On Minimum Entropy Segmentation

David L. Donoho
Department of Statistics
Stanford University

September 1993
Revised March 1994

Abstract

We describe segmented multiresolution analyses of $[0,1]$. Such multiresolution analyses lead to segmented wavelet bases which are adapted to discontinuities, cusps, etc., at a given location $\tau \in [0,1]$. Our approach emphasizes the idea of *average-interpolation* – synthesizing a smooth function on the line having prescribed boxcar averages. This particular approach leads to methods with *subpixel resolution* and to wavelet transforms with the advantage that, for a signal of length n, all n pixel-level segmented wavelet transforms can be computed simultaneously in a total time and space which are both $O(n \log(n))$.

We consider the search for a segmented wavelet basis which, among all such segmented bases, minimizes the "entropy" of the resulting coefficients. Fast access to all segmentations enables fast search for a best segmentation.

When the "entropy" is Stein's Unbiased Risk Estimate, one obtains a new method of edge-preserving de-noising. When the "entropy" is the ℓ^2-energy, one obtains a new multi-resolution edge detector, which works not only for step discontinuities but also for cusp and higher-order discontinuities, and in a near-optimal fashion in the presence of noise.

We describe an iterative approach, *Segmentation Pursuit*, for identifying edges by the fast segmentation algorithm and removing them from the data.

Key Words and Phrases. Segmented Multi-Resolution analysis. Edge-Preserving Image processing methods. Edge detection. Sub-pixel resolution. De-Noising. Stein's Unbiased Risk estimate.

Acknowledgements. Supported by NSF DMS 92-09130 and by ONR N00014-92-0066 (Statistical Sciences, Inc.). The author's interest in this topic was stimulated by good-natured complaints against wavelet de-noising lodged by Jeff Scargle and Stéphane Mallat. The author thanks Iain Johnstone for suggestions leading to the Minimum Energy Segmentation edge detector, and Wim Sweldens for informing us of the priority of Deng, Jawerth, Peters, and Sweldens in introducing fast pixel-level segmentations.

1 Introduction

1.1 Improved De-Noising

Several recent papers (see [25] and references therein) have shown that wavelet methods can be used to de-noise data of various kinds, obtaining a level of theoretical performance not approached by pre-existing methods. In general, these methods have the following character: first, one takes the empirical wavelet transform of the noisy data; next one subjects the coefficients to a simple coordinatewise nonlinearity, applying specially-chosen thresholding to wavelet coefficients; finally one inverts the empirical wavelet transform, obtaining de-noised coefficients.

While the theoretical benefits of this approach are now well-established, and the actual reconstructions obtained by wavelet de-noising methods have seemed to us quite good, particularly in comparison with pre-existing methods, we have received comments from users and others that indicate some improvement is to be desired. These comments include

1. *Gibbs Phenomenon.* In the neighborhood of strong jump discontinuities, wavelet shrinkage methods often exhibit alternating overshoot. Although the phenomenon is much more localized than in the case of Fourier series, it would be desirable to improve further.

2. *Peak Shrinkage.* In the analysis of data such as NMR spectra, there is a tendency of wavelet shrinkage to "pull down" strong peaks, thereby distorting amplitudes. It would be desirable to reduce this tendency.

3. *Edge Erosion.* In the analysis of certain edge data, there is a tendency of wavelet shrinkage to erode weak edges, reducing the sharpness of transitions. It would be desirable to reduce or avoid this tendency.

4. *Inter-Scale Correlations.* The theory underlying the optimality of wavelet shrinkage techniques shows quite clearly that from a minimax-theoretic point of view, the wavelet transform plays the role of a de-correlating transform, mapping the object into a space where the different coordinates (wavelet coefficients) have no information about each other, and therefore scalar processing (e.g. coordinatewise thresholding) cannot essentially be improved upon. On the other hand, in "real world" objects, containing edges, one can expect to see nonzero wavelet coefficients at the same locations across several scales. Therefore, in real objects, the information that a certain wavelet coefficient is large leads to the presumption that similarly located coefficients at other scales be large. One expects that by exploiting such correlations, the accuracy of reconstruction might be improved over what simple thresholding offers. It would be desirable to develop a method to exploit such inter-level coefficient correlations and improve on ordinary wavelet shrinkage.

All of these problems seem to call for improving on wavelet de-noising, in a second pass, to clean up the "mess" left by the presence of singularites in 1 and 2-dimensional data. For example, if a user complains of Peak Shrinkage and asks for an improvement, there ought to be a simple, automatic procedure to correct it.

1.2 Segmented MRA, and Best Segmentation

In this paper, we develop an approach to these problems based on the concept of *segmented multi-resolution analysis* for the interval $[0,1]$, which allows a kind of wavelet decomposition and reconstruction adapted to the presence of segmentation. The functions in a segmented MRA need not be continuous across a certain point τ internal to the interval $[0,1]$. We develop in section 2 a special MRA, based on a biorthogonal system of wavelets called average-interpolating wavelets by Donoho (1993b); these derive from smooth wavelets which are biorthogonal to Haar wavelets, and they lead to fast algorithms for computing the segmented wavelet transform.

The idea is that if an object contains a sharp separation between one "phase" and another, we can develop an MRA adapted to that two-phase structure, and a corresponding wavelet transform, so as to avoid the presence of nonzero wavelet coefficients associated with the inter-phase transition.

Of course, the application of such a segmented MRA depends heavily on information about the location τ of the inter-phase boundary. In empirical work, this is not generally available a-priori. Hence, in order to exploit segmented MRA's, we must have some way to *infer* τ from data.

In section 3 we approach this problem from a *best-basis* point of view; compare Coifman and Wickerhauser (1992). Section 2 makes available to us a collection of *segmented* wavelet expansions

$$f \sim \sum_{j,k} \alpha^t_{j,k} \psi^{(t)}_{j,k};$$

each expansion determined by a segmentation point t. We seek among all these representations of f for one with minimal representation cost; we call the optimizing basis a best basis for f, and label the optimizing segmentation point $\hat{\tau}$. To measure representation cost we depart slightly from Coifman and Wickerhauser (1992), who were analyzing noiseless data and selecting best bases in a different collection of bases; they proposed the use of a $-\theta^2 \log(\theta^2)$ entropy as a measure of the cost of a representation. For measuring the cost of a representation in the noiseless cases we consider here other entropies, including θ^2, $|\theta|$, and $|\theta|^{1/2}$. In dealing with noisy data, we adapt ideas of Donoho and Johnstone (1993), and suggest the use of a certain Stein Unbiased Estimate of Risk as an entropy measure. This measures the quality of a given segmented-wavelet basis in which to de-noise the data. We test the functioning of the SURE Best-basis paradigm on some simple examples.

The practicality of any best-basis method depends on the existence of a fast search algorithm for searching through a collection of bases. In Section 4 we describe a fast algorithm for obtaining the optimizer $\hat{\tau}$ when the entropy measure is an additive measure of information. This algorithm searches through all n pixel-level segmentations and identifies the optimum one in order $n \log(n)$ space and time.

When we infer τ from noisy data, we are actually identifying edges. In fact, using the θ^2-entropy leads to a new multiresolution edge locator which adapts to the type of edge (step edge; cusp; etc.); we hope to show elsewhere that this has near-optimal properties in locating such types of edges in noise. Section 5 describes in heuristic terms some properties of this locator.

How can one handle the presence of multiple segmentation points? In section 6 we propose an iterative method, *segmentation pursuit*, based on iteratively identifying the best current segmentation point and "stripping away" the singularity at the identified position. We illustrate by a computational example.

This is not a "math paper". There are no theorems here, only computational experiments, motivated by our work in other "math papers". We aim here only to develop a collection of fast computational tools which may be employed as a "vacuum cleaner" to remove structure from residuals in ordinary de-noising caused by the presence of singularities in the object to be recovered. The algorithms described here are all available for use in MATLAB and may be obtained by anonymous FTP to playfair.stanford.edu.

1.3 The Challenge of Higher Dimensions

We view these 1-d results as only a modest start on a much more ambitious program of dealing with segmentation in higher dimensions. Such segmentations would potentially lead to improvements on wavelet de-noising of image data. To understand why, note once again that arguments on the optimality of wavelet de-noising depend heavily on the idea that the wavelet transform is a de-correlating transform. Yet when the wavelet transform is used in two dimensions, the presence of edges in "real world" images guarantees long connected "strings" of nonzero coefficients at each scale. The nonzero coefficients correspond to wavelets concentrated near dyadic boxes intersecting the edge curve. In two dimensions, one expects that wavelet de-noising can be improved on, by exploiting these inter-coefficient correlations, and that the effect is quantitatively much more important in two dimensions than in one-dimension.

To understand this point more fully, we exploit a connection between the de-noising goals of interest to us and certain data compression goals of interest to others. (For more on the connection between statistical estimation and data compression see Donoho(1993c)).

Wavelet compression of piecewise smooth objects, $f(t)$, defined on an interval $[0,1]$ in dimension 1, is very satisfactory. For example, a signal which is piecewise a polynomial of degree D, with only P pieces, has intrinsically

$(D+2) \times P$ parameters. If we take N equispaced samples of the signal, these N numbers may be efficiently recovered from just these $(D+2) \times P$ parameters. For comparison, the discrete wavelet transform based on N equispaced samples, (using wavelets with $> D$ vanishing moments) will have at most $P \cdot \log(N) \cdot C$ nonvanishing coefficients. Hence, the number of data which must be stored to recover via the wavelet transform is much less than N; in fact it is within a logarithmic factor of the ideal $(D+2) \times P$.

For dealing with objects $f(t)$ defined on a cube $[0,1]^d$ in dimension $d > 1$, wavelet compression of piecewise smooth objects is less satisfactory. Suppose we have an object defined by equispaced sampling on a grid, with spacing of size $1/N$ on a side. Then N^d samples naively characterize the object. The wavelet transform of a d-dimensional piecewise smooth object, with smooth $d-1$-dimensional boundaries between pieces, achieves an improvement over the N^d parameter representation, but only to something like N^{d-1} parameters.

The fact that N^d data are typically required to represent an object in d dimensions is sometimes called the *curse of dimensionality.* We may say that the wavelet transform lightens the curse of dimensionality by changing an exponent d to $d-1$. We would like, ideally, to find methods to further reduce the exponent $d-1$ by exploiting the "regularity of the singularity."

To make these issues concrete, consider the following *two-phase* image model in 2-dimensions:

$$f(x,y) = \begin{cases} s_1(x,y) & y > h(x) \\ s_0(x,y) & y \leq h(x) \end{cases} . \tag{1.1}$$

Here $h(x)$ denotes a *horizon*, and the s_i are smooth, for example, polynomials. In principle, such an object might be parametrized by the parameters of the polynomials s_1 and s_0, and the parameters of the boundary. In fact, if the boundary is piecewise smooth, it can itself be parametrized in terms of relatively few parameters. Therefore, instead of representing such an object by the nominally required N^2 numbers, we might instead get good reconstructions using a constant number of parameters, or else a number of parameters growing logarithmically with the scale of the finest resolution.

On the other hand, suppose we use a 2-d wavelet transform to represent the object, and suppose the number of vanishing moments of the wavelets under study is greater than the degree of s_1 and s_0. Then wavelets whose support is disjoint from the horizon will have zero coefficients, but every wavelet which "feels" the horizon is eligible to be significantly nonzero. There are order $Arclength(h) \cdot 2^j$ such coefficients at resolution level j, and so there are roughly

$$Arclength(h) \cdot N$$

nonzero wavelet coefficients in the 2-d wavelet transform. While this is better than N^2, it is not nearly as good as the order 1 or order $\log(N)$ we might think appropriate for a "good transform" of "very simple objects".

A very natural goal in this setting is to obtain compression to many fewer than $Arclength(h) \cdot N$ nonzero coefficients.

To see how this may be possible, we consider below a specific *horizon-adapted wavelet transform.* This is a 1.5d wavelet transform, which involves a segmented wavelet transform on each column of a 2-d array, followed by a traditional wavelet transform on each row. The segmentation point varies from column-to-column according to a horizon parameter $h = h(x)$. This gives us a *horizon-adapted* wavelet expansion

$$f \sim \sum_I \alpha_I^{(h)} \psi_I^{(h)}$$

Ideally, this achieves the following. *If* the object is of the horizon form (1.1), and *if* the transform is segmented with the correct horizon, there will be only $O(1)$ nonzero wavelet coefficients. Moreover, *if* the horizon itself is very smooth and simple, we get a representation of the horizon itself with only

$$Complexity(h, N)$$

nonzero coefficients, where $Complexity(h, N)$ should be much smaller than $Arclength \cdot N$ for simple smooth curves. For example, if the horizon is Lipschitz, we would expect that something like $N^{1/3}$ parameters suffice. With more regularity, even fewer parameters should suffice, and with less regularity, somewhat more. Hence a horizon-adapted transform can "compress away" data caused by the horizon.

The ability to "compress away" wavelet coefficients associated with a horizon therefore might offer benefits in statistical estimation. Suppose we observe an N-by-N collection of block averages of an image observed in a white noise of standard deviation σ. It nominally costs σ^2/N^2 in risk to estimate one normalized parameter. Suppose the object has a jump discontinuity across a horizon. Under the usual 2-d wavelet transform, the noiseless object has roughly $Arclength(h) \cdot N \cdot \log(N)$ nonzero wavelet coefficients of the noiseless and so there are $Arclength(h) \cdot N \cdot \log(N)$ "parameters" which need to be estimated. Because of this, the best de-noising can do in dimension 2 is a per-pixel mean-squared error going to zero no faster than

$$\sigma^2 \log(N)/N.$$

De-Noising a correctly horizon-adapted wavelet transform of a horizon object (1.1) involves estimating only order $O(1)$ nonzero parameters. Therefore, ignoring the cost of estimating the horizon itself, the component of the risk caused by the horizon drops to

$$\sigma^2 O(1)/N^2.$$

Therefore *if* the horizon can be estimated from data, horizon-adaptive methods promise a significant reduction in mean-squared errors.

1.4 First Steps

In practice, the author knows at the moment of no elegant method for reaching the full goal of identifying a horizon from data based on clear principles, a fully 2-dimensional approach, and using fast non-heuristic algorithms.

In Section 7 we cobble together computational tools from the 1-d case, adapting ideas of segmented transforms and best-basis search to a higher-dimensional setting. We study the problem of representing an image containing a sharp horizon. A segmented 1.5-d transform gives us a representation

$$f \sim \sum_I \alpha_I^{(h)} \psi_I^{(h)},$$

each such representation dependent on a parameter $h = h(x)$. In general we *should* seek among all these representations for one with minimal representation cost, where cost includes the cost in representing h.

Our simplest cobbling-together ignores the cost of representing $h(x)$ and attempts to segment each column of a 2-dimensional array as an independent 1-dimensional dataset, without using any information about adjacent columns. Computational experiments show that such an adaptively segmented 1.5-dimensional wavelet transform can work acceptably even in the presence of noise. We suspect that an approach exploiting local continuity of $h(x)$ works better at low signal-to-noise ratio, but are not aware of fast algorithms for such an approach.

Another way to look at the problem of wavelet de-noising in high dimensions is in terms of error structure. As we show below, ordinary de-noising of objects containing a horizon leads to errors "original surface" - "reconstructed surface" which contain considerable structure in the neighborhood of a horizon. This is due to the "correlation" of wavelet coefficients induced by edges. Computational experiments show that segmented de-noising in the 1.5-dimensional setting can significantly reduce the correlation of errors caused by edges and so can improve the behavior of wavelet de-noising near edges and creases.

The challenge of better compression and de-noising in high dimensions is still largely open. We briefly describe, in section 8, what is involved in implementing a "fully 2-d" refinement algorithm. Perhaps such discussion will stimulate further progress.

1.5 Credit where credit is due

The idea to recognize the special role of edges in images for data compression purposes is not new. It is related to existing edge-coding ideas in image processing as well to the nonlinear multi-scale edge reconstruction ideas of Mallat and Zhong (1992) and Mallat and Froment (1992). The idea that wavelets improve the curse of dimensionality, but only by dropping the exponent from d to $d-1$ is also related to comments in the article on compression of operators by Beylkin, Coifman, and Rokhlin (1991).

The idea to use some sort of wavelet transform adapted to the presence of edges has been mentioned by Prof. Björn Jawerth of the University of South Carolina at several conference presentations in 1992 and 1993. After the work reported here was done, the author learned that Deng, Jawerth, Peters, and Sweldens (1993) have independently and somewhat earlier come up with fast algorithms for computing all pixel-level segmented 1-d transforms. Their method is based on "breaking" the dataset into pieces and computing boundary-adjusted transforms of the left and right pieces, while ours is based on segmented refinement schemes. The underlying logic is somewhat different, while the resulting algorithms are similar.

2 1-d Segmented Wavelet Transforms

In this section we briefly describe a method for constructing 1-d segmented multi-resolution analyses.

2.1 Refinement by Average-Interpolation

Suppose we have an array $(a_{j,k})_{k=-\infty}^{\infty}$ which represents averages of a function f on dyadic intervals $I_{j,k} = [k/2^j, (k+1)/2^j]$. We may synthesize mock-averages at finer scales by the following procedure (see Figure 2.1). Let D be an *even* integer greater than 0.

[1] At each site k, find the polynomial $\pi_{j,k}$ of degree D which generates the same averages in the neighborhood $(a_{j,k'}, k' = k - D/2, \dots, k + D/2)$, i.e.

$$Ave_{j,k'}\pi_{j,k} = a_{j,k'}, \qquad k' = k - D/2, \dots, k + D/2.$$

As the polynomial has $D+1$ coefficients and there are $D+1$ constraints to satisfy, the polynomial is uniquely determined.

[2] Define the mock-averages at the next finer scale as averages of that polynomial. On the left half of the sub-interval we get

$$a_{j+1,2k} = Ave_{j+1,2k}\pi_{j,k}$$

on the right half

$$a_{j+1,2k+1} = Ave_{j+1,2k+1}\pi_{j,k}.$$

[3] After having synthesized all the $a_{j+1,k}$'s, set $j := j+1$ and goto [1]

This refinement scheme, which is analogous to the interpolating refinement scheme of Deslauriers-Dubuc [13, 26], is discussed at length in Donoho(1993b). The main point is that it defines a sequence of refinements which in some sense converge: the function $A_{j',D}(t) = \sum_k a_{j',k} 1_{[k/2^{j'},(k+1)/2^{j'}]}(t)$ converges to a continuous limit $A_D(t)$ on the line, which has the averages $a_{j,k}$ at scale 2^{-j}. In fact, the limit has C^R regularity, where $R = R(D)$ increases with D.

The above describes average-interpolation on the line. On the interval $[0,1]$, we have only averages $(a_{j,k})_{k=0}^{2^j-1}$. At the heart of the interval, refinement can proceed exactly as above: at the edges we redefine the set of neighboring intervals in step [1] to refer only to the $D+1$ nearest intervals fitting inside the interval $[0,1]$.

Return again to the case of functions on the line.

2.2 Average-Interpolating Multi-Resolutions

The vector space V_j of functions obtainable by refining sequences $(a_{j,k})_k$ has an alternate description. Refining the Kronecker sequence $a_{0,k} = \delta_{k,0}$ yields fundamental functions $\phi = \phi_D$. These functions and their integer translations and dyadic dilations $\phi_{j,k}(t) = 2^{j/2}\phi(2^j t - k)$ generate the spaces $V_j = \{f : f = \sum_k \beta_{j,k}\phi_{j,k}(t)\}$. The parameters are identical to rescaled averages: $\beta_{j,k} = 2^{-j/2}a_{j,k}$. The V_j make up a multiresolution analysis which is therefore biorthogonal to the usual Haar MRA. The operations of calculating the averages $(a_{j,k})_k$ of f at scale 2^{-j} and then refining those averages to produce a limit function $\tilde{f}$; the linear operator implicitly defined by $\tilde{f} = P_j f$ acts as the identity on V_j and is therefore a non-orthogonal projection. Because the average interpolation scheme is exact on polynomials of degree D, we have

$$P_j \pi = \pi$$

whenever π is a polynomial of degree D.

Given the averages at a scale $j+1$, we of course know the averages at scale j, because $a_{j,k} = (a_{j+1,2k} + a_{j+1,2k+1})/2$. The vector space W_j obtained by refining sequences $(a_{j+1,k})_k$ where $a_{j+1,2k} = -a_{j+1,2k+1}$, consist entirely of functions whose coarser-scale averages are zero; it is in fact the difference space $W_j = V_{j+1} - V_j$. This space has an alternate description. Refining the Kronecker sequence $a_{0,k} = (\delta_{k,1} - \delta_{k,0})/\sqrt{2}$ yields Mother wavelets $\psi = \psi_D$. These functions and their integer translations and dyadic dilations $\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k)$ generate the difference spaces: $W_j = \{f : f = \sum_k \alpha_{j,k}\psi_{j,k}(t)\}$. Let $h(t)$ be the Haar function $h(t) = 1_{(1/2,1]} - 1_{(0,1/2]}$, and $h_{j,k}(t) = 2^{j/2}h(2^j t - k)$. The parameters $\alpha_{j,k}$ of an object $f \in W_j$ are identical to Haar coefficients of f: $\alpha_{j,k} = 2^{-j/2}(a_{j+1,2k+1} - a_{j+1,2k})/2 = \int f h_{j,k}$. The difference space W_j is therefore biorthogonal to the usual Haar detail spaces. Now consider the operator which, given the averages of f at scale 2^{-j-1}, calculates the averages one scale coarser, refines those coarser averages, producing mock averages $(\hat{a}_{j+1,k})$; then forms the residuals $r_{j+1,k} = (a_{j+1,k} - \hat{a}_{j+1,2k})$ and average-interpolates that residual sequence. This produces an element of W_j, which we denote $Q_j f$; Q_j is a non-orthogonal projection on W_j.

This multi-resolution system arose before, without the average-interpolation interpretation, in Cohen, Daubechies, and Feauveau (1990), where it was called a system biorthogonal to spline of degree 0; see the discussion in [19].

Corresponding to these schemes on the line are boundary-corrected multiresolutions on the interval $[0,1]$. These are built from a boundary-corrected refinement scheme for the interval. This scheme has spaces $V_j^{[]}$ and $W_j^{[]}$, with projectors $P_j^{[]}$ and $Q_j^{[]}$. These retain key properties from the line, such as biorthogonality with respect to the Haar system, and the polynomial exactness $P_j^{[]}\pi = \pi$, valid whenever π is a polynomial of degree D on $[0,1]$. There are 2^j basis elements of $V_j^{[]}$, obtained by refinement of appropriately normalized Kronecker sequences and also 2^j elements of $W_j^{[]}$. We call these functions $\phi_{j,k}$ and $\psi_{j,k}$, respectively. See Figure 2.2. Fix j_0 so that $2^{j_0} > 2(D+2)$. Then every function in $L^2[0,1]$ has an expansion

$$f = \sum_{k=0}^{2^{j_0}-1} \beta_{j_0,k}\phi_{j_0,k} + \sum_{j \geq j_0} \sum_{k=0}^{2^j-1} \alpha_{j,k}\psi_{j,k}$$

unconditionally convergent in L^2 norm.

Here the coefficients are computed by

$$\beta_{j,k} = \int_0^1 \chi_{j,k}(t)(f - P_j^{[]}f)(t)dt$$

where $\chi_{j,k} = 2^{j/2}1_{(2^{-j}k, 2^{-j}(k+1)]}$ is a normalized boxcar and

$$\alpha_{j,k} = \int_0^1 h_{j,k}(t)(f - P_j^{[]}f)(t)dt.$$

The mapping from f to its coefficients $((\beta_{j_0,k})_k, (\alpha_{j_0,k})_k, (\alpha_{j_0+1,k})_k, \ldots))$ is the non-segmented wavelet transform we are interested in.

This transform has the property that its basis elements are C^R regular, with an $R = R(D)$, and the $\alpha_{j,k}$ coefficients all vanish for polynomials of degree D, that the $\alpha_{j,k}$ coefficients of a function in C^r, $0 < r < R$, are of order $2^{-j(r+1/2)}$; see Donoho (1993b) for more information.

Moreover, the transform has associated with it a fast algorithm. Given the boxcar integrals $\beta_{j_1,k}$ at a fine scale $2^{-j_1} \ll 2^{-j_0}$, coefficients $(\beta_{j_0,k})$ and $(\alpha_{j,k})$ at all coarser levels $j_0 \leq j < j_1$ can be computed in order n time, where $n = 2^{j_1}$.

2.3 Segmented Refinement

Consider now the following segmented refinement procedure, with segmentation point τ. We assume that the segmentation point is in the heart of the interval, so that $D/2^j < \tau < (2^j - D/2^j)$. Given a sequence of averages $a_{j,k}$, $0 \le k < 2^j$, as in Figure 2.3 we synthesise mock averages at finer scales by the following procedure:

[1] At each site k which is more than $D/2$ sites away from the boundaries 0 and 1 and more than $D/2$ sites away from the segmentation point τ, use the earlier procedure to find the polynomial $\pi_{j,k}$ of degree D which generates the same averages in the neighborhood $(a_{j,k'}, k' = k - D/2, \dots, k + D/2)$.

[2] At each site k which is at most $D/2$ sites away from the boundaries 0 and 1 find the polynomial $\pi_{j,k}$ of degree D which generates the same averages in the neighborhood $(a_{j,k'})_{k' \in N(k)}$, where $N(k)$ consists of the $D + 1$ nearest neighbors of k.

[3] At each site k which is at most $D/2$ sites away from the segmentation point τ, we distinguish two cases. [3a] If $\tau \notin [2^{-j}k, 2^{-j}(k+1)]$, then find the polynomial $\pi_{j,k}$ of degree D which generates the same averages in the neighborhood $(a_{j,k'})_{k' \in N(k)}$, where $N(k)$ consists of the $D+1$ nearest neighbors of k which are all on the same side of the segmentation point as k. [3b] If $\tau \in [2^{-j}k, 2^{-j}(k+1)]$, then fit, by constrained least squares, left and right polynomials $\pi^L_{j,k}$ $\pi^R_{j,k}$ of degree D to the block averages in the neighborhoods on the right and left of the segmentation point, respectively; the constraint is that the piecewise polynomial $\pi^\tau_{j,k}$ which is π^L on the left of τ and π^R on the right of τ should have an average equal to the average $a_{j,k}$.

[4] In cases [1], [2], and [3a], define the mock-averages at the next finer scale as averages of the polynomial. On the left half of the sub-interval we get

$$a^\tau_{j+1,2k} = Ave_{j+1,2k}\pi_{j,k}$$

on the right half

$$a^\tau_{j+1,2k+1} = Ave_{j+1,2k+1}\pi_{j,k}.$$

In case [3b], the steps are the same, only using the piecewise polynomial $\pi^\tau_{j,k}$.

[5] After having synthesized all the $a^\tau_{j+1,k}$'s, set $j := j+1$ and goto [1]

This segmented average-interpolating resolution has a variety of properties; a key one being that *if π^τ is a piecewise polynomial of degree D, with one knot, at τ, then the refinement process recovers π^τ exactly.*

To see the possible benefit of this procedure consider a piecewise linear function, with jump discontinuity at $\tau = \lfloor .37 \cdot 256 \rfloor / 256$. Figure 2.4a depicts the boxcar averages at scale $j = 8$; Figure 2.4b depicts the same averages at scale $j = 4$. Refinement from the coarse data at scale $j = 4$ by the usual nonsegmented method produces Figure 2.4d; the attempted refinement misses the fine-scale structure entirely; in fact, the transition in the synthesized fine-scale data retains the same slope as in the coarse scale data. In contrast, Figure 2.4c illustrates the result of segmented refinement; this has perfectly reconstructed the fine-scale data, despite being derived from much coarser-scale data.

2.4 Fast Segmented Transforms

Combining results of the last two sections allows us to define fast segmented transforms, as follows. Given a segmentation point τ at the "Heart" of the interval, and a j_0 satisfying $D/2^{j_0} < \tau < (2^{j_0} - D)/2^{j_0})$, we transform f to coefficients

$$\beta_{j_0,k} = \int_0^1 \chi_{j_0,k}(t) f dt, \qquad k = 0, \dots, 2^{j_0} - 1,$$

and

$$\alpha^\tau_{j,k} = \int_0^1 h_{j,k}(t)(f - P^\tau_j f)(t) dt.$$

Given integrals at a fine scale $\beta_{j_1,k}$, $j_1 \gg j_0$, we can calculate all needed coefficients at coarser scales $j_0 \le j < j_1$ in order 2^{j_1} time. Given $(\beta_{j+1,k})_k$ we calculate $\beta_{j,k} = (\beta_{j+1,2k} + \beta_{j+1,2k+1})/\sqrt{2}$ exactly as in the fast algorithm for the Haar transform. We then calculate $(\alpha^\tau_{j,k})$ as follows: refine the coarser sequence $(\beta_{j,k})$, getting $(\hat{\beta}^\tau_{j+1,k})_k$, (this takes $O(2^j)$ time), and form the residuals $r^\tau_{j+1,k} = \beta_{j+1,k} - \hat{\beta}^\tau_{j+1,k}$. These residuals obey $r^\tau_{j+1,2k} = -r^\tau_{j+1,2k+1}$ and so

$$\alpha^\tau_{j,k} = (r^\tau_{j+1,2k+1} - r^\tau_{j+1,2k})/\sqrt{2}.$$

As all computations at level j can be performed in a time proportional to 2^j, this algorithm for computing at all levels $j_0 \leq j < j_1$ is order 2^{j_1}.

2.5 Examples of Segmented Transforms

Now assume that we have been given a segmentation point τ and n boxcar averages, $(a_{j_1,k})_k$, of f, where $n = 2^{j_1}$. We have seen that we can calculate the first n wavelet coefficients of f in order n time. (Incidentally, the reconstruction of boxcar averages from those wavelet coefficients is also order n).

We now give some examples of this fast transform in action. Figure 2.5 presents four objects: *Ramp, Cusp, Junk*, and *HeaviSine*. They all are segmented at the point $\tau = .3696$. Data consist of boxcar averages at resolution $j = 11$. We use $D = 2$ and $j_0 = 4$ below.

Figure 2.6 presents the traditional wavelet coefficients $(\alpha_{j,k})$ of these objects. In panels a,b, and d, the presence of the singularity is clearly signaled by the significant wavelet coefficients in the vicinity of τ. Similar information is contained in Figure 2.7, which presents the multi-resolution displays $P_{j_0} f$ and $Q_j f$ for these objects.

Figure 2.8 presents the segmented wavelet coefficients $(\alpha^{\tau}_{j,k})$ of these objects. In Panels a, and b, they are all essentially zero; in Panel d they are essentially zero after two resolution levels. The segmented coefficients of object *Junk* scarcely differ from the ordinary non-segmented ones. Figure 2.9 portrays the same information in the form of multi-resolution displays $P^{\tau}_{j_0} f$ and $Q^{\tau}_j f$ for these objects. The discontinuity in the MRA's is visible.

2.6 Application Areas

We now indicate three possible applications of segmented wavelet transforms

2.6.1 Data Compression

It is evident on comparing Figures 2.6 and 2.8 that coefficients which are significant in the ordinary wavelet expansion become zero in the segmented wavelet expansion. This is a consequence of the fact that if π^{τ} is a piecewise polynomial of degree D, with breakpoint at τ, then

$$\alpha^{\tau}_{j,k}(\pi^{\tau}) = 0, \qquad j \geq j_0, \quad k = 0, \ldots, 2^j - 1.$$

Ramp is piecewise linear; *Cusp* and *HeaviSine* are piecewise analytic, and so well approximated by polynomials; hence the segmented wavelet transform really should be sparser than the ordinary one.

To quantify sparsity, we use the following approach. Suppose that $\theta = T(f)$ is a transform of f into a sequence space; we measure sparsity in the transform domain as follows. Let $|\theta|_{(i)}$ denote the i-th from largest coefficient in θ, so that $|\theta|_{(0)} = \max_k |\theta_k|$, and define the compression number

$$c_m = \sum_{i>m} |\theta|_{(i)}^2.$$

The compression numbers measure how well we can approximate the vector θ by a vector with only m nonzero entries. If c_m tends to zero rapidly with m, then there are very few big coefficients in θ.

Figure 2.10 portrays the compression numbers c_m for the two different transforms of each of the objects. In each case, the dashed line is the ordinary transform and the solid line is the properly segmented transform. Evidently, segmented compression is much better than ordinary compression in cases a, b, and d, while it performs about the same as ordinary compression for object *Junk*.

2.6.2 Subpixel resolution

As indicated in Figure 2.4, a properly segmented multi-resolution operator P_j^τ has the ability to reconstruct jumps much more precisely than the usual $\Omega(2^{-j})$ resolution of un-segmented operators. In principle this can even continue at scales finer than the data gathering, so that if we know the segmentation point τ with extreme precision, we can reconstruct at a resolution which is just as accurate as our knowledge of τ, and much more accurately than our measurement model seems naively to permit.

2.6.3 De-Noising

As indicated in the introduction, one of our main interests in the present topic is in improving the behavior of wavelet shrinkage de-noising. To illustrate how this can be done using segmented transforms, we present in Figure 2.11 a noisy version of the *Ramp* object, its segmented wavelet transform, a thresholded version of the transform, and the reconstruction which was obtained by inverting the transform. The result displays a clean break, with no messy Gibbs phenomena, nor any appreciable shrinkage of the jump.

In contrast, Figure 2.12 gives a side-by-side comparison of the segmented recovery with the usual non-segmented method described in [25], using a periodized wavelet transform. The difference is pronounced; the non-segmented method is plagued by Gibbs artifacts.

We also present, in Figure 2.13, a noisy version of the *Cusp* object, its segmented wavelet transform, a thresholded version of the transform, and the reconstruction which was obtained by inverting the transform. The result displays a clean cusp in the correct location and amplitude.

Figure 2.14 gives a superposed comparison of the segmented recovery with the usual non-segmented method described in [25], using a periodized wavelet transform. The difference is not very pronounced; the non-segmented method is plagued by downward shrinkage of peak amplitudes.

For fairness, it must be emphasized at this point that we are using *ideally* segmented transforms, in which the exact value τ of the break is known and employed. Such exact knowledge would not be available in most situations.

2.7 Variations

Before leaving the topic of 1-dimensional segmented MRA's, it is worth remarking that many variations on these ideas are possible.

First, the specific average-interpolating refinement scheme we are studying is not the only one we could have used. Here we have obtained a polynomial by interpolating averages at blocks in a neighborhood of a point. We could equally well have fit, by constrained least squares, the averages at blocks in a larger neighborhood, with the constraint that the polynomial in question had an average that agreed exactly with the block average to be refined. This would give a method with perhaps more numerical stability, at the expense of longer filters.

Second, it is not necessary to use average-interpolating wavelets; for example, rather than modelling the system on biorthogonality with respect to the Haar system, one could have used higher-order spline systems – at the expense of greater complexity in certain refinement calculations.

Third, the system considered here is only biorthogonal. To be maximally consistent with the best-basis notions we will present below, it would be very interesting to consider segmented orthogonal wavelet expansions. The ideas of Anderssen, Hall, Jawerth, and Peters (1993) may be useful in this connection.

3 Adapting by Minimum Entropy

There is one obvious objection to the direction we have been headed. This would argue that while segmented transforms may be attractive in the ideal case where the appropriate point of segmentation is known exactly, one never knows this point in advance, and so the concept of of segmented transform is of uncertain usefulness.

In this section we will investigate the idea of selecting, adaptively, from data, an appropriate segmentation. Let $\mathcal{E} = \mathcal{E}(\theta)$ be an *entropy*, which in our terms means merely a functional which is small for sparse vectors containing very few nonzero components and which is large for vectors containing very many nonzero components all of the same size. An example is the ℓ^1 entropy

$$\mathcal{E}^1(\theta) = \sum_i |\theta_i|.$$

Other examples will be given below. We use the convention that if x denotes a vector of dyadic length $n = 2^{j_1}$ containing block averages at scale 2^{-j_1}, then $W_n^t x$ denotes the segmented wavelet coeficients $((\beta_{j_0,k})_k, (\alpha_{j,k}^t)_k)$ obtained with segmentation point t. The *Minimum Entropy Segmentation* principle (MES) is to select, from among all possible segmented bases for representing x, that basis which gives the coefficients with smallest entropy in the wavelet coefficient domain:

$$\hat{\tau} = \text{arg } \min_{t\in[0,1]} \mathcal{E}(W_n^t x)$$

One ought to choose the entropy $\mathcal{E}$ so that the resulting segmentation reflects the task at hand. Consequently, there will be different implementations of this principle, depending on whether one's goal is data compression or de-noising.

3.1 Data Compression

Coifman and Wickerhauser (1992) in now-classic work, have proposed a method of best-basis selection which, translated into the present framework, goes as follows. First, given wavelet coefficients $W_n^t x$, define $p_{j,k} = (\alpha_{j,k}^t)^2$. Then $p_{j,k} \geq 0$.

Second, one defines the Coifman-Wickerhauser entropy by

$$\mathcal{E}^{CW}(\theta) = -\sum_{j,k} p_{j,k} \log(p_{j,k})$$

(We differ here from Coifman-Wickerhauser in two ways. First, they were working with orthogonal transformations, and it was therefore natural to normalize the object to unit ℓ^2-norm 1. We do *not* adopt this convention here. Second, our sum does not include the $(\beta_{j_0,k})$ terms, which anyways are the same regardless of segmentation point t.)

To the original C-W entropy we add the following general family of entropies $\mathcal{E}^\alpha$, $\alpha \in [0,2]$,

$$\mathcal{E}^\alpha(\theta) = \sum_{j,k} p_{j,k}^{\alpha/2}.$$

We are particularly interested in the ℓ^1 entropy $\mathcal{E}^1$, the $\ell^{1/2}$ entropy $\mathcal{E}^{1/2}$, and the ℓ^2 entropy $\mathcal{E}^2$.

All of these measures are measures of anti-sparsity. The limit, as $\alpha \to 0$, is simply the numerosity,

$$\mathcal{E}^0 = \#\{(j,k) : \alpha_{j,k}^t \neq 0\}.$$

At the other limit, as $\alpha \uparrow 2$, we can obtain the C-W entropy:

$$\frac{d}{d\alpha}\mathcal{E}^\alpha(\theta)|_{\alpha\to 2^-} = \mathcal{E}^{CW}(\theta).$$

Hence the Coifman-Wickerhauser entropy is the tangent on a curve which measures sparsity; the other entropies are simply points on that curve. See Figure 3.1.

Note that in the original best-basis setting, one was considering a choice among orthogonal bases, and the ℓ^2 entropy would not vary among bases; but here, one is considering a choice among non-orthogonal bases, and the ℓ^2 entropy is more reasonable.

We now compare the use of these entropies in choice of segmentation. First we consider object *Ramp*; here n = 2048 and the correct segmentation is at $\tau = 757/2048$. Figure 3.2 shows the unsegmented wavelet transform and three segmentations at pixel boundaries 756, 757, and 758. Because the object is piecewise linear, at the correct segmentation 757, the wavelet coefficients $\alpha_{j,k}$ all vanish. The figure shows that at nearby segmentations, the segmented wavelet transform entropy is intermediate between an unsegmented one and the appropriately segmented one. Figure 3.3 shows entropy profiles for pixel-level segmentations running from 749 to 765. At the correct segmentation 757, all of the entropies vanish. As we move away from the correct segmentation, the entropy more or less increases, but is more well-behaved for the 2, 1- and 1/2 entropies than for the C-W entropy, which seems erratic.

Next we consider the object *Cusp*; here $n = 2048$ and the correct segmentation is again $\tau = 757/2048$. Figure 3.4 shows the unsegmented wavelet transform and three segmentations at pixel boundaries 756, 757, and 758. Because the object is piecewise analytic, at the correct segmentation 757, the wavelet coefficients $\alpha_{j,k}$ nearly vanish. The figure shows that at nearby segmentations, the segmented wavelet transform entropy is intermediate between an unsegmented one and the appropriately segmented one. Figure 3.5 shows entropy profiles for pixel-level segmentations running from 749 to 765. At the correct segmentation 757, all of the entropies vanish. As we move away from the correct segmentation, the entropy more or less increases, but is again more well-behaved for the 2, 1- and 1/2 entropies than for the C-W entropy, which has rather a broad minimum, and fails to point to a unique optimum.

In both examples, the 1/2-entropy indicates a sharper preference for a specific segmentation than the other entropies.

3.2 De-Noising

We now consider adaptive choice of basis in the presence of noise. With $n = 2^{j_1}$, we suppose that we have noisy block-averages

$$d_k = Ave\{f|I_{j_1,k}\} + \epsilon \cdot z_k, \qquad k = 0, \dots, n-1$$

where the z_k are a Gaussian white noise. We process these data as if they were noiseless block averages, obtaining coarse-scale empirical block averages

$$v_{j_0,k} = \beta_{j_0,k} + \epsilon \cdot \xi_{j_0,k}, \qquad k = 0, \dots, 2^{j_0} - 1,$$

and empirical wavelet coefficients

$$w^t_{j,k} = \alpha^t_{j,k} + \epsilon\zeta^t_{j,k}, \qquad k = 0, \dots, 2^j - 1.$$

We act provisionally as if the ξ's and ζ's were independent and constant variance 1, which they are not, owing to the lack of orthogonality of the transforms.

We consider the problem of recovering the vector of coefficients $\theta^t = \big((\beta_{j_0,k})_k, (\alpha^t_{j_0,k})_k, \dots, \big)$, and we group the noisy empirical wavelet coefficients together into a vector $y^t = \big((v_{j_0,k})_k, w^t_{j,k}, \dots, \big)$.

Initially, consider the following ideal problem (compare Donoho and Johnstone (1992a)). We have available an oracle which furnishes optimal weights (w_i) for use in a diagonal linear estimator $\hat{\theta}^t = (w_i y^t_i)_i$; these weights being optimal in the sense that they minimize the mean squared error

$$E \sum (w_i y_i - \theta_i)^2.$$

In reality such an oracle and such optimal weights are never available to us. The risk of such an ideal procedure is within a factor of 2 of the following proxy:

$$\mathcal{R}(\hat{\theta}^t) = \sum_i \min((\theta^t_i)^2, \epsilon^2).$$

In terms of the compression number introduced earlier, we have

$$\mathcal{R}(\hat{\theta}^t) = c_{N(\epsilon)} + \epsilon^2 N(\epsilon),$$

where $N(\epsilon) = \#\{i : |\theta_i| > \epsilon\}$. As this "ideal risk"is therefore large for dense vectors containing lots of entries and small for sparse vectors containing only a few nonzero coefficients, it is an entropy.

Figure 3.6, panel (a) displays the behavior of this ideal risk measure in segmenting the *Ramp* object; panel (b) displays the behavior in segmenting the *Cusp* object. In both cases the minimum risk segmentation is at the natural location. There is a sharper preference for the minimum in the case of the *Ramp* object, which is connected with the object's sharper discontinuity.

Now consider the behavior of a "real" de-noising procedure, as follows. Setting a threshold $\lambda = \sqrt{2\log(n)}\epsilon$ we apply soft thresholds, getting estimates

$$\hat{\theta}_i = \eta_\lambda(y_i), i = 1, \dots, n.$$

(Again we are acting provisionally as if the y_i all have the same variance ϵ^2). As in Donoho and Johnstone (1993), we can estimate the risk of this estimator using Stein's Unbiased Estimate of Risk for this nonlinear estimator:

$$SURE(y) = \epsilon^2 \cdot \left(n - 2\sum_i 1_{|y_i|<\lambda} \right) + \sum_i \min(y_i^2, \lambda^2).$$

This risk measure is smaller for sparse vectors and larger for dense vectors, and so represents a kind of entropy.

Figure 3.6, panel (c) displays the behavior of this empirical risk measure in segmenting the noisy *Ramp* object; panel (d) displays the behavior in segmenting the noisy *Cusp* object. In both cases the SURE profile is much noisier than the Risk profile (as expected); and the minimum SURE segmentation is near, but not exactly at, the natural location. There is again sharper preference for the minimum in the case of the *Ramp* object, which is connected with the object's sharper discontinuity.

Apparently, even in the presence of noise, one can adaptively select a transform which preserves the structure of strong discontinuities.

4 Fast Computation of all Segmentations

A further objection to the direction we have been headed is computational. The segmented wavelet transform is an order n operation; to calculate all n pixel-level segmentations therefore seems naively to require an $O(n^2)$ procedure; this is unsuitable for many applications.

Fortunately, there is a method for calculating all n pixel-level segmentations in order $n \log(n)$ time and space. The method is based on the following observation. In performing the segmented refinement, only those blocks within a distance $D/2$ of the block containing the segmentation point are affected by the segmentation. That is to say, the resulting values are the same as they would be in a non-segmented refinement based on the average-interpolating wavelets of section 2.1. We therefore propose to calculate, for each pixel-level segmentation $t = i/n, i = 0, \dots, n-1$, only those specific coefficients which differ from the non-segmented transform.

Label these coefficients

$$\nu_{j,l}(t), j = j_0, \dots, j_1 - 1, l = -D/2, \dots, D/2,$$

where the subscript j again indicates resolution level and the subscript l indicates offset from the block $\lfloor t2^j \rfloor$ containing the segmentation point.

Suppose these coefficients are all available for a given segmentation point t, and that we also have available the unsegmented transform. By copying values from the array of ν into the appropriate locations of the array of unsegmented wavelet coefficients, we obtain the segmented wavelet coefficients.

Each one of these coefficients depends in a linear fashion on a fixed number of block averages at scale $j+1$ in a neighborhood of a given block. Therefore each coefficient can be computed from scratch in order $C \cdot D$ work.

There are order $log_2(n) \cdot D$ $\nu's$ attached to a given t; therefore the calculation of all the $\nu's$ attached to that t, starting from scratch, is an order $D^2 \log_2(n)$ operation.

It follows that we can evaluate, in sequence, all n pixel-level segmented transforms by the following approach.

- 1. Compute the unsegmented AI wavelet transform. Make an extra copy of the transform array.
- 2. For $i = 0, \ldots, n-1$ do:
- 2.a. Calculate the ν-coefficients for $t = i/n$.
- 2.b. Copy them into the unsegmented array at the appropriate positions (these depend on t).
- 2.c. Evaluate the entropy of the resulting array
- 2.d. Restore the unsegmented array with the original unsegmented wavelet coefficients.
- 3. Using the best i arising in step 2, perform steps 2.a and 2.b once more at that i.

The cost of this procedure, excepting the evaluations of entropy, is order $n \cdot \log(n) \cdot D^2$.

We now point out how to rapidly minimize the entropy functional. Let θ^0 denote the coordinates of the unsegmented wavelet transform. Define the differential entropy

$$\delta\mathcal{E}(t) = \mathcal{E}(\theta^t) - \mathcal{E}(\theta^0).$$

The minimum of the entropy $\mathcal{E}(\theta^t)$ will be at the same value of t as the minimum of the differential entropy, so it is sufficient to minimize differential entropy.

Now we remark that all the entropy functionals we have discussed are coordinatewise sums; in addition, most of the coefficients of θ^0 and θ^t agree. Therefore most terms in the entropy difference $\mathcal{E}(\theta^t) - \mathcal{E}(\theta^0)$ disappear, and only those coefficients which are potentially different need be considered. The differential entropy $\delta\mathcal{E}(t)$ is, up to a quantity which does not depend on t, simply a functional of the ν coefficients, and of the unsegmented wavelet coefficients that they replace. Call the coefficients being replaced

$$\mu_{j,l}(t), j = j_0, \ldots, j_1 - 1, l = -D/2, \ldots, D/2.$$

(Of course, these are all present in the single n-element array of wavelet coefficients, so that one does not actually store the μ's; it is convenient to have a notation referring to them). One can therefore simply evaluate the *entropy of the ν-coefficients*, subtract the entropy of the μ-coefficients, and minimize this difference as a function of t.

We therefore have the following streamlined algorithm, which requires less time and space.

- 0. Calculate the ordinary unsegmented transform.
- 1. For $i = 0, \ldots, n-1$ do:
- 2.a. Calculate the ν-coefficients for $t = i/n$.
- 2.b. Evaluate the entropy difference between those ν coefficients, and the corresponding μ coefficients from the unsegmented transform.
- 3. Using the best i arising in step 2, calculate the segmented wavelet transform for $t = i/n$.

The complexity of the unsegmented transform is again $O(n)$, and the whole algorithm is order $n \cdot \log_2(n) \cdot D^2$.

Remarks.

1. With the exception of offsets $l = 0$, each $\nu_{j,l}$ is actually the result of a filtering operation – simple convolution – applied to the block averages. Thus $\nu_{j,l}$ is constant in blocks of size $2^{(j_1 - j)}$. By exploiting this remark, all

the $\nu_{j,l}(t)$ for $l \neq 0$ can be computed simultaneously in order $O(n)$ time (rather than $n \log(n)$).

2. The vector $\nu(t)$ is, in fact, a kind of multiresolution filter bank, with $\approx (D+1)\log_2(n)$ outputs at each "time" i. Therefore, *we are searching for an optimal segmentation by applying multiresolution filters, and evaluating the entropy of the output, searching for a minimum entropy output.* Applying this idea, we present, in figure 4.1, a display of the entire filter bank output for object *Ramp*.

3. Although we started from the point of view of looking for a single segmentation point, we can instead look for several. By displaying the whole differential entropy profile $\delta\mathcal{E}(t)$ as a function of t, we may perhaps identify several points of segmentation. To illustrate this fact, we display in Figure 4.2, panel (a), the object *Blocks*, which has many edges, and in panel (b), the corresponding differential entropy profile.

4. Although the method we are discussing is order $n \log(n)$, the constants involved are worse than, say, the constants involved in the Fast Fourier transform, as order $n \log(n)$ inversions of a $2D+2$ by $2D+2$ matrix must be made. (The matrices are all the same except for two rows, however, so systematic use of the Sherman-Morrison-Woodbury formulas could improve the constants over naive inversion.)

5. Most fundamentally, the current ideas completely change our opinion of what makes sense in treating noisy data. We began with the prejudice that using SURE to select bases was the most sensible approach. This was based on the favorable experience of Donoho and Johnstone (1993) in using SURE to adaptively select parameters of a de-noising procedure. In particular, we would not have considered the use of an ℓ^2 entropy, because the noise in such entropy would seemingly swamp any signal. But from the present point of view, it is clear that minimizing the simple ℓ^2 entropy criterion involves in an essential way only quadratic forms in order $\log_2(n)$ noise variables. Therefore, the noise variance in such a criterion grows with n in a moderate way, and selecting a segmentation by minimizing the ℓ^2 entropy is no longer ruled out.

5 MES as an Edge Locator

The observation just made suggests a new multi-resolution filter bank edge locator: *find the t minimizing the ℓ^2-entropy of the segmented empirical transform.* There is of course a massive literature on edge detection and location; it is interesting to compare this *Minimum Energy Segmentation* method with existing approaches.

1. The segmented wavelet approach allows a definition of edges which is very broad, including not only step discontinuities, but also cusps, and, if high polynomial degrees D are employed in wavelet construction, discontinuities in higher derivatives. Moreover, the segmented wavelet approach allows considerable variety in behavior near the edge – a jump discontinuity needn't be a simple Heaviside; it could also be a jump with different slopes on the two sides of the jump. In contrast, many existing schemes depend on a specific shape of discontinuity (e.g. Heaviside); they may miss more subtle effects and may produce biased locations if the simple models they assume do not fit.

2. The segmented wavelet approach is based on a multi-resolution filter, whereas the traditional approaches are mono-resolution – based on filtering at one fixed scale. It is evident that if one filters at one fixed scale, a variety of tuning parameters need to be specified; if these are misspecified, the results will be poor. The segmented wavelet approach based on the ℓ^2 entropy has no tuning parameters.

3. The segmented wavelet approach seems to be near-optimal from the point of view of statistical theory. Roughly speaking, when the object contains a step discontinuity, the Minimum Energy criterion seems to give a localization of the edge at a rate approaching the $O_P(1/n)$ which statistical theory says is optimal; when the object contains a cusp discontinuity, the Minimum-Energy criterion seems to give a localization of the edge of accuracy $O_P(1/\sqrt{n})$, which statistical theory says is again optimal, and so on for other discontinuity types.

This is not the place for an extended analysis or proof of the asymptotic properties of this edge locator. However, we do offer a simple heuristic analysis which may be persuasive, and with work can be refined into a rigorous analysis. For simplicity, we argue below as if the various coefficient functionals had equal, unit, norms. A rigorous argument would allow for the fact that they do not.

Suppose that we have a function like *Ramp* with jump discontinuity at τ, and poynomial behavior on both sides (polynomial of degree D). Consider the segmented wavelet coefficients $\nu_{j,l}(t)$ for $t \neq \tau$. We say that a non-central coefficient ($l \neq 0$) is "contaminated" if the formula producing it involves using data from blocks containing the segmentation point, or on the opposite side.

For "uncontaminated" coefficients with $\ell \neq 0$, the magnitude of the coefficient $\nu_{j,l}(t)$ is $O(2^{-j(D+1/2)})$. For "contaminated" coefficients, the magnitude of the coefficient, unless a happy accident intervenes, is $O(2^{-j/2})$.

The ℓ^2 entropy is an additive measure, so we may partition the risk measure by resolution level:

$$\delta\mathcal{E}^2(t) = \sum_j \left(Q_j^\nu(t) - Q_j^\mu(t) \right),$$

with component at level j

$$Q_j^\nu = \sum_l \nu_{j,l}^2(t); \qquad Q_j^\mu = \sum_l \mu_{j,l}^2(t).$$

Roughly speaking therefore, Q_j^ν is of order

$$Q_j^\nu \approx \#\{\text{uncontaminated } \nu_{j,l}\} \cdot 2^{-j(2D+1)} + \#\{\text{contaminated } \nu_{j,l}\} 2^{-j}$$

Now, again roughly speaking

$$\#\{\text{contaminated } \nu_{j,l}\} \approx \min(2^j|t-\tau|, D/2+1)$$

and

$$\#\{\text{uncontaminated } \nu_{j,l}\} \approx D - \#\{\text{contaminated } \nu_{j,l}\}.$$

We conclude that each Q_j^ν has a "well" of order 2^{-j} wide, and a depth which is of order 2^{-j} also. Combining over all scales, we get for expected behavior that $\delta\mathcal{E}^2(t)$ has a well with sides behaving like $\asymp |t-\tau|$, for $|t-\tau| \geq 1/n$.

In case the underlying function has a cusp, "contaminated" coefficients are of size $2^{-j(3/2)}$. Repeating the above analysis, we get an expected behavior that $\mathcal{E}^2(t)$ has a well with sides behaving like $\asymp |t-\tau|^2$, for $|t-\tau| \geq 1/n$.

In general, for a discontinuity in the m-th derivative, $\mathcal{E}^2(t)$ has a well with sides behaving like $\asymp |t-t|^{m+1}$, for $|t-\tau| \geq 1/n$.

Let us now consider the noise in the objective function. Define the noise process $Z(t) = \delta\mathcal{E}^2(y^t) - E\{\delta\mathcal{E}^2(y^t)\}$. This is a continuous zero-mean stochastic process, at each t a diagonal quadratic form in $O(\log_2(n))$ random variables, each one a Gaussian with variance σ^2/n. Hence $Z(t)$ has tail probabilities bounded by a double exponential distribution with variance parameter $C(\sigma^2 \log_2(n)/n)^2$. We ignore, in this heuristic treatment, the issue of the noncentrality parameters of various quadratic forms.

Now for the minimum to occur at a certain, fixed t, it is necessary that the noise in $Z(t) - Z(\tau)$ exceed the drift $Q(t) - Q(\tau)$. The chance that the noise is smaller than some multiple of $\sigma^2 \log_2(n)/n$ is overwhelming. Therefore a given t has a non-negligible chance to be better than τ only if $Q(t) - Q(\tau)$ is smaller than some multiple of $\sigma^2 \log_2(n)/n$. This suggests that

$$Q(\hat{\tau}) - Q(\tau) = O_P(\sigma^2 \log_2(n)/n).$$

(Rigorous proof of such a relation of course requires the use of techniques from the theory of empirical processes.) Combining these relations with the fact that, in the presence of a jump discontinuity, $Q(\hat{\tau}) - Q(\tau) \asymp |\hat{\tau} - \tau|$, and that, in the presence of a cusp, $Q(\hat{\tau}) - Q(\tau) \asymp |\hat{\tau} - \tau|^2$, and one gets the following heuristic predictions.

First, the rate of convergence of the minimizer at a simple discontinuity with polynomial behavior on either side is predicted to be

$$\hat{\tau} - \tau = O_P(\sigma^2 \log_2(n)/n).$$

Second, the rate of convergence of the minimizer at a simple cusp with polynomial behavior on each side is predicted to be

$$\hat{\tau} - \tau = O_P(\sigma\sqrt{\log_2(n)/n}).$$

Third, the rate of convergence of the minimizer at an m-th order discontinuity, with polynomial behavior on each side, is predicted to be

$$\hat{\tau} - \tau = O_P((\sigma^2 \log_2(n)/n)^{1/(m+1)}),$$

provided $D > m$.

We know from asymptotic decision theory that this behavior is essentially the best one may expect. It is possible that if we knew that the discontinuity were of a certain type, say a ramp, we could invent a method which converges at a slightly faster rate – avoids the logarithm terms. But the new method makes no assumptions whatever, and achieves a near-optimal rate for the given type of singularity without advance knowledge of the type of singularity. We conjecture (based on related experience in [22]) that if one wants to adapt to an *unknown* type of singularity, the logarithm terms can not be avoided.

We carried out a small simulation experiment to assess the performance of the estimator. In the simulation, we attempted to locate the segmentation point for objects *Ramp* and *Cusp* at various signal-to-noise ratios and sample sizes.

The simulations show that the estimator had an all-or-nothing character. At sufficiently high signal-to-noise ratio, the methods give accuracy at the pixel level, while at signal-to-noise ratio below some critical threshold, the methods fail completely. There was very little evidence of continuous or gradual degradation in the estimator's quality with decreasing SNR. We did not succeed in identifying a heuristic formula which would predict the SNR at which the pixel-level resolution degraded completely.

6 Multi-Segmented Analysis

Figure 4.2 shows that, if one evaluates the differential entropy profile on an object with several discontinuities, the profile will exhibit several local minimizers. This suggests that tools for finding a single best segmentation might profitably be employed in the case of multiply-segmented objects.

The key issue in such an undertaking is that some kind of sequential unmasking is necessary. Figure 6.1 shows an object, *Bumps*, together with its differential entropy profile. Evidently, not all the bumps result in visible local minima of the entropy profile. Figure 6.2 displays its segmentation-coefficients $\nu_{j,l}$.

6.1 Sharp- and Flat-Components

Any function $f_j \in V_j^\tau$ is, in principle, smooth except at τ. Hence we can decompose the function into "potentially singular" and "certainly smooth" parts solely by location. If $K_j(\tau)$ denotes those indices k where $closure(support(\phi_{j,k}))$

contains τ, then for an $f_j \in V_j$ we may write

$$f_j = f_j^{\#,\tau} + f_j^{\flat,\tau}$$

with "potentially singular" (sharp) part

$$f_j^{\#,\tau} = \sum_{k \in K_j(\tau)} \beta_{j,k}^{\tau} \phi_{j,k}$$

and "certainly smooth" (flat) part

$$f_j^{\flat,\tau} = \sum_{k \notin K_j(\tau)} \beta_{j,k}^{\tau} \phi_{j,k}.$$

We note also that the mapping $f \to f_j^{\#,\tau}$ is a non-orthogonal projection, and similarly for $f \to f_j^{\flat,\tau}$

We can do the same sort of thing for a function d in W_j, getting non-orthogonal projection operators $S_j^{\sharp,\tau} d$ and $S_j^{\flat,\tau} d$.

If we now write $f = f_{j_0} + \sum_{j_0 \leq j < J} Q_j^t f$, we can decompose each individual term into sharp- and flat- components, producing

$$f = f^{\#,\tau} + f^{\flat,\tau},$$

where

$$f^{\#,\tau} = f_{j_0}^{\#,\tau} + \sum_{j_0 \leq j < J} S_j^{\sharp,\tau} Q_j f.$$

The function $f^{\#,\tau}$ is potentially singular at τ; and of compact support; the complementary function $f^{\flat,\tau}$ is zero at τ and also in a vicinity of width $\asymp 2^{-J}$.

To illustrate these ideas, we present in Figure 6.3 the corresponding functions $f^{\#,\tau}$ for object *Bumps*, where τ runs through the points t_i underlying the construction of the *Bumps* object.

We may think of the functions $f^{\#,\tau}$ as representing the "part of" f "explained by" any singularity at τ.

6.2 Segmentation Pursuit

The ability to identify the part of a function "explained by" a singularity at a fixed point suggests a sort of iterative cleaning operation, analagous to Friedman and Stuetzle's Projection Pursuit in statistics and Mallat's Matching Pursuit is Signal Analysis.

1. Set $r := f$ and $i := 1$.

2. Identify a point of likely segmentation via
$$t_i := \arg\,\min_t \mathcal{E}(W^t r)$$

3. Calculate $f^{\natural,t_i}$, the component of r "explained by" the segmentation.

4. Remove this component.
$$r := r - f^{\natural,t_i}$$

5. Unless satisfied, set $i := i + 1$ and go to 2.

We call this "segmentation pursuit"; it is in formal analogy with "projection pursuit" [27] and "matching pursuit" [31].

Figure 6.4 gives the result of applying segmentation pursuit to object *Bumps*; shown are the functions $f^{\#,t_i}$ extracted in the first ten iterations of the procedure. Several issues deserve comment. First, that while some of the functions extracted are indeed sharp peaks, corresponding to sharp peaks in the original object, some of the extracted objects are rather "dull" and not exactly what one expects. The reason is that the points of segmentation do not always correspond to actual singularities; the extracted components in those cases are smooth rather than peaked. Second, the method appears, in general, to leave "peaky" residuals even when peaks are being successfully extracted; this is caused by the influence of peak shapes which differ from piecewise polynomial. Figure 6.5 displays the residual vector at several stages.

Based on experience with projection pursuit [27], a variety of simple modifications to the above should also be useful, and should address the two objectionable features just seen. One example is "backfitting", where, after adding a new term into the equation, we cycle through all previous terms; on each cycle we add back to r the term under consideration, and then we locate and extract a singular component all over.

1. Set $r := f$ and $i := 1$.

2. Identify a point of likely segmentation via
$$t_i := \arg\,\min_t \mathcal{E}(W^t r)$$

3. Calculate $f^{\sharp,t_i}$, the component of r "explained by" the segmentation.

4. Remove this component.

$$r := r - f^{\sharp,t_i}$$

5. for $j := 1$ to $i - 1$, set $r := r + f^{\sharp,t_j}$, and perform the analog of steps 2 and 3, extracting an "improved" $f^{\sharp,t_j}$

6. Unless satisfied, set $i := i + 1$ and go to 2.

The idea is that we can thereby adjust the locations of segmentations to allow for improved segmentation after neighboring peaks are unmasked.

Efficient implementation of this idea requires the implementation of an efficient updating scheme for the all-segmentations algorithm. When extracting a sharp-component of f, the $\nu_{j,l}$ coefficients of the new residual r differ from the previous coefficients only in order $O(\log(n))$ positions. If we develop a method to efficiently update just those coefficients, the intrinsic computational complexity of this iterative scheme can be made quite small. We have not yet implemented this scheme and so have little experience with the method.

6.3 A Vacuum Cleaner

The computational toolkit we have assembled consists of several dozen procedures, expressed as MATLAB m-files.

The principal application we have in mind for this toolkit at the present time is the one indicated in the introduction: improving on ordinary wavelet de-noising by adapting to the presence of a few singularities.

1. Apply several iterations of Segmentation pursuit to remove edges.

2. Apply standard wavelet de-noising to the edge-less object.

3. Apply segmented wavelet de-noising to each segmentation-component.

4. Superpose the results.

Armed with a fast all-segmentations algorithm and a fast updating method for extracting sharp-components, such ideas are practical and bear further study.

7 The Horizon problem

Now we analyze the horizon problem posed in the introduction. For definiteness, consider Figure 7.1, which displays object *HalfDome*. This object is defined by the following 1.5-dimensional imaging model:

$$d_{i,k} = Ave\{f(x,y)|y \in I_{j_1,k}\} + \epsilon \cdot z_{i,k}, \qquad 0 \le i,k < 64, \quad x = i/64.$$

The object itself has the following form. With horizon function

$$h(x) = 1/4 + x(1-x), \qquad x \in [0,1],$$

we have, above the horizon, $f = s_1(x,y) = 0$, and, below the horizon, $f = s_0(x,y) = x(1-x)y(1-y)$.

The 2-dimensional wavelet transform of this object is portrayed in Figure 7.2. Figure 7.3 portrays a 1.5-dimensional wavelet transform of the object. Here the "1.5-dimensional transform" is obtained by first, applying the 1-dimensional wavelet transform along each column, then applying the 1-dimensional wavelet transform along each row. Symbolically,

$$W_{1.5}d = W_x[W_y d_x],$$

where d_x denotes a column of the data array – all the samples corresponding to a single x-value. The resulting transform has separable basis functions obtained as simple tensor products of wavelets in x and wavelets in y.

It will be noted that in both transforms the nonzero wavelet coefficients tend to cluster, so that if a certain coefficient is large, one expects that coefficients at nearby sites will also be large.

Figure 7.4 compares compression numbers of the two transforms. They are roughly similar, though the full 2-d transform offers somewhat higher accuracy at large m.

Consider now the segmented 1.5-dimensional wavelet transform, obtained by applying an ideally- segmented wavelet transform to each column, followed by a standard wavelet transform to each row.

$$W_{1.5}^h d = W_x[W_y^{h(x)} d_x],$$

Results are depicted in Figure 7.5. Note that the transform is by and large missing the long strings of correlated wavelet coefficients against a near-zero background. Figure 7.4 also compares the compression numbers of this transform with the compression numbers of the other two transforms. Obviously, the results are much better with ideal segmentation.

How to infer a horizon from data? In determining a minimum entropy segmentation for this 1.5-dimensional case, two different approaches suggest themselves.

Ignoring Horizon Cost. Here one simply applies the 1-dimensional ideas used so far on each column of the 2-d data array, without borrowing strength from any apparent relationship between neighboring columns. Simply put, one searches for the best segmentation point for each column separately; only after this segmentation is found does one make the transition to a 1.5-dimensional transform.

Enforcing Horizon Cost. Here one measures the entropy of the full transform array, and attempts to find an appropriate segmentation for optimizing this target.

As an example of the first approach, we consider the noisless case. Figure 7.6 portrays Entropy profile arrays as 2-d surfaces. Each column of the underlying array is a 1-dimensional profile of the type seen before, for a column of the corresponding *HalfDome* object. For clarity, we set positive values to zero, only negative values being important for the minimization. In all cases, we get a strong minimum near the true horizon. Figure 7.7 portrays the located horizon under each criterion, with the true horizon.

Second, we consider the noisy case. Figure 7.8 portrays the ideal risk and SURE surfaces for the noisy *HalfDome* object of Figure 7.14. Figure 7.9 portrays the ideal-risk located horizon and the MES- located horizon. Evidently, the quality of the horizon estimate has an all-or nothing character. Either the estimate is accurate even at the pixel level, or else it is wildly scattered about. This example shows that the noise level is too high and the discretization too coarse for adaptively segmented methods to radically improve things.

Finally, we display the results of de-noising with the ideally-specified horizon, in Figure 7.10.

For comparison purposes, we depict in Figure 7.11 a reconstruction by de-noising the non-segmented wavelet transform. The result is somewhat smoother looking, but it is less accurate. The mean squared reconstruction error by segmented search is 308/4096; the mean-squared reconstruction error by non-segmented transform is 812/4096. A good way to see the improvement is to compare Figures 7.12 and 7.13. These portray the reconstruction errors made by each method. Evidently, the errors made by the segmented de-noising are rather balanced in spatial distribution. On the other hand, the errors made by non-segmented de-noising are very large near the horizon discontinuity.

In some sense, the ideal segmentation has cleaned-up the structure in the large errors associated with curves, giving errors which are more spatially random. It is an interesting question whether for problems of this scale, performance of realistic algorithms can possibly approach this ideal.

8 Topics for Further Work

In this section we briefly mention some areas which are natural continuations of work done here, but which we have not pursued. We aim mainly to make the reader aware of the difficulties involved.

8.1 Multi-Segmented Multi-resolutions

In section 6 we aimed to treat multiply-segmented objects by iterative applications of operators derived from the single-segmentation case. Instead, we could have developed an Multi-Resolution Analysis based on multiple segmentations, and corresponding wavelet transforms.

The appropriate generalization of our earlier methods leads to the following.

```
Algorithm: Multi-Segmented Refinement

Inputs:
    Block averages: a[k], 0 <= k < 2^j.
    Segmentation Points: 0 = t_0 < t_1 < ... < t_M = 1.
Outputs:
    Refined Block Averages: r[k], 0 <= k < 2^(j+1)

for k=0 to 2^j-1 {
 Break_in_box = True, if a breakpoint falls inside box k.
 Nleft = Number of consecutive ‘unbroken’ boxes on left of box k.
 Nright = Number of consecutive ‘unbroken’ boxes on right of box k.

 if(!break_in_box){
    if(Nleft >= D/2 && Nright >= D/2){
         Fit polynomial of degree D to the D+1 nearest box averages
         Impute box-averages to finer scale using fitted polynomial
    } else {
         Impute as in Haar refinement
    }
 } else {
   if(Nleft >= (D+1) && NRight >= (D+1)){
         Fit left and right polynomials of degree D
           to (D+1) averages on left and right sides of break point.
         Impute box-averages to finer scale using left and right polynomials.
     } else {
         Impute as in Haar refinement
     }
 }
}
```

This refinement scheme, iterated across levels, leads to leads to a sequence of spaces $V_j^{(t_0,t_1,\ldots)}$, and refinement operators $P_j^{(t_0,t_1,\ldots)}$ much as before. However, unlike before, it will not always be the case that the spaces V_j contain all piecewise polynomials of degree D. In the fortunate case where $2^j(t_i - t_{i-1}) > (D+1)$ for each relevant i, this will be the case; that is, if $\pi^{(t_0,t_1,\ldots)}$ denotes a piecewise polynomial with knots at the t_i, we will have

$$P_j^{(t_0,t_1,\ldots)}\pi^{(t_0,t_1,\ldots)} = \pi^{(t_0,t_1,\ldots)}.$$

However, if two breakpoints fall in the same box, then no such relation will hold, in general.

It would be interesting to consider a variation of the ideas in section 6, in which, when we decide that segmentation is occuring at t, rather than removing the “sharp part” of a function, we augment the MRA by inserting another point into the segmentation list. It is at the moment unclear to the author how to search for segmentations “masked” by other, more pronounced, segmentations.

8.2 Enforcing Horizon Cost

In section 7 we ignored the cost of representing the horizon. In data-compression terminology, in an N by N image, the position of the horizon gives N free variables that need to be obtained, so unless we consider the issue, the compressed storage can never be smaller than $O(N)$, even when the object is very simple, with a very simple horizon. In statistical-estimation terminology, the horizon obtained from noisy data will itself be noisy; better horizon estimates will be obtained by imposing some sort of smoothness, which will help block the noise.

In the data compression setting, one might consider the use of an entropy involving the sum of the representation cost in the horizon-segmented transform and the representation cost of the horizon.

$$\mathcal{E}(f,h) = \mathcal{E}(W^h_{1.5}x) + \mathcal{E}(W_1 h)$$

This means that a horizon which is simple to represent, but not exactly the "true" horizon, might be preferred over the "true" horizon, if that horizon is complex and requires many coefficients to represent.

In the statistical-estimation setting, one might consider the use of two kinds of entropies. First, the unpenalized SURE method:

$$SURE(d,h) = SURE(W^h_{1.5}d);$$

second, the penalized SURE method

$$P - SURE(d,h) = SURE(W^h_{1.5}d) + \sum \min(\alpha_{j,k}(h)^2, \delta^2_{j,k});$$

Here one calculates a purely formal uncertainty $\delta_{j,k}$, for example by examining the fluctuations of the unpenalized SURE minimizer at the signal-free model. That formal uncertainty is used to penalize fluctuations in the horizon model.

Much can be said about the attractiveness of these measures; however, the computational issues involved in minimizing them are extremely thorny. It seems natural to try gradient descent from a starting horizon estimate, and to exploit the multi-resolution nature of the transform; this leads to a framework as follows:

[1] Use the Univariate Segmentation as a starting guess for $h(x)$.

[2] Expressing $h(x)$ in a standard Schauder Basis (piecewise-linear wavelets), attempt multiresolution gradient descent to improve the guess.

[2a] Express the gradient of the objective in terms of the Schauder coefficients.

[2b] Operating at coarsest scales, do a line search in the restricted gradient direction to get a better horizon.

[2c] Express the new gradient of the objective in terms of the Schauder coefficients.

[2d] Operating at finer scales, do a line search in the restricted gradient direction to get a better horizon.

This framework is vague, and does not lead to any concrete algorithms. We are skeptical that such ideas will ever lead to practical methods.

For one specific choice of entropy, however, there does seem to be an algorithm with some hope of working. Suppose the entropy applied to the two-2 array is just the L^2-energy, and that the horizontal wavelet transform W_x is orthogonal. Then by Parseval,

$$\|W_x[W_y^{h(x)}d_x]\|_2^2 = \sum_x \|W_y^{h(x)}d_x\|_2^2$$

and the above general form of entropy becomes

$$\mathcal{E}(d,h) = (\sum_x \|W_y^{h(x)}d_x\|_2^2) + \mathcal{E}(W_x h).$$

The fast-all-segmentations algorithm in dimension 1 allows us to calculate and store, in order $n^2\log(n)$ time, the surface $S(x,y) = \|W_y^{h(x)}d_x\|_2^2$ at the grid of $O(n^2)$ pixel values. In fact, surfaces of exactly this sort were presented in Figures 7.6 and 7.7. The objective function then becomes

$$\mathcal{E}(d,h) = (\sum_x S(x,h(x))) + \mathcal{E}(W_x h).$$

8.3 2-d Refinement Schemes

The computational difficulty of carrying out a fully two-d segmented refinement scheme is fairly high. In order to maintain the possibility of sub-pixel resolution, one must faithfully model the sub-pixel details of an edge in two dimensions. To see how this goes, we describe a simple two-d refinement scheme for horizons of a special type.

Assume the horizon is monotone increasing, and that we wish to refine averages at the scale of the coarse grid, producing imputed averages at the scale of the fine grid. The horizon is modelled as a piecewise linear curve with knots only at grid points.

The average-refinement paradigm we have been using in 1-d fits low-order polynomials to block averages in a neighborhood of the block to be refined. It is important that these neighborhoods themselves consist only of blocks which are unsegmented. In 2-d, with a monotone increasing phase boundary, the northwest quadrant and southwest quadrant are always unsegmented. Therefore at any block which is segmented, we propose refinement using *quarter-plane filters* as follows;

- Fit polynomials of degree D to block averages in the northwest and southeast quadrants; call these the top and bottom polynomials.
- Use these polynomials to impute averages at each of the four subblocks.

The algorithm is similar at blocks with other geometries. For example, at a block which is unsegmented, but whose immediate northern neighbor is segmented, we may use the quarter-plane filter facing southeast, with vertex (northwest corner) at the block in question.

9 Discussion

This article describes computational experiments applying the persuasive best-basis heuristic to the segmentation problem. It does not try to prove anything, but to develop the implications of the wavelet/best-basis formalism. It also represents a report on the development of a considerable body of software, which the reader may wish to obtain for experimental purposes.

The author sees two principal issues raised by this work.

1. There are many existing edge-detection schemes. A number of these are based on wavelets themselves [32, 33]. The method we have discussed here is different, in that it tries to be true to the internal logic of the wavelets-wavelet packets paradigm.

2. Fidelity to the internal logic of wavelets puts us in a kind of straight jacket. We are limited in this paper to certain methods and attitudes; this makes some questions, like how to handle fully two-d segmentation problems, seem very difficult.

The author also wishes he had the time to repeat these experiments using more stable refinement schemes. This would be his first priority for further work.

References

[1] Andersson, L., Hall, N., Jawerth, B., and Peters, G. Wavelets on closed subsets of the real line. in *Recent Advances in Wavelet Analysis*, eds. L.L. Schumaker and G. Webb. Academic Press. pp. 1-62.

[2] Antonini, M., Barlaud, M., Mathieu, P. and Daubechies, I. (1991) Image coding using wavelet transforms, *IEEE Proc. Acoustics, Speech, Signal Processing*, to appear.

[3] Beylkin, G., Coifman, R. and Rokhlin, V. (1991) Fast wavelet transforms and numerical algorithms. *Comm. Pure Appl. Math.* **43**, 141-183.

[4] Cavaretta, A.S., Dahmen, W., and Micchelli, C.A. (1991) Stationary Subdivision. *Mem. Amer. Math. Soc.* **453**

[5] Chui, C. (1991). An Introduction to Wavelets. Academic Press, N.Y.

[6] Chui, C. and Quak, E. (1992) Wavelets on a bounded interval. *Numerical Methods of Approximation Theory*, Dietrich Braess and Larry L. Schumaker, Eds. Birkhauser Verlag, Basel, pp. 1-24.

[7] Cohen, A. (1990) Thèse. Paris IX (Dauphine).

[8] Cohen, A., Daubechies, I., Feauveau, J.C. (1990). Biorthogonal bases of compactly supported wavelets. *Comm. Pure Appl. Math.* **45**, 485-560.

[9] Cohen, A., Daubechies, I., Jawerth, B., and Vial, P. (1992). Multiresolution analysis, wavelets, and fast algorithms on an interval. To appear, *Comptes Rendus Acad. Sci. Paris* (A).

[10] Coifman, R.R. and M.V. Wickerhauser (1992) Entropy-based algorithms for best-basis selection. *IEEE Trans. Info. Thry* **38**, 713-718.

[11] Daubechies, I. (1992) *Ten Lectures on Wavelets.* Philadelphia: SIAM.

[12] Daubechies, I. (1993) Two recent results: Wavelet bases for the interval and biorthogonal wavelets diagonalizing the derivative operator. in *Recent Advances in Wavelet Analysis*, eds. L.L. Schumaker and G. Webb. Academic Press. pp. 237-259.

[13] Deslauriers, G. and Dubuc, S. (1989) Symmetric iterative interpolation processes. *Constructive Approximation*, **5**, 49-68.

[14] Deng, B., B. Jawerth, G. Peters, and W. Sweldens. (1993) Wavelet Probing for Compression-based segmentation. *Proc. SPIE Symp. Math. Imaging: Wavelet Applications in Signal and Image Processing.* Proceedings of SPIE conference July 1993, San Diego.

[15] DeVore, R.A., Jawerth, B., and Lucier, B.J. (1992) Image compression through wavelet transform coding. *IEEE Trans. Info Theory.* **38**,2,719-746.

[16] DeVore, R.A. and Lucier, B.J. (1992) Fast wavelet techniques for near-optimal image processing. *Proc. IEEE Mil. Commun. Conf.* Oct. 1992. IEEE Communications Society, NY.

[17] Donoho, D.L. (1992) De-Noising via Soft-Thresholding. to appear *IEEE Trans. Info. Thry.*.

[18] Donoho, D.L. (1993a) Wavelet Shrinkage and W.V.D.: a ten-minute tour. in *Progress in Wavelet Analysis and Applications*, Y. Meyer and S. Roques, eds. Editions Frontières: Gif-sur-Yvette, pp. 109-128.

[19] Donoho, D.L. (1993b) Smooth wavelet decompositions with blocky coefficient kernels. in *Recent Advances in Wavelet Analysis*, L. Schumaker and G. Webb, eds.

[20] Donoho, D.L. (1993c) Unconditional bases are optimal bases for data compression and for statistical estimation. *Applied and Computational Harmonic Analysis* **1**, 100-115.

[21] Donoho, D.L. & Johnstone, I.M. (1992a). Ideal spatial adaptation via wavelet shrinkage. to appear *Biometrika.*

[22] Donoho, D.L. & Johnstone, I.M. (1992b). Neo-classical minimax theorems, thresholding, and adaptation. Technical Report, Department of Statistics, Stanford University.

[23] Donoho, D.L. & Johnstone, I.M. (1992c). Minimax estimation by wavelet shrinkage. to appear *Ann. Stat.*

[24] Donoho, D.L. & Johnstone, I.M. (1993). Adaoting to unknown smoothness via wavelet shrinkage. to appear *J. Amer. Stat. Assn.*.

[25] Donoho, D.L., Johnstone, I.M., Keryacharian, and Picard, D. (1993). Wavelet Shrinkage: Asymptopia? to appear, *J. Roy. Stat. Soc.* ser (B).

[26] Dubuc, S. (1986) Interpolation through an iterative scheme. *J. Math. Anal. and Appl.* **114** 185-204.

[27] Friedman, J.H. and W. Stuetzle (1981). Projection Pursuit Regression. *Journ. Amer. Stat. Assn.* **76** 817-823.

[28] Lucier, B.J. (1992) Wavelets and Image Compression. in *Mathematical Methods in CAGD and Image Processing*, T. Lyche and L.L. Schumaker, eds. pp. 1-10. Academic Press, Boston.

[29] Mallat, S. and J. Froment (1992) Second-Generation Compact Coding from Wavelet Edges. in *Wavelets: a tutorial in theory and applications* C.K. Chui, ed. Jones and Bartlett: Boston.

[30] Mallat, S. and S. Zhong (1992) Wavelet Transform Maxima and Multiscale Edges. in *Wavelets and Their Applications* M.B. Ruskai et al., eds. Academic Press: Boston.

[31] Mallat, S. and S. Zhong (1993) Matching Pursuits with Time-Frequency Dictionaries. in *IEEE Trans. Signal Proc.* **41**, 3397-3415.

[32] Moreau, E., Charbassier, G., and Lassau, J.C. (1993) Détection et localisation d'un obstacle grâce à l'utilisation d'une transformée en ondelettes. in *Progress in Wavelet Analysis and Applications*, Y. Meyer and S. Roques, eds. Editions Frontières: Paris.

[33] Serir, A. and Sansal, B. (1993) Détecteurs de contours optimaux basés sur la transformation en ondelettes. in *Progress in Wavelet Analysis and Applications*, Y. Meyer and S. Roques, eds. Editions Frontières: Paris.