavelet basis. It is generated by the Haar mother wavelet function

$$\psi(x) = \begin{cases} 1 & \text{if } 0 \leq x < 1/2 \\ -1 & \text{if } 1/2 \leq x \leq 1 \\ 0 & \text{elsewhere} \end{cases} .$$

Haar wavelets form an orthonormal basis and they are compactly supported. Their main drawback is that they are not smooth, which results in wavelet representations that are not very sparse, contrary to what was desired.

- The Meyer wavelet is one of the first constructions of a smooth mother wavelet function that generates an *orthogonal basis*, but it is not compactly supported, which is a drawback for computational purposes.
- The Daubechies wavelet family comes next chronologically, it consists of wavelet bases that are both smooth, compactly supported and orthogonal.
- There are wavelets that form a *biorthogonal* basis (defined in Subsection 2.4), one of which is illustrated in Figure 2.1, along with the above examples.

2.2 Multiresolution analyses and fast wavelet transforms

For the wavelet expansion

$$f = \sum_{j \in \mathcal{J}, m \in \mathcal{M}(j)} \gamma_{j,m} \psi_{j,m},$$

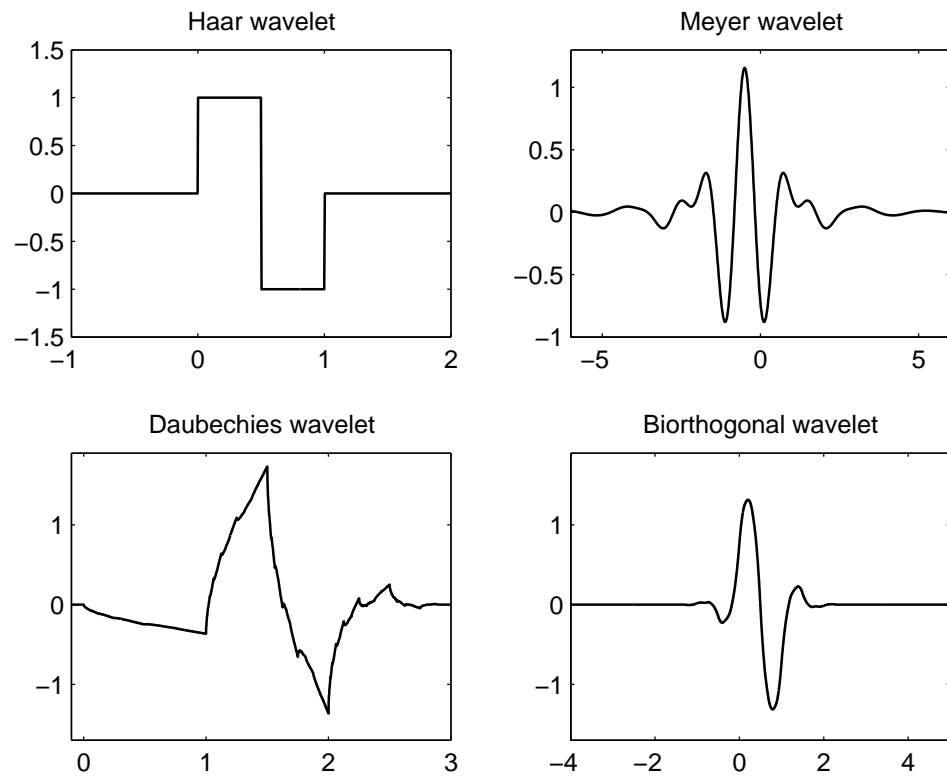


Figure 2.1: Examples of mother wavelets, belonging to some of the most commonly used wavelet bases: Haar, Meyer, Daubechies and Biorthogonal (average interpolating) wavelets.

orthogonal and biorthogonal wavelets have the relation

$$\gamma_{j,m} = \langle f, \tilde{\psi}_{j,m} \rangle,$$

that makes the computation of the coefficient functionals possible with a simple expression. However taking this inner product is computationally very slow.

Many of the wavelet bases, on the other hand, are implicitly tied to a structure formed by a multilayered sequence of subspaces, which is called a *multiresolution analysis*.

A multiresolution analysis brings about a much faster way of computing the wavelet coefficients, which is called the *fast wavelet transform* or the *cascade algorithm*.

2.2.1 Multiresolution analysis

Definition 2.2.1. *A sequence of closed subspaces $\mathbf{M} = \{V_j : j \in \mathcal{J} \subset \mathbb{Z}\}$ of $L^2(X, \Sigma, \mu)$ for some measurable space $X \subset \mathbb{R}^n$ is called a multiresolution analysis if it satisfies the following:*

1. $V_j \subset V_{j+1}$
2. $\bigcap_j V_j = \{0\}$
3. $\overline{\bigcup_j V_j} = L^2(X, \Sigma, \mu)$
4. *For each $j \in \mathcal{J}$, there exists a Riesz basis $\{\varphi_{j,k} : j \in \mathcal{K}(j)\}$ that spans V_j . The functions $(\varphi_{j,k})$ are called scaling functions.*

Here, $\mathcal{K}(j)$ is an index set indicating location, and in general it is assumed that $\mathcal{K}(j) \subset (\mathcal{K}(j+1))$. The scale index \mathcal{J} is taken to be $\mathcal{J} = \mathbb{Z}$ when X is an unbounded set of infinite measure, and $\mathcal{J} = \mathbb{N}$ when X is bounded and of finite measure (which is the case for the focus of this thesis).

Dual multiresolution analysis

Let $\mathbf{M} = \{V_j : j \in \mathcal{J} \subset \mathbb{Z}\}$ be a multiresolution analysis. Then, another multiresolution analysis

$$\widetilde{\mathbf{M}} = \{\widetilde{V}_j : j \in \mathcal{J} \subset \mathbb{Z}\}$$

is called a *dual multiresolution analysis* of \mathbf{M} , if its scaling functions $(\widetilde{\varphi}_{j,k})$ satisfy

$$\langle \varphi_{j,k}, \widetilde{\varphi}_{j,k'} \rangle = \delta_{k,k'} \quad \text{for } k, k' \in \mathcal{K}(j). \quad (2.1)$$

For any given function $f \in V_j$, the coefficients in the expansion

$$f = \sum_{k \in \mathcal{K}(j)} \lambda_{j,k} \varphi_{j,k},$$

can be obtained by

$$\lambda_{j,k} = \langle f, \widetilde{\varphi}_{j,k} \rangle.$$

Remark 2.2.2. *Note that, according to this definition, a dual multiresolution is not unique. We also note that, since $(\varphi_{j,k})$ is a Riesz basis for V_j , it also has a dual basis within V_j that is biorthogonal to it, as mentioned in Subsection 2.4. However a set of dual scaling functions that are denoted by $(\widetilde{\varphi}_{j,k})$ here are not necessarily contained in V_j , and they do not necessarily coincide with the dual Riesz basis of $(\varphi_{j,k})$ within V_j .*

2.2.2 Refinement equation, finite filters and set of partitionings

Because of the inclusion $V_j \subset V_{j+1}$, $\varphi_{j,k}$ can be written as a linear combination of $(\varphi_{j+1,l})$'s:

$$\varphi_{j,k} = \sum_{l \in \mathcal{K}(j+1)} h_{j,k,l} \varphi_{j+1,l}, \quad (2.2)$$

for any $j \in \mathcal{J}$ and $k \in \mathcal{K}(j)$. This is a relation that must be satisfied in any multiresolution analysis. On the other hand, we shall see that a multiresolution analysis will be uniquely determined once the coefficients $(h_{j,k,l})$ are defined. In order for this to be possible, we need two definitions. One is the concept of a *finite filter*, which puts some constraints on the coefficients $(h_{j,k,l})$ for them to result in a well defined multiresolution analysis. The other is the *set of partitionings*, using which one will be able to pass from the finite filter to scaling functions in $L^2(X, \Sigma, \mu)$.

Finite filters

Definition 2.2.3. *A set of real numbers $\{h_{j,k,l} : j \in \mathcal{J}, k \in \mathcal{K}(j), l \in \mathcal{K}(j+1)\}$ is called a finite filter if the following are satisfied.*

1. *For each j and k , $h_{j,k,l}$ is nonzero for only finitely many l 's. Hence, the set defined as*

$$\mathcal{L}(j, k) = \{l \in \mathcal{K}(j+1) | h_{j,k,l} \neq 0\}$$

is finite.

2. *For each j and l , $h_{j,k,l}$ is nonzero for only finitely many k 's. Hence, the set defined as*

$$\mathcal{K}(j, l) = \{k \in \mathcal{K}(j) | h_{j,k,l} \neq 0\}$$

is finite.

3. *The sizes of sets $\mathcal{L}(j, k)$ and $\mathcal{K}(j, l)$ are uniformly bounded over all j, k and l .*

Dual filter

We note that, for the dual scaling functions, there also exists a refinement relation

$$\tilde{\varphi}_{j,k} = \sum_{l \in \mathcal{K}(j+1)} \tilde{h}_{j,k,l} \tilde{\varphi}_{j+1,l}, \quad (2.3)$$

and using (2.1), (2.2) and (2.3), one obtains that the filter $(h_{j,k,l})$ and the dual filter $(\tilde{h}_{j,k,l})$ must satisfy the relation

$$\sum_l h_{j,k,l} \tilde{h}_{j,k',l} = \delta_{k,k'}, \quad \text{for } j \in \mathcal{J}, \quad k, k' \in \mathcal{K}(j). \quad (2.4)$$

Nested set of partitionings

In order to be able to define scaling functions on $L^2(X, \Sigma, \mu)$, the next structure that is needed is a *set of partitionings*. In the following definition, we assume that $\mathcal{K}(j) \subset \mathcal{K}(j+1)$.

Definition 2.2.4. *A nested set of partitionings is a collection $\{S_{j,k}: j \in \mathcal{J}, k \in \mathcal{K}\}$ of subsets of X that satisfy*

1. *For all $j \in \mathcal{J}$, the collection $\{S_{j,k}: k \in \mathcal{K}(j)\}$ is disjoint and $X = \overline{\bigcup_{k \in \mathcal{K}(j)} S_{j,k}}$,*
2. *$S_{j+1,k} \subset S_{j,k}$,*
3. *For all $j \in \mathcal{J}$, and $k \in \mathcal{K}(j+1)$, there exists a $k' \in \mathcal{K}(j)$ such that $S_{j+1,k} \subset S_{j,k'}$,*
4. *For a fixed $k \in \mathcal{K}(j_0)$, the infinite intersection $\bigcap_{j>j_0} S_{j,k}$ is a single point set, whose unique element will be denoted by x_k .*

2.2.3 Synthesizing the scaling functions with the cascade algorithm

Having defined a set of partitionings $\{S_{j,k}: j \in \mathcal{J}, k \in \mathcal{K}(j)\}$, and a finite filter $\{h_{j,k,l}: j \in \mathcal{J}, k \in \mathcal{K}(j), l \in \mathcal{K}(j+1)\}$, one is now ready to synthesize the scaling function φ_{j_0,k_0} . Initially, setting $(\lambda_{j_0,k})$ to a Kronecker sequence as

$$\lambda_{j_0,k} = \delta_{k,k_0}, \quad \text{for } k \in \mathcal{K}(j_0),$$

we define the collection of sequences $(\lambda_{j,k})_{k \in \mathcal{K}(j)}$ for $j > j_0$, using the following recursive formula:

$$\lambda_{j+1,l} = \sum_{k \in \mathcal{K}(j,l)} h_{j,k,l} \lambda_{j,k}.$$

Then a sequence of functions $(f_{j_0,k_0}^j)_j$ for $j \geq j_0$ is defined as

$$f_{j_0,k_0}^j = \sum_{k \in \mathcal{K}(j)} \lambda_{j,k} \chi_{S_{j,k}},$$

which also satisfies

$$f_{j_0,k_0}^j = \sum_{l \in \mathcal{K}(j_0+1)} h_{j_0,k_0,l} f_{j_0+1,l}^j. \quad (2.5)$$

Now the function φ_{j_0,k_0} is defined to be

$$\varphi_{j_0,k_0} = \lim_{j \rightarrow \infty} f_{j_0,k_0}^j,$$

assuming the limit exists in L^2 .

2.2.4 Wavelets

Given a multiresolution analysis $\mathbf{M} = \{V_j : j \in \mathcal{J} \subset \mathbb{Z}\}$, and a dual $\widetilde{\mathbf{M}} = \{\widetilde{V}_j : j \in \mathcal{J} \subset \mathbb{Z}\}$, the *wavelet subspace* W_j is the complement of V_j in V_{j+1} which is orthogonal to \widetilde{V}_j . The *wavelets* $\psi_{j,m}$ are the functions that span W_j .

Definition 2.2.5. *Let $\mathcal{M}(j) = \mathcal{K}(j+1) \setminus \mathcal{K}(j)$ be the index set complementing $\mathcal{K}(j)$ in $\mathcal{K}(j+1)$, and let $\{\psi_{j,m} : j \in \mathcal{J}, m \in \mathcal{M}(j)\} \subset V_{j+1}$ be a collection of functions, and W_j be the closure of its span. Then the collection $\{\psi_{j,m} : m \in \mathcal{M}(j)\}$ is a set of wavelet functions if the following are satisfied.*

1. $V_j \cap W_j = \{0\}$ and $W_j \perp \widetilde{V}_j$.
2. If $\mathcal{J} = \mathbb{Z}$, the set $\{\psi_{j,m} : j \in \mathcal{J}, m \in \mathcal{M}(j)\}$ is a Riesz basis for L^2 .

If $\mathcal{J} = \mathbb{N}$, the set $\{\psi_{j,m} : j \in \mathcal{J}, m \in \mathcal{M}(j)\} \cup \{\varphi_{0,k} / \|\varphi_{0,k}\| : k \in \mathcal{K}(0)\}$ is a Riesz basis for L^2 .

The Riesz bases mentioned above will be referred as *wavelet bases* from now on.

Dual wavelets

Given a wavelet basis $(\psi_{j,m})$, there exists a corresponding dual Riesz basis (as explained in Subsection 2.4), whose elements will be denoted by $(\tilde{\psi}_{j,m})$, and for a given j , the closed span of $(\tilde{\psi}_{j,m})$ will be denoted by \widetilde{W}_j . The dual wavelets satisfy

$$\langle \psi_{j,m}, \tilde{\psi}_{j',m'} \rangle = \delta_{m,m'} \delta_{j,j'},$$

by their definition. The dual wavelet subspaces also satisfy $\widetilde{W}_j \cap \widetilde{V}_j = \{0\}$ and $\widetilde{W}_j \perp V_j$.

The dual wavelets can also be defined by reversing the roles of \mathbf{M} and $\widetilde{\mathbf{M}}$, in Definition 2.2.5.

2.2.5 Sets of biorthogonal filters

The finite filters $(h_{j,k,l})$ and $(\tilde{h}_{j,k,l})$ were defined in Subsection 2.2.2. These filters take place in the refinement equations

$$\varphi_{j,k} = \sum_{l \in \mathcal{K}(j+1)} h_{j,k,l} \varphi_{j+1,l},$$

and

$$\tilde{\varphi}_{j,k} = \sum_{l \in \mathcal{K}(j+1)} \tilde{h}_{j,k,l} \tilde{\varphi}_{j+1,l}.$$

Now similarly, since $\psi_{j,m} \in V_{j+1}$ and $\tilde{\psi}_{j,m} \in \widetilde{V}_{j+1}$, we can define the filters $g_{j,m,l}$

and $\tilde{g}_{j,m,l}$ to be the filters that satisfy

$$\psi_{j,m} = \sum_{l \in \mathcal{K}(j+1)} g_{j,m,l} \varphi_{j+1,l},$$

and

$$\tilde{\psi}_{j,m} = \sum_{l \in \mathcal{K}(j+1)} \tilde{g}_{j,k,l} \tilde{\varphi}_{j+1,l}.$$

The functions $(\varphi_{j,k}), (\tilde{\varphi}_{j,k}), (\psi_{j,m})$ and $(\tilde{\psi}_{j,m})$ satisfy the biorthogonality relations

$$\begin{aligned} \langle \varphi_{j,k}, \tilde{\varphi}_{j,k'} \rangle &= \delta_{k,k'}, \\ \langle \psi_{j,m}, \tilde{\psi}_{j,m'} \rangle &= \delta_{m,m'}, \\ \langle \varphi_{j,k}, \tilde{\psi}_{j,m} \rangle &= 0, \\ \langle \psi_{j,m}, \tilde{\varphi}_{j,k} \rangle &= 0. \end{aligned} \tag{2.6}$$

As a consequence of the above relations, the corresponding filters must satisfy

$$\begin{aligned} \sum_{l \in \mathcal{K}(j+1)} h_{j,k,l} \tilde{h}_{j,k',l} &= \delta_{k,k'}, \\ \sum_{l \in \mathcal{K}(j+1)} g_{j,m,l} \tilde{g}_{j,m',l} &= \delta_{m,m'}, \\ \sum_{l \in \mathcal{K}(j+1)} g_{j,m,l} \tilde{h}_{j,k,l} &= 0, \\ \sum_{l \in \mathcal{K}(j+1)} h_{j,k,l} \tilde{g}_{j,m,l} &= 0. \end{aligned} \tag{2.7}$$

Definition 2.2.6. *A set of finite filters $\{h, \tilde{h}, g, \tilde{g}\}$ is called a set of biorthogonal filters, if the relations in (2.7) are satisfied.*

Remark 2.2.7. *The collection of wavelet and scaling functions that satisfy (2.6) results in finite filters that satisfy (2.7). But also, thinking in the reverse direction, if one starts with a set of biorthogonal filters, one can obtain the wavelet and scaling functions using the cascade algorithm of Subsection 2.2.3, and they will have the*

desired biorthogonal relations of (2.6). The purpose of the lifting scheme is actually to design a set of biorthogonal filters and make successive improvements on them without disturbing their biorthogonality.

2.2.6 Fast wavelet transforms

The fast wavelet transform is an efficient algorithm that finds the expansion coefficients in the N -level wavelet expansion

$$f = \sum_{k \in \mathcal{K}(0)} \lambda_{0,k} \varphi_{0,k} + \sum_{j=0}^{N-1} \sum_{m \in \mathcal{M}(j)} \gamma_{j,m} \psi_{j,m}, \quad (2.8)$$

for a positive integer N and $f \in V_N$, by exploiting the refinement relations.

If we take a function f in V_N , it can be written as a linear combination of the scaling functions of V_N as

$$f = \sum_{k \in \mathcal{K}(N)} \lambda_{N,k} \varphi_{N,k}. \quad (2.9)$$

At the beginning of a fast wavelet transform, it is assumed that the scaling function coefficients $(\lambda_{N,k})_k$ at this finest level are given. Since W_{N-1} is a complement of V_{N-1} in V_N , f can be written as $f = f_a + f_d$ with $f_a \in V_{N-1}$ and $f_d \in W_{N-1}$. If we expand f_a and f_d in terms of the scaling functions and wavelets, we get

$$\begin{aligned} f &= \sum_{k \in \mathcal{K}(N)} \lambda_{N,k} \varphi_{N,k} \\ &= \sum_{k \in \mathcal{K}(N-1)} \lambda_{N-1,k} \varphi_{N-1,k} + \sum_{m \in \mathcal{M}(N-1)} \gamma_{N-1,m} \psi_{N-1,m}. \end{aligned}$$

The key observation at this point is that the coefficient sequences $(\lambda_{N-1,k})_k$ and $(\gamma_{N-1,m})_m$ can be obtained by applying filters \tilde{h} and \tilde{g} to $(\lambda_{N,k})_k$, respectively. That

is to say, the relations

$$\lambda_{N-1,k} = \sum_{l \in \mathcal{K}(N)} \tilde{h}_{N,k,l} \lambda_{N,l}$$

and

$$\gamma_{N-1,k} = \sum_{l \in \mathcal{K}(N)} \tilde{g}_{N,k,l} \lambda_{N,l}$$

can be obtained as a result of the biorthogonality relations. Recursive application of this relation to the $(\lambda_{N-i,k})$ after each step results in (2.8).

The inverse transform of the above step is given by

$$\lambda_{N,l} = \sum_{k \in \mathcal{K}(N-1)} h_{N-1,k,l} \lambda_{N-1,k} + \sum_{m \in \mathcal{M}(N-1)} g_{N-1,m,l} \gamma_{N-1,m}.$$

We recall that, in the present setting, it is assumed that all the summations in the above equations are finite, since the filters are assumed to be finite.

2.3 The lifting scheme

The lifting scheme is the framework through which one can obtain new biorthogonal filters from old ones, and tailor them according to the desired smoothness and vanishing moment properties of the corresponding scaling functions and wavelets.

2.3.1 Operator notation

Continuing to follow [28], we will introduce the operator notation corresponding to the filters of the previous section. The operator notation will allow a much compact expression for the lifting scheme.

First we will give a general definition for an operator H and its adjoint H^* ,

corresponding to a filter $(h_{k,l})$. Then the definitions specific to present case will follow.

Let \mathcal{I}_1 and \mathcal{I}_2 be two index sets that are at most countable, and let $h = \{h_{k,l} : k \in \mathcal{I}_2, l \in \mathcal{I}_1\}$ be a filter with real coefficients. Let $\ell^2(\mathcal{I}_1)$ and $\ell^2(\mathcal{I}_2)$ be the spaces of the square summable, real valued discrete functions, defined on the index sets. The operator H corresponding to the filter $(h_{k,l})$ is be defined to be

$$H: \ell^2(\mathcal{I}_1) \rightarrow \ell^2(\mathcal{I}_2)$$

$$(a_k)_{m \in \mathcal{I}_1} \mapsto \left(\sum_{l \in \mathcal{I}_1} h_{k,l} a_l \right)_{k \in \mathcal{I}_2}. \quad (2.10)$$

The adjoint operator of H , which is denoted by H^* , is an operator from $\ell^2(\mathcal{I}_2)$ to $\ell^2(\mathcal{I}_1)$ given by

$$H^*: \ell^2(\mathcal{I}_2) \rightarrow \ell^2(\mathcal{I}_1)$$

$$(a_k)_{m \in \mathcal{I}_2} \mapsto \left(\sum_{k \in \mathcal{I}_2} h_{k,l} a_k \right)_{l \in \mathcal{I}_1}. \quad (2.11)$$

Given an operator H , the adjoint operator is also the unique operator that satisfies

$$\langle x, Hy \rangle = \langle H^*x, y \rangle$$

for all $x \in \ell(\mathcal{I}_1)$ and $y \in \ell(\mathcal{I}_2)$.

The above notation will be used for the rest of the section. Note that, in (2.10), the summation runs over the first index of $(h_{k,l})$, whereas in (2.11), it runs over the second index. For the case of matrices, the adjoint operator corresponds to the transpose, and the order of indexes for the filters has been set accordingly.

Now, let us fix a level $j \in \mathcal{J}$. We have three index sets $\mathcal{K}(j+1)$, $\mathcal{K}(j)$ and $\mathcal{M}(j)$,

which satisfy

$$\mathcal{K}(j+1) = \mathcal{K}(j) \cup \mathcal{M}(j).$$

Let us consider the spaces $\ell^2(\mathcal{K}(j+1))$, $\ell^2(\mathcal{K}(j))$ and $\ell^2(\mathcal{M}(j))$. We define the operators \tilde{H}_j , \tilde{G}_j , H_j^* and G_j^* as follows:

$$\begin{aligned} \tilde{H}_j: \ell^2(\mathcal{K}(j+1)) &\rightarrow \ell^2(\mathcal{K}(j)) \\ (a_k)_{k \in \mathcal{K}(j+1)} &\mapsto \left(\sum_{l \in \mathcal{K}(j+1)} \tilde{h}_{j,k,l} a_l \right)_{k \in \mathcal{K}(j)}, \end{aligned}$$

$$\begin{aligned} \tilde{G}_j: \ell^2(\mathcal{K}(j+1)) &\rightarrow \ell^2(\mathcal{M}(j)) \\ (a_k)_{k \in \mathcal{K}(j+1)} &\mapsto \left(\sum_{l \in \mathcal{K}(j+1)} \tilde{g}_{j,m,l} a_l \right)_{m \in \mathcal{M}(j)}, \end{aligned}$$

and similarly,

$$\begin{aligned} H_j^*: \ell^2(\mathcal{K}(j)) &\rightarrow \ell^2(\mathcal{K}(j+1)) \\ (a_k)_{k \in \mathcal{K}(j+1)} &\mapsto \left(\sum_{l \in \mathcal{K}(j)} h_{j,k,l} a_l \right)_{k \in \mathcal{K}(j)}, \end{aligned}$$

$$\begin{aligned} G_j^*: \ell^2(\mathcal{M}(j)) &\rightarrow \ell^2(\mathcal{K}(j+1)) \\ (a_m)_{m \in \mathcal{M}(j+1)} &\mapsto \left(\sum_{l \in \mathcal{K}(j)} g_{j,k,l} a_l \right)_{k \in \mathcal{K}(j)}. \end{aligned}$$

2.3.2 Fast wavelet transform and biorthogonality with the operator notation

Let us denote the sequence $(\lambda_{j,k})_{k \in \mathcal{K}(j)}$ simply by λ_j , and similarly $(\gamma_{j,m})_{m \in \mathcal{M}(j)}$ by γ_j . The fast wavelet transform is the process that starts with λ_N , and continues as

$$\begin{aligned} \lambda_N &\mapsto (\lambda_{N-1}, \gamma_{N-1}) \\ &\mapsto (\lambda_{N-2}, \gamma_{N-2}, \gamma_{N-1}) \\ &\quad \vdots \\ &\mapsto (\lambda_0, \gamma_0, \gamma_1, \dots, \gamma_{N-2}, \gamma_{N-1}), \end{aligned}$$

where

$$\lambda_j = \tilde{H}_j \lambda_{j+1} \quad \text{and} \quad \gamma_j = \tilde{G}_j \gamma_{j+1},$$

at each step.

The inverse of each step can be obtained by the operators H^* and G^* as

$$\lambda_{j+1} = H_j^* \lambda_j + G_j^* \gamma_j.$$

The biorthogonality relations (2.7) in Subsection 2.2.5 can be written with the operator notation as

$$\tilde{G}_j H_j^* = \tilde{H}_j G_j^* = 0 \tag{2.12}$$

$$\tilde{H}_j H_j^* = \tilde{G}_j G_j^* = I, \tag{2.13}$$

where I is the identity operator. From these, it follows that

$$\tilde{H}_j^* H_j + G_j^* \tilde{G}_j = I,$$

provided the union of the ranges of the operators H_j^* and G_j^* span entire $\ell^2((K(j+1)))$. In finite dimensions, it is enough that $|\mathcal{K}(j+1)| = |\mathcal{K}(j)| + |\mathcal{M}(j)|$, in order for this to be satisfied.

2.3.3 Obtaining a new biorthogonal set of filters from old ones

Lifting

The key point of the lifting scheme is to obtain new biorthogonal filters from old ones without losing the biorthogonality.

Theorem 2.3.1. *Let $H^{\text{old}}, \tilde{H}^{\text{old}}, G^{\text{old}}, \tilde{G}^{\text{old}}$ be a set of biorthogonal filter operators. Then the following gives a new set of biorthogonal filter operators:*

$$\begin{aligned} H_j &= H_j^{\text{old}}, \\ \tilde{H}_j &= \tilde{H}_j^{\text{old}} + S\tilde{G}_j^{\text{old}}, \\ G_j &= G_j^{\text{old}} - S^*H_j^{\text{old}}, \\ \tilde{G}_j &= \tilde{G}_j^{\text{old}}, \end{aligned}$$

where S is any operator from $\ell^2(\mathcal{M}(j))$ to $\ell^2(\mathcal{K}(j))$.

Proof. For (2.12) we have

$$\begin{aligned} \tilde{G}_j H_j^* &= \tilde{G}_j^{\text{old}} H_j^{\text{old}} = 0, \\ \tilde{H}_j G_j^* &= \left(\tilde{H}_j^{\text{old}} + S\tilde{G}_j^{\text{old}} \right) \left(G_j^{\text{old}} - S^*H_j^{\text{old}} \right)^* \\ &= \left(\tilde{H}_j^{\text{old}} + S\tilde{G}_j^{\text{old}} \right) \left(G_j^{*\text{old}} - H_j^{*\text{old}}S \right) \\ &= \tilde{H}_j^{\text{old}} G_j^{*\text{old}} - \tilde{H}_j^{\text{old}} H_j^{*\text{old}}S + S\tilde{G}_j^{\text{old}} G_j^{*\text{old}} - S\tilde{G}_j^{\text{old}} H_j^{*\text{old}}S \\ &= 0 - S + S + 0 = 0. \end{aligned}$$

Similarly, one can verify that (2.13) also holds. \square

The modification step to the original operators is called a *lifting step*. This modification changes \tilde{H} and G^* , but it does not change H^* and \tilde{G} . This implies that, after such a modification, the scaling functions remain the same, but the dual scaling functions and the wavelets change. The dual wavelets also change, since the dual scaling functions change, although the coefficients $(\tilde{g}_{j,m})$ in their refinement relation stay the same.

Dual lifting

There also exists another version of Theorem 2.3.1, in which the filters H^* and \tilde{G} are modified and \tilde{H} and G^* remain unchanged, which we call as the *dual lifting*.

Theorem 2.3.2. *Let $H^{\text{old}}, \tilde{H}^{\text{old}}, G^{\text{old}}, \tilde{G}^{\text{old}}$ be a set of biorthogonal filter operators. Then the following gives a new set of biorthogonal filter operators:*

$$\begin{aligned} H_j &= H_j^{\text{old}} + R^* G_j^{\text{old}}, \\ \tilde{H}_j &= \tilde{H}_j^{\text{old}}, \\ G_j &= G_j^{\text{old}}, \\ \tilde{G}_j &= \tilde{G}_j^{\text{old}} - R \tilde{H}_j^{\text{old}}, \end{aligned}$$

where R is any operator from $\ell^2(\mathcal{K}(j))$ to $\ell^2(\mathcal{M}(j))$.

2.3.4 Heuristics for a good discrete wavelet transform

Now, we can focus on a (single-level) discrete wavelet transform T_j , which maps a given discrete function $\lambda_{j+1} \in \ell^2((K(j+1)))$ into two discrete functions (λ_j, γ_j) ,

where $\lambda_j \in \ell^2((K(j)))$ and $\gamma_j \in \ell^2((M(j)))$ as

$$T_j : \ell^2\mathcal{K}(j+1) \rightarrow \ell^2(\mathcal{K}(j)) \times \ell^2(\mathcal{M}(j))$$

$$\lambda_{j+1} \mapsto (\lambda_j, \gamma_j).$$

Here, the discrete functions λ_j and γ_j are interpreted as approximation and detail functions, respectively.

Now, we can list the properties generally expected from a wavelet transform. In the following, we will implicitly identify the discrete function (λ_j) , with the corresponding function $f \in L^2(X, \Sigma, \mu)$ given by $f = \sum_k \lambda_{j,k} \varphi_{j,k}$, when talking about concepts like *smoothness*.

- W1: If λ_{j+1} belongs to a class of smooth functions to be specified (most commonly, a class of polynomials up to a certain degree), then the detail function γ_j is expected to be exactly zero.
- W2: λ_j is expected to be an approximation of λ_{j+1} , in the sense that setting $\gamma_j = 0$ and taking the inverse of $(\lambda_j, 0)$ under T_j should give an approximation of λ_{j+1} .
- W3: The transform is expected to be stable, i.e., a small change in λ_{j+1} should correspond to small changes in λ_j and γ_j , both in the transform and its inverse.
- W4: The transform is expected to be local. Changes in a localized area of the function should only affect a corresponding localized region of the transformed functions.
- W5: As a consequence of the above conditions, if λ_{j+1} is locally well approximated by the class of smooth functions, then many entries of the detail output γ_j will be close to zero. This is the reason for the *sparsity* property of the wavelet transforms, a key point of their success.

2.3.5 Tailoring a wavelet transform in successive lifting steps

In order to design a wavelet transform, one can start with a very simple transform, and in successive steps, improve it to meet the guidelines of Subsection 2.3.4. In doing so, Theorems 2.3.1 and 2.3.2 will be utilized.

The lazy wavelet transform

The lazy wavelet transform simply splits the given signal λ_{j+1} into two, i.e. λ_j and γ_j are just restrictions of λ_{j+1} to $\mathcal{K}(j)$ and $\mathcal{M}(j)$ respectively,

$$T_j^{\text{lazy}} : \ell^2\mathcal{K}(j+1) \rightarrow \ell^2(\mathcal{K}(j)) \times \ell^2(\mathcal{M}(j))$$

$$\lambda_{j+1} \mapsto (\lambda_{j+1}|_{\mathcal{K}(j)}, \lambda_{j+1}|_{\mathcal{M}(j)}).$$

In other words, the corresponding filters are given by $h_{j,k,l}^{\text{lazy}} = \delta_{k,l}$ and $g_{j,m,l}^{\text{lazy}} = \delta_{m,l}$. We will denote the corresponding filter operators as $\tilde{H}_j^{\text{lazy}}$ and $\tilde{G}_j^{\text{lazy}}$, and use the ordered pair notation for the operator as

$$T_j^{\text{lazy}} = (\tilde{H}_j^{\text{lazy}}, \tilde{G}_j^{\text{lazy}}).$$

It is easy to see that the lazy wavelet transform does not satisfy property W1. There is no reason to expect that the signal would have values close to zero on the complementary grid $\mathcal{M}(j)$, which are typically uniformly scattered through the initial grid $\mathcal{K}(j+1)$. However, the lazy wavelet transform is very helpful as an initial step in the lifting framework, to be followed by a sequence of improvement steps.

Prediction

The first improvement to the lazy wavelet transform comes with the help of a prediction operator. Let $\lambda_j^{(0)} = \tilde{H}_j^{\text{lazy}} \lambda_{j+1}$ and $\gamma_j^{(0)} = \tilde{G}_j^{\text{lazy}} \lambda_{j+1}$ be the outputs of the lazy

wavelet transform. The output of the prediction operator is a function of $\lambda_j^{(0)}$ and under the smoothness assumption, it approximates the signal $\gamma_j^{(0)}$

$$P: \ell^2(\mathcal{K}(j)) \rightarrow \ell^2(\mathcal{M}(j))$$

$$P\lambda_j^{(0)} \approx \gamma_j^{(0)}.$$

The improved transform now becomes

$$T_j^{(1)}: \ell^2(\mathcal{K}(j+1)) \rightarrow \ell^2(\mathcal{K}(j)) \times \ell^2(\mathcal{M}(j))$$

$$\lambda_{j+1} \mapsto (\lambda_j^{(0)}, \gamma_j^{(0)} - P\lambda_j^{(0)})$$

If the smoothness assumptions are satisfied, and the prediction successfully satisfies $P\lambda_j^{(0)} \approx \gamma_j^{(0)}$, then the desired property W1, follows immediately, as intended. The new filter operators are $\tilde{H}_j^{(1)} = \tilde{H}_j^{lazy}$ and $\tilde{G}_j^{(1)} = \tilde{G}_j^{lazy} - PH_j^{lazy}$. The complete set of biorthogonal filters can be obtained by Theorem 2.3.2.

Locality of the Prediction

In order to have the property W4, we must impose a condition on the prediction operator P : the m th entry of $P\lambda_j^{(0)}$ should depend only on the *neighbors* of the m th element of the complimentary grid $\mathcal{M}(j)$, noting that $\mathcal{K}(j) \cup \mathcal{M}(j) = \mathcal{K}(j+1)$. Without this condition, the transform would not be local, and would not possess the advantages of wavelet transforms.

The Update Step

At this point, we have a transform $T^{(1)}$ that satisfies property W1. However, there is something unsatisfying about the *approximation output* $\lambda_j^{(1)} := \tilde{H}_j^{(1)}\lambda_{j+1}$, which is still simply a restriction of λ_{j+1} to the subgrid $\mathcal{K}(j)$. This constitutes a violation of

property W2. In order to see this, one can take an extreme example for λ_{j+1} , such as

$$\lambda_{j+1,k} = \begin{cases} 0 & \text{if } k \in \mathcal{K}(j) \\ 1 & \text{if } k \in \mathcal{M}(j) \end{cases},$$

whose reconstruction after setting the detail coefficients γ_j to 0 will result in the 0 function, which is obviously not a good approximation for the initial λ_{j+1} . As a result, a new step is required, which will ensure that the local averages of the function reconstructed with λ_j will match those of λ_{j+1} . It can be accomplished by means of an operator U that is local, and acts on the detail output $\gamma_j^{(1)} := \tilde{G}_j^{(1)}\lambda_{j+1}$ of the previous step. The new operators will be $\tilde{H}_j^{(2)} := \tilde{H}_j^{(1)} + U\tilde{G}_j^{(1)}$, and $\tilde{G}_j^{(2)} := \tilde{G}_j^{(1)}$. The condition that is desired to be satisfied by the output $\lambda_j^{(2)}$ of $\tilde{H}_j^{(2)}$ is to preserve the weighted sum of the coefficients as

$$\sum_{k \in \mathcal{K}(j+1)} \lambda_{j+1,k} \mu(S_{j+1,k}) = \sum_{k \in \mathcal{K}(j)} \lambda_{j,k}^{(2)} \mu(S_{j,k}).$$

The modifications to get a complete set of biorthogonal filters is given by Theorem 2.3.1. Note also that the update step for the primal filters corresponds to a prediction step for the dual filters.

There can be as many “predict” and “update” steps as one desires, to be added with similar rules. However, in general additional steps with improved predictions requires the use of filters with a higher number of nonzero coefficients. This, in turn, results in less localized wavelets and scaling functions.

In this chapter, we gave an overview of wavelets and the lifting scheme, a flexible framework that enables designing wavelets on unstructured domains. This chapter was essentially a preparation for the next chapter, in which we give a concrete con-

struction that works for arbitrary subsets of \mathbb{R}^2 and \mathbb{R}^3 .

2.4 Appendix: Orthogonal bases, Riesz bases and frames

2.4.1 Riesz bases and orthogonal bases

In a Hilbert space \mathcal{H} , a basis (x_n) is called a *Riesz basis* if there exist two constants A and B such that

$$A \sum |a_n|^2 \leq \left\| \sum a_n x_n \right\|^2 \leq B \sum |a_n|^2, \quad (2.14)$$

for all scalar sequences (a_n) . A and B are called the Riesz basis constants.

An *orthogonal basis* is a special Riesz basis in which $A = B = 1$. In this case, the basis satisfies

$$\langle x_n, x_{n'} \rangle = \delta_{n,n'},$$

and for an $f \in \mathcal{H}$, the scalar coefficients (a_n) in the basis expansion

$$f = \sum_n a_n x_n$$

can be computed by taking inner products with the corresponding basis elements:

$$a_n = \langle f, x_n \rangle.$$

2.4.2 Biorthogonal bases

For each Riesz basis (x_n) that is not orthogonal, there exist another Riesz basis (\tilde{x}_n) that satisfies

$$\langle x_n, \tilde{x}_{n'} \rangle = \delta_{n,n'},$$

and therefore one has the relation

$$a_n = \langle f, \tilde{x}_n \rangle,$$

useful for the computation of the coefficients. The bases (x_n) and (\tilde{x}_n) are called *biorthogonal* bases. For the proof of existence of a biorthogonal basis for every Riesz basis, we refer to [1].

2.4.3 Frames

A frame is a countable collection of vectors (e_n) in a Hilbert space \mathcal{H} , for which there exists constants A and B such that

$$A\|f\|^2 \leq \sum_n |\langle f, e_n \rangle|^2 \leq B\|f\|^2.$$

Frames are generally regarded as redundant families of vectors: a frame is not necessarily linearly independent, and a frame expansion of a function is not unique.

Chapter 3

Randomized domain-adapted wavelets

In this chapter we give details about our construction of two and three dimensional wavelets on an arbitrary domain, and study their properties. This construction constitutes one of our main contributions to the thesis.

3.1 The structures needed for the construction

In order to start defining and shaping filter operators within the lifting framework, we first need the embedded index sets (grids) $\mathcal{K}(j)$ for $j \in \mathcal{J}$, as defined in Chapter 2. The scale index set will be $\mathcal{J} = \mathbb{N}$, but in practice we will only consider the finite subset $\mathcal{J} = \{0, 1, 2, \dots, N\}$ for some $N \in \mathbb{N}$. After constructing $(\mathcal{K}(j))$, we can give the corresponding detail index sets $(\mathcal{M}(j))$ and the nested partitionings $(S_{j,k})$. We will also need *neighbor*, *sibling*, *parent* and *child* relations on $(\mathcal{K}(j))$, to be used in the *prediction* and *update* steps of the lifting.

3.1.1 The domain

The domain \mathcal{X} , which will be an input to the algorithm, is a subset of \mathbb{Z}^2 or \mathbb{Z}^3 . For the case of \mathbb{Z}^3 , the domain \mathcal{X} actually corresponds to the intersection of a set $X \subset \mathbb{R}^3$ with the discrete grid $\{(n_x d_x, n_y d_y, n_z d_z) : (n_x, n_y, n_z) \in \mathbb{Z}^3\}$, where d_x , d_y and d_z are the resolution parameters in the corresponding directions. Each $n \in \mathcal{X}$ represents the rectangular region

$$\begin{aligned} S_n = & [n_x d_x - d_x/2, n_x d_x + d_x/2] \\ & \times [n_y d_y - d_y/2, n_y d_y + d_y/2] \\ & \times [n_z d_z - d_z/2, n_z d_z + d_z/2], \end{aligned} \tag{3.1}$$

which is called a *voxel*. A similar definition holds for two dimensions, in which the set S_n is called a *pixel*. An example of a discrete domain and the corresponding pixels is displayed in Figure 3.1.

The measure

The measure of each set S_n , denoted by $\mu(S_n)$, is another required input. In the regular setting, one takes $\mu(S_n) = 1$ for all $n \in \mathcal{X}$, but in some applications it may be more meaningful to weight each voxel according to some related measure.

Any set that we shall consider in the finite setting will be a union of the sets (S_n) , i.e sets of the form

$$S = \bigcup_{n \in \mathcal{X}'} S_n$$

for some $\mathcal{X}' \subset \mathcal{X}$, whose measure is given by

$$\mu(S) = \sum_{n \in \mathcal{X}'} \mu(S_n).$$

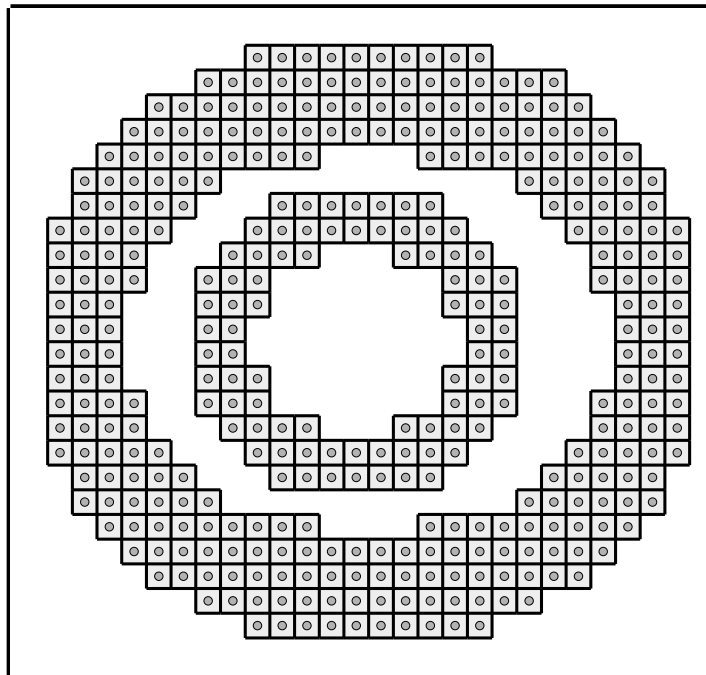


Figure 3.1: A two dimensional discrete domain \mathcal{X} , whose elements are displayed as dots, and the corresponding pixels (S_n) are displayed as the squares surrounding the dots.

Neighboring elements

We define two elements $n, m \in \mathcal{X} \subset \mathbb{Z}^3$ to be *neighbors*, if

$$|n_x - m_x| + |n_y - m_y| + |n_z - m_z| = 1,$$

where $n = (n_x, n_y, n_z)$ and $m = (m_x, m_y, m_z)$. This is equivalent to the condition that the rectangular prisms bounding regions S_n and S_m share a face. A similar definition also holds for two dimensions.

3.1.2 Embedded grids and the random merging algorithm

We choose an $N \in \mathbb{N}$, which is the number of decomposition levels. We take the scale index set \mathcal{J} to be $\{0, 1, 2, \dots, N\}$, where $j = N$ represents the scale of highest resolution available.

We set $\mathcal{K}(N)$ to be equal to the discrete domain \mathcal{X} . For every $k \in \mathcal{K}(N)$, we denote the set of neighbors of k with $\text{Nbr}(N, k) \subset \mathcal{K}(N)$, based on the neighborhood structure of \mathcal{X} . Also we set $S_{N,k} := S_k$, where S_k is the set defined in (3.1). After defining $\mathcal{K}(N)$, we will define $\mathcal{K}(N-1), \mathcal{K}(N-2), \dots, \mathcal{K}(0)$, and the corresponding neighborhood structure with a random merging algorithm, inductively.

For a given j , assume we have $\mathcal{K}(j)$, the sets $(S_{j,k})_{k \in \mathcal{K}(j)}$, and a collection of sets $\{\text{Nbr}(j, k) : k \in \mathcal{K}(j)\}$ of the neighbors for each element.

1. Start with $\mathcal{K}(j-1) = \mathcal{M}(j-1) = \emptyset$.
2. Compute the centroids of all sets $S_{j,k}$, which is defined as

$$C(S_{j,k}) = \frac{1}{\mu(S_{j,k})} \sum_{S_{N,k} \subset S_{j,k}} (k_x d_x, k_y d_y, k_z d_z) \mu(S_{N,k}),$$

which gives a vector $C(S_{j,k}) \in \mathbb{R}^3$.

3. Declare all elements of $\mathcal{K}(j)$ as available, and put them in a linear order that is determined by a random permutation,
4. For the first available $k \in \mathcal{K}(j)$, select at most p of the available neighbors, $s_1, s_2, \dots, s_q \in \mathcal{K}(j)$, where $q \leq p$, in a random way, with probability of being selected being inversely proportional to the distance between the centroids,
5. Add k to $\mathcal{K}(j-1)$ and each of s_1, s_2, \dots, s_q to $\mathcal{M}(j-1)$,
6. Mark each of $k, s_1, s_2, \dots, s_q \in \mathcal{K}(j)$ to be *unavailable*, and declare them to be *siblings*, a relationship denoted as

$$\text{Sib}(j, k) = \text{Sib}(j, s_1) = \dots = \text{Sib}(j, s_q) = \{k, s_1, s_2, \dots, s_q\},$$

7. Define the set $S_{j-1, k} := \bigcup_{l \in \text{Sib}(j, k)} S_{j, l}$,
8. Go to Step 4, until all elements of $\mathcal{K}(j)$ are marked as unavailable,
9. Declare two elements $k_1, k_2 \in \mathcal{K}(j-1)$ to be neighbors, if there exists $s_1 \in \text{Sib}(j, k_1)$ and $s_2 \in \text{Sib}(j, k_2)$ such that s_1 and s_2 are neighbors in $\mathcal{K}(j)$, i.e., $s_1 \in \text{Nbr}(j, s_2)$.

There are two sources of randomness in this algorithm: one is in the random permutation of $\mathcal{K}(j)$ in Step 3 and the other is in the random choice of neighboring sets to be selected to be merged in Step 4.

It can be verified that the set of partitionings $(S_{j, k})$ that is output by the algorithm, after being extended for levels $j = N+1, N+2, \dots$ in some suitable way, satisfies Definition 2.2.4.

The parameter p in Step 4 is mostly set to 3 in our experiments.

The formation of siblings is displayed in Figure 3.2, an example output of the algorithm in Figure 3.3, and the graphs indicating the neighbors is shown in Figure 3.4.

3.2 The wavelet transform

After obtaining the necessary structures, such as the nested index sets $(\mathcal{K}(j))_j$, their complements $(\mathcal{M}(j))_j$, and the neighborhood and sibling relations on them, we are now ready to define the filter operators that identify the wavelet transform. Our wavelet transform is basically a version of David Donoho's *average interpolating wavelets* [10], which can be obtained with a three-stage lifting framework [29]. Following the splitting step (lazy wavelet transform), it continues with a prediction, an update and another prediction step. The first two steps are similar to Haar decomposition, and result in wavelets similar to the ones called *unbalanced Haar wavelets* [15]. These wavelets are orthogonal, and the transform can be considered to be a meaningful wavelet transform in its own right.

The second prediction step is a version of the average interpolation prediction, which converts the Haar-like wavelets of the previous step into smoother functions. With this step, we lose orthogonality, in return for smooth wavelet and scaling functions, which entails better sparsity properties for the output.

3.2.1 The finest available resolution

In the practical implementation, the scale index $\mathcal{J} = \{1, 2, \dots, N\}$ is finite, there is a highest available resolution, which is determined by the measuring device's capabilities. The discrete function input to the transform $f_d \in \ell^2(\mathcal{X})$ is assumed to come from a corresponding function $f \in L^2(X)$, with

$$f_d(n) = \frac{1}{\mu(S_n)} \int_{S_n} f d\mu, \quad \text{for } n \in \mathcal{X} \quad (3.2)$$

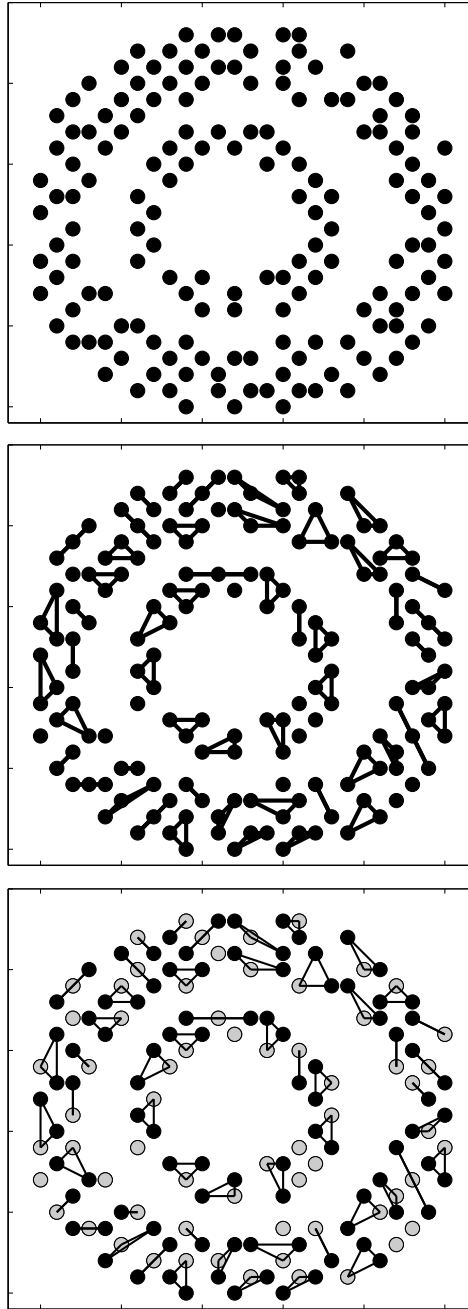


Figure 3.2: Illustration of merging and splitting of the grid $\mathcal{K}(j)$. The top row is the original grid $\mathcal{K}(j)$. In the second row, elements of the grid are randomly merged with neighbors, and they form groups consisting of one, two or three elements. In the last row, one element in each group, determined in random way, is displayed in a lighter color. The elements shown in light color make up $\mathcal{K}(j - 1)$, and grid elements that remain dark-colored belong to the complimentary grid $\mathcal{M}(j - 1)$.

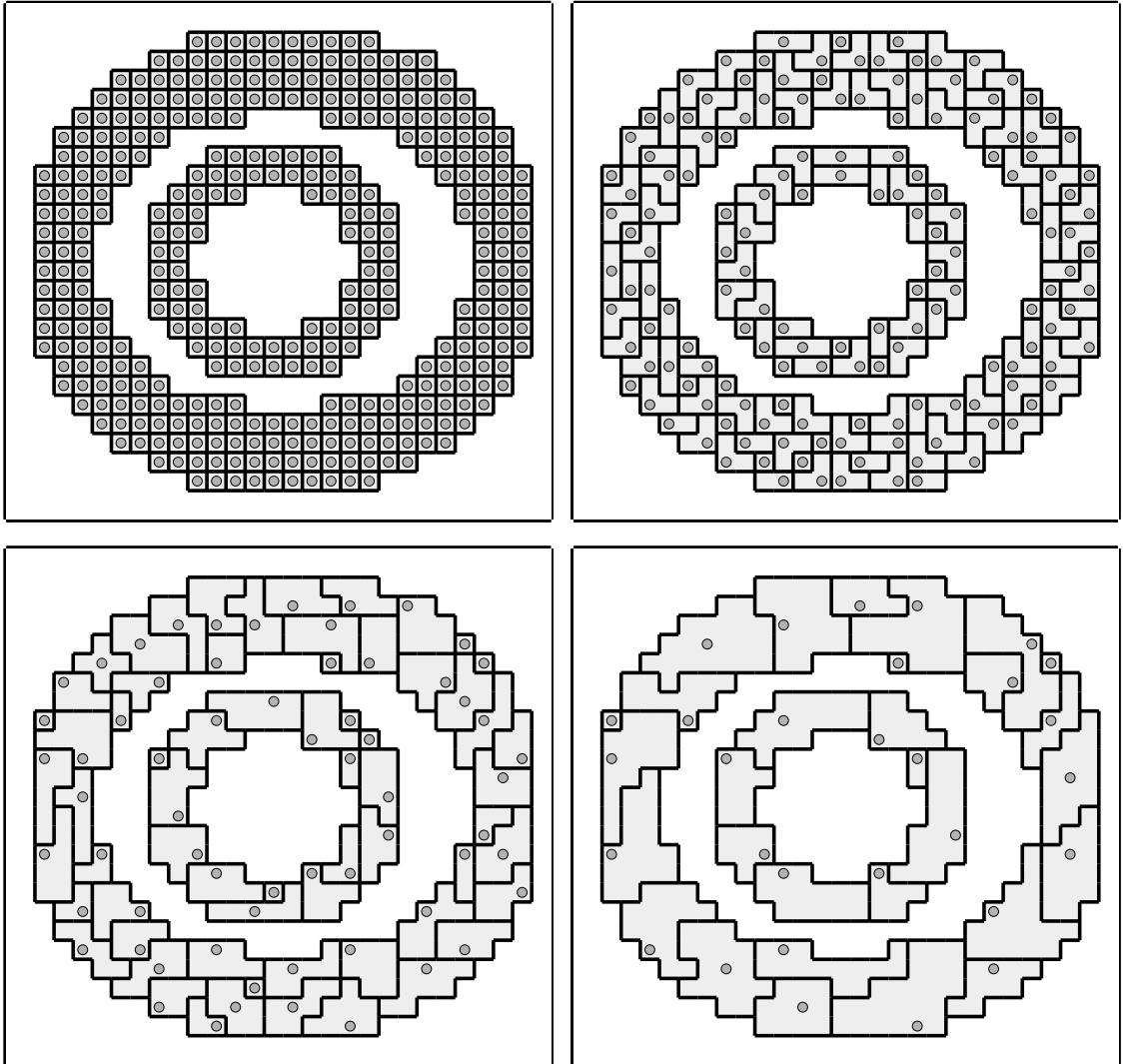


Figure 3.3: The embedded grid and the corresponding regions for a three level decomposition. In each figure, the dots represent $\mathcal{K}(3)$, $\mathcal{K}(2)$, $\mathcal{K}(1)$ and $\mathcal{K}(0)$ in the reading order. The shaded regions bounded by lines represent $(S_{3,k})_{k \in \mathcal{K}(3)}$, $(S_{2,k})_{k \in \mathcal{K}(2)}$, $(S_{1,k})_{k \in \mathcal{K}(1)}$ and $(S_{0,k})_{k \in \mathcal{K}(0)}$.

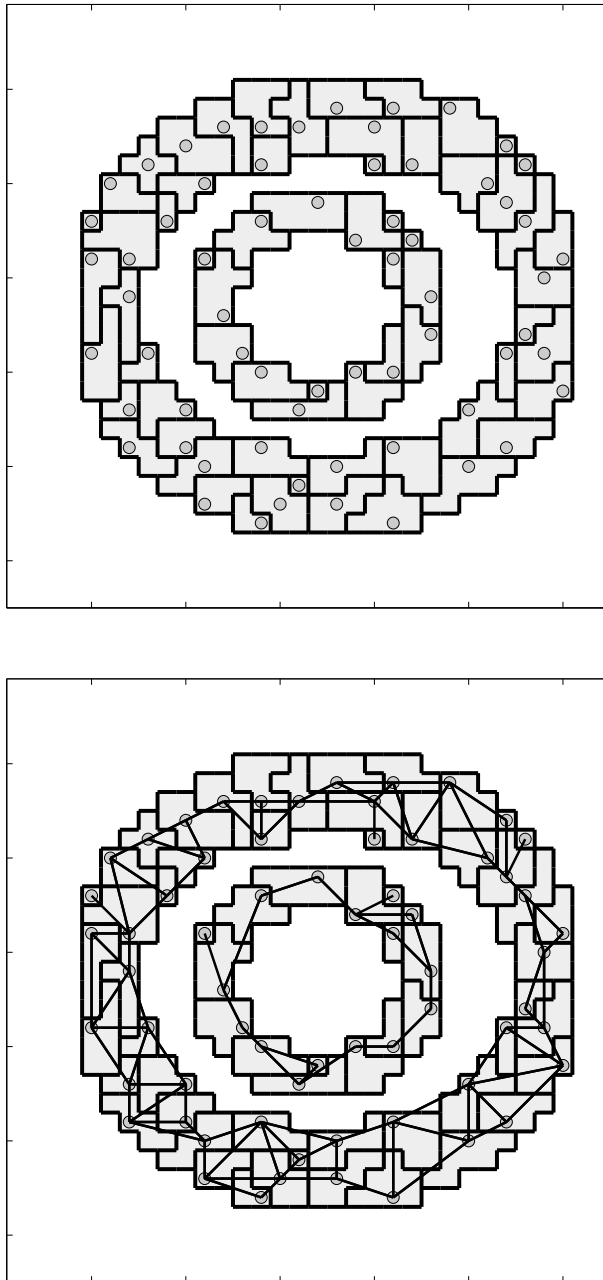


Figure 3.4: Grid neighbors and regions. In the top each grid element $k \in \mathcal{K}(j)$ is shown inside the region $S_{j,k}$ it represents. The bottom row displays the neighborhood graph, where the condition for two elements to be neighbors is equivalent to the regions they represent sharing a nontrivial boundary. Note that there are grid element pairs that are geographically very close but that nevertheless are not neighbors of each other.

for some measure that agrees with the $(\mu(S_n))_n$ of Subsection 3.1.1. As such, the dual scaling functions of the finest level N are defined to be

$$\tilde{\varphi}_{N,k} = \frac{1}{\mu(S_{N,k})} \chi(S_{N,k}),$$

and the primal scaling functions $(\varphi_{N,k})_{k \in \mathcal{K}(N)}$ are

$$\varphi_{N,k} = \chi(S_{N,k}),$$

where χ denotes the characteristic function, which takes 1 inside the set and 0 elsewhere. Accordingly, the finest resolution scaling function coefficients $\lambda_{N,j}$ are simply taken to be equal to the $f_d(n)$'s, and as a result

$$\sum_{k \in \mathcal{K}(N)} \lambda_{N,k} \varphi_{N,k} = \sum_{n \in \mathcal{X}} f_d(n) \chi(S_n) \approx f,$$

noting that $\mathcal{K}(N) = \mathcal{X}$ and $S_k = S_{N,k}$.

In synthesizing the scaling functions and the wavelets with the algorithm in Subsection 2.2.3, we will have to stop at $j = N$. Hence, in a sense, the scaling functions $(\varphi_{N,k})_k$ will be the atomic building blocks for the wavelets and scaling functions that we consider, in the same sense that the sets $(S_{N,k})_k$ are the building blocks for all sets that we consider, in the construction.

3.2.2 The lifting steps

We assume an input $\lambda_{j+1} = (\lambda_{j+1,k})_{k \in \mathcal{K}(j)}$ is given. The analysis filters will initially perform the lazy wavelet transform, which is a simple splitting. This initial pair of analysis filters will be denoted as $(\tilde{H}_j^{\text{lazy}}, \tilde{G}_j^{\text{lazy}})$ and their outputs as $(\lambda_j^{(0)}, \gamma_j^{(0)})$. Three lifting steps (a prediction, an update and another prediction) will modify the filters. The resulting analysis filter pairs will be denoted as $(\tilde{H}_j^{(1)}, \tilde{G}_j^{(1)})$, $(\tilde{H}_j^{(2)}, \tilde{G}_j^{(2)})$ and

$(\tilde{H}_j^{(3)}, \tilde{G}_j^{(3)})$. The outputs of these filters will be denoted as $(\lambda_j^{(1)}, \gamma_j^{(1)})$, $(\lambda_j^{(2)}, \gamma_j^{(2)})$ and $(\lambda_j^{(3)}, \gamma_j^{(3)})$, respectively. The lifting operators will be denoted by P_1 , U and P_2 , respectively, and the filters will satisfy

$$\begin{aligned}\tilde{H}_j^{(1)} &= \tilde{H}_j^{\text{lazy}} \\ \tilde{G}_j^{(1)} &= \tilde{G}_j^{\text{lazy}} - P_1 \tilde{H}_j^{\text{lazy}} \\ \tilde{H}_j^{(2)} &= \tilde{H}_j^{(1)} + U \tilde{G}_j^{(1)} \\ \tilde{G}_j^{(2)} &= \tilde{G}_j^{(1)}\end{aligned}$$

and,

$$\begin{aligned}\tilde{H}_j^{(3)} &= \tilde{H}_j^{(2)} \\ \tilde{G}_j^{(3)} &= \tilde{G}_j^{(2)} - P_2 \tilde{H}_j^{(2)}.\end{aligned}$$

In the implementation, one does not need to explicitly compute these filters. Starting with λ_{j+1} and, one can obtain $\lambda_j^{(3)}$ and $\gamma_j^{(3)}$ in a sequence of operations

$$\begin{aligned}\lambda_j^{(0)} &= \tilde{H}_j^{\text{lazy}} \lambda_{j+1}, & \gamma_j^{(0)} &= \tilde{G}_j^{\text{lazy}} \lambda_{j+1}, \\ \lambda_j^{(1)} &= \lambda_j^{(0)}, & \gamma_j^{(1)} &= \gamma_j^{(0)} - P_1 \lambda_j^{(0)}, \\ \lambda_j^{(2)} &= \lambda_j^{(1)} + U \gamma_j^{(1)}, & \gamma_j^{(2)} &= \gamma_j^{(1)}, \\ \lambda_j^{(3)} &= \lambda_j^{(2)}, & \gamma_j^{(3)} &= \gamma_j^{(2)} - P_2 \lambda_j^{(2)},\end{aligned}$$

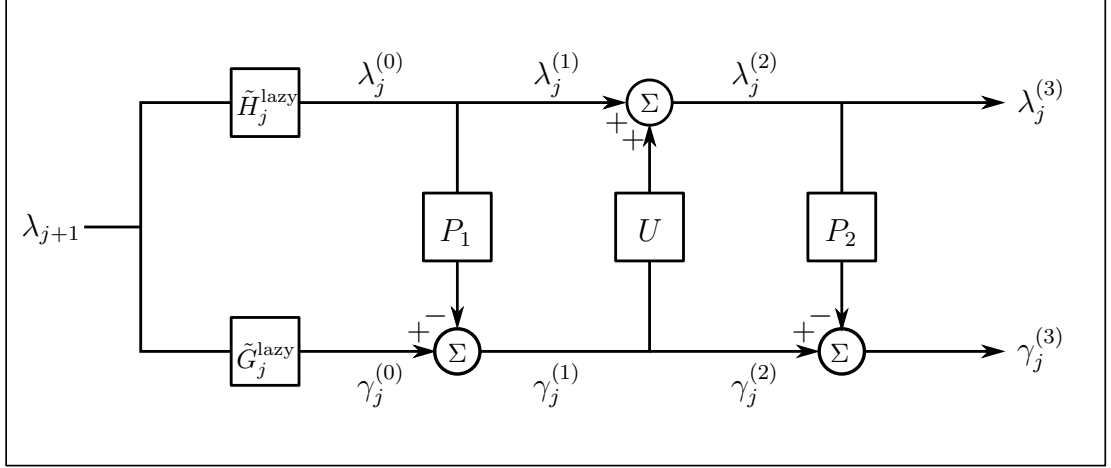


Figure 3.5: Diagram illustrating the implementation of the three-stage lifting.

as displayed in Figure 3.5. Similarly, the inverse transform is implemented by reversing the steps:

$$\begin{aligned}
 \lambda_j^{(2)} &= \lambda_j^{(3)}, & \gamma_j^{(2)} &= \gamma_j^{(3)} + P_2 \lambda_j^{(2)}, \\
 \lambda_j^{(1)} &= \lambda_j^{(2)}, & \lambda_j^{(1)} &= \lambda_j^{(2)} - U \gamma_j^{(1)}, \\
 \lambda_j^{(0)} &= \lambda_j^{(1)}, & \gamma_j^{(0)} &= \gamma_j^{(1)} + P_1 \lambda_j^{(0)}, \\
 \lambda_{j+1} &= H_j^{*lazy} \lambda_j^{(0)} + G_j^{*lazy} \gamma_j^{(0)}.
 \end{aligned}$$

The filters $(\tilde{H}_j^{(2)}, \tilde{G}_j^{(2)}) := (\tilde{H}_j^{\text{Haar}}, \tilde{G}_j^{\text{Haar}})$ that come after the second lifting step actually corresponds to the unbalanced Haar transform, which we will give in the next subsection. The third lifting step is based on *average interpolation*, and the resulting filters after this step, $(\tilde{H}_j^{(3)}, \tilde{G}_j^{(3)}) := (\tilde{H}_j^{\text{AI}}, \tilde{G}_j^{\text{AI}})$, are called average interpolating filters, to be explained in Subsection 3.2.4.

3.2.3 Unbalanced Haar wavelets

First prediction

The first prediction operator P_1 , is a function from $\ell^2(\mathcal{K}(j))$ to $\ell^2(\mathcal{M}(j))$. First we note that, for each $m \in \mathcal{M}(j)$ there exists a unique $k_m \in \mathcal{K}(j)$ such that $k_m \in \text{Sib}(j+1, m)$. If $a = (a_k)_{k \in \mathcal{K}(j)}$ is the input to P_1 , the m th entry in the output of P_1 is simply defined to be equal to a_{k_m} , i.e.,

$$P_1: \ell^2(\mathcal{K}(j)) \rightarrow \ell^2(\mathcal{M}(j)), \quad P_1 a = b \text{ means}$$

$$b_m = a_{k_m},$$

where k_m is the unique element of $\mathcal{K}(j) \cap \text{Sib}(j+1, m)$.

The first prediction step is based on predicting the value of an entry $\lambda_{j+1,k}$ to be the same value as λ_{j+1,k_m} , for a specific neighbor k_m in $\mathcal{K}(j+1)$. The idea is very similar to *nearest neighbor interpolation*, except that the neighbor used in prediction is not necessarily the nearest one, but is determined by the structure of the embedded grids. For the example in Figure 3.2, the value at the dark colored spots in the lowest row is to be predicted with the value in the corresponding light colored spot.

We note that, after this first prediction step, one has

$$\begin{aligned} \gamma_{j,s_1}^{(1)} &= \lambda_{j+1,s_1} - \lambda_{j+1,k} \\ &\vdots \\ \gamma_{j,s_q}^{(1)} &= \lambda_{j+1,s_q} - \lambda_{j+1,k}. \end{aligned} \tag{3.3}$$

where $k \in \mathcal{K}(j)$ and $\text{Sib}(k, j+1) = \{k, s_1, s_2, \dots, s_q\}$, with q being the number of siblings.

Update

The update operator U is a function from $\ell^2(\mathcal{M}(j))$ to $\ell^2(\mathcal{K}(j))$ that modifies the output of the previous step as $\lambda_j^{(2)} = \lambda_j^{(1)} + U\gamma_j^{(1)}$, in order to guarantee that

$$\sum_{k \in \mathcal{K}(j+1)} \lambda_{j+1,k} \mu(S_{j+1,k}) = \sum_{k \in \mathcal{K}(j)} \lambda_{j,k}^{(2)} \mu(S_{j,k}).$$

It is sufficient, and usually expected, that this condition is satisfied locally, by having

$$\begin{aligned} \lambda_{j,k}^{(2)} \mu(S_{j,k}) &= \lambda_{j+1,k} \mu(S_{j+1,k}) + \lambda_{j+1,s_1} \mu(S_{j+1,s_1}) \\ &+ \cdots + \lambda_{j+1,s_q} \mu(S_{j+1,s_q}), \end{aligned} \quad (3.4)$$

for all $k \in \mathcal{K}(j)$, and $s_1, s_2, \dots, s_q \in \text{Sib}(k, j+1)$. This can be accomplished with an update operator U given by

$$\begin{aligned} U_j: \ell^2(\mathcal{M}(j)) &\rightarrow \ell^2(\mathcal{K}(j)), \quad Ua = b \text{ means} \\ b_k &= \frac{a_{s_1} \mu(S_{j+1,s_1}) + \cdots + a_{s_q} \mu(S_{j+1,s_q})}{\mu(S_{j+1,k}) + \mu(S_{j+1,s_1}) + \cdots + \mu(S_{j+1,s_q})}, \end{aligned}$$

where $\{k, s_1, s_2, \dots, s_q\} = \text{Sib}(j+1, k)$ for all $k \in \mathcal{K}(j)$. It can be verified that after this step, one has

$$\lambda_{j,k}^{(2)} = \frac{\lambda_{j+1,k}^{(0)} \mu(S_{j+1,k}) + \lambda_{j+1,s_1}^{(0)} \mu(S_{j+1,s_1}) + \cdots + \lambda_{j+1,s_q}^{(0)} \mu(S_{j+1,s_q})}{\mu(S_{j+1,k}) + \mu(S_{j+1,s_1}) + \cdots + \mu(S_{j+1,s_q})}, \quad (3.5)$$

which is the same as (3.4), noting that

$$\mu(S_{j,k}) = \mu(S_{j+1,k}) + \mu(S_{j+1,s_1}) + \cdots + \mu(S_{j+1,s_q}).$$

Wavelets and scaling functions

Using (3.2) and (3.5), it can be shown inductively that the dual scaling functions $\tilde{\varphi}_{j,k}$ are given by

$$\tilde{\varphi}_{j,k} = \frac{1}{\mu(S_{j,k})} \chi(S_{j,k}),$$

and using (3.3) one gets the dual wavelets ($\tilde{\psi}_{j,m}$) as

$$\tilde{\psi}_{j+1,m} = \frac{1}{\mu(S_{j+1,k_m})} \chi(S_{j+1,k_m}) - \frac{1}{\mu(S_{j+1,m})} \chi(S_{j+1,m}),$$

where k_m is the unique element of $\text{Sib}(j+1, m) \cap \mathcal{K}(j)$.

It is easy to verify that the collection of dual scaling functions and wavelets are orthogonal to each other, which implies that the primal scaling functions and wavelets are the same as the dual ones, up to normalizing constants. These type of wavelets are studied in [15], where they are named *unbalanced Haar wavelets*.

3.2.4 Average interpolating wavelets

The unbalanced Haar wavelets have the advantage of being orthogonal, and they are relatively simple to design, but they have the drawback of not being smooth, not even continuous. In a wavelet representation

$$f = \sum_{j,m} \gamma_{j,m} \psi_{j,m},$$

we usually seek *sparse representations*, meaning that most of the coefficients $\gamma_{j,m}$'s are zero or at least negligible. This is usually not achieved by Haar wavelets to the most pronounced degree, because there is a link between the smoothness of wavelets and fast decay of wavelet coefficients. Intuitively, we can say that Haar wavelets, not being smooth, find it difficult to represent smooth functions and require a larger number of basis elements than is needed by a more smooth basis. Therefore we would like to

have a wavelet basis without sharp discontinuities and certain smoothness properties.

By adding a second prediction operator P_2 to the lifting scheme, it is possible to obtain smoother wavelets out of Haar wavelets, and that give smaller detail coefficients for smooth signals. Or, when we think the other way around, if the prediction operator P_2 is designed to make the entries of the detail output smaller for smooth functions, then the corresponding wavelets will turn out to be smoother than Haar wavelets.

The first prediction step P_1 was based on a version of *nearest neighbor interpolation*, while this second prediction step is based on the notion of *average interpolation*, which we explain next. The use of average interpolation within the context of wavelets is due to Donoho [10].

Average interpolation

In the following, all sets considered are subsets of \mathbb{R}^n , and μ is a measure on \mathbb{R}^n . The average $\frac{1}{\mu(S)} \int_S f d\mu$ is denoted shortly as $f_S f$, for a given $S \subset \mathbb{R}^n$.

Let $S_1, S_2, \dots, S_r, C_1, C_2, \dots, C_t \subset \mathbb{R}^n$ be mutually disjoint sets that are geographically close, for some integers r and t . Let f be an unknown function, which is assumed to be locally well approximated by polynomials.

It is assumed that the averages $f_{S_1} f, \dots, f_{S_r} f$ are known, and the goal is to estimate $f_{C_1} f, \dots, f_{C_t} f$.

The approach is to find a polynomial p whose average values agree with f on S_1, \dots, S_r , i.e.,

$$\int_{S_i} f = \int_{S_i} p \tag{3.6}$$

for $i = 1, 2, \dots, r$. After such a polynomial is found, the estimates for $f_{C_1} f, \dots, f_{C_t} f$ are given to be $f_{C_1} p, \dots, f_{C_t} p$, respectively.

After this brief description of average interpolation, we can define the second prediction operator that is used to obtain the average interpolating wavelets.

Second prediction

The second prediction operator P_2 maps $\ell^2(\mathcal{K}(j))$ to $\ell^2(\mathcal{M}(j))$, aiming to estimate $\gamma_j^{(2)}$ using $\lambda_j^{(2)}$. Here using (3.5) and assuming the finest level coefficients $\lambda_{N,j}$ are local averages of some input function f as in (3.2), we get

$$\lambda_{j,k}^{(2)} = \int_{S_{j,k}} f.$$

Also for $m \in \mathcal{M}(j)$, by (3.3), one has

$$\begin{aligned} \gamma_{j,m}^{(2)} &= \gamma_{j,m}^{(1)} = \lambda_{j+1,m} - \lambda_{j+1,k_m} \\ &= \int_{S_{j+1,m}} f - \int_{S_{j+1,k_m}} f, \end{aligned}$$

where k_m is the unique element of $\mathcal{K}(j) \cap \text{Sib}(j+1, m)$. We aim to estimate this value, using $\lambda_{j,n_1}^{(2)}, \dots, \lambda_{j,n_r}^{(2)}$, where $\{n_1, n_2, \dots, n_r\} = \text{Nbr}(j, k_m)$. This is actually the problem of estimating $\int_{S_{j+1,m}} f$ and $\int_{S_{j+1,k_m}} f$ using the values $\int_{S_{j,n_1}} f, \dots, \int_{S_{j,n_r}} f$, which is to be handled by average interpolation.

In our implementation, for each $k \in \mathcal{K}(j)$ we solve for the first degree polynomial

$$p_{j,k}(x, y, z) = a_{j,k} + b_{j,k}x + c_{j,k}y + d_{j,k}z \tag{3.7}$$

that minimizes

$$\sum_{n \in \text{Nbr}(j,k)} \left| \lambda_{j,n}^{(2)} - \int_{S_{j,n}} p_{j,k} \right|^2.$$

This is a linear problem, and each of the polynomial coefficients in (3.7) is a linear combination of $\lambda_{j,n_1}^{(2)}, \dots, \lambda_{j,n_r}^{(2)}$. So we can define P_2 to be the linear operator such that the m th entry of $P_2 \lambda_j^{(2)}$ is given by

$$\int_{S_{j+1,m}} p_{j,k_m} - \int_{S_{j+1,k_m}} p_{j,k_m},$$

noting that this quantity is also a linear combination of $\lambda_{j,n_1}^{(2)}, \dots, \lambda_{j,n_r}^{(2)}$. If the input f to the algorithm is itself a first degree polynomial, then the fit would be perfect and the prediction will give exact quantities.

After this final prediction step, one can add a normalizing step to the transform that makes each of the wavelets and the dual wavelets of unit norm.

3.3 Numerical experiments

3.3.1 The potential to reduce noise by sparse representation

The underlying point in transform-based noise reduction methods is the sparse representation property. A good transform enables a representation that puts most of the information about a smooth function into a relatively small fraction of the coefficients, while spreading out a signal uniformly to all components if the signal is noise-like, having little spatial correlation. More explicitly if we denote the wavelet representation of a function with a simplified single indexed notation as

$$f = \sum_{m \in \mathcal{M}} \gamma_m \psi_m,$$

then we assume that for the f of interest to us, there exists a set $\mathcal{A} \subset \mathcal{M}$ with $|\mathcal{A}| \ll |\mathcal{M}|$ that gives

$$f_{\text{approx}} := \sum_{m \in \mathcal{A}} \gamma_m \psi_m \approx f. \quad (3.8)$$

For the case of orthogonal wavelets, the best choice for such sets are generally of the form

$$\mathcal{A} = \{m \in \mathcal{M} : |\gamma_m| > \tau\}, \quad (3.9)$$

for some threshold τ . These type of sets give the best approximation in the L^2 sense, among the sets of the same size. This way of choosing \mathcal{A} usually still works even when

the wavelet basis is not orthogonal for some signal subclasses, if the corresponding frame bounds are close to tight.

Comparing adaptive wavelets with standard wavelets

In this numerical experiment, we started with a domain X consisting of concentric rings, contained in a rectangle R , as displayed in the top row of Figure 3.6. We generated a smooth function on R , which is a different random linear combination of two dimensional Gaussians on each connected component of the X . On $R \setminus X$ the domain also contains another smooth function of the same type. That is to say, if the domain can be written as $X = S_1 \cup \dots \cup S_n$, where each of S_1, \dots, S_n is a single, connected ring, and if f_1, \dots, f_{n+1} are smooth functions on R , then our input f is taken to be

$$f = f_1 \chi_{S_1} + f_n \chi_{S_n} + f_{n+1} \chi_{R \setminus X},$$

χ_S denoting the characteristic function of a set S .

This choice of f is motivated by the fMRI problem, where the brain cortex, which is the natural domain of the measured data, has a convoluted structure, and geographically close parts of it may carry signals that are of different nature.

We computed the coefficients under different wavelet transforms, and computed reconstructed approximations in the form of (3.8), the coefficient sets \mathcal{A} being selected with thresholding as in (3.9). For each different threshold, we computed the approximation error as

$$\frac{\|f - f_{\text{approx}}\|}{\|f\|}.$$

Although the standard wavelet transforms give reconstructions over the whole rectangle R , we compute the norms only over the domain X , i.e.,

$$\|f\| = \sqrt{\int_X |f(x)|^2 dx}.$$

We also generated multiple realizations of the noise n over the whole square, whose samples are i.i.d. standard Gaussian random variables. We transformed each of the realizations of the noise with the same wavelets, and reconstructed it using only entries from the coefficient set \mathcal{A} that was determined by the function f , and computed the expected norm $E[\|n_{\text{rec}}\|]$, of the reconstructed noise. That is, we process the signal f and the noise n separately, and plot $\frac{\|f-f_{\text{approx}}\|}{\|f\|}$ versus $E[\|n_{\text{rec}}\|]$, while the threshold parameter τ is being reduced. This gives a measure of the potential to reduce noise of the related transform on the domain X .

The results are plotted in Figure 3.6, where the experiment is done with three standard wavelet bases that have the domain R , and the domain-adapted wavelets that are constructed for X . For each choice of wavelets, the experiment is repeated for the decomposition levels 1,2 and 3. The results show that increasing the decomposition level gives a much more pronounced positive effect with the domain adapted wavelets than the standard wavelets.

Comparing different domains

We repeated the experiment while changing the thickness of each of the rings of the domain, or the gap between the rings corresponding to the off-domain regions. The results are given in Figure 3.7. This experiment shows that, in order for the adapted wavelets to have a clear advantage over the standard ones, the rings or the gaps between them must be sufficiently thin. For an example like the one given in the first row, the adapted wavelets do not have a considerable advantage over the wavelets that have the whole rectangle R as their domain.

3.3.2 Improvement due to averaging

The randomness in our wavelet algorithm allows us to repeat a signal processing task with multiple realization of the wavelets, and then take the average of the results. In

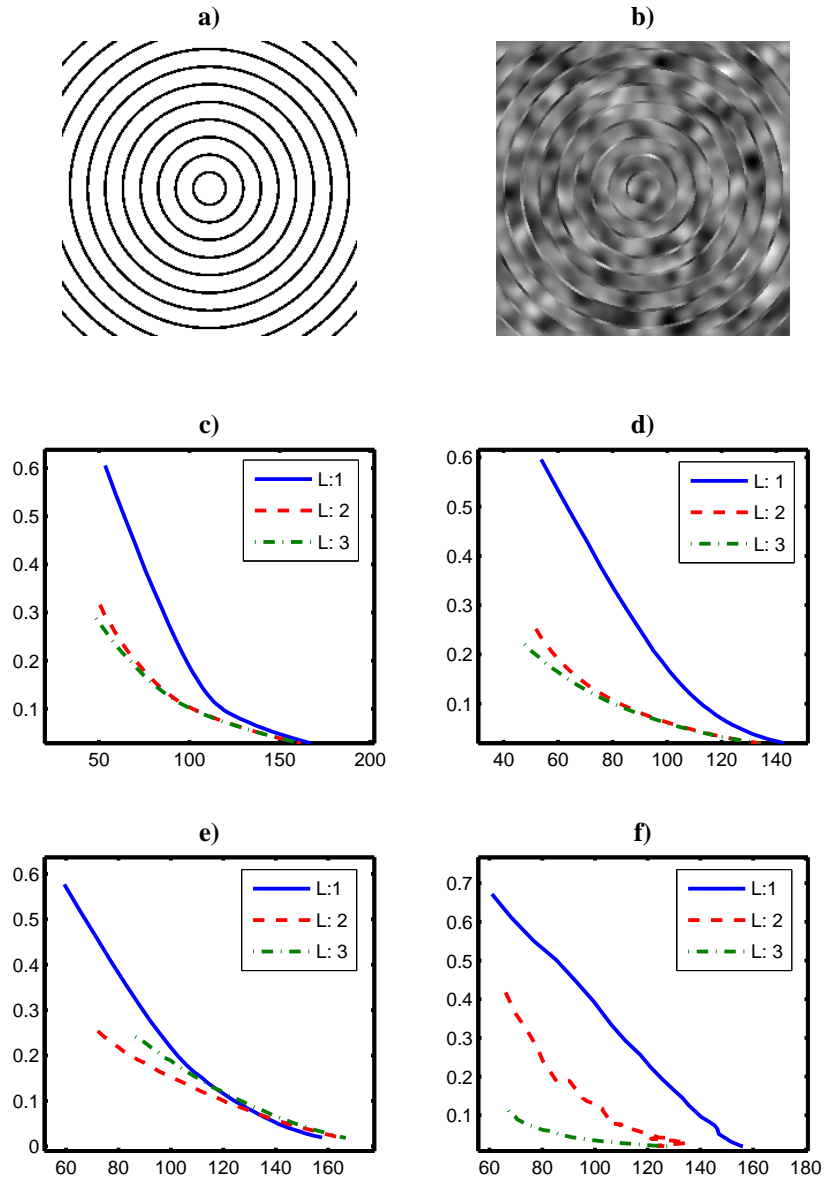


Figure 3.6: Comparing denoising potentials of different conventional wavelets, and the domain-adapted wavelets. Relative approximation error (vertical) versus expected noise norm (horizontal), as the threshold is reduced. All error norms are obtained from the surviving coefficients of the original image, and are computed only on the annular domain. A curve in the lower part of the plane implies a better performance. a) Domain, b) Image, c) Haar wavelets d) Daubechies-3 wavelets e) Biorthogonal 3.3 wavelets f) Domain-adapted wavelets. The vertical axis in figures c-g is the relative L_2 approximation error, which is $\frac{\|f - f_{\text{approx}}\|}{\|f\|}$. Note that increasing the level of the transform has a stronger effect with the domain-adaptive wavelets.

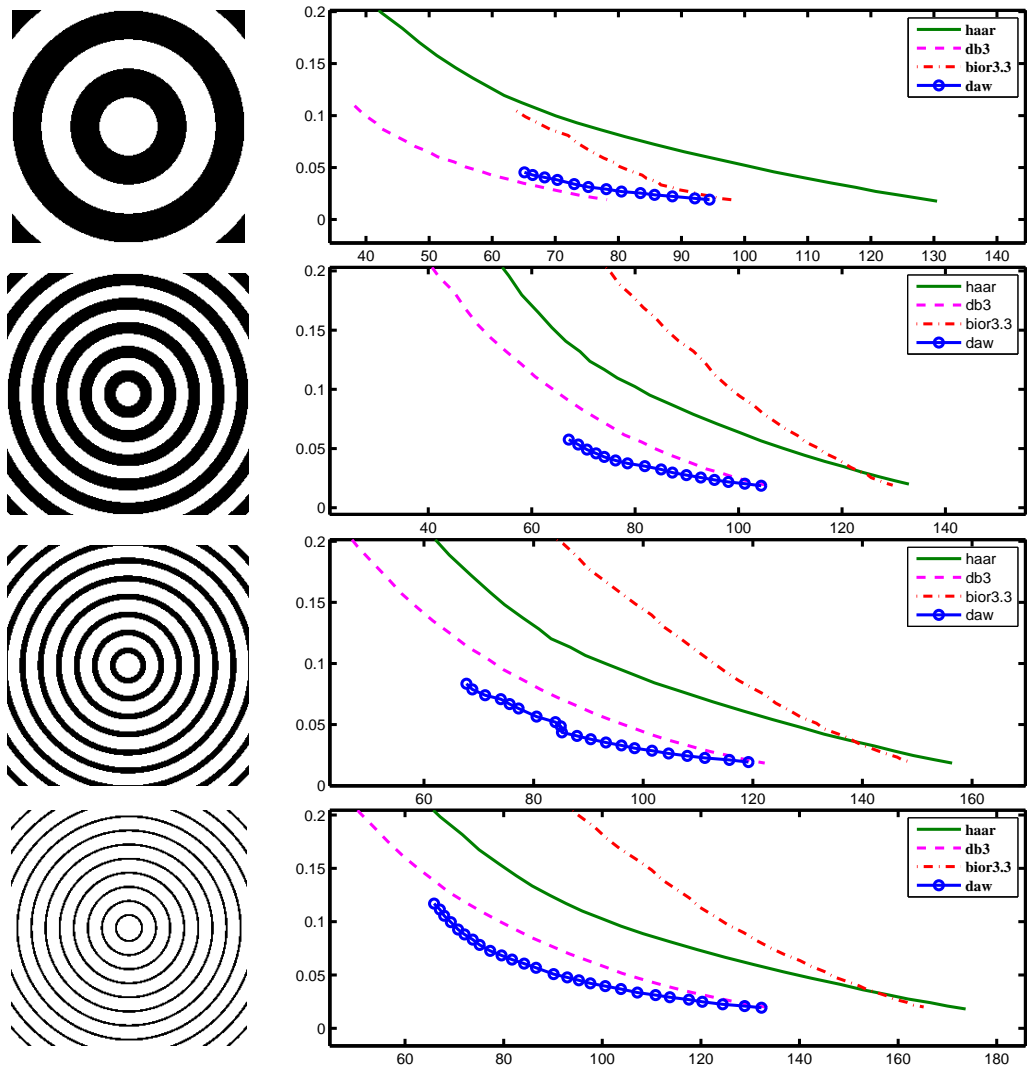


Figure 3.7: Comparison of the denoising potential of different wavelets on four different two-dimensional domains. Each of the wavelet transforms is computed at Level-3. Figures show the relative approximation error (horizontal) versus expected noise norm (vertical), as the threshold is reduced. As the domain gets thinner, or the gaps between different circles gets smaller, the domain-adaptive wavelets outperform the conventional wavelets.

this experiment, we work with a domain consisting of concentric circles and generate smooth signals on them, similar to the examples in the previous subsection. We add i.i.d. Gaussian noise, and perform a wavelet denoising. The signal-to-noise-ratio (SNR) versus number of realizations plot is given in Figure 3.8. For chosen domain, with the given noise, the tensor product Daubechies-3 wavelet transform initially performs better denoising than the domain adapted wavelets. However, as we average over multiple realizations, the domain-adapted wavelets result in better SNR values.

3.3.3 The effect of the second prediction

The second prediction step of the transform is designed to make the detail coefficients smaller, or the wavelets and scaling functions smoother, as explained in Subsection 3.2.4. The wavelet transform before this step is the *unbalanced Haar wavelet* transform, which is orthogonal. After the second prediction, the wavelets are *average interpolating wavelets*, and they are smooth but not orthogonal.

We tested the effect of this step in noise reduction with an experiment similar to the one in the previous subsection. The SNR versus number of wavelet realizations is given in Figure 3.9. This result shows an improvement of nearly 2.5 dB as a result of the second prediction step. A fixed threshold is empirically determined, and used in all wavelet realizations for the same type. Since the Haar wavelets are orthogonal, the thresholded approximation gives always the best approximation that can be achieved with the same number of surviving coefficients. However this is not necessarily the case for the average interpolating wavelets, because they are not orthogonal. This explains the sudden drops in the SNR curve corresponding to some outlying wavelet realizations.

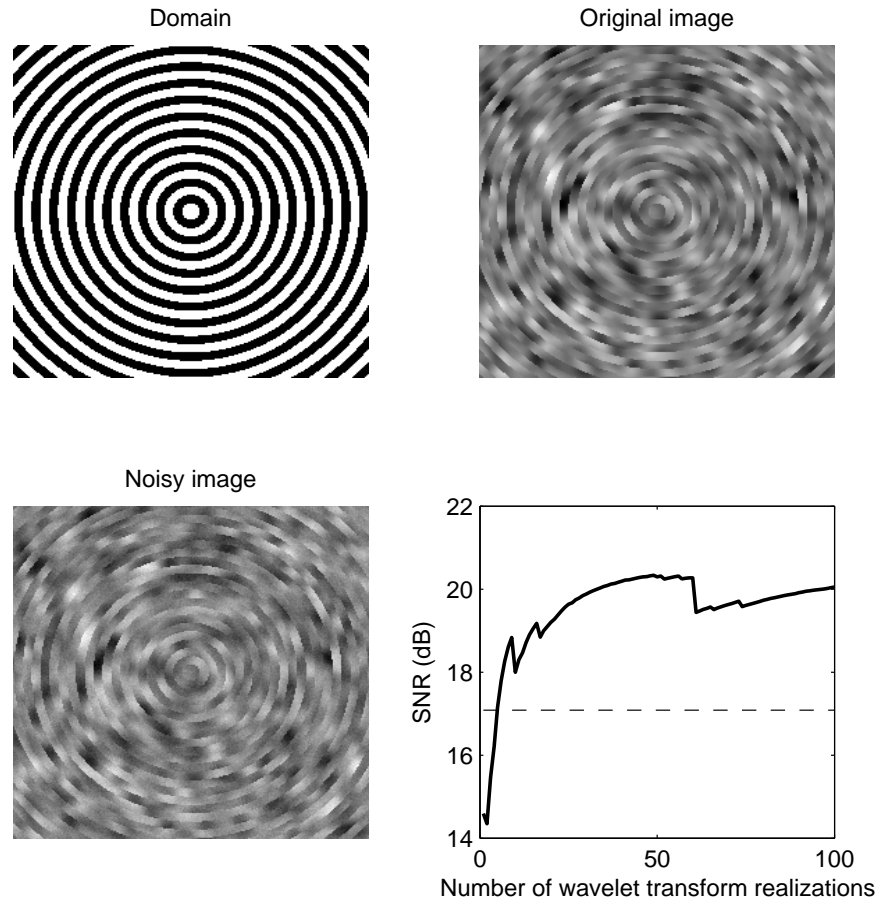


Figure 3.8: Denoising with multiple wavelet realizations. The performance improves as we average the results over multiple realizations. The dashed line in the plot represent the performance of the Daubechies-3 wavelet, which performed the best among the standard wavelets we tested in the previous subsection. SNR values computed only over the domain.

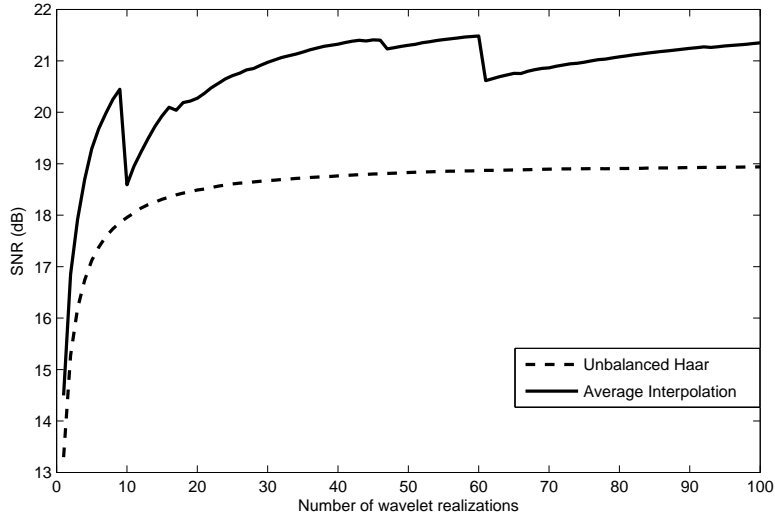


Figure 3.9: Comparison of unbalanced Haar wavelets and average interpolating wavelets in a noise reduction experiment. Performance improves as the results obtained by newer wavelet realizations are included in the cumulative average. The average interpolating wavelets give better performance than the unbalanced Haar wavelets, but occasional drops in performance are observed when an outlying wavelet realization is encountered.

3.3.4 Translation-invariant processing

In a reliable signal processing algorithm, one would intuitively expect a translation-invariance property. If we translate the signal without distorting, process it and then translate it back, we would expect the result to be the same as if it were processed without being translated. The simplest version of a multiresolution-based wavelet algorithm does not have this invariance, which is why Donoho and Coifman propose the wavelet spin cycle algorithm in [4], which achieves translation invariant denoising in one dimension. Translational invariance in higher dimensions can be achieved in the same way. In dimensions two and higher, one can similarly desire rotational invariance, which can be achieved by introducing redundancy in angular resolution as well, e.g., via steerable filters [13] or dual-tree wavelet transforms [27].

We tested the invariance property under translations and rotations of our randomized wavelet transform algorithm, when the results are averaged over multiple

realizations, as follows. We took an image on a 64×64 square domain, and we considered a 5-level randomized wavelet decomposition. We chose a processing task of projecting an image onto the spaces $V_0, W_0, W_1, \dots, W_5$, which are described in Subsection 2.2.1 and Subsection 2.2.4. This corresponds to transforming the image, and reconstructing from only a single level of coefficients. We took averages of the results according to 100 different wavelet realizations, and found that the realization-averaged transform is (almost) invariant under translations and rotations. This is illustrated in The result is shown in Figure 3.10, for rotations (which is the harder of the two tasks); and it is seen that the process commutes with a rotation of 45 degrees, as desired.

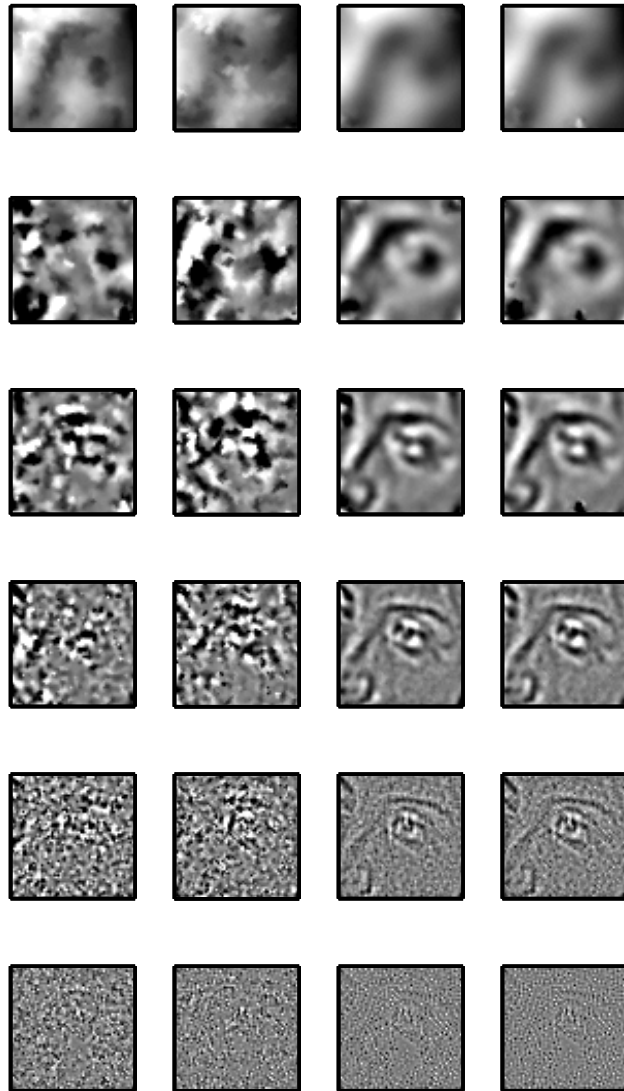


Figure 3.10: An image transformed and reconstructed back from a single level of wavelet coefficients, i.e, projected on to the corresponding subspaces for a 5-level wavelet decomposition. The six rows correspond to the projection onto the spaces $V_0, W_0, W_1, \dots, W_5$, respectively. In the first and third columns, images are processed without being translated whereas in the second and fourth columns they are first rotated 45 degrees, and then processed, and rotated back. The left two columns show the result of a single realization, and the right two columns show the result of computing the average of 100 realizations. We see that the processing becomes much more powerful, and almost rotationally invariant when averaged over multiple realizations.

Chapter 4

Application to wavelet-based statistical analysis of fMRI data

One of the most widely used and recognized methods for fMRI analysis is Statistical parametric mapping (SPM) [12]. In SPM, a key step is spatial prefiltering with a Gaussian window, as a means of reducing noise. However, Gaussian filtering has the drawback of destroying the fine spatial details. A wavelet-based alternative of SPM, which is called WSPM, is due to Van De Ville et al. [33, 32, 31, 35]. It replaces the Gaussian filtering step with wavelet filtering, and it involves thresholding in the wavelet domain as a denoising step, followed by a thresholding in the spatial domain. The threshold parameters in WSPM are selected so as to control the false positive rate, while minimizing the reconstruction error.

Standard wavelets used in this framework are defined on rectangular domains, typically a square (in two dimensions) or a cube (in three dimensions). On the other hand, the natural domain of the neural activity is the brain cortex, which is an intricately convoluted three dimensional domain.

This chapter essentially consist of the application of the domain-adapted wavelets of the previous chapter to the WSPM framework. The wavelets are constructed so

as to have the brain cortex as their natural domain.

4.1 Statistical testing of fMRI data

4.1.1 One sample t-test

One of the most basic tasks in fMRI data analysis is to determine whether there is any brain response to a given stimulus or a given train of stimuli, and if there is, to determine its location. In a simplified paradigm, we may assume the measurement we get from a given voxel is a random variable of the form

$$v_n = \mu_n + e_n,$$

where e_n is a noise random variable with zero mean and unknown variance, and μ_n is the true activation in the voxel n , i.e. $\mu_n = 0$ when there is no activation and $\mu_n > 0$ when there is activation. For each voxel $n \in V \subset \mathbb{Z}^3$ in the region of interest, one may consider using a standard one-sample t-test to decide whether that voxel is active or not. Given a series of measurements $v_n(1), v_n(2), \dots, v_n(N)$ from voxel n , the goal is to decide whether μ_n is zero or strictly positive, i.e., to decide between the hypotheses

$$\mathbf{H}_0: \mu_n = 0$$

$$\mathbf{H}_1: \mu_n > 0.$$

To decide which of the hypotheses is true, one passes to another random variable, which is called as a t -statistic, and distributed with Student's t -distribution of N degrees of freedom. In order to obtain a t -variable, the population mean $\hat{\mu}_n$ and an

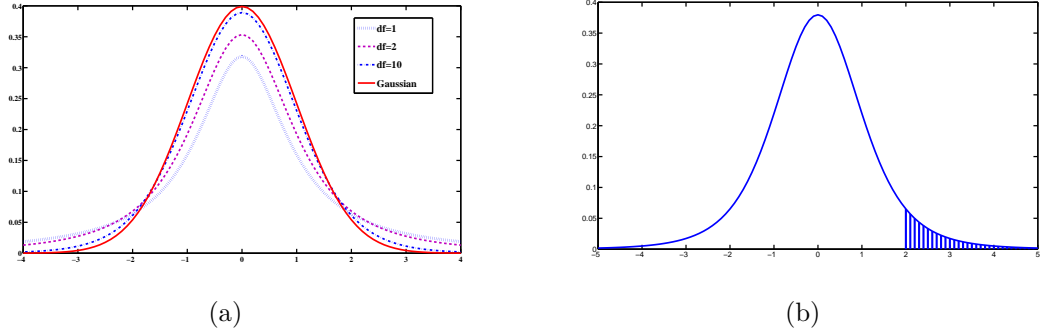


Figure 4.1: (a) The density functions of Student's t -distribution with $n = 1, 2$ and 10 degrees of freedom, respectively, and the standard Gaussian distribution, which can also be viewed as a t -distribution with infinitely many degrees of freedom. (b) The 5 % region to reject the null hypothesis, for number of degrees of freedom $n = 5$.

unbiased estimate of the variance s_N^2 are computed from the samples as follows:

$$\hat{\mu}_n = \frac{1}{N} \sum_{k=1}^N v_n(k) \quad s_N^2 = \frac{1}{N-1} \sum_{k=1}^N (v_n(k) - \hat{\mu}_n)^2.$$

Then the ratio $t = \frac{\hat{\mu}}{\sqrt{s_N^2/N}}$ has a Student's t -distribution with N degrees of freedom. A preselected extremely unlikely region is used to reject the null hypothesis. In our case one would use the so called one-sided tail as in Figure 4.1, since the mean is assumed to be nonnegative. This region is chosen to have a small probability of occurrence, a typical value for the significance level α being 0.05 , which is also the false-positive rate, i.e., the chance that null hypothesis will be rejected incorrectly.

4.1.2 The general linear model

Although the model that is described in subsection 4.1.1 may be helpful for illustration purposes, it was not realistic, since data from a real voxel is never silent, due to the highly active nature of the human brain. That is why one can observe only a slight increase in the activity of a voxel in response to stimuli. In Figure 4.2 data from a real experiment by Haxby et al. [17] are plotted, in which signals from 577

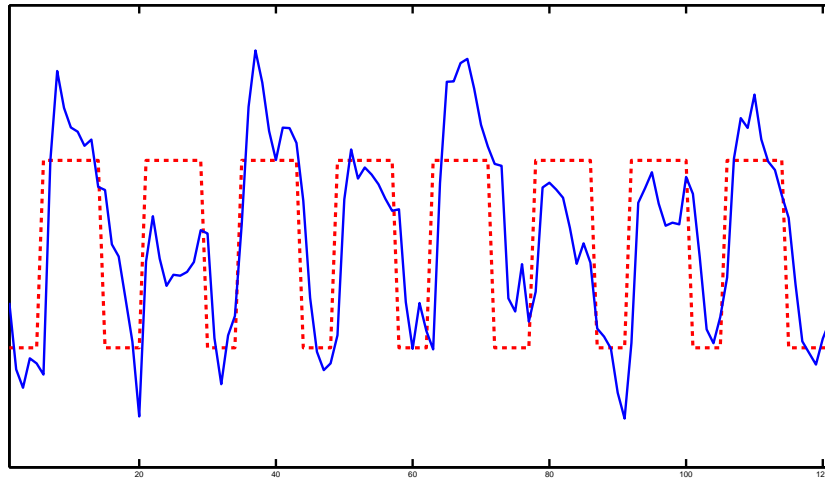


Figure 4.2: The stimuli and fMRI response. Data from a visual stimuli experiment by Haxby et.al [17]. The dashed line represents the on/off timing of the visual stimuli, and the solid line is the fMRI response, averaged over 577 voxels from the VT cortex. The linear trend in the stimuli is also removed. The correlation between the stimuli and the response is clearly visible in the figure, although in the signal from a single voxel the correlation would be virtually impossible to see.

voxels are averaged for the correlation to be visually detectable. In this setting, the question becomes whether there is a positive correlation between the stimulus and the response from the voxel and the decision is less straightforward. Fortunately, there exist straightforward extensions of the one sample t -test into this setting.

Let $v_n(1), v_n(2), \dots, v_n(N)$ be the time samples obtained from the voxel n at times $t = 1, 2, \dots, N$, and let us represent these samples as an $N \times 1$ column vector,

$$\mathbf{v}_n = \begin{bmatrix} v_n(1) \\ v_n(2) \\ \vdots \\ v_n(N) \end{bmatrix}.$$

In a typical setting the subject is presented with stimuli during certain intervals for a certain amount of time. Let the voxel location n be fixed, and consider the time

series $v(t)$, denoted as simply v . We set up a model, known as the *general linear model*, or GLM, to analyze the time course of this voxel:

$$\mathbf{v}_n = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}_n$$

where \mathbf{X} is a matrix each column of which is the expected time course if the subject had responded only to a particular type of stimulus, with different stimulus type time courses given by the different columns of \mathbf{X} . After the linear regression, each component of the estimate of $\boldsymbol{\beta}$ will correspond to the level of response to the corresponding category.

A simple example for a single, on-off type stimulus with only eight time points can be given as follows: Let us present the subject with stimuli for two time points, and then give a rest for the next two time points, and repeat this procedure one more time. Then a simple choice for our model would be

$$\mathbf{v}_n = \begin{bmatrix} 1 & t_0 & 1 \\ 1 & t_1 & 1 \\ 1 & t_2 & -1 \\ 1 & t_3 & -1 \\ 1 & t_4 & 1 \\ 1 & t_5 & 1 \\ 1 & t_6 & -1 \\ 1 & t_7 & -1 \end{bmatrix} \begin{bmatrix} \beta_{0,n} \\ \beta_{1,n} \\ \beta_{2,n} \end{bmatrix} + \mathbf{e}. \quad (4.1)$$

The constant vector in the first column accounts for the mean. The second column is the time vector, and its coefficient $\beta_{1,n}$ accounts for the linear trend. The last column corresponds to the actual stimulus, having value 1 when the stimulus is on, and -1

when it is off. The only parameter of interest here would be $\beta_{2,n}$, corresponding to this last column, which would be interpreted as the magnitude of response from the current voxel to the stimulus train.

Then this analysis is repeated for each voxel, and the resulting map $\{\beta_{2,n} : n \in R\}$ is then the activation parameter map for the given stimulus, where R is the region of interest. Significance of the response can be determined with the t -test, and the voxels passing the test are marked to be active. This is illustrated with sample time courses from a real experiment, in Figure 4.4.

Remark 4.1.1. *A more advanced approach would be to convolve the box functions in the columns of interest in \mathbf{X} with what is called as the hemodynamic response function, which makes it resemble the actual expected time course more, taking account the dynamics of the blood-oxygenation in the brain. See Figure 4.3 for the plot of the convolved regressors and the hemodynamic response function, and [25] for more information.*

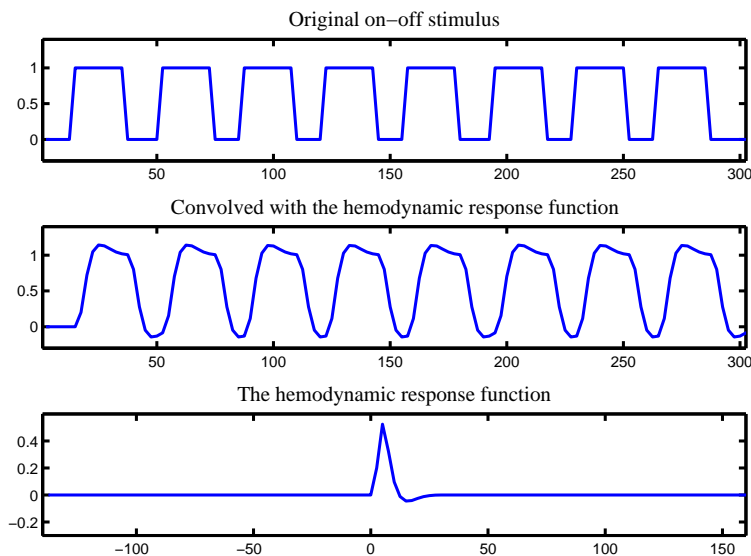


Figure 4.3: Original on-off stimulus (top), convolved with the homodynamic response function (middle), and the hemodynamic response function (bottom).

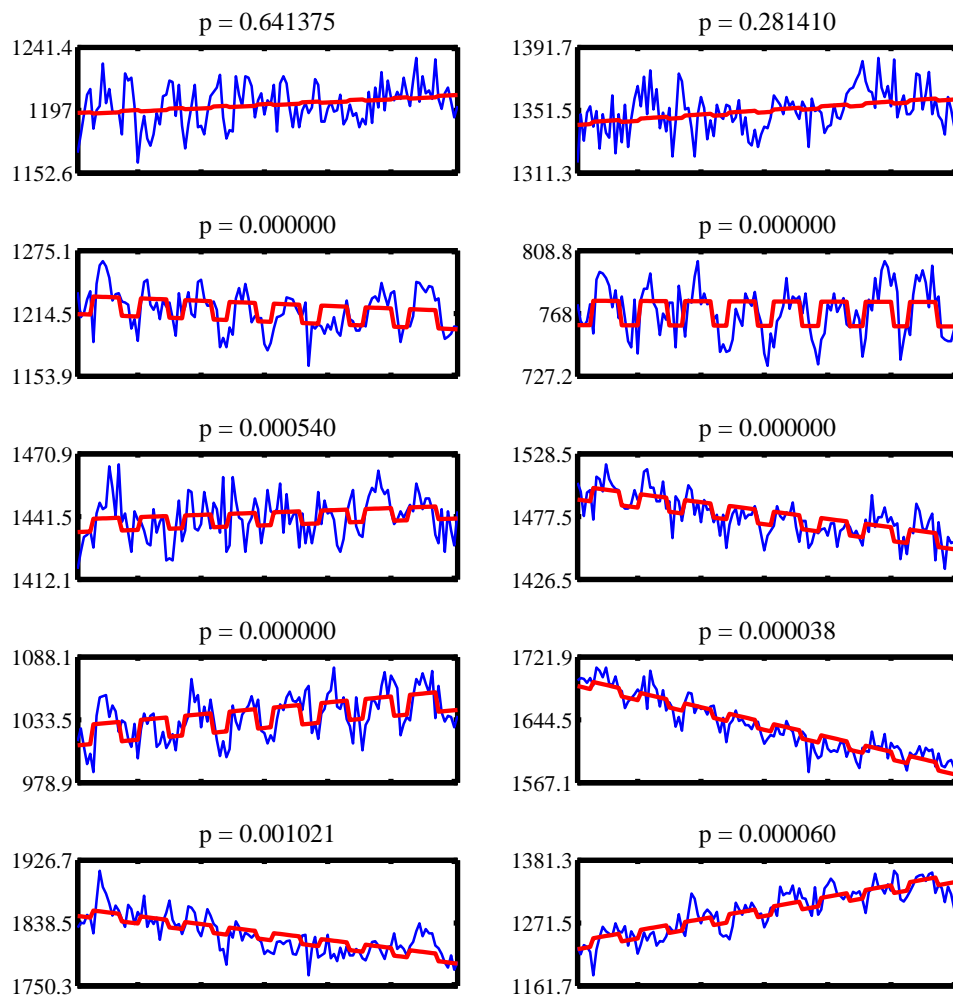


Figure 4.4: Plots of time series from various voxels, from the experiment by Haxby et al. [17]. Each voxel is fitted to the model, as a linear combination of a straight line and the box train function corresponding to the on-off stimuli. A t variable is derived and corresponding p values are computed. The smaller values of p imply a stronger rejection of the null hypothesis.

Statistical testing

The general linear model (GLM) for the time course of the voxel n is

$$\mathbf{v}_n = \mathbf{X}\boldsymbol{\beta} + \mathbf{e},$$

where $\boldsymbol{\beta}$ is an $N \times 1$ vector of unknown parameters, and \mathbf{e} is an $N \times 1$ Gaussian random vector with zero mean and unknown variance. The matrix \mathbf{X} is $N \times L$, which is known beforehand and is called the *design matrix*.

Usually not all components of $\boldsymbol{\beta}$ are of interest. One usually needs to know whether the quantity $\mathbf{c}^T\boldsymbol{\beta}$ is zero or strictly positive, and \mathbf{c} is called the *contrast vector*. For the example of (4.1), the contrast vector would be taken as $\mathbf{c}^T = [0 \ 0 \ 1]^T$, indicating that one is interested in whether the last column of the design matrix has positive correlation with the data, after removing the effects of the first two columns. In another case when one is interested in whether column i of the design matrix has more correlation with the data than column j , the contrast vector \mathbf{c} would be a vector of all zeros except for the i th and j th entries, which would be 1 and -1 , respectively.

We observe a realization of the random vector \mathbf{v}_n , and would like to decide between the two hypothesis about $\boldsymbol{\beta}$:

$$\mathbf{H}_0: \mathbf{c}^T\boldsymbol{\beta} = 0$$

$$\mathbf{H}_1: \mathbf{c}^T\boldsymbol{\beta} > 0.$$

In order to do a t -test, we first need to obtain the t -variable. One starts with computing an unbiased estimate for $\boldsymbol{\beta}$ by an ordinary least squares formula

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}\mathbf{y},$$

implicitly assuming the columns of \mathbf{X} are linearly independent. Then an estimate for

the error would be

$$\hat{\mathbf{e}} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}.$$

Now if one defines

$$g_n = \mathbf{c}^T \hat{\boldsymbol{\beta}}, \quad (4.2)$$

$$s_n^2 = \hat{\mathbf{e}}_n^T \hat{\mathbf{e}}_n \mathbf{c}^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{c}. \quad (4.3)$$

then the scalar random variable g would have a Gaussian distribution with mean $\mathbf{c}^T \boldsymbol{\beta}$, and s^2 will follow a Chi-square distributions with $J = N - \text{rank}(X)$ degrees of freedom, and they are independent [31]. Following this, one can obtain a t variable by

$$t_n = \frac{g_n}{\sqrt{s_n^2/J}}, \quad \text{with } J = N - \text{rank}(\mathbf{X}),$$

and one can then proceed similarly to the t -test of Subsection 4.1.1 for a statistical test of the significance of the activation.

4.1.3 Multiple comparison problem

When multiple statistical tests are to be performed based on the same data set, the problem of *multiple comparison* needs to be dealt with. In the case of fMRI, a typical dataset has of the order of 10^4 voxels. A significance level of $\alpha = 0.05$ would result in 10^2 to 10^3 false detections; this number may be as large as the total number of voxels in the entire region of activation! In general, for a dataset of K voxels, one expects to get αK false positives. One solution offered to remedy this problem is to use the *Bonferroni correction*, which is a conservative method to reduce the expected number of total false positives by reducing α , and considering voxels in (sub)collections rather than individually. For a total number of K voxels, the Bonferroni correction asks to reduce the significance level from α to α/K , which in

turn reduces the expected number of total false positives from αK to α . By the *union bound*, getting a single falsely detected voxel within the set of K voxels would have a rate smaller than α . For a set with, e.g., 10^2 voxels and an initial α selected to be 0.05, Bonferroni correction would ask to reduce α from 5×10^{-2} to 5×10^{-4} . Although it can be improved by restricting the region of interest to a smaller set of voxels, this correction has the obvious drawback of extremely reduced sensitivity for fMRI datasets, almost to the level of detecting no activation [33]: although the Bonferroni argument reduces the number of false positives, it *increases* the number of false negatives. Such small choices of sensitivity rates make the tests useless, because fMRI data sets are very noisy and there are not enough samples to get detections at such a conservative specificity.

4.2 Spatial filtering for improving statistical power

4.2.1 Statistical parametric mapping and wavelets

The problem with the Bonferroni correction is that it does not make use of the spatial correlation between voxels [33]. There are methods that remedy this situation by essentially performing a transformation, which practically maps the large number of noisy and highly correlated voxels into a small number of uncorrelated and less noisy transform domain coefficients. The most widely used approach in this category is the Statistical parametric mapping (SPM) package [14], in which the essential step is prefiltering by a Gaussian window, which corresponds to going to the Fourier domain, and throwing out the high frequency components and reconstructing. Note that a spatial convolution with a Gaussian window is equivalent to multiplying the Fourier transform of the data with the Fourier transform of the Gaussian window; since this is another Gaussian, this multiplication is a form of weighted thresholding. As depicted in Figure 4.5, Gaussian and wavelet filtering can be viewed as two different forms

of the same idea. In Gaussian filtering, the coefficients to be discarded are the high frequency coefficients that are predetermined, which makes it linear, while in wavelet filtering, coefficients are to be discarded are determined by thresholding, therefore it is a form of nonlinear filtering.

4.2.2 Gaussian versus wavelet filtering

While Gaussian filtering has the advantages of being a well established linear noise reduction method, its main disadvantage is that it destroys all fine spatial details, which might be important in fMRI images. In Figure 4.6, this phenomenon is illustrated with a single dimensional example. That is the main motivation of exploring wavelet based methods, which are the subject of the rest of this chapter.

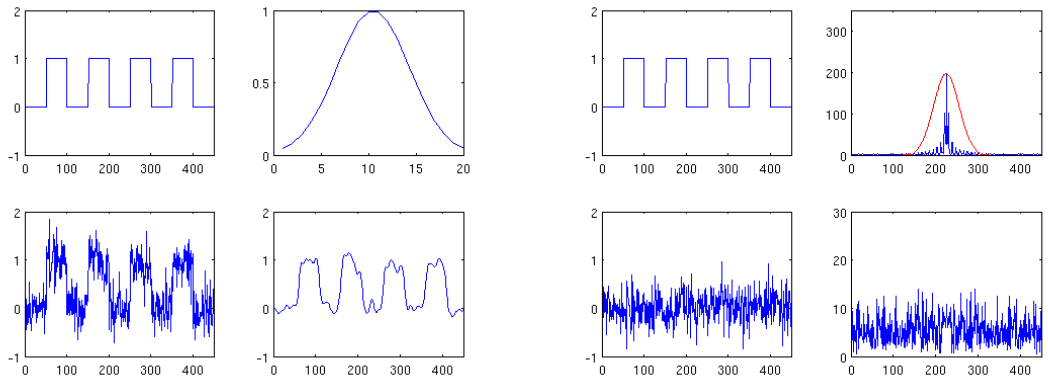
4.2.3 Classical wavelet-based analysis

The classical wavelet-based methods [24], mainly transfer each spatial volume to the wavelet domain, perform a t -test on each wavelet coefficient, and reconstruct the volumes back by only using the wavelets whose coefficients survive the t -test.

Let us denote an fMRI data set with $v_{\mathbf{n}}(t)$, where $\mathbf{n} \in \mathbb{Z}^3$ is the spatial index, and $t \in \mathbb{Z}$ is the temporal index. With a simplified notation, the wavelet transform corresponds to representing the data under the form

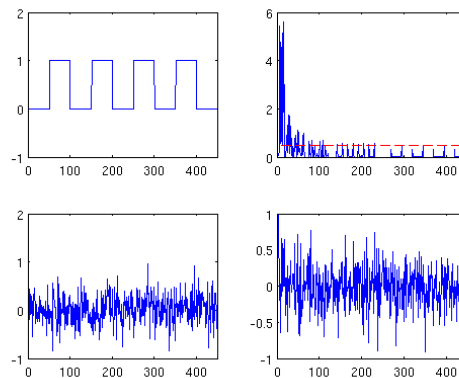
$$v_{\mathbf{n}}(t) = \sum_{\mathbf{k}} w_{\mathbf{k}}(t) \psi_{\mathbf{k}}(\mathbf{n}),$$

where $\{\psi_{\mathbf{k}}\}$ is the wavelet basis. The basis functions are translated and dilated versions of some prototype in the case of standard wavelets, but they take more arbitrary forms in the adapted case.



(a)

(b)



(c)

Figure 4.5: Depiction of noise reduction by Gaussian and wavelet filtering. Spatially correlated functions have their transform domain concentrated to a few coefficients, while noise processes have their transforms spread out to all coefficients uniformly. (a) First Row: A square wave function, and a Gaussian which will be used in filtering, Second Row: Noisy (left) and filtered (right) versions of the square wave function. (b) First row: The square wave function and its Fourier transform, overlaid by the Fourier transform of the Gaussian window, Second Row: White noise and its Fourier transform. (c) First Row: The square wave function and its Wavelet transform, Second Row: White noise and its wavelet transform.

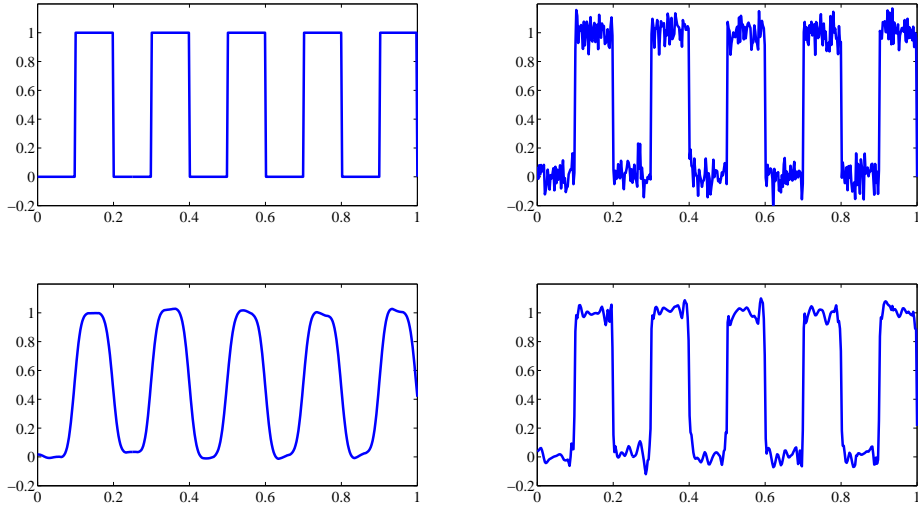


Figure 4.6: Wavelet denoising. A square wave signal is contaminated with noise. In the second row, it is denoised by Gaussian filtering and by wavelets, respectively. Although Gaussian filtering achieves a visually detectable noise reduction, it also destroys the fine spatial details, which corresponds, in this example, to the steepness of the edges; this steepness is much better preserved by wavelet denoising.

Now let \mathbf{w}_k be the vector of wavelet coefficients corresponding to ψ_k ; i.e.,

$$\mathbf{w}_k = [w_k(1) \cdots w_k(N_t)]^T,$$

where N_t is the total number of time samples. Due to linearity of the wavelet transform, we can write the same general linear model as in the spatial domain

$$\mathbf{w}_k = \mathbf{X}\mathbf{y}_k + \mathbf{e}_k,$$

where \mathbf{X} is the $N_t \times L$ design matrix, \mathbf{y}_k is the $N \times 1$ vector of unknown parameters, and \mathbf{e}_k is the residual error. Assuming the noise to be independent and identically distributed Gaussian, the unbiased estimate of \mathbf{y}_k is given by

$$\hat{\mathbf{y}}_k = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{w}_k.$$

Corresponding to each index \mathbf{k} of the wavelet basis, we obtain two scalar values:

$$g_{\mathbf{k}} = \mathbf{c}^T \hat{\mathbf{y}}_{\mathbf{k}},$$

$$s_{\mathbf{k}}^2 = \hat{\mathbf{e}}_{\mathbf{k}}^T \hat{\mathbf{e}}_{\mathbf{k}} \mathbf{c}^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{c},$$

where $\mathbf{g}_{\mathbf{k}}$ and $\mathbf{s}_{\mathbf{k}}$ follow a Gaussian and Chi-squared distribution, respectively, and \mathbf{c} is the contrast vector. From these one can obtain a t value for each wavelet coefficient \mathbf{k} :

$$t_{\mathbf{k}} = \frac{g_{\mathbf{k}}}{\sqrt{s_{\mathbf{k}}^2/J}}, \text{ with } J = N_t - \text{rank}(\mathbf{X}),$$

which can be tested against a threshold τ_w , which is chosen in accordance with the desired significance level. After testing, the detected coefficients are reconstructed as

$$\mathbf{r}_{\mathbf{n}} = \sum_{\mathbf{k}} T_{\tau_w}(t_{\mathbf{k}}) g_{\mathbf{k}} \psi_{\mathbf{k}}(\mathbf{n}), \quad (4.4)$$

where T is the thresholding function corresponding to the two sided t test; i.e., $T(t_{\mathbf{k}}) = 1$ if $|t_{\mathbf{k}}| \geq \tau_w$, and zero otherwise. The volume $\mathbf{r}_{\mathbf{n}}$ contains many nonzero voxels, each of which is a function of many voxels from the original data. One must rely on heuristic thresholds on $\mathbf{r}_{\mathbf{n}}$ to obtain acceptable detection maps. Moreover, $\mathbf{r}_{\mathbf{n}}$ does not have a direct statistical interpretation. These disadvantages are overcome with the WSPM, as explained in the next section.

4.2.4 WSPM: Joint spatio-wavelet statistical analysis

Wavelet based Statistical Parametric Mapping (WSPM) is a method proposed by Van De Ville et al. [33, 32, 31, 35], which is a modification of SPM where the denoising step is performed by thresholding in the spatial wavelet domain. This makes the typical advantage of wavelets, which is about reducing noise while keeping high frequency detail information, apparent in the results. The underlying theorem guar-

antees control over the false-positive rate by a bound of the null-hypothesis rejection probability. Moreover, empirical results show similar sensitivity to that obtained by SPM with improved spatial detail. The resulting map of active voxels can be seen to align with the cortex, as a demonstration of preserving the detail information.

The main idea in the joint spatio-wavelet statistical analysis is to perform two consecutive thresholding operations: first in the wavelet domain and then in the spatial domain. There are two corresponding threshold parameters, τ_w and τ_s , to be determined [32, 31]. A spatial map is obtained after the first thresholding as in (4.4). Then $r_{\mathbf{n}}$ is weighted by $1/\sum_{\mathbf{k}}\sigma_{\mathbf{k}}|\psi(\mathbf{n})|$ and thresholded by τ_s . That is, the set of voxels that are declared to be active would be

$$\left\{ \mathbf{n}: \frac{|\sum_{\mathbf{k}} T_{\tau_w}(t_{\mathbf{k}})g_{\mathbf{k}}\psi_{\mathbf{k}}(\mathbf{n})|}{\sum_{\mathbf{k}}\sigma_{\mathbf{k}}|\psi(\mathbf{n})|} \geq \tau_s \right\}.$$

Given the desired significance level α , an optimal choice for τ_s and τ_w , which minimizes the approximation error between the reconstruction from the fitted parameters and two times thresholded reconstruction, can be computed to be

$$\tau_w = \sqrt{-W_{-1}\left(-\frac{\alpha^2\pi}{2}\right)}, \quad \tau_s = 1/\tau_w,$$

where W_{-1} is the -1 -branch of the Lambert- W function [32].

4.2.5 Anatomically adapted wavelets

In the methods mentioned above methods, the wavelet transforms are either performed slice by slice as two dimensional wavelet transforms, or are applied to the whole volume as a three dimensional wavelet transform. In either case, the domain of the signals are assumed to be the rectangular. However, in reality the activity takes place in a subset corresponding to the brain cortex, which is a highly convoluted

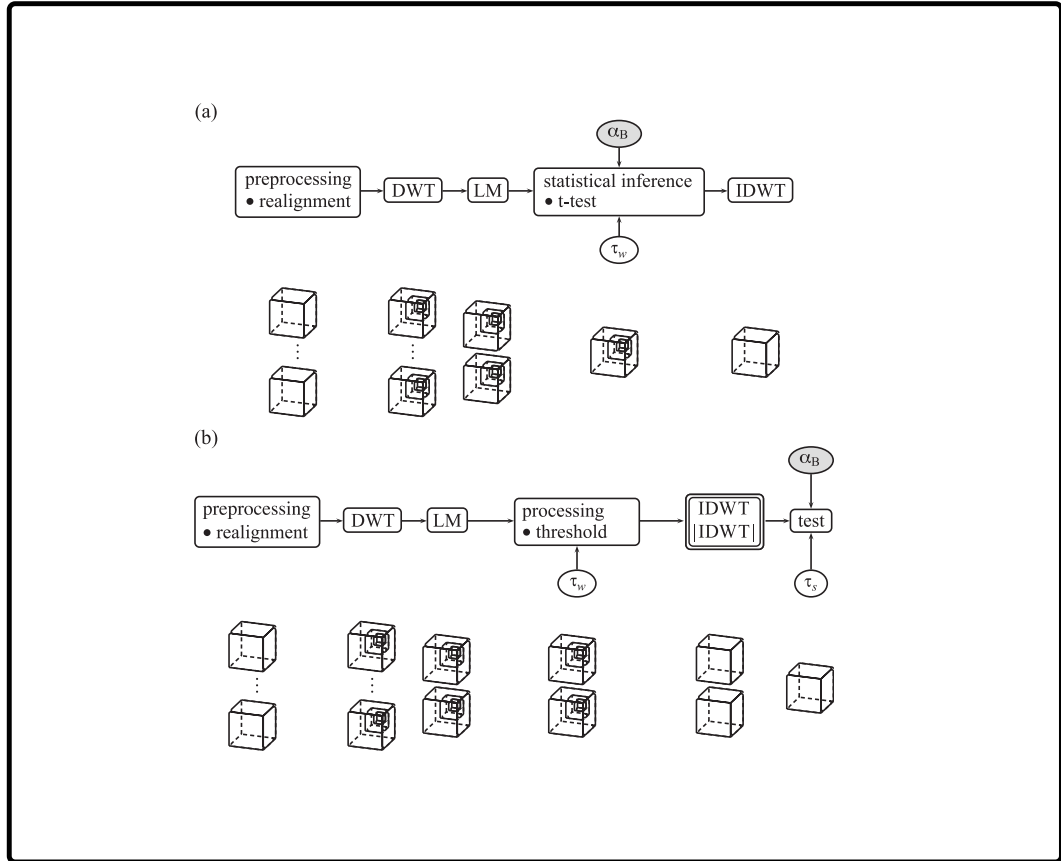


Figure 4.7: Diagram illustrating the wavelet-based approach proposed in [31]. a) Classical wavelet based method, and b) Integrated wavelet based method (WSPM). In both methods, the wavelet transform of each volume is computed (DWT), the variables $g_{\mathbf{k}}$ and $s_{\mathbf{k}}^2$ are computed for each wavelet coefficient (LM), a t variable is obtained for each wavelet coefficient and the t -test is performed, and the inverse wavelet transform is computed after throwing out the coefficients failing the test (IDWT). In the classical method, one must rely on heuristic thresholds for the reconstructed volume. In the integrated framework, the thresholds τ_w and τ_s are determined together, as a function of the input sensitivity parameter α_B . In the integrated method, one also needs to compute a reconstruction with the absolute value of the wavelets, before the spatial thresholding. Figure replicated from [31].

three dimensional structure. This motivates us to obtain domain-adaptive wavelets for fMRI data, using the construction in Chapter 3. We input the segmented brain cortex to the algorithm, and use the resulting wavelets in the statistical framework for analyzing fMRI data. We refer to these type of wavelets as *anatomically adapted wavelets*.

4.3 Experimental results

4.3.1 Simulated data

We generated smooth functions over a domain consisting of concentric circles, to represent the fine layered structure of the gray-matter cortical layer with different widths. We contaminated the data with white Gaussian noise, whose magnitude is decreasing from left to right, as shown in Figure 4.8. The adapted wavelets showed an improvement in the sensitivity, while keeping the specificity within the theoretical limits. As the level of the wavelet decomposition increases, the sensitivity keeps increasing when adapted wavelets are used, while it remains unchanged for standard wavelets; this is illustrated by the comparison in Figure 4.9 .

4.3.2 Real data

We tested the proposed method with data obtained from a visual stimulation experiment, with 16 slices of 128×128 voxels, the size of which is $1.8 \text{ mm} \times 1.8 \text{ mm} \times 5 \text{ mm}$. We performed segmentation with the SPM software package, and generated the adaptive wavelets using the domain corresponding to the gray-matter layer. The thresholded binary domain is shown in Figure 4.10. Using the standard orthogonal wavelets, the analysis resulted in 1032 detected voxels, with the adapted wavelets it resulted in 1214 active voxels. In both cases the sensitivity parameter α is taken to be 0.001. This suggests an improved sensitivity, with a detection of larger number of

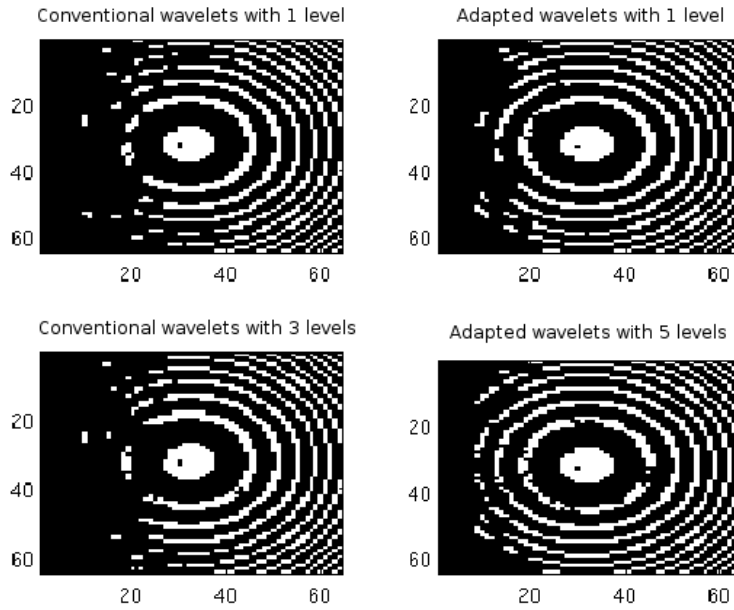


Figure 4.8: The detected map of active voxels in an example with simulated data. The domain consists of concentric rings. Gaussian white noise is superimposed on the data, with σ kept constant within each “column”, but as the “row” number r increases, σ decreases, with $\sigma r = \text{constant}$. For $r = 2$, σ is such that the noise overpowers the signal completely; for $r = 64$, σ has decreased to a level where the signal is detected without problems. The left column shows results with standard orthogonal wavelets, and the right column shows the result with the anatomically adapted wavelets, which detect the signal in some locations where standard wavelets don’t. In the second row the level of wavelet decomposition is increased, and the performance of adapted wavelets has increased correspondingly while the performance of standard wavelets remain unchanged.

voxels, as shown in Figure 4.11.

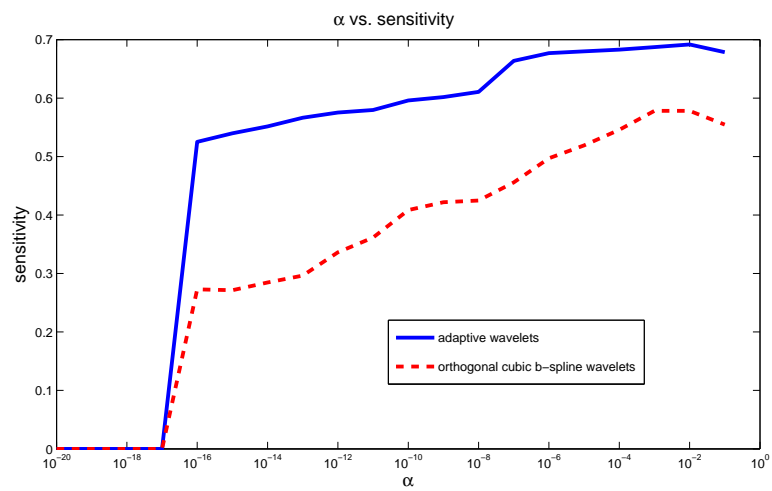


Figure 4.9: The ROC curves with standard (tensor-product orthogonal cubic B-spline wavelets) and anatomically adapted wavelets. The α -value represents the type-I error control rate that is input to the algorithm.

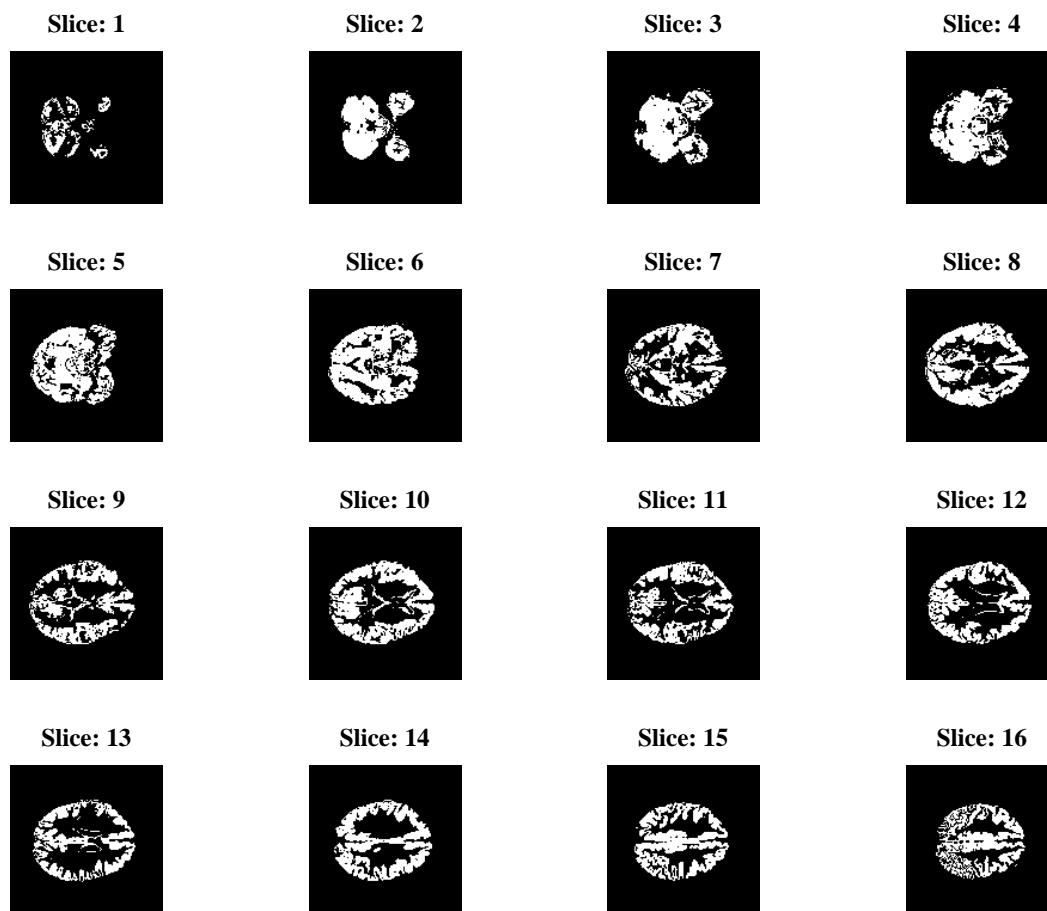


Figure 4.10: The binary mask used in the algorithm, which identifies the brain cortex. The volume corresponding to the brain cortex is displayed slice by slice. This map is obtained by thresholding the output of SPM's brain segmentation algorithm. This domain information is input to the wavelet construction algorithm, as a binary volume.

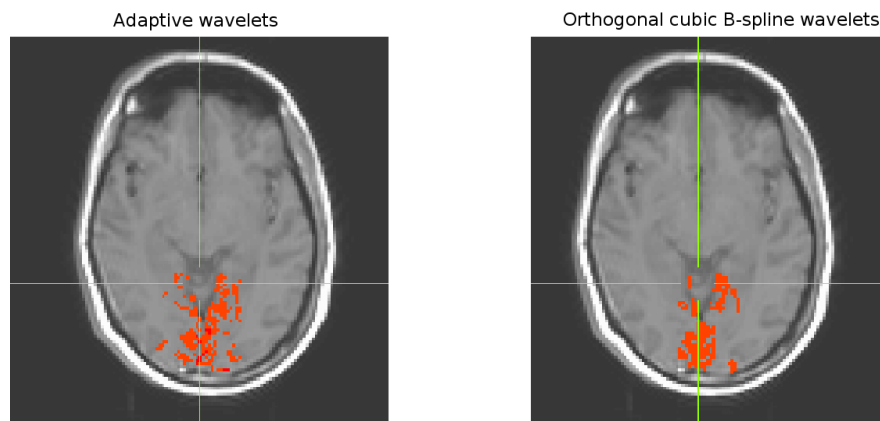


Figure 4.11: Slices from detected activation maps with real visual stimulation data. The left image is obtained with adaptive wavelets, and the right image is obtained with orthogonal cubic B-spline wavelets.

Chapter 5

Summary and future work

In this thesis, we constructed two and three dimensional wavelets for arbitrary domains, using the lifting scheme. When the original domain of the signals to be analyzed has a significant proportion of boundary pixels or voxels, our adapted wavelets demonstrate better sparsity properties and a superior performance in denoising, compared to standard wavelets that have rectangular domains.

In the construction, a nested grid structure is required to be defined on the domain. In defining the nested grid, we used a randomized algorithm. This randomization gives us the chance to have multiple sets of wavelet bases on the same domain, which allows one to process the data multiple times and then take the average of the results. This, in general, improves the results whenever it is possible to average the output of the wavelet application. Our test with two dimensional images showed that this algorithm is nearly translation and rotation invariant.

The new class of wavelets are then used in the brain imaging problem, after being adapted to the anatomy of the brain cortex. In the wavelet-based Statistical Parametric Mapping framework, we have observed an improved sensitivity, while retaining the same amount of control over type-I errors, compared to wavelet transforms having rectangular domains. In simulated data, contrary to what is observed with the

standard wavelet transform, use of the adapted wavelets shows clear improvement as the level of the wavelet decomposition increases.

As the high resolution fMRI scanners are becoming more widely available, spatial filtering tools like ours, which treat the brain cortex as an arbitrary three dimensional volume, may become an alternative to the well-established spatial filtering tools that are currently used in brain imaging analysis.

Future work

In utilizing multiple sets of wavelet bases, we processed the data separately with each individual non-redundant basis, and took the average of the results. However, the average may include results from some bad realizations of the basis, which may deteriorate the performance. As an alternative, we will explore considering of the union of the multiply generated bases as a single overcomplete basis, and use it together with sparsity-constrained risk minimization algorithms.

In another direction, we will test the performance of the anatomically adapted wavelets in multi-subject fMRI studies.

Bibliography

- [1] Ole Christensen. *Frames and Bases: An Introductory Course (Applied and Numerical Harmonic Analysis)*. Birkhauser, illustrated edition edition, July 2008.
- [2] A. Cohen, Ingrid Daubechies, and J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, 45(5):485–560, 1992.
- [3] Albert Cohen, Wolfgang Dahmen, and Ronald DeVore. Adaptive wavelet methods for elliptic operator equations: convergence rates. *Math. Comp.*, 70(233):27–75 (electronic), 2001.
- [4] R. R. Coifman and D. L. Donoho. Translation-invariant de-noising. Technical report, Department of Statistics, Stanford University, 1995.
- [5] Ronald R. Coifman and Mauro Maggioni. Diffusion wavelets. *Appl. Comput. Harmon. Anal.*, 21(1):53–94, 2006.
- [6] Wolfgang Dahmen. Wavelet and multiscale methods for operator equations. In *Acta numerica, 1997*, volume 6 of *Acta Numer.*, pages 55–228. Cambridge Univ. Press, Cambridge, 1997.
- [7] I. Daubechies. *Ten lectures on wavelets*, volume 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.

- [8] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.*, 4(3):245–267, 1998.
- [9] David L. Donoho. Unconditional bases are optimal bases for data compression and for statistical estimation. *Appl. Comput. Harmon. Anal.*, 1(1):100–115, 1993.
- [10] D.L. Donoho. Smooth wavelet decompositions with blocky coefficient kernels. In *Recent advances in wavelet analysis*, volume 3 of *Wavelet Anal. Appl.*, pages 259–308. Academic Press, Boston, MA, 1994.
- [11] Raanan Fattal. Edge-avoiding wavelets and their applications. *ACM Trans. Graph.*, 28(3):1–10, 2009.
- [12] R.S.J. Frackowiak, K.J. Friston, C. Frith, R. Dolan, C.J. Price, S. Zeki, J. Ashburner, and W.D. Penny. *Human Brain Function*. Academic Press, 2nd edition, 2003.
- [13] William T. Freeman and Edward H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:891–906, 1991.
- [14] K. J. Friston, A. P. Holmes, K. J. Worsley, J. P. Poline, C. D. Frith, and R. S. J. Frackowiak. Statistical parametric maps in functional imaging: A general linear approach. *Hum. Brain Mapp.*, 2(4):189–210, 1994.
- [15] M. Girardi and W. Sweldens. A new class of unbalanced Haar wavelets that form an unconditional basis for L_p on general measure spaces. *J. Fourier Anal. Appl.*, 3(4):457–474, 1997.
- [16] M. Griebel and D. Oeltz. A sparse grid space-time discretization scheme for parabolic problems. *Computing*, 81(1):1–34, 2007.

- [17] James V. Haxby, Ida M. Gobbini, Maura L. Furey, Alunit Ishai, Jennifer L. Schouten, and Pietro Pietrini. Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science*, 293(5539):2425–2430, September 2001.
- [18] M. Jansen and P. Oonincx. *Second generation wavelets and applications*. Springer, 2005.
- [19] Andrei Khodakovsky and Igor Guskov. Compression of normal meshes. In *Geometric modeling for scientific visualization*, Math. Vis., pages 189–206. Springer, Berlin, 2004.
- [20] John Michael Lounsbery. *Multiresolution analysis for surfaces of arbitrary topological type*. ProQuest LLC, Ann Arbor, MI, 1994. Thesis (Ph.D.)–University of Washington.
- [21] Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier/Academic Press, Amsterdam, third edition, 2009. The sparse way, With contributions from Gabriel Peyré.
- [22] K. Norman, S. Polyn, G. Detre, and J. Haxby. Beyond mind-reading: multi-voxel pattern analysis of fmri data. *Trends in Cognitive Sciences*, 10(9):424–430, September 2006.
- [23] S. Gorkem Ozkaya and Dimitri Van De Ville. Anatomically adapted wavelets for integrated statistical analysis of fmri data. In *ISBI*, pages 469–472. IEEE, 2011.
- [24] U.E. Ruttimann, M. Unser, R.R. Rawlings, D. Rio, N.F. Ramsey, V.S. Mat-tay, D.W. Hommer, J.A. Frank, and D.R. Weinberger. Statistical Analysis of Functional MRI Data in the Wavelet Domain. *IEEE Transactions on Medical Imaging*, 17(2):142–154, 1998.

- [25] G.E. Sarty. *Computing Brain Activity Maps from fMRI Time-Series Images*. Cambridge University Press, 2007.
- [26] Peter Schröder and Wim Sweldens. Spherical wavelets: Efficiently representing functions on the sphere. *Computer Graphics Proceedings (SIGGRAPH 95)*, pages 161–172, 1995.
- [27] I. W. Selesnick, R. G. Baraniuk, and N. C. Kingsbury. The dual-tree complex wavelet transform. *Signal Processing Magazine, IEEE*, 22(6):123–151, November 2005.
- [28] W. Sweldens. The lifting scheme: a construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2):511–546 (electronic), 1998.
- [29] W. Sweldens and P. Schröder. Building your own wavelets at home. In *ACM SIGGRAPH Course Notes*.
- [30] Wim Sweldens. The lifting scheme: a custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.*, 3(2):186–200, 1996.
- [31] D. Van De Ville, T. Blu, and M. Unser. Integrated Wavelet Processing and Spatial Statistical Testing of fMRI Data. *NeuroImage*, 23(4):1472–1485, 2004.
- [32] D. Van De Ville, T. Blu, and M. Unser. Wavelet-based fmri statistical analysis and spatial interpretation: A unifying approach. In *ISBI*, pages 1167–1170. IEEE, 2004.
- [33] D. Van De Ville, T. Blu, and M. Unser. Surfing the brain—An overview of wavelet-based techniques for fMRI data analysis. *IEEE Engineering in Medicine and Biology Magazine*, 25(2):65–78, March-April 2006.
- [34] D. Van De Ville, T. Blu, and M. Unser. Wspm or how to obtain statistical parametric maps using shift-invariant wavelet processing. In *Acoustics, Speech and*

Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on, volume 5, page V, may 2006.

- [35] D. Van De Ville, M.L. Seghier, F. Lazeyras, T. Blu, and M. Unser. WSPM: Wavelet-based statistical parametric mapping. *NeuroImage*, 37(4):1205–1217, 2007.
- [36] Oleg V. Vasilyev and Christopher Bowman. Second-generation wavelet collocation method for the solution of partial differential equations. *J. Comput. Phys.*, 165(2):660–693, 2000.
- [37] Li-Yi Wei and Marc Levoy. Texture synthesis over arbitrary manifold surfaces. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, pages 355–360, New York, NY, USA, 2001. ACM.
- [38] P. Yu, X. Han, F. Ségonne, R. Pienaar, R.L. Buckner, P. Golland, P.E. Grant, and B. Fischl. Cortical surface shape analysis based on spherical wavelets. *IEEE Transaction on Medical Imaging*, 26(4):582–597, 2007.