# Multiscale methods for data on graphs and irregular multidimensional situations.
# University of Bristol
# Statistics Group
# Technical Report 08:07

Maarten Jansen, Guy P. Nason and Bernard W. Silverman

21st November 2006

## Summary

For regularly spaced one-dimensional data, wavelet shrinkage has proven to be a compelling method for nonparametric function estimation. We create three new multiscale methods that provide wavelet-like transforms for both data arising on graphs and for irregularly spaced spatial data in more than one dimension. The concept of scale still exists within these transforms but as a continuous quantity rather than dyadic levels. Further, we adapt recent empirical Bayesian shrinkage techniques to enable us to perform multiscale shrinkage for function estimation both on graphs and for irregular spatial data. We demonstrate that our methods perform very well when compared to several other methods for spatial regression for both real and simulated data. Although our article concentrates on multiscale shrinkage (regression) we present our new 'wavelet transforms' as generic tools intended to be the basis of methods that might benefit from a multiscale representation of data either on graphs or for irregular spatial data.

## 1 Introduction

### 1.1 Background

Over the last decade a large variety of wavelet methods have been introduced to several different areas of statistics such as curve estimation (regression, density estimation, intensity estimation, survival function estimation), time series analysis, functional data analysis, and image warping (see for example, Vidakovic (1999), Silverman and Vassilicos (2000), Percival and Walden (2000), Abramovich *et al.* (2000) for reviews). Nearly all work in the statistical area has been based on the fast discrete wavelet transform (DWT) invented by Mallat (1989). The major exception being work in statistical inverse problems which has relied on Fourier transformation and Meyer wavelets, see Johnstone *et al.* (2004) for a recent review.

Existing work in wavelet-based function estimation has typically made use of the following model and assumptions. Let $x(t)$ be some function that we are interested in for some $t$ either on

$\mathbb{R}$ or some interval $[a, b]$. Suppose $\epsilon_i$ is iid Gaussian with mean zero and constant variance $\sigma^2$. Let $t_i = i/n$. We observe

$$y_i = x_i + \epsilon_i \tag{1}$$

where $x_i = x(t_i)$, $y_i = y(t_i)$ and $i = 1, \ldots, n$. Key features of this model are that

1. the number of observations, $n$, is a power of two, say $n = 2^J$ for some $J \in \mathbb{N}$. This restriction is not too difficult to overcome even when using fast wavelet transforms.

2. the data are observed on the regular grid $t_i = i/n$. This assumption enables direct use of standard wavelet (and Fourier) discrete transforms. When data are irregularly distributed various methods, such as binning or interpolation to a regular grid, have been proposed. For example, in one dimension, Antoniadis *et al.* (1997), Hall and Turlach (1997), Cai and Brown (1999), Sardy *et al.* (1999), Kovac and Silverman (2000), Antoniadis and Fan (2001), Pensky and Vidakovic (2001), Nason (2002), and Kohler (2003).

   In two dimensions Herrick (2000) extended the interpolation method of Kovac and Silverman (2000) to two dimensions but found the resulting procedure too computationally intensive to be of any practical use.

   Recently a new "second-generation" wavelet-like paradigm called "lifting" has been developed which can handle multidimensional irregularly spaced data which arise commonly in statistics. For a quick introduction to lifting see Sweldens (1996). Lifting is the mathematical foundation of our work and it is described in more detail, with references, in Section 2.

   Adaptions of lifting to curve estimation problems in one dimension are discussed in Delouille *et al.* (2001) and Vanraes *et al.* (2002). For lifting half-regular designs (tensor product of two one-dimensional irregular designs) see Delouille and von Sachs (2002). In two-dimensions curve estimation with lifting has been tackled by Delouille (2002) and Delouille *et al.* (2003): this work and the current article both develop and build on Jansen *et al.* (2001).

3. the error distribution is iid Gaussian with zero mean and constant variance. Various authors have weakened these assumptions. For example, see Johnstone and Silverman (1997) for correlated noise; Neumann and von Sachs (1995) and Averkamp and Houdré (2003) for non-Gaussian noise.

The main advantages of using wavelets are their excellent theoretical properties, excellent empirical performance both for smooth functions and also those with discontinuities or other inhomogeneities (even when, *a priori*, it is not explicitly known whether the function is smooth or not) and very fast computational speed.

## 1.2 Our main contributions

The main contributions of our work can be summarized as follows. We introduce:

- a wavelet-like transform for data on a graph;
- wavelet-like transforms for irregularly spaced data in two- or higher-dimensional space;
- statistical methods for function estimation adapted to these new wavelet-like transforms.

Our proposed methods perform very well, they are rotationally invariant, extremely fast and memory efficient, can provide credible intervals as well as 'point estimates' through empirical Bayes and can very easily be extended to use smoother basis functions. See the end of this section for a discussion of the pros and cons of our methods compared to other techniques).

The multiscale concept is particularly powerful for data that arise on networks permitting, for the first time, the description and quantification of structure within a graph at several scales and locations simultaneously. From now we shall be solely concerned with Gaussian iid noise but several of the techniques mentioned above for generalizing the distributional assumptions could be made to work efficiently with our technique.

A key concept in many spatial regression contexts, including ours, is that of neighbourhoods. That is, given a point which other points are "close" and which are its neighbours. In one dimension, with the order relation on $\mathbb{R}$, neighbourhoods can be more straightforwardly defined. The closest points to a given point are smallest/largest point greater/less than the given point. In more than one dimension there are many possible neighbourhood concepts that could be used. Some problems come with their own neighbourhood structure. Where there is no *a priori* neighbourhood structure we use either Voronoi polygons or minimal spanning trees (MSTs) to define neighbourhoods which are utilized by a lifting technique.

We also carefully analyze the variance structure of the lifted wavelet coefficients and develop a novel Bayesian wavelet shrinkage technique which works in the absence of formal scales (for irregularly spaced data the dyadic scale concept is artificial).

### 1.3 Other methods for function estimation

As the previous section highlights one of our goals is to use our newly created lifting/wavelet transforms for function estimation. For function estimation there exists an enormous range of alternatives developed across a huge range of disciplines including many in statistics. The ones that we have considered, and compared to our methods, in writing this paper are: loess by Cleveland and Devlin (1988), triograms, see Hansen *et al.* (1998) and Koenker and Mizera (2004), locfit, see Loader (1997), thin-plate splines, see Wahba (1990) and Green and Silverman (1994), and kriging, see Cressie (1993). The latter two sets of comparisons are to be found in Heaton and Silverman (2006) the others in section 7. There are many more possibilities: for example partition models, Denison *et al.* (2002), stationary and non-stationary Gaussian processes, Gaussian Markov random fields, see Rue and Held (2005) and empirical orthogonal functions (EOFs), see Jolliffe (2002) and, for graphs, network kriging, see Chua *et al.* (2006).

Although our methods compare favourably to the first list of methods listed above our main aim is not to conduct a 'regression olympics'. As well as developing a new regression method our main goal is to introduce new multiscale algorithms (for graph and irregular data) and several of the techniques listed above could be used in conjunction with our new multiscale algorithms. For example, one might wish to construct a Gaussian Markov random field model on the 'wavelet coefficients' of a structure.

However, we do believe that our methods have a strong set of advantages:

1. our methods are fast and efficient in storage and for the multiscale part require $\mathcal{O}(n)$ operations for $n$ sites. For the Voronoi version the Voronoi tessellation can be computed in $\mathcal{O}(n \log n)$ operations, see, e.g., Fortune (1987). It is not always easy to discover the computational complexity of some of the methods listed above. However, EOFs are based on

eigenvector determination ($\mathcal{O}(n^3)$), loess is quadratic in storage and some of the above algorithms rely on variants of MCMC which do not scale well to large problems.

2. our methods are rotationally invariant. Some of the above methods are not.

3. our methods are easily extendable to smoother 'predict' and 'update' steps (see later for an explanation of these). For methods such as triograms extensions to smoother basis functions are not trivial, see Hansen *et al.* (1998). Moreover, our methods can even be further developed to adapt to local smoothness conditions by use of *adaptive* lifting, see Nunes *et al.* (2006) for this in one dimension.

4. on a range of real and simulated examples reported in Section 7, our methods work well. The examples considered include both discontinuous and smooth functions. It is reassuring that a method developed to allow for possible discontinuities in the function of interest also work well in the smoother case.

The main disadvantage is that, apart from analogies with regular wavelets, there is currently no substantial body of theory behind our methods. We discuss the reasons for this in section 8.

### 1.4 Krill intensity estimation example

Before we discuss lifting it is instructive to consider an example that existing wavelet techniques would find hard to solve and other statistical techniques, such as kriging, might find challenging.

Goss and Everson (1996) describe an experiment designed to quantify the amount and distribution of krill in the south Atlantic ocean around South Georgia. Figure 1 shows the interesting sampling design and a depiction of the detected krill density. Clearly the design is very far from being a regular grid but it *does* have a very strong structure which one might wish to take into account when performing spatial regression. For example, in some applications one might be interested in regression on the transect itself, or one might be interested in regression over the whole domain of definition excluding, presumably, the island, where it is known *a priori* that the krill intensity is zero. Indeed, the presence of structure or a hole in the data (e.g. island) would be challenging for more global multivariate regression techniques. Our techniques can take account of various kinds of structure of this sort and are applied to this data set in Section 7.1.

### 1.5 Structure of the article

Section 2 first reviews lifting and then introduces our variation on the theme: "lifting one coefficient at a time". We then elaborate by introducing the multiresolution analysis underlying our scheme for irregular spatial data and elicit the basis and dual basis functions. Section 2 closes with a description of an efficient method for computing the variance of our lifting coefficients.

Sections 3 and 4 set out two different approaches for lifting for irregularly located spatial data. Section 3 describes a version of our approach that enables lifting to be applied to a function on a graph (a network). Such a network might be constructed from, e.g. irregularly spaced data in Euclidean space or the data itself might naturally arise in the form of a network. For example, in a rail transportation network one might think of stations either as irregularly spaced points in two-dimensional space or one might think of them as nodes in a network where the edges are railway lines.

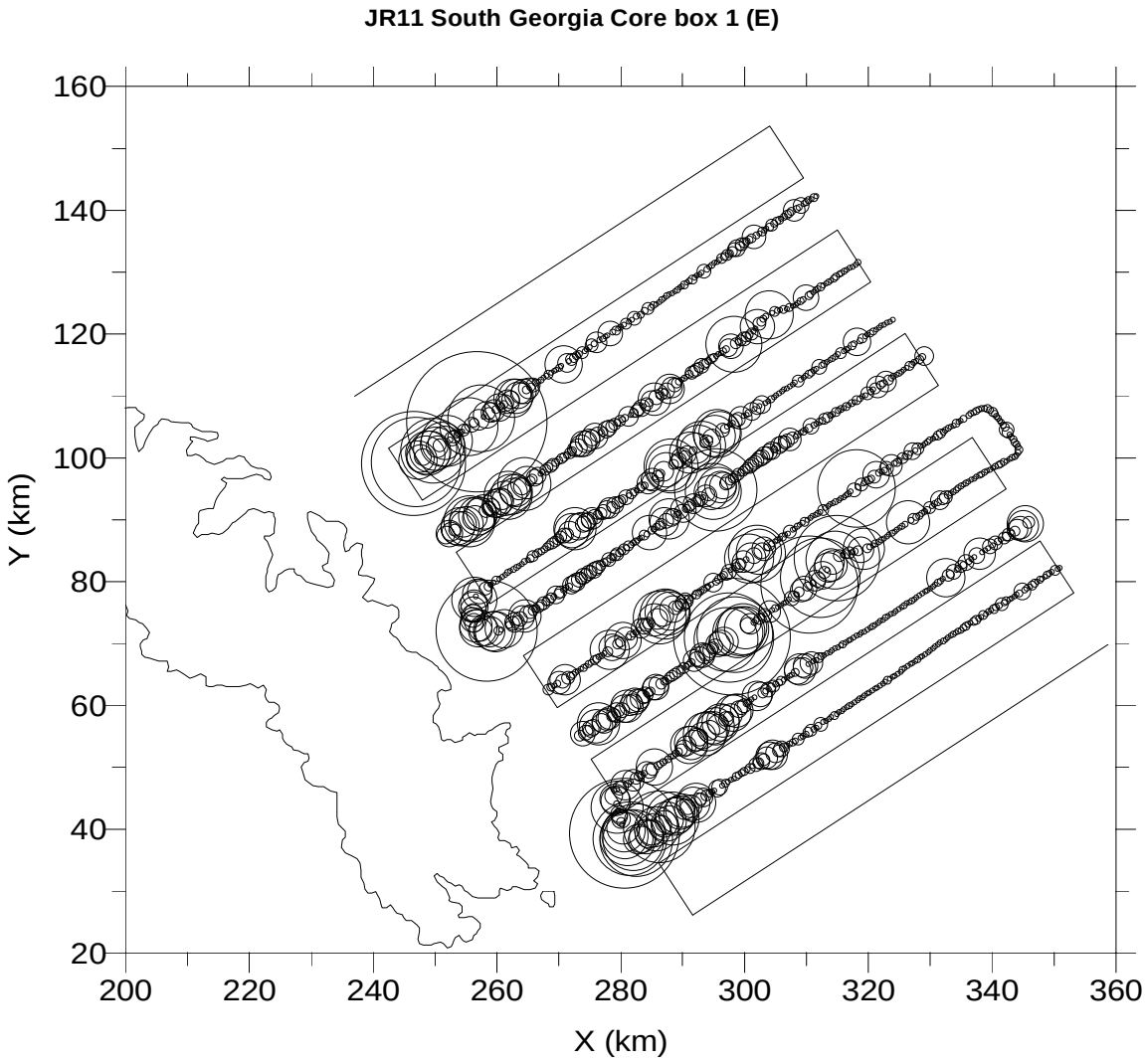


Figure 1: An example krill intensity sampling scheme. The island of South Georgia is shown in the bottom left of the plot. Each sample is indicated by a circle and the diameter which is proportional to the density of krill detected at that location. (Figure kindly supplied by Alistair Murray, British Antarctic Survey)

For irregularly spaced data in Euclidean space Section 4 uses a Dirichlet tessellation to define neighbourhoods and constructs a lifting transform using those neighbourhoods.

Successful wavelet shrinkage depends on good compression abilities of the underlying wavelet transform. We compare the compression abilities of a standard 2D DWT on a regular grid to our lifting techniques on sites with varying degrees of irregularity in Section 5.

Section 6 details the new techniques that we use to perform coefficient shrinkage on "one coefficient at a time" lifting transforms. "Scale" in lifting can be more of a continuous concept and the fixed dyadic scales of the regular DWT no longer exist in our work. We describe several empirical Bayes methods designed to work with the more general concept of scale. Section 7 contains two real life examples and summarizes several simulation studies. The real examples consider: regression of the krill data where coordinate information is used and regression of average train delay data where station coordinates are not relevant but the inter-station distances are. Finally, section 8 concludes and provides ideas for further work.

# 2 General discussion of lifting

## 2.1 The lifting approach to the standard discrete wavelet transform

Let us begin with a general specification of the lifting scheme as it has been considered previously. Given a vector $x$ of data, we divide the indices of $x$ into two subsets, denoted $I$ and $J$ for the moment. For example, in one dimension, $I$ might be the odd indices and $J$ the even. Denote by $x^I$ the vector $(x_i, i \in I)$ and $x^J$ the vector $(x_j, j \in J)$.

A single step of the lifting transform works in the following stages:

**Predict** Use $x^J$ to yield an appropriate predictor $\tilde{x}^I$ of $x^I$, and set

$$(x^I)^* = x^I - \tilde{x}^I,$$

the residual from this prediction.

**Update** Update $x^J$ by adding to $x^J$ a suitable linear transform of $(x^I)^*$.

A specific example is the Haar transform of the data. Suppose the original vector $x$ is of length 16 (for definiteness). Initially, define $I$ to be the odd indices $\{1, 3, 5, 7, 9, 11, 13, 15\}$, and $J$ to be the even indices $\{2, 4, 6, 8, 10, 12, 14, 16\}$. The prediction is carried out by estimating each odd-indexed element by the next element in the sequence, so

$$\tilde{x}_{2m-1} = x_{2m} \text{ for } m = 1, \ldots, 8.$$

Hence the modified coefficients $(x^I)^*$ are given by

$$x^*_{2m-1} = x_{2m-1} - x_{2m}.$$

These correspond to the 'detail' coefficients in the Haar transform of the data. The update step is defined by

$$x^*_{2m} = x_{2m} + \tfrac{1}{2}x^*_{2m-1} = \tfrac{1}{2}(x_{2m-1} + x_{2m})$$

so the $(x^J)^*$ represent the 'scale' coefficients at the next level, a smoothed-out version of the original data.

The lifting steps can be performed 'in place' by the two assignments

$$\begin{array}{rcl} x^I & := & x^I - x^J \\ x^J & := & x^J + \frac{1}{2}x^I \end{array} \tag{2}$$

For the next step of the Haar transform, we proceed in exactly the same way, setting $I = \{2, 6, 10, 14\}$ and $J = \{4, 8, 12, 16\}$. These correspond to the odd and even indices of the scale coefficients at the previous level. We then continue the cascade by setting $I = \{4, 12\}$ and $J = \{8, 16\}$, and for the final step $I = \{8\}$ and $J = \{16\}$. This completes the entire multiresolution analysis of the original vector $x$, and the coefficients obtained are, in a suitable order, rescaled versions of those obtained by the Mallat discrete wavelet transform. At each stage of the process, the current scale coefficients are divided into two equal sets, one of which is processed in the predict step to give the detail coefficients, and the other is updated to give the scale coefficients for the next stage.

The description we have given uses the Haar transform for simplicity, but all classical wavelet filter banks can be factored into a sequence of lifting steps, see Daubechies and Sweldens (1998).

An attractive feature of the lifting scheme is that the inverse transform can be constructed mechanically. The individual step (2) is inverted by reversing the order of the assigments and changing the signs, to give

$$\begin{array}{rcl} x^J & := & x^J - \frac{1}{2}x^I \\ x^I & := & x^I + x^J. \end{array} \tag{3}$$

To invert the whole transform, the steps are considered in the opposite order, starting with $I = \{8\}$ and $J = \{16\}$ and finishing with $I = \{1, 3, 5, 7, 9, 11, 13, 15\}$, and $J = \{2, 4, 6, 8, 10, 12, 14, 16\}$.

## 2.2 Lifting one coefficient at a time

When considering the standard wavelet transform, the sets $I$ and $J$ correspond to odd and even indices at the current level. We shall consider a different approach, where each set $I$ is just a single coefficient. The general paradigm we adopt will be as follows.

The first step is to construct an order $i_n, \ldots, i_{\ell+1}$ in which the wavelet coefficients, or their equivalents, will be obtained. Our reason for numbering in reverse order is the analogy with scale levels in the standard wavelet transform; the first coefficients to be found will be those corresponding to the finest level of detail in the function, and at the end of the process $\ell$ coefficients will remain, corresponding to the scaling coefficients at level $\ell$.

For each $i_r$, we construct, by some appropriate means, a set of $n_r$ 'neighbours' $J_r$, which may not contain any $i_s$ for $s > r$. The underlying notion is that the values $x_j$ for $j \in J_r$ may reasonably be used to construct at least an approximate prediction of $x_{i_r}$. For each $r$, our lifting transform requires the definition of two vectors $a^r$ and $b^r$, each of length $n_r$.

At each stage, the transform consists of the same two steps as previously, firstly redefining $x_i$ to be its residual from the prediction from its neighbours, and then updating the neighbour values appropriately. To avoid notational clutter, we suppress the explicit dependence on $r$ of $i$, $J$, $a$ and $b$. The step of the transform can then be written

$$\begin{array}{rcl} \textbf{Predict} \quad x_i & := & x_i - a' x^J \\ \textbf{Update} \quad x^J & := & x^J + x_i b \end{array} \tag{4}$$

Again, just as before, the inverse of this transform can be written down mechanically, by reversing the order of the steps and changing the signs:

$$\begin{array}{rcl} x^J & := & x^J - x_i b \\ x_i & := & x_i + a' x^J. \end{array} \tag{5}$$

For computational purposes, it is convenient to specify and store the transform in a standard format, as a ragged array with $n - \ell$ rows. We call this the *lifting coefficient array*. The $s$th row of the array corresponds to $r = n + 1 - s$ and consists of the sequence of $3n_r + 2$ integers

$$i_r \quad n_r \quad J^r \quad a^r \quad b^r.$$

The computational burden of the lifting scheme is the same in order of magnitude as the number of elements in the lifting coefficient array, and is certainly $O(Mn)$ where $M = \max\{n_r\}$.

In the remainder of the paper we will consider ways of constructing the lifting coefficient array, with particular attention paid to the case of spatial irregular data. Even the Haar transform as already discussed can be calculated one coefficient at a time. The order in which the indices are considered would be first the odd indices, in any order, then the indices not divisible by 4, then those not divisible by 8, and so on. In every case each index would have a single neighbour, so that $n_r = 1$, and we would have $a_r = 1$ and $b_r = \frac{1}{2}$. The neighbour $J_r$ would be, in every case, the smallest integer $j > i_r$ that is not a member of $i_{r+1}, \ldots, i_n$.

Further information on lifting in more than one dimension for data not on a lattice can be found in Daubechies *et al.* (1999). For data on a lattice see Uytterhoeven and Bultheel (1997) and Kovačević and Sweldens (2000)

## 2.3 Aspects of lifting transforms for spatial irregular data

In this section, some specific issues relevant to lifting transforms for spatial irregular data are considered, but the discussion has wider validity for methods based on neighbours in any sense.

Suppose that we have values $f_i$ of a function at $n$ points, or *sites*, $\mathbf{t}_i$ Initially, we assume that the function is approximated by an expansion of the form

$$f(\mathbf{t}) = \sum_{k=1}^{n} c_{nk} \phi_{nk}(\mathbf{t}) \tag{6}$$

where $\phi_{nk}$ are scaling functions such that

$$\phi_{nk}(\mathbf{t}_i) = \delta_{ik}, \tag{7}$$

where $\delta_{ik}$ is the Kronecker delta, at least approximately. If the scaling functions satisfy (7) exactly then the function $f$ will interpolate the values $f_i$ if we set $c_{nk} = f_k$. Denote by $I_{nk}$ the integral of $\phi_{nk}$ with respect to some suitable measure.

The stages of our procedure are numbered *downwards* from $n$, so the first stage to be carried out is stage $n$, followed by $n-1, n-2, \ldots$. At stage $r$, let $\mathcal{S}_r$ be the indices of the scaling coefficients, in other words those indices for which no wavelet coefficient has yet been calculated. Initially $\mathcal{S}_n = \{1, \ldots, n\}$. Let $\mathcal{D}_r = \{i_{r+1}, \ldots, i_n\}$, the indices of the detail coefficients already found.

We assume that we have an expression for $f$ of the form

$$f(\mathbf{t}) = \sum_{\ell \in \mathcal{D}_r} d_\ell \psi_\ell(\mathbf{t}) + \sum_{k \in \mathcal{S}_r} c_{rk} \phi_{rk}(\mathbf{t}) \tag{8}$$

where the $\psi_\ell$ are wavelet functions with zero integral, and the $\phi_{rk}$ are scaling functions at level $r$, with integral $I_{rk}$. We now set out the process whereby the various quantities, functions and sets are updated to the next stage, whereby we find an expression corresponding to (8) but with $r$ replaced by $r-1$.

Firstly, choose $i_r$ to be the value of $k$ that minimizes $I_{rk}$ over $k$ in $\mathcal{S}_r$; writing $i = i_r$, the next wavelet coefficient to be constructed is $d_{i_r}$, say. At every stage, we eliminate the scaling function with smallest integral. Set $\mathcal{S}_{r-1} = \mathcal{S}_r \setminus i_r$ and $\mathcal{D}_{r-1} = \mathcal{D}_r \cup i_r$.

Let $J_r = J$ be the set of neighbours of $i_r$ as specified in the lifting coefficient array. The specification of $J_r$ and the weight vector $a^r$ will depend on the particular lifting strategy we adopt, and will be discussed in subsequent sections of the paper. We calculate the coefficient $d_{i_r}$ in the way specified in (4), setting

$$d_{i_r} = c_{ri_r} - \sum_{j \in J_r} a_j^r c_{rj} \tag{9}$$

and, for $j$ in $J_r$,

$$c_{r-1,j} = c_{rj} + b_j d_{i_r}. \tag{10}$$

For all other $j$ in $\mathcal{S}_{r-1}$ we set $c_{r-1,j} = c_{rj}$.

If the function $f(\mathbf{t})$ is constant in the neighbourhood of the site $\mathbf{t}_{i_r}$ we would wish the wavelet coefficient to be zero, so we conduct the predict step with a set of weights satisfying $\sum a_j^r = 1$. With judicious choice of weights we can obtain a zero coefficient for locally linear functions and a near-zero coefficient for locally smooth functions, but this will be discussed below.

We next set out the way the scaling functions are updated. For any fixed $j \in J_r$, consider the special case $f(\mathbf{t}) = \phi_{r-1,j}(\mathbf{t})$. For this $f$, from (8), we have $c_{r-1,j} = 1$ and all other $c_{r-1,s}$, $s \neq j$ and $d_s$ equal 0 for $s = i_r, \ldots, i_n$. Hence, inverting the lifting steps, $c_{rj} = 1$, from (10), and $c_{ri_r} = a_j$ from (9). Therefore, by the expansion (8) for $f$,

$$\phi_{r-1,j} = \phi_{rj} + a_j^r \phi_{ri_r}. \tag{11}$$

To find the integrals of the scaling functions at the next stage, integrate (11) to obtain

$$I_{r-1,j} = I_{rj} + a_j^r I_{ri_r} \text{ for each } j \in J_r. \tag{12}$$

For $j$ in $\mathcal{S}_{r-1}$ that are not members of $J_r$, the same argument with $a_j^r = 0$ this gives $c_{rj} = c_{r-1,j}$ as well as $c_{ri_r} = 0$. This implies that $\phi_{r-1,j} = \phi_{rj}$ and $I_{r-1,j} = I_{rj}$.

To find an expression for the wavelet, we now consider $f = \psi_{i_r}$, so that $d_{i_r} = 1$ and all other coefficients at stage $r-1$ are equal to zero. From (10) we then have $c_{rj} = -b_j^r$ for $j$ in $J_r$. Equation (9) then gives $c_{ri_r} = 1 - \sum_{j \in J_r} a_j^r b_j^r$. Therefore we have

$$\begin{aligned} \psi_{i_r}(\mathbf{t}) &= (1 - \sum_{j \in J_r} a_j^r b_j^r)\phi_{ri_r}(\mathbf{t}) - \sum_{j \in J_r} b_j^r \phi_{rj}(\mathbf{t}) \\ &= \phi_{ri_r}(\mathbf{t}) - \sum_{j \in J_r} b_j^r (\phi_{rj}(\mathbf{t}) + a_j^r \phi_{ri_r}(\mathbf{t})) \\ &= \phi_{ri_r}(\mathbf{t}) - \sum_{j \in J_r} b_j^r \phi_{r-1,j}(\mathbf{t}), \end{aligned} \tag{13}$$

by substituting the expression (11).

The weights $b_j^r$ are found from the requirement that the integral of the wavelet is zero. By integrating (13), this requirement is equivalent to

$$\sum_{j \in J_r} b_j^r I_{r-1,j} = I_{ri_r}, \tag{14}$$

where the integrals $I_{r-1,j}$ have been found using (12). For reasons of numerical stability, we use the minimum norm solution of the equation (14), setting

$$b_j^r = I_{ri_r} I_{r-1,j} / \sum_{k \in J_r} I_{r-1,k}^2. \tag{15}$$

Note that within the process it is never necessary to express the wavelets or scaling functions explicitly, but the integrals of the scaling functions are used to choose the coefficient $i_r$ and to specify the weight vector $b^r$. Therefore, in order to start the process off, it is necessary to specify the integrals $I_{nj}$ of the original scaling functions. Apart from these integrals, we also need appropriate ways of choosing the vectors $J^r$ and $a^r$ of neighbours and prediction weights at each stage. We shall consider two particular approaches in detail later in the paper, the first based on Voronoi polygons and the second on minimal spanning trees.

Finally, there are circumstances within which it is helpful to have a notion of the scale of each wavelet function. A convenient measure of this scale for the wavelet $\psi_i$ for $i_r$ is the integral $I_{ri_r}$ of the scaling function for site $i_r$ at the last stage before $i_r$ is removed from future consideration. We denote this scale by $\alpha_{i_r}$. In the natural neighbour method described later, $\alpha_{i_r}$ will be the area of the last Voronoi cell based on site $i_r$. In general, for any fixed $r$, and assuming all the weights $a_j \geq 0$ we have

$$\alpha_j = I_{r-1,i_{r-1}} \geq I_{r,i_{r-1}} \geq I_{r,i_r} = \alpha_i$$

and so the scales $\alpha_i$ are a monotonic function of the index $r$ and the order in which the lifting scheme determines the coefficients.

## 2.4 The dual basis functions

The lifting procedure can be thought of in two separate ways. On the one hand, if we have a function $f$ of the form (6), the expansion (8) gives an expression of $f$ in terms of a multiresolution basis, where effects of different scales are captured by different wavelet coefficients. On the other hand, an alternative way of thinking is to consider the lifting scheme as a linear tranformation of a vector of *values* $x$, yielding a coefficient vector $\tilde{x}$, say, whose elements have a multiresolution interpretation. In either case the relation between the original function or data, and the derived coefficients, can be elucidated by investigating the dual basis functions or vectors.

Both cases can be covered by considering suitable inner products $\langle \cdot, \cdot \rangle$. In the 'function' case, the natural inner product between functions $g_1$ and $g_2$ is an integral of the form

$$\langle g_1, g_2 \rangle = \int_\Omega g_1(\mathbf{t}) g_2(\mathbf{t}) d\mathbf{t} \tag{16}$$

over some fixed bounded region $\Omega$. In the 'vector' case, given vectors $x$ and $y$ of values at the data points, the standard inner product $\langle x, y \rangle = \sum_i x_i y_i$ can be used. If $x$ and $y$ are the values of

functions $g_1$ and $g_2$ at the data sites $\mathbf{t}_i$, then we can equivalently use the inner product

$$\langle g_1, g_2 \rangle = \sum_i g_1(\mathbf{t}_i) g_2(\mathbf{t}_i). \tag{17}$$

Suppose we have an expansion of $f$ of the form (8) in terms of the basis $\phi_r$ made up of the functions $\psi_\ell$ for $\ell$ in $\mathcal{D}_r$ and $\phi_{rk}$ for $k$ in $\mathcal{S}_r$. Let $c_r$ be the corresponding vector of coefficients $c_{rk}$ and $d_\ell$. We can set out the derivation of a dual basis $\phi_r^* = \{\psi_\ell^*, \phi_{rk}^*\}$ having the properties

$$d_\ell = \langle \psi_\ell^*, f \rangle \text{ for } \ell \text{ in } \mathcal{D}_r, \qquad \text{and} \qquad c_{r,k} = \langle \phi_{rk}^*, f \rangle \text{ for } k \text{ in } \mathcal{S}_r, \tag{18}$$

which may be written in vector form as $c_r = \langle \phi_r^*, f \rangle$.

The interest of the dual basis functions is that they give the weight functions that are applied to $f$ or to $x$ to yield the corresponding coefficients. In this sense they make explicit the contribution of values of $f$ or $x$ at various points to particular wavelet coefficients $d_\ell$ and scaling coefficients $c_{rk}$.

The dual basis functions are constructed inductively. Suppose, we have constructed the functions $\{\psi_\ell^*, \ell \in \mathcal{D}_r\}$ and $\{\phi_{rk}^*, r \in \mathcal{S}_r\}$. To construct the functions at stage $r-1$, we use exactly the updates in the lifting scheme itself, with $i = i_r$ and $J = J_r$, setting

$$\psi_i^* = \phi_{ri}^* - \sum_{j \in J} a_j \phi_{rj}^* \tag{19}$$

and, for $j$ in $J$,

$$\phi_{r-1,j}^* = \phi_{rj}^* + b_j \psi_i^*, \tag{20}$$

with $\phi_{r-1,k}^* = \phi_{rk}^*$ for all other $j$ in $\mathcal{S}_{r-1}$.

To see why these relations hold, let $L^r$ be the matrix corresponding to the lifting step that yields the vector $c^{r-1}$ from $c^r$. Suppose, as an inductive hypothesis, that the conditions (18) hold. Then

$$c_{r-1} = L^r c_r = L^r \langle \phi_r^*, f \rangle = \langle L^r \phi_r^*, f \rangle = \langle \phi_{r-1}^*, f \rangle,$$

as required. Therefore, as long as the original vector of dual basis functions or vectors $\phi_n^*$ satisfies $c_n = \langle \phi_n^*, f \rangle$, the lifting scheme will produce the required dual basis functions at every stage.

To obtain the values at the data sites of dual basis functions relative to the vector inner product (17) we start the process with vectors $\phi_{nj}^*$ with elements $(\phi_{nj}^*)_i = \delta_{ji}$. To find dual basis functions relative to the function inner product (16), it is necessary to find a suitable initial dual basis. For example, if the initial basis functions are constant over non-overlapping regions, then an initial dual basis will be given by $\phi_{ni}^* = \phi_{ni} / \int (\phi_{ni})^2$.

## 2.5 The variance of the sample coefficients

In this section, we set out an approach, which operates in $O(Mn)$ time and storage, for finding, approximately, the variance of each wavelet and scaling coefficient as obtained by a lifting scheme. Of course, because the lifting scheme operates linearly, for reasonably small data sets it is possible to calculate the full covariance matrix of the coefficients by successively carrying out on the covariance matrix the row and column operations corresponding to the lifting steps. This is a much more burdensome calculation, requiring $O(Mn)$ vector operations on vectors of length $n$, but makes it possible to evaluate the usefulness of the approximate method.

Suppose that the original data $x_k$ are independent random variables with variances $V_k$. Consider a single lifting step of the form (4), writing $x^*$ for the values after the lifting has taken place. Since $x_i^* = x_i - \sum_{j \in J} a_j x_j$, we have

$$\text{var } x_i^* = V_i + \sum_{j \in J} a_j^2 V_j \tag{21}$$

and

$$\text{cov}(x_i^*, x_j) = -a_j V_j. \tag{22}$$

Since $x_j^* = x_j + x_i^* b_j$, it follows that

$$\text{var } x_j^* = V_j + b_j^2 \text{var } x_i^* + 2 b_j \text{cov}(x_i^*, x_j) = (1 - 2 a_j b_j) V_j + b_j^2 \text{var } x_i^*. \tag{23}$$

It follows that the effect of a single lifting step is to replace the variances by $V_k^*$, where

$$\begin{array}{rcl} V_i^* & = & V_i + \sum_{j \in J} a_j^2 V_j \\ V_j^* & = & (1 - 2 a_j b_j) V_j + b_j^2 V_i^* \text{ for } j \in J. \end{array} \tag{24}$$

The approximation we use is to neglect any correlations between the coefficients obtained at the next stage, but simply to iterate the calculations (24). This will yield an algorithm essentially of the same complexity as the lifting algorithm itself, and indeed that can similarly be carried out in place. Some experiments on lifting arrays obtained from Voronoi polygons, in the way discussed later in the paper, demonstrate that only a little accuracy is lost, mostly in the large-scale wavelet coefficients and in the final scaling function coefficients, which tend to have small variance anyway.

In some practical situations the assumption of independent $x_k$ variables is not tenable. Such a situation is beyond the scope of the present paper. However, we can envisage prior or estimated information on the covariance structure can be fed into the calculation of the coefficients' variance along the lines of methods used for regular wavelet shrinkage such as Kovac and Silverman (2000).

# 3 A lifting scheme for graphs

We introduce a lifting scheme that essentially provides a kind of 'wavelet transform on a network'. Here we mean a 'network' to be a 'function on a graph'. We consider our graphs to have arisen in one of two ways. One way is that the graph is supplied to us predefined — for example a transportation network or communications network. The other way is that data is supplied in a form that can be converted into a network. For example, irregularly spaced data in $K$-dimensional space on which a graph can be induced by calculating interpoint distances and constructing, say, a minimal spanning tree. We elaborate on these next.

## 3.1 Minimal spanning trees and other tree-based approaches

Consider data observed at an irregular set of points in $K$ dimensions, for some $K > 2$. For data sets in two dimensions, approaches based on Voronoi cells are attractive, but in higher dimensions they become both computationally infeasible and philosophically inappropriate. The number of Voronoi neighbours of each point will typically be large and the computations will become burdensome.

In this section we consider an alternative lifting approach based on trees, and in principle any tree can be used as the basis of our scheme. In the case of $K$-dimensional data, useful trees are

those that reflect the neighbourhood structure of the points. The notion is that trees, not Voronoi polygons, are used to incorporate the "neighbourhood" structure of the data at each point of the lifting scheme. If the original data sites $\mathbf{t}_i$ lie in a $K$-dimensional Euclidean space, a natural approach is to use *minimal spanning trees* (MST), see e.g. Krzanowski and Marriott (1995), which are easily computed. Other types of tree may be useful for particular applications, and these would be a possible topic for future work.

There are some data sets where the data themselves naturally live on a tree rather than in a Euclidean space. For example, the data collection transects for the krill data depicted in Figure 1 constitute a tree. More generally, we can extend our "lifting on a tree" to more general graphs as long as there is a suitable neighbourhood structure. For example, in protein modelling, a tree could be defined by the chemical bonds in a large molecule. In this case, wherever it is necessary to determine distances between points, it may be appropriate to use distances in the original tree or graph.

For data that live naturally on a network (graph) our methods effectively provide a kind of 'wavelet transform on a network'. By restricting the analysis to a narrow range of scales our methodology provides a kind of 'coarse Fourier transform' of a function on a network (in the same way that a single scale level of wavelet coefficients acts as a bandpass filter isolating information about a function around a narrow range of frequencies).

See Smola and Kondor (2003) and Belkin *et al.* (2004) for work on regularization of functions on graphs.

### 3.2 General aspects of the tree-based lifting scheme

The first step in the lifting scheme as set out in Section 2.3 was to specify the initial scaling functions $\phi_{nk}$ and to find their integrals. In the tree context, we define the scaling function $\phi_{ni}$ to be 1 at the node $i$ and zero at all other nodes of the tree. At each stage of our process, we consider the scaling functions and wavelets as being defined on the original nodes. We define a set of weights $w_i$ and then define the 'integral' of any function having value $f_i$ at node $i$ as the weighted sum $\sum_i w_i f(i)$. In order to relate the weights to the tree on which we are working, we define $w_i$ to be the sum of the lengths of the edges from the node $i$ to its immediate neighbours. We arbitrarily use the sum of the lengths but the average of the lengths is another possibility that we have used.

At each stage $r$, we calculate the wavelet coefficient corresponding to the node $i$ with the smallest current value of $I_{ri}$. Letting $J$ be the set of current neighbours of $i$, we have to define a suitable set of weights $a$. We may either let $J$ be the immediate neighbours within the tree, or we may include second- or even higher-order neighbours in the set $J$.

Once the set $J$ is defined, we need to define the prediction weight vector $a$. For reasons explained below, we mostly use *inverse distance prediction weights*, setting $a_{ij} = c\delta_{ij}^{-1}$, where $\delta_{ij}$ is the distance from point $i$ to point $j$, and $c$ is chosen so the weights sum to 1. In the extreme case where $J$ contains only one index $j$, the value at node $j$ is used as the predictor at node $i$.

Alternatively, in some circumstances, e.g. the krill data, the nodes do possess *bona fide* Euclidean coordinates. In which case the tree can be used to define the neighbours but the coordinates are used by least squares to form prediction weights. To distinguish between these two variants we refer to them either as "tree with inverse distances weights" or "tree with least squares coordinate weights". As an example of these two algorithms in action see Figure 4.

Having defined the weight vector $a$, we can update the integrals using equation (12), and calculate the update weights $b_j$ using the equation (15).

The final step is to update the neighbourhood structure. We shall assume that as a point $i$ is eliminated from consideration, the spanning tree is modified locally, only changing the linkage structure between points previously linked directly to $i$. If the point $i$ to be removed has immediate neighbours $j_1, \ldots, j_m$, say, then we replace the links between $i$ and the $j_k$ by the links of the minimum spanning tree of the points indexed by $j_1, \ldots, j_m$. This procedure maintains the tree structure of the pattern of links between points under current consideration.

How many orders of neighbours should be used in the prediction part of the lifting scheme? "Mixed scale" points cause minor practical problems for our method based on Voronoi tessellations, mostly near the boundaries. They are the source of the long and thin Delaunay triangles that we discuss, with some solutions to the resulting problems, in Section 4.3.

On average points in a tree have fewer neighbours than those from a Voronoi tessellation. For example, compare the Voronoi mosaic for the krill data in Figure 2 (right) with the ship track in Figure 4 (bottom left). This can be made precise: there are $(n-1)$ edges in a tree constructed on $n$ points so the average number of neighbours for a point in a tree is $2(1-\frac{1}{n})$ irrespective of dimension or distribution of the points, or the method of construction of the tree. For Voronoi tessellations the average number of neighbours is higher, nearer 6 in two-dimensions for moderate numbers of points (see Penrose (1996) and Penrose and Yukich (2003)). In a tree, therefore, if only immediate neighbours are considered in $J$ there is less opportunity for "mixed scales" to occur. On the other hand we may wish to include higher-order neighbours in $J$ in order to obtain better predictions. If it is decided to use higher-order neighbours, one could either use neighbours up to a given order, or one could increase the order of the neighbours until the size of $J$ reached a certain level.

Finally, our algorithm is not just restricted to trees. The same steps can be followed for any general graph where distances and integrals can be sensibly defined. For example, with the UK rail network, see section 7.2.

### 3.3 Why use inverse-distance prediction weights?

In this section, we explore a correspondence between inverse distance prediction weights and local linear prediction. Suppose we are working on a tree, that we are predicting the value at a point $i$, and that $J = \{j_1, j_2, \ldots, j_r\}$ for some $r \geq 2$. Work on the philosophy that a tree is defined only by its linkage structure and the lengths $\delta_{ij}$ of its edges. We consider a particular Euclidean embedding of the tree near the point $i$.

Define $r$ unit vectors $\mathbf{u}_j$ in $(r-1)$-space to be as far as possible from one another on the unit sphere, so that the end points of the $\mathbf{u}_j$ form a line segment, equilateral triangle, regular tetrahedron, or higher-dimensional regular simplex, in all cases centred at the origin. We will then have $\sum_{j \in J} \mathbf{u}_j = 0$. Now place the vertex $i$ at the origin, and place vertex $j$ at $\delta_{ij}\mathbf{u}_j$ for $j \in J$. In the case where there are two neighbours, this is simply placing $i$ on a straight line between its two neighbours. More generally, this corresponds to arranging the edges around vertex $i$ to be as far as possible in different directions.

Given values $y_j$ at vertex $j$ for each $j$ in $J$, define the linear function $L(\mathbf{t}) = a'\mathbf{t} + b$ in $(r-1)$-space to be the interpolant of the values $y_j$ at the points $\delta_{ij}\mathbf{u}_j$; the graph of this function will be the unique hyperplane through the $r$ points $(\delta_{ij}\mathbf{u}_j, y_j)$ in $r$-space. Define $y^*$ to be the value obtained

by inverse-distance weighting the values $y_j$. We now have, setting $c$ such that $c\sum_j \delta_{ij}^{-1} = 1$,

$$\begin{aligned} y^* &= c\sum_{j\in J} \delta_{ij}^{-1} y_j = c\sum_{j\in J} \delta_{ij}^{-1} L(\delta_{ij}\mathbf{u}_j) \\ &= c\sum_{j\in J} \delta_{ij}^{-1}(\delta_{ij}a'\mathbf{u}_j + b) = ca'\sum_{j\in J}\mathbf{u}_j + b = b = L(0). \end{aligned}$$

It follows that, with this particular embedding of the tree in Euclidean space, the linear interpolant at the vertex $i$ to the values $y_j$ at the vertices $j$ is precisely the inverse-distance weighted average $y^*$.

# 4 A lifting scheme based on Voronoi polygons

In this section we consider a lifting scheme for spatial irregular data based around Voronoi polygons and Delaunay triangulations. The basic idea is to construct, at each stage, a triangulation of the data sites. The neighbours of any site are then the sites joined to that site by edges within the triangulation. Once a detail coefficient corresponding to a particular site has been found, the triangulation is appropriately modified to remove that site.

## 4.1 Voronoi polygons, Delaunay triangulations and Dirichlet tessellations

Consider a set of sites in the plane. Let $\Omega$ be a suitable region in the plane containing all the sites under consideration. The region $\Omega$ may, for example, be the whole plane, or a suitable rectangle, or the convex hull of the sites. Comments about the precise choice of $\Omega$ will be made later. The *Voronoi cell* of any particular site is the set of points in $\Omega$ nearer to that site than to any other. Because the boundaries of each cell are all perpendicular bisectors of lines joining two sites, the Voronoi cells are polygons, and the *Dirichlet tessellation* is the partition of the $\Omega$ into these polygons. See Figure 2 for an example. Two sites are neighbours if their Voronoi cells have a boundary in common, and the joins of all pairs of neighbours forms the *Delaunay triangulation*. There are algorithms for finding the Delaunay triangulation in the first place, and for updating the triangulation when a site is removed. For further detailed information see Okabe *et al.* (1992); for more information on these methods in statistics see Herrmann *et al.* (1995) or Allard and Fraley (1997) for example.

At each stage of the lifting scheme, the neighbours $J$ of a site $i$ under consideration are the neighbours of $i$ within the current Delaunay triangulation, and the values at these neighbours are used in the predict and update steps. More sophisticated prediction methods could be based on higher order neighbours.

The paradigm set out in Section 2.3 requires two more ingredients, the integrals of the initial scaling functions $\phi_{nk}$, and a method of specifying the prediction weights $a^r$ at each stage. Provided $\Omega$ is a finite region, a natural definition of the initial scaling function $\phi_{nk}$ is the indicator function of the Voronoi cell of the site $\mathbf{t}_k$, and so the integral of the scaling function is the area of this Voronoi cell.

We consider two main methods of prediction, the *natural neighbour* method as proposed by Sibson (1981), and local least squares.

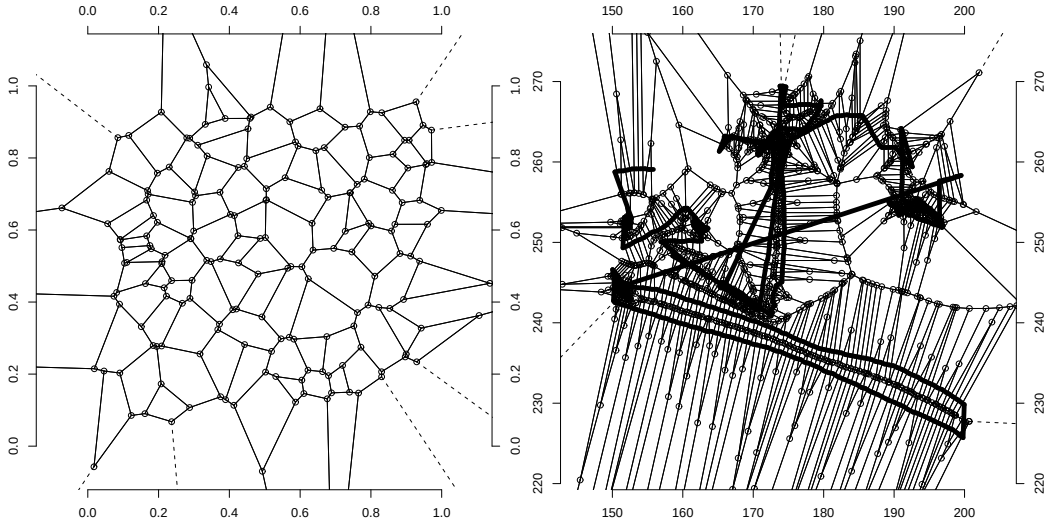


Figure 2: Left: Voronoi mosaic of 100 points uniformly distributed in $[0,1]^2$. Right: Voronoi mosaic of krill data portion used in later examples, e.g. Figure 4. The thick line indicates the track of the ship. Both plots constructed using the R package `tripack`.

## 4.2 Natural neighbour interpolation

If site $i$ is removed and the Dirichlet tessellation recomputed, the Voronoi cell of that site will be divided among its neighbours. Assume the region $\Omega$ is finite. Let $A_i$ be the cell corresponding to site $i$ and let $A_{ij}$ be the part of the cell made up of points whose next nearest site, after $i$, is the site $j$. If site $i$ is removed, then $A_{ij}$ will form part of the new cell of site $j$. If $j$ is not a neighbour of $i$ then $A_{ij}$ will be empty.

The lifting scheme using natural neighbour interpolation works by setting $a_j = |A_{ij}|/|A_i|$ for each neighbour $j$ of $i$, where $|\cdot|$ denotes area. Provided the cell $A_i$ does not intersect the boundary of $\Omega$, the prediction weights thus obtained through natural neighbour interpolation will predict a constant or linear function perfectly, and have other attractive regularity, continuity and stability properties. A corollary of the perfect prediction of linear functions is that if a function is linear, then its wavelet coefficients will be zero except for possible boundary effects. If the function is approximately linear in the region of the site $\mathbf{t}_i$ and its neighbours $\{\mathbf{t}_j : j \in J\}$, then the linear prediction based on the neighbours will be quite good and so the wavelet coefficient will be small. Another good property is that the scheme is *interpolating*; if the site $\mathbf{t}_i$ is very close to one of its neighbours $\mathbf{t}_j$ then the prediction at site $\mathbf{t}_i$ will be close to the value at site $\mathbf{t}_j$, and will tend to this value in the limit as site $\mathbf{t}_i$ coincides with site $\mathbf{t}_j$.

One disadvantage of the natural neighbour method is its computational intensity, though the method does remain linear in the number of sites.

## 4.3 Local least squares prediction

A computationally simpler approach to prediction uses local least squares. A least squares plane is fitted to the values at the sites $\mathbf{t}_j$ for $j$ in $J$, and used to interpolate at the site $\mathbf{t}_i$. This scheme

obviously has the property that if the function $f$ is linear over the site $\mathbf{t}_i$ and its neighbours, then the wavelet coefficient is zero. Therefore it shares some of the good properties of the natural neighbour method.

There are, however, some numerical and conceptual issues with the local least squares method which require careful attention. For example, unlike the natural neighbour method, the local least squares method is not interpolating. The residuals from the least squares plane through the values at the sites with indices $J$ will not, in general, be zero. Therefore, even if the site $\mathbf{t}_i$ is very close to one of its neighbours, the predicted value will not necessarily be close to the value at that neighbour, and more distant neighbours will still have a relatively heavy impact on the prediction. This is in contrast with the natural neighbour method, where more distant neighbours are automatically downweighted in the prediction, because of having small values of $|A_{ij}|$. In the local least squares approach it is desirable to avoid neighbour configurations with a mixture of short and long edges, because these rise to neighbour relationships between sites that are a long way apart on the scale currently being considered. Because distant neighbours will influence the prediction, for a smooth function the magnitude of a wavelet coefficient at a site will be affected by the distance to its furthest neighbour, and so the method may have worse compression properties than the natural neighbour approach. Triangles which are very far from equilateral are particularly likely to occur near the boundary, where two fairly distant sites may still have Voronoi cells that touch one another, particularly if the boundary of $\Omega$ is some distance from the actual boundary of the data. This behaviour can be seen in the right hand plot in Figure 2.

One way of dealing with this issue is to remove from the triangulation those narrow triangles with two vertices on the boundary where the opposite angle is obtuse. This corresponds to re-defining $\Omega$ to be the convex hull of the sites under current consideration, so that sites will only be considered to be neighbours if their Voronoi cells touch within the convex hull. A more relaxed policy could allow obtuse triangles, but only up to 120 degrees, say. In any event, the approach may need some modification at the corners of the configuration, where the approach described may leave sites with a single neighbour, and in this case it may be appropriate to re-introduce narrow triangles.

A related matter is the treatment of sites lying some distance from the remainder of the configuration, so that the angle subtended by all the site's neighbours is quite small. In this case, prediction is more like extrapolation, and can be quite unstable. A good, if fairly ad hoc, way of dealing with this is to project both the site $\mathbf{t}_i$ and the set of neighbours $\{\mathbf{t}_j : j \in J\}$ onto the first principal component direction of the set $\{\mathbf{t}_j : j \in J\}$. This is equivalent to using a least squares fitting plane constrained to have gradient in this direction. Especially in this case, the raw local linear least squares weights may fall outside the range $[0, 1]$, though it will be only in rather pathological cases that this will happen in the modified method. The natural neighbour approach cannot suffer from this instability because its weights are necessarily in $[0, 1]$.

### 4.4 Conclusions and further comparisons

Whichever method is used, it is necessary to retriangulate the configuration each time a site is removed. If the natural neighbour method is used, then the Dirichlet tessellation within the region $\Omega$ will be needed for the next stage, though of course only the cells neighbouring the site $i_r$ will have to be modified. It is conceivably possible to modify $\Omega$ at each stage but there is not usually any particular point in doing so. Overall, the natural neighbour method is more stable and more

elegant, but at a considerable computational cost which is usually not warranted.

# 5 Compression

Figure 3 illustrates the varying compression performances for two different 2D multiscale methods (Voronoi lifting and regular 2D Daubechies wavelets). The plots are constructed as follows: an equally spaced $16 \times 16$ grid is constructed; 2D analogues of the *Blocks*, *Bumps*, *Heavisine* and *Doppler* signals from Donoho and Johnstone (1994) as well as a piecewise linear function (called $\mathrm{mfa}$) are evaluated on the grid on $[0,1] \times [0,1]$ (these analogues are defined in Nason *et al.* (2004) and illustrated in this article in Figure 9); a wavelet or lifting transform is performed; then a certain number of the largest coefficients are retained, the rest are set to zero; the inverse transform is applied; the error between the inversion and the original is computed. For the transforms we have also added varying amounts of jitter: to each $x$ and $y$ grid coordinate a uniform random variable on the interval $[-\eta, \eta]$ is added for three values of $\eta$: 0.1, 0.01, and 0.001: the results are medians over 100 simulations with different jittered grids (only the smallest and largest jitter is shown for clarity).

Good compression is about having the smallest error for a given number of coefficients removed. Generally speaking the Voronoi method has much better compression abilities than the graph-based lifting. This is not surprising as Voronoi makes much more use of neighourhood information that graph-based lifting (although remember that tree-based lifting can even be used when only inter-point distance information is present).

Figure 3 shows the compression performance for the 2D Daubechies wavelet with two vanishing moments (both for jittered and unjittered values). Readers acquainted with the excellent compression properties of 1D wavelets may be surprised at the poor looking compression performance of the discrete wavelet transform (DWT). The fact that wavelets do not compress 2D images particularly well is known and has spurred the field of multiscale geometric image processing, see Starck *et al.* (2000) for example. The lifting methods seem to do particularly well then for the $\mathrm{mfa}$ and *Heavisine* function, maybe not too bad for *Bumps* but less well for *Doppler* and *Blocks*.

For large jitter (0.1, code C for Daubechies, and dashed line for Voronoi) the Voronoi lifting has better compression abilities. This is not really surprising as the jittered values, when transferred back to a regular grid, mean that the Daubechies wavelets are trying to compress a very irregular function. However, the point here is that compression performance is much better for Voronoi which is designed to take account of the irregularity. In the case of the $\mathrm{mfa}$ function, *Bumps* and *Doppler* our performance is somewhat better and in these cases it is somewhat more surprising that the compression performance of our methods with jitter is better than even regular wavelets with no jitter.

With small, or no, jitter, regular wavelets perform better than Voronoi lifting on *Blocks* and *Heavisine*. The latter signal is mostly very smooth, the former is blocky and the Haar wavelets adapt extremely well.

We have also drawn similar plots but retaining more coefficients and the conclusions are broadly the same. The overall conclusion we draw from this plot is that for data that are reasonably irregular our Voronoi lifting methods are no worse than Daubechies wavelets, and sometimes much better. In reality it would not, in any case, be possible to use Daubechies wavelets as one would not know how to map irregular data back to a grid and/or one might not have the correct number of points

unless the data were obviously only slightly jittered. However, it is somewhat reassuring to learn that our methods have compression abilities broadly in line with regular wavelets.

# 6 Bayesian shrinkage

Now consider the following model of observations subject to noise:

$$Z_i = f(\mathbf{t}_i) + \epsilon_i, \tag{25}$$

where the noise $\epsilon_i$ are independent $N(0, \sigma_i^2)$ random variables. The grid locations are irregular but considered fixed for the purposes of the analysis. Wavelet based smoothing algorithms estimate $f$ by taking an appropriate wavelet transform, modifying the coefficients in order to reduce noise, and finally inverse transforming the updated coefficients. Because of the notion that the wavelet transform of the unknown function is likely to be in some sense 'economical', some form of thresholding or shrinkage procedure is used to process the observed coefficients. Soft and hard thresholding are the best known thresholding methods, but more sophisticated shrinking may follow (among others) from a Bayesian analysis of the noisy coefficients.

## 6.1 Prior model and posterior density

The essence of the thresholding problem is the following. Suppose we have a parameter $\theta$ and an observation $Z \sim N(\theta, 1)$. In the wavelet smoothing case, $\theta$ would be an individual coefficient rescaled so that the empirical coefficient had unit variance. Following papers such as, Clyde *et al.* (1998), Abramovich *et al.* (1998) and Johnstone and Silverman (2004) the assumption that $\theta$ is a coefficient from an economical expansion is modelled by using a mixture prior for $\theta$ of the form

$$\theta \sim (1 - \pi)\delta_0 + \pi\gamma \tag{26}$$

where $\gamma$ is a symmetric density.

Johnstone and Silverman (2004) explore the advantages of using a heavy-tailed density for $\gamma$, such as the density

$$\gamma(u) = (2\pi)^{-1/2}\{1 - |u|\tilde{\Phi}(|u|)/\phi(u)\} \tag{27}$$

where $\tilde{\Phi}(u)$ is the upper tail probability of the standard normal distribution. This density has tails that decay as $u^{-2}$, the same weight as those of the Cauchy distribution. For this reason we refer to the density (27) as the *quasi-Cauchy* density.

Suppose $\theta \sim (1 - \pi)\delta_0 + \pi\gamma$ and $Z \sim N(\theta, 1)$. Johnstone and Silverman (2004) set out details of the calculation of the posterior density $f(\theta|Z)$ and also of the marginal density $f(Z) = \int\{(1 - \pi)\delta_0(u) + \pi\gamma(u)\}\phi(z - u)du$.

## 6.2 Bayesian decision rule: posterior median

Once we have the expression for the posterior density $f_{\theta|Z}$, we have various choices of possible point estimates of $\theta$. The posterior mean

$$\hat{\theta} = E(\theta|Z = z)$$

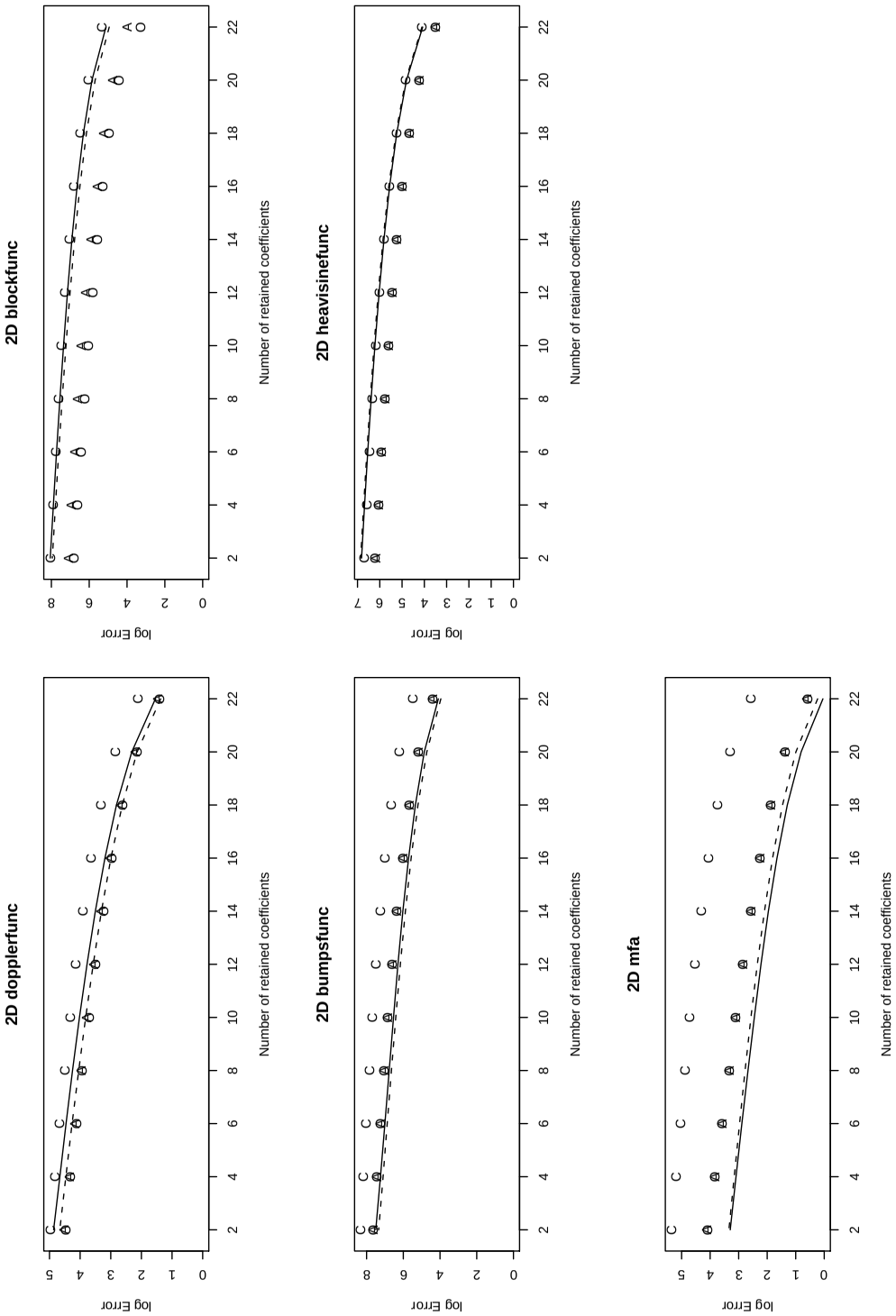


Figure 3: Compression abilities of Voronoi lifting and regular 2D Daubechies wavelets. Key: solid line: Voronoi lifting with jitter=0.1; dashed line: Voronoi lifting with jitter=0.001. O=Daubechies wavelet with no jitter; A=Daubechies wavelet with jitter=0.001; C=Daubechies wavelet with jitter=0.1. For all plots Daubechies extremal phase wavelet with 2 vanishing moments was used apart from *Blockfunc* where we used Haar wavelets.

is a popular decision rule but it lacks the thresholding property. Unless $Z = 0$ the estimate will be non-zero, which does not accord with the notion that the coefficient may well be zero. An alternative is the posterior median $\tilde{\theta}(z)$, satisfying $\tilde{F}_{\theta|Z=z}(\tilde{\theta}) = 0.5$. With the quasi-Cauchy distribution for $\gamma$, this leads to a tractable expression for $\tilde{\theta}(z)$ in terms of the standard normal distribution function and its inverse. See Johnstone and Silverman (2005a) for details and for a computer implementation.

The posterior median rule is a strict thresholding rule, with the property that, for any given $\pi$, there is a threshold $\tau(\pi)$ such that $\tilde{\theta}(z) = 0$ if and only if $|z| \leq \tau(\pi)$. An alternative to the use of the full posterior median is to use hard or soft thresholding with threshold $\tau(\pi)$. The smaller the probability $\pi$ the larger the threshold $\tau(\pi)$, and the choice of prior probability $\pi$ that $\theta \neq 0$ corresponds to the choice of threshold. It is this choice that we consider next.

### 6.3 Estimating the parameters (MLE)

Suppose that we have sequence $\theta_i$ of coefficients and a sequence of observations $Z_i \sim N(\theta_i, 1)$, for $i = 1, 2, \ldots, n$. Suppose, initially, that the $\theta_i$ have independent prior distributions (26) all with the same value of $\pi$ and that the observations $Z_i$ are themselves independent conditional on the $\theta_i$. Let $g$ be the convolution of $\gamma$ with the standard Normal density, so that the marginal density of the $Z_i$ is $(1 - \pi)\phi(z) + \pi g(z)$. Johnstone and Silverman (2004, 2005b) explore attractive practical and theoretical features of a marginal maximum likelihood approach to the choice of $\pi$, where $\pi$ is chosen to maximize the log likelihood

$$\ell(\pi) = \sum_i \log\{(1 - \pi)\phi(z_i) + \pi g(z_i)\}.$$

This procedure is an empirical Bayes approach. First of all, the whole data set is used to estimate the parameter $\pi$. The estimated value is then used as a prior probability in the model (26) and the inference carried out for each coefficient separately. For theoretical and practical reasons, the maximisation is usually carried out over a range of $\pi$ bounded below at a point corresponding to the threshold taking the 'universal threshold' value $\sqrt{2 \log n}$.

In the case of a classical orthogonal wavelet estimate, the coefficients are arranged into levels, and it is appropriate for the probability $\pi$ to be constant within levels but to be allowed to vary between levels. To this end, each level of the transform is treated separately by the marginal maximum likelihood method, and an estimated parameter $\pi_j$ is obtained for each level $j$. Typically, the parameter decreases as the resolution increases. At the levels of the transform corresponding to fine-scale effects, the prior probability $\pi_j$ is small and an observed coefficient has to pass a high threshold in order not to yield an estimate of zero. At the coarser-scale levels, a smaller threshold will usually be appropriate.

In the lifting case, for example, the division into 'dyadic' levels is no longer appropriate, and instead one of a number of other possible approaches can be pursued. Overall, it can be assumed that the prior used for coefficient $\theta_i$ has probability $\pi_i$ of being nonzero. The criterion for choosing the $\pi_i$ is still the maximization of the marginal log likelihood

$$\ell(\pi_1, \ldots, \pi_n) = \sum_i \log\{(1 - \pi_i)\phi(z_i) + \pi_i g(z_i)\}.$$

but subject to appropriate constraints on the parameters $\pi_i$.

Various different possibilities arise, for example

**Parametric dependence** The coefficients are constrained to belong to a particular low-dimensional parametric family. For example, for the lifting scheme one might constrain $\pi_i$ to be proportional to the scale $\alpha_i$, or perhaps to some power $\alpha_i^\lambda$. This accords with the notion that there are singularities of some sort in the underlying function. If the singularities are points, $\alpha_i$ is proportional to the probability of the wavelet encountering one of these singularities. For line singularities a more appropriate model for this probability is $\alpha_i^{1/2}$, and so on for spaces of singularities of different fractal dimension.

**Artificial levels** This approach is an adaptation of the dyadic structure of the standard discrete wavelet transform. One splits up the coefficients into levels in some arbitrary way, and one possibility is simply to impose an artificial dyadic split, with the highest level containing the half of the coefficients with finest scale, and subsequently lower levels successively one-quarter, one-eighth, and so on of the total number of coefficients in the order defined by the lifting scheme. An alternative is to group the coefficients taking account of the values of their pseudo-scales. For example, if $\alpha_0$ is the median scale of the coefficients, then levels could be defined with coefficients with scales in ranges $(2^j\alpha_0, 2^{j-1}\alpha_0]$ for $j \geq 1$, with the highest level consisting of all those coefficients with scales up to and including $\alpha_0$.

**Parametric dependence within artificial levels** The simplest approach using artificial levels is to constrain $\pi_i$ to be constant within levels. An alternative is to allow a parametric dependence, for example $\pi_i$ proportional to $\alpha_i^{1/2}$, with a constant of proportionality that is allowed to depend on the level. Finally, whatever method is chosen, it may be appropriate to smooth or interpolate the estimated $\pi_i$.

**Monotone dependence** Conceptually the simplest constraint on the $\pi_i$ would be to require only that $\pi_i$ increases as the individual scale $\alpha_i$ increases. Because of the convexity properties of the log likelihood function, estimation of $\pi_i$ subject to this constraint can be carried out using an iteratively reweighted least squares isotone regression algorithm. Part of the standard theory of least squares isotone regression is a convexity argument showing that the least squares isotone regression function is piecewise constant. The same argument shows that the resulting estimated $\pi_i$ are also piecewise constant functions of the scales $\alpha_i$, and so this method indirectly splits the coefficients up into levels, with constant $\pi_i$ within each level. Further details are available from Johnstone and Silverman (2005b). See Figure 4 (Bottom right) for an example of using such an algorithm.

The calculations for maximizing the log likelihood are easily set out. Define

$$\beta(w) = \{g(w) - \phi(w)\}/\phi(w) = w^{-2}(e^{w^2/2} - 1) - 1.$$

Then, by simple calculus, we have

$$\frac{\partial \ell}{\partial \pi_i} = \frac{\beta(z_i)}{1 + \pi_i \beta(z_i)}$$

which is a decreasing function of $\pi_i$. Obviously we always constrain $\pi_i \leq 1$. In addition, to avoid excessively high thresholds, and in line with the theory developed in Johnstone and Silverman (2004), we impose a lower limit on $\pi_i$ corresponding approximately to a threshold value equal

to the universal threshold $\sqrt{2\log n}$. For simplicity, we choose the lower limit $\pi_{lo}$ to satisfy the condition

$$P(\theta_i = 0 | z_i = \sqrt{2\log n}) = 1/2$$

which is equivalent to setting

$$\pi_{lo}^{-1} = 1 + (n-1)/(2\log n).$$

Details of the algorithms used to make the constrained maximum likelihood choice of the $\pi_i$ for the parametric and monotone dependence cases are set out in Johnstone and Silverman (2005a).

### 6.4 Parametric dependence within artificial levels

Full details of the parametric dependence algorithm can be found in Johnstone and Silverman (2005a). We consider the modifications necessary to adapt the procedure to the artificial levels case for lifting.

**General setup:** Suppose we have data $z_i$ for $i = 1, \ldots, n$, and consider the basic model $\pi_i = c_i\zeta$ where $c_i$ are known constants. In order to enforce the constraints $\pi_{lo} \leq \pi_i \leq 1$ we refine this to

$$\pi_i(\zeta) = \text{median}\{\pi_{lo}, c_i\zeta, 1\}. \tag{28}$$

Letting $g$ be the convolution of $\gamma$ with $\phi$, the marginal log likelihood function is then given by

$$\ell(\zeta) = \sum_i \log[\{1 - \pi_i(\zeta)\}\phi(z_i) + \pi_i(\zeta)g(z_i)] \tag{29}$$

By the definition of $\pi_i$ there is no loss of generality in considering $\zeta$ only over the interval

$$[\pi_{lo}(\max c_i)^{-1}, (\min c_i)^{-1}] = [\zeta_{lo}, \zeta_{hi}],$$

say. If $\zeta < \zeta_{lo}$ then all the $\pi_i$ will be $\pi_{lo}$ and if $\zeta > \zeta_{hi}$ the all the $\pi_i$ will be 1, regardless of how far outside the interval $\zeta$ lies.

**For artifical levels:** All of the artificial levels cases reduce to the same general form. Within a particular level $\mathcal{L}$, we have (28), where $c_i$ are known constants such as 1 or $\alpha_i^{1/2}$, and $\zeta$ is a parameter to be estimated. The likelihood, $\ell_{\mathcal{L}}$, for the level $\mathcal{L}$ is now (29) but where the sum is now over $i \in \mathcal{L}$.

In the straightforward artificial levels case, all the $c_i = 1$, and $\ell_{\mathcal{L}}$ is a concave function of $\zeta$ in $[\pi_{lo}, 1]$. We have

$$\ell'_{\mathcal{L}}(\zeta) = \sum_{i \in \mathcal{L}} \beta(z_i)/\{1 + \zeta\beta(z_i)\},$$

a decreasing function of $\zeta$. By checking the signs of $\ell'_{\mathcal{L}}(\zeta)$ at the ends of the range it can be discovered whether $\ell_{\mathcal{L}}(\zeta)$ has its maximum at one end or the other; if not, a binary search on the decreasing function $\ell'_{\mathcal{L}}(\zeta)$ will find the maximum likelihood estimate.

If the $c_i$ are not all the same, then we apply the 'parametric dependence' approach within each artificial level as described in Johnstone and Silverman (2005a).

# 7 Examples and comparisons

## 7.1 Multiscale lifting for krill data

*Background.* Goss and Everson (1996) report that as a by-product of a fish stock assessment study an opportunity was taken to estimate the biomass of Antarctic krill on the South Georgia shelf by the British Antarctic Survey (BAS). Goss and Everson (1996) state that krill biomass determination is important because they are basic part of the "food web". Krill are consumed by large numbers of birds, mammals and fish but it is also increasingly being harvested for both human and animal consumption. As well as potential over-fishing krill stocks are also under pressure from a variety of other sources such as sea temperature rise or increased UV penetration of sea water.

Since the study was a by-product of another study the sampling points took little account of the expected distribution of krill. Indeed, stations were selected for the fish abundance study and the shortest overall track was selected that visited all of the sampling stations. Figure 1 shows a selection from the transects and the sample values of krill taken along it. Figure 4 shows a different portion of the krill data subjected to regression analyses using lifting with trees using both least squares coordinate and inverse distance weights. Figure 6 shows estimates obtained using Voronoi lifting.

*Fitting.* For all of the regression estimates a small proportion of small negative values were replaced by zero. In all estimates a lot of the original zero data values have been replaced by very small intensity values. In Figure 4 it is interesting to note the differences between the two estimates around the [175km, 262km] location. The estimate based on the MST estimates some "lumps" of intensity, whereas the one based on the ships track estimates small intensities following the ships path. There are at least two reasons for these differences: (i) the ships track only uses neighbours from the previous and next sample in the track whereas the MST algorithm will use nearest neighbours irrespective of the track; (ii) the total time that the ship takes to cover points in the region (within a $25\text{km}^2$ box centred on [175, 262]) is approximately 12 hours and the ship crosses near to the centre about 5 times and the actual krill density over this time may change.

With regards to the second point if the density field of a system is subject to rapid change then maybe the estimate that follows the ship's track would be more reliable. Otherwise, if the field is slowly changing then estimates that take more account of geographical spread, like the MST, or even Voronoi might be more appropriate.

*Model Verification.* Let us take the MST lifted using least squares coordinate weights analysis further. The estimate from this procedure is shown in the top right of Figure 4. We examined the residuals from the fit and discovered that the residuals were approximately normally distributed (both by inspecting a histogram and through a Kolmogorov-Smirnov test $p$-value of 0.18) with a standard deviation of about 11.4. The variance of the residuals appears remarkably constant over the plane. All of this indicates a very good fit to model (1).

*Comparisons.* Our results are in direct contast to results generated by *loess* and the MATLAB 'triogram' function. Both of these methods did not deal with the 'clumpiness' of the krill data at all well. Both methods smoothed out some features and missed others completely. Hence the residuals also did not look satisfactory either. These results concur with our simulated comparisons in section 7.3 below.

*Physical Interpretation.* Figure 5 shows the piecewise constant thresholds which are derived from the piecewise constant weight estimates $\pi_i$ arising from the monotone dependence constraints

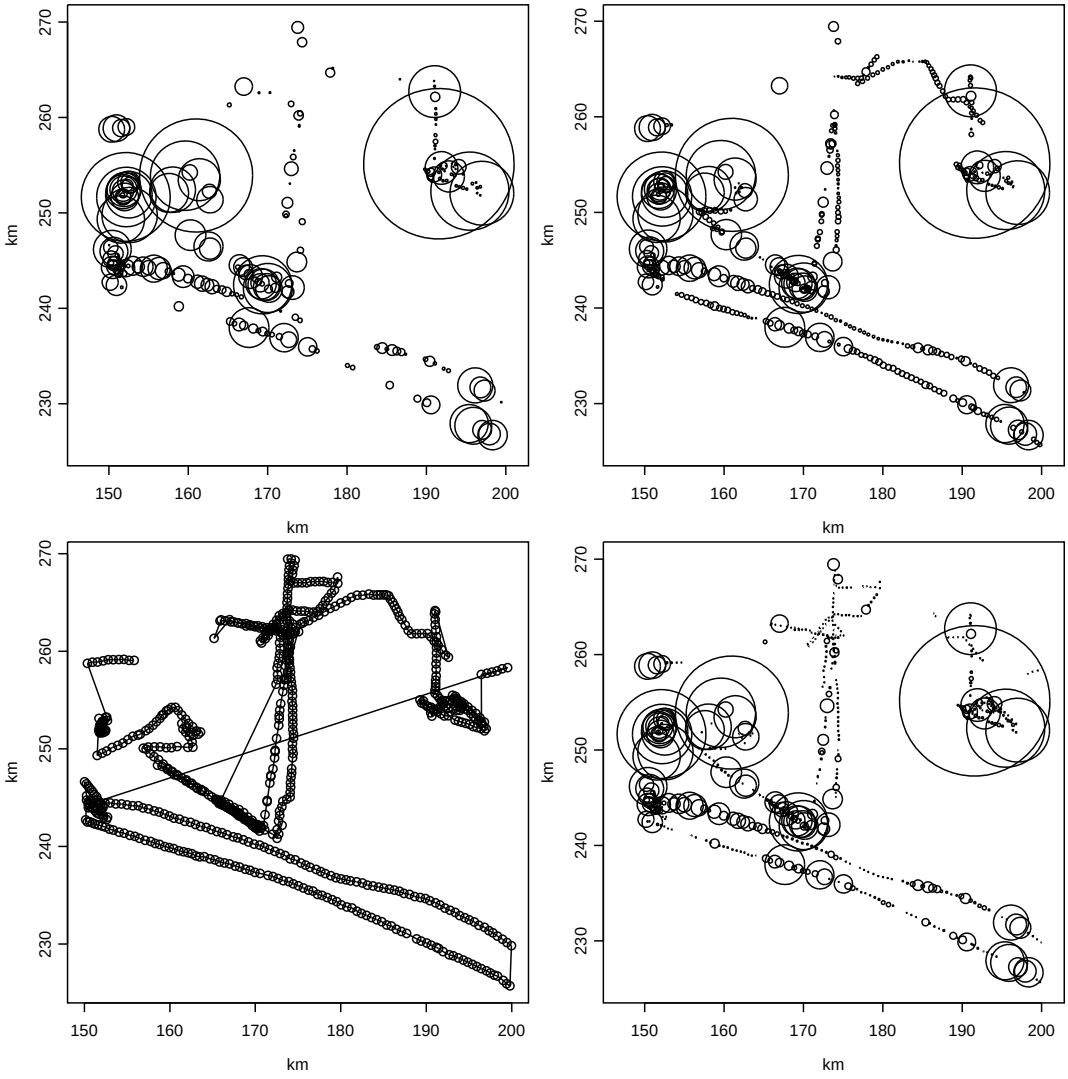


Figure 4: Analyses of selected portion of krill data set. Circle radius (not Bottom Left) encodes square root of krill density estimate in $\mathrm{gm}^{-2}$: largest value is $14981\mathrm{gm}^{-2}$. Top left: krill density supplied by BAS. Top right: MST lifted estimate with least squares coordinate weights and eBayesThresh applied to lifting coefficients at all scales. Bottom left: circles indicate krill sample locations, line indicates tree determined by ship transect. Bottom right: ship-determined transect tree lifted estimate using inverse distance weights and eBayesThresh with monotone dependence of $\pi_i$.

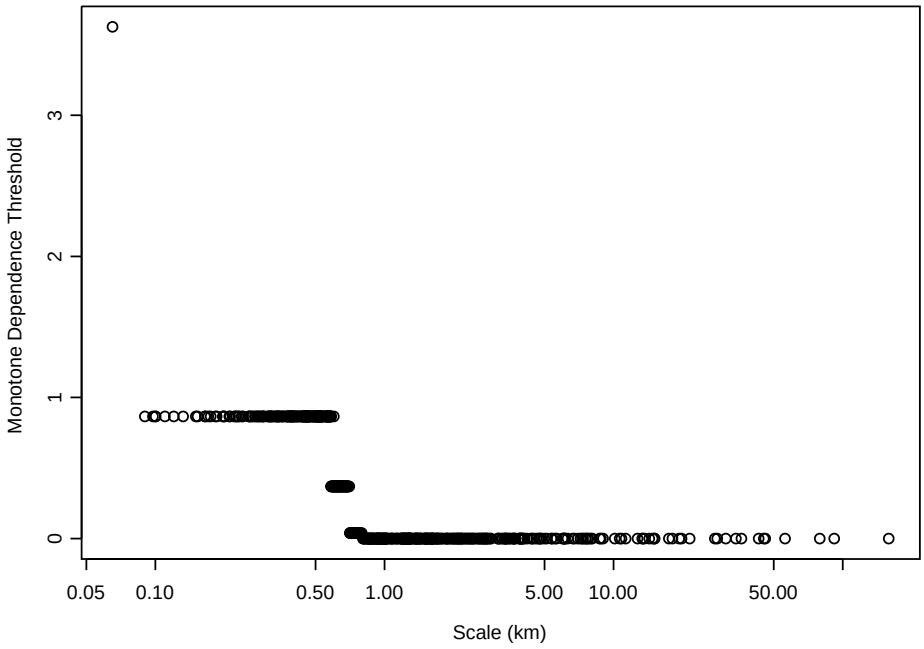


Figure 5: Piecewise constant thresholds arising from the monotone dependence constraints when maximizing the marginal maximum likelihood for $\pi_i$ for the Krill data fit at the bottom right of Figure 4. Large thresholds are applied to finer scales.

that are applied to likelihood maximization described in Section 6.3. The figure is particularly interesting as the piecewise constant functions implicitly divide the scale space into a number of *data-defined resolution levels*. (For those familiar with regular wavelet methods Figure 5 is an example of level-dependent thresholding but where the resolution levels are not fixed dyadic but arise from, and depend on, the data). The smallest threshold value is approximately $4.6 \times 10^{-9}$ for the coarsest 345 coefficients. This means wavelet coefficients in scale ranges from 0.8km and up are essentially not thresholded. Another way of interpreting this, familiar to wavelet shrinkage researchers, is to say that 0.8km is the "primary resolution". Finer scales than this get monotonically higher thresholds in bands $[0.71, 0.8)$, $[0.58, 0.71)$, $[0.09, 0.58)$ and less than 0.09. Figure 5 and these bands statistically indicate that these is little or no variation in the 'true' intensity pattern at less than 100m and there is reduced variation at less than 600m. This information could be then cross-referenced with individual clusters of wavelet coefficients to provide estimated information about particular cluster groupings and locations.

In summary, we obtain information in terms of the estimate *but also* information on the variation of the 'true' intensity via the thresholds.

Finally, the krill data distribution does not look particularly Gaussian. Figure 6 shows two more estimates using Voronoi based lifting with and without the log transformation. In future the Haar-Fisz transform, see Fryzlewicz and Nason (2004) or Jansen (2006) might be used.

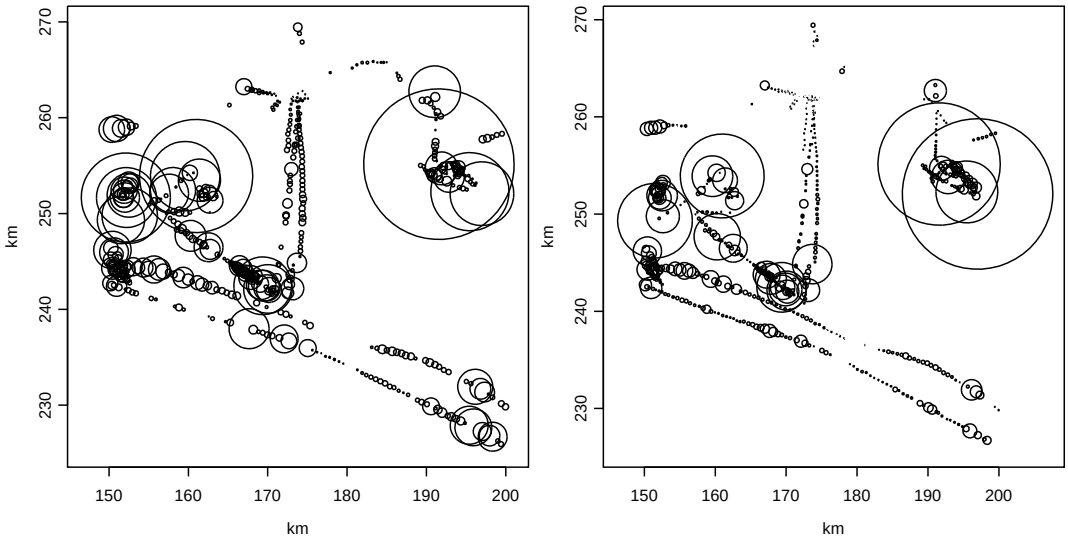


Figure 6: Krill density estimates computed using Voronoi least-squares lifting with regular eBayesThresh. Left: estimate on raw data; Right: estimate on log transformed data.

## 7.2 Multiscale lifting for rail network delay data

Figure 7 (top) shows a portion of the UK railway network concentrating on main lines in the south-west of England and South Wales. The station locations are not geographical positions but are a 2D projection of a classical multidimensional scaling solution computed from distances obtained from a list of rail routes and inter-station distances compiled by Butler (1999). In this example the graph, which is not a tree, arises as an integral part of the data.

For each train scheduled to arrive at a station Network Rail reports its estimated arrival time. We assume that the reported delays are the actual delays subjected to additive error. The error is due to a number of factors including the discrete nature of train monitoring points, trains making up time, further delays occuring. Figure 8 (top) shows the average delay for trains arriving into each station at 1655 on 30th July 2004. Stations *Bristol Temple Meads, Westbury, Gloucester, Exeter* and *Newton Abbott* have the largest average delays. Clearly there are many other kinds of network similar to this one for which it is useful to estimate the actual delays or other statistics.

Figure 8 (bottom) depicts the estimated average delay at each station using network lifting with inverse distance weights and the monotone dependence selection of $\{\pi_i\}$ as described in section 6.3. Figure 7 (bottom) shows the residuals from this fit.

A picture (not shown) of the lifting (wavelet) coefficients of the raw data in Figure 7 tends to characterise discontinuities in a similar way to, e.g., the 2D regular wavelet transform often highlights edges in images. So, for example, in Figure 7 *Bristol Temple Meads* has a very high average delay but the surrounding stations have a low value and hence a spatial discontinuity exists at this location. On a picture of the lifting coefficients those coefficients immediately surrounding (but not including) *Bristol Temple Meads* form a circular ridge that characterises the 'edge' between the large delay at *Temple Meads* and the small delays in surrounding stations.

There are many other interesting tasks that one might consider using this new 'wavelet transform' for graphs rather than using the raw data on the nodes. In some situations it might be advanta-

Table 1: Median (MAD) of 100 simulated sums of squares error values for `loess`, Tree based lifting (`pictree`) using coordinate information, and Voronoi based lifting (`liftvorLS`). Jitter $\eta = 0.01$, SNR=5, $n_g = 16^2$, monotone dependence EBayesThresh, ($\times 1000$).

| Signal | Loess | | Tree | | Voronoi | |
|---|---|---|---|---|---|---|
| mfc | 18 | (1.6) | 75 | (46) | 26 | (4) |
| *Doppler* | 130 | (5.9) | 35 | (26) | 8 | (1.0) |
| *Heavisine* | 530 | (49) | 410 | (200) | 72 | (20) |
| *Blocks* | 2300 | (53) | 190 | (91) | 160 | (37) |
| *Bumps* | 3000 | (160) | 770 | (500) | 210 | (32) |

geous for information to be represented using the lifting coefficients scale-location characterisation rather than on the nodes directly. For example, propagating information throughout a network at different scales, or forecasting future network behaviour.

It would be inappropriate to replace the given rail network graph with one calculated, say, from just the inter-station distances since the computed edges might not correspond to actual rail lines. Moreover, it does not really make sense to ask questions about the behaviour of the underlying function *over a region*. For example, it makes no sense to ask questions about the average delay at a location where no station exists. Compare this to the krill data set where one can ask about the density of krill in a location in the sea but not on the ship transect. However, it might make sense to ask hypothetical questions about planned stations that might come to be. We discuss later the methods of Heaton and Silverman (2006) that would permit this prediction to be achieved.

## 7.3 Comparisons

### 7.3.1 Comparing our lifting methods with themselves and `loess`

We carried out a large simulation study with our new methods and compared them to `loess` a well-known statistical smoothing method using the `R` implementation (see Cleveland and Devlin (1988) for more information on `loess`, see R Development Core Team (2005) for R). We evaluated these methods on 2D analogues of the *Blocks*, *Bumps*, *Heavisine* and *Doppler* test functions introduced by Donoho and Johnstone (1994) and the piecewise linear function `mfc`. Pictures of the test functions appear in Figure 9. Full mathematical definitions of these functions along with comprehensive simulation results appear in Nason *et al.* (2004).

Every simulation run was based on estimating one of the test functions on a jittered $16 \times 16$ grid and adding iid Gaussian noise. Varying amounts of jitter (distributed as Unif$[-\eta, \eta]$ for $\eta = 0.1, 0.01, 0.001$, varying signal-to-noise ratios. Sensitivity to "primary resolution" (the number of points that get removed in the lifting transform) was also explored. We also explored the performance of our different ways of carrying out our MLE as described in Section 6.3.

Table 1 shows a selection of results from Nason *et al.* (2004). One can see that for the very simple piecewise linear function mfc the loess procedure does very well, but the Voronoi lifting is not far behind. For all other signals the lifting procedures do better or much better. However, note that the performance for the tree based lifting is highly variable (large MAD values) this is because of the fewer neighbours it uses in constructing neighbours. The excellent performance of the Voronoi based lifting is seen throughout all simulations. Primary resolution does not appear to dramatically

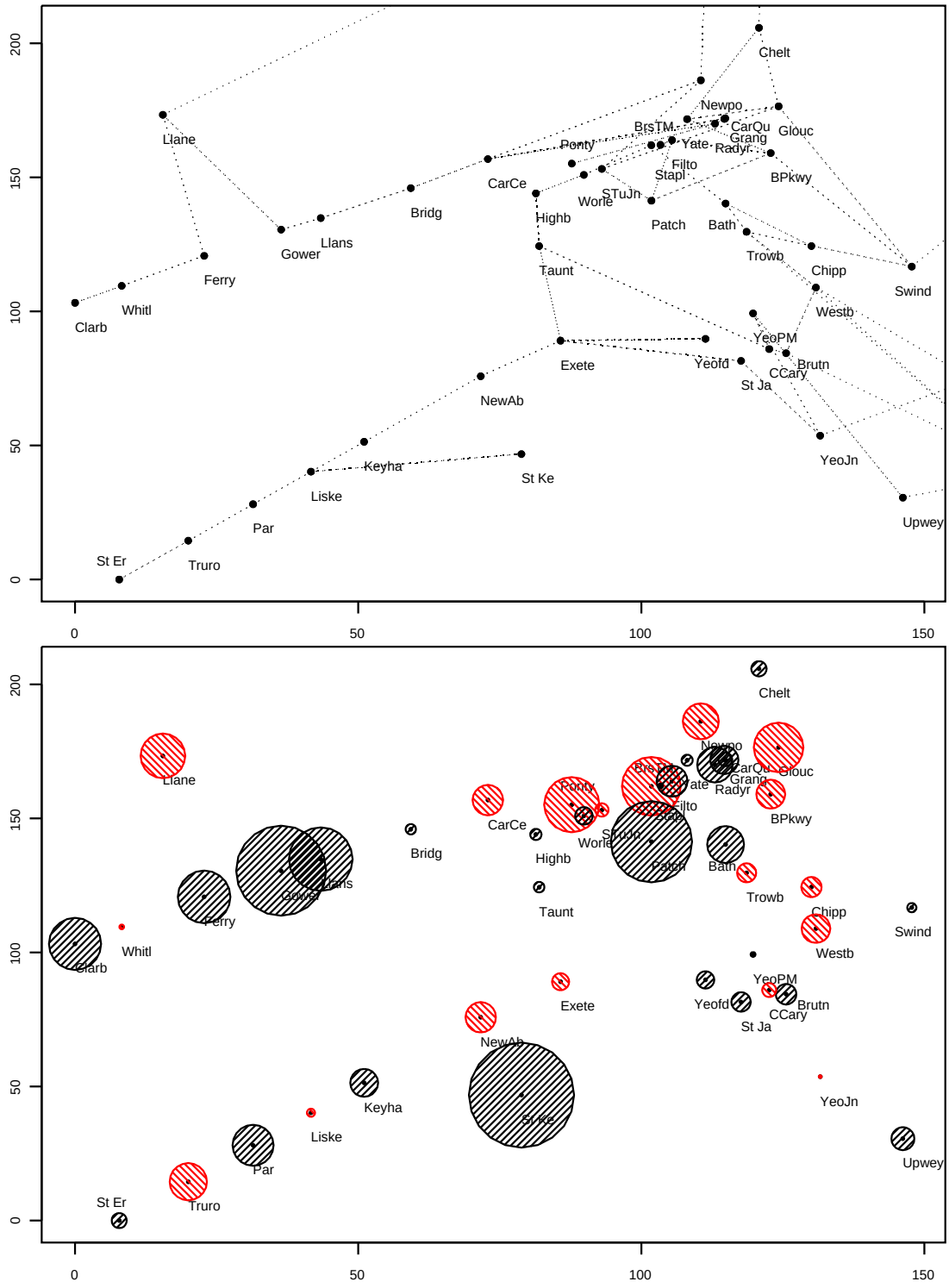


Figure 7: Top: diagram showing our selected stations and their connections. Bottom: residuals from the fit described in the text. Circles with lines oriented at 45° are positive, those at 135° are negative. The largest residual in absolute size is *St Keyne* at 55 seconds. Axes are arbitrary from multidimensional scaling.

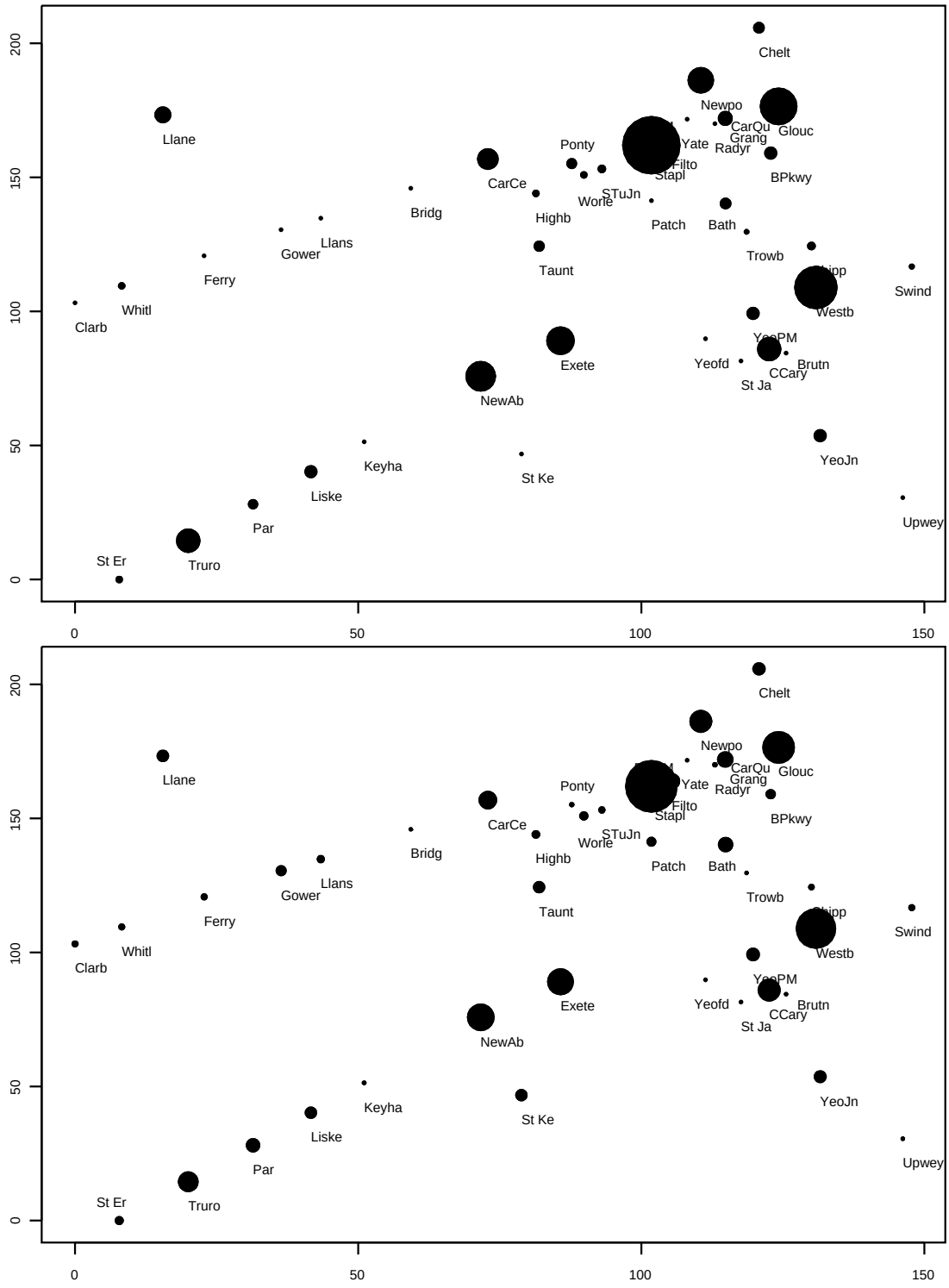


Figure 8: Top: Raw average delay data. Radius of each circle is proportional to average delay for each station. Largest average delay is 15 minutes at *Bristol Temple Meads* (unfortunately). Bottom: Network lifted regression estimate using inverse distance weights and monotone dependence eBayesThresh. Largest *estimated* average delay at *Bristol TM* is 14 minutes 28 seconds. Axes are arbitrary from multidimensional scaling.

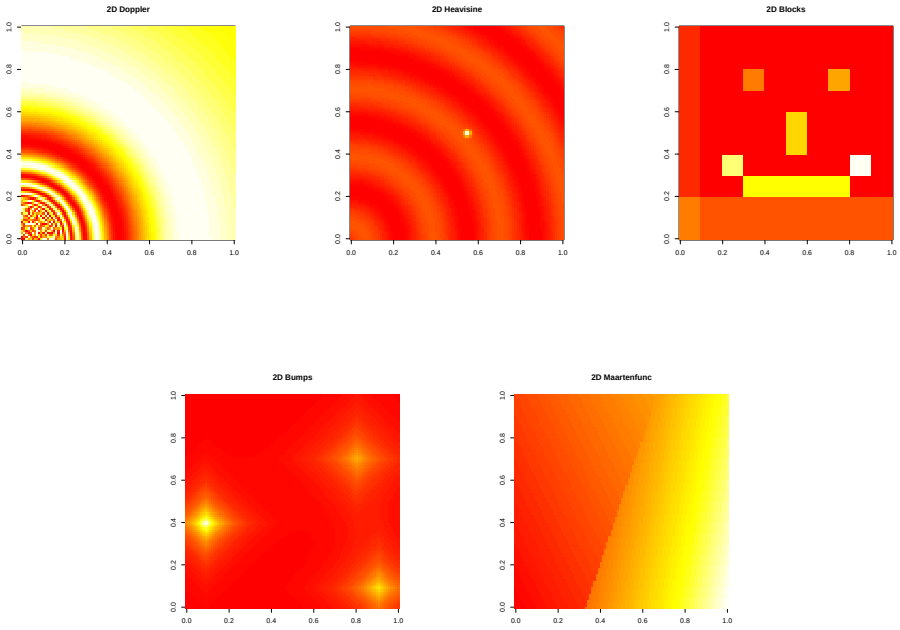


Figure 9: 2D analogues of Donoho and Johnstone test functions. From top left clockwise: Doppler, Heavisine, Blocks, $\mathrm{mfc}$(not an analogue), Bumps.

influence performance but small differences appear, especially with the tree-based lifting. Likewise, amongst all of the methods for carrying out MLE (all coefficients, parametric dependence, artificial levels, parametric dependence within artificial levels, and monotone dependence) there seems to be no clear winner. Each method seemed to do better than the others on occasion. If forced to select one method then monotone dependence usually seemed to do well.

### 7.3.2 Comparing Voronoi lifting with Triograms

Hansen *et al.* (1998) introduced the triogram method for function estimation using piecewise linear, bivariate splines based on an adaptively constructed triangulation (see also Koenker and Mizera (2004) for a smoothing spline approach to triograms based on the Delaunay triangulation). We compare our Voronoi lifting method to Triograms using the `quantreg` package.

We used two test functions for this simulation study. First define the generic function:

$$\mathrm{gf}(x, y, \mathrm{horizon}) = (2x + y)\mathbb{I}\left\{\mathrm{horizon}(x, y) \leq 0\right\} + (10 - x)\mathbb{I}\left\{\mathrm{horizon}(x, y) > 0\right\}, \qquad (30)$$

where $\mathbb{I}$ is the usual indicator function and then define horizons

$$\mathrm{horizon}_A(x, y) = 3x - y - 1 \text{ and } \mathrm{horizon}_B(x, y) = (x - 1/2)^2 + (y - 1/2)^2 - 1/16, \qquad (31)$$

and then our test functions are

$$\mathrm{mfa}(x, y) = \mathrm{gf}(x, y, \mathrm{horizon}_A(x, y)) \text{ and } \mathrm{mfb}(x, y) = \mathrm{gf}(x, y, \mathrm{horizon}_B(x, y)), \qquad (32)$$

Table 2: Mean averaged squared errors resulting from 50 simulation runs for denoising of $\mathrm{mfa}$ and $\mathrm{mfb}$ by triogram and Voronoi lifting method.

| | Function $\mathrm{mfa}$ | | Function $\mathrm{mfb}$ | |
|---|---|---|---|---|
| Method | 15dB | 18dB | 15dB | 18dB |
| Triograms | 20.9 (0.04) | 20.0 (0.04) | 19.9 (0.04) | 19.3 (0.04) |
| Voronoi | 16.4 (0.02) | 11.1 (0.02) | 14.3 (0.03) | 9.7 (0.02) |

in words: two different piecewise functions defined on two different 'horizons' for each function (one a line, the other a disc).

For each simulation run in this section we generated 1000 $(x, y)$ locations from a 2D uniform density on $[0, 1] \times [0, 1]$. We then generated noisy observations by adding Gaussian noise with two signal to noise ratios (SNRs) of 18dB and 15dB. In each case we performed 50 simulations. The results are shown in Table 2 and indicate the superior performance of the Voronoi lifting method *for these functions and SNRs*. Further experiments show that for very low SNRs triogram methods do better.

### 7.3.3 Comparing Voronoi lifting with thin-plate splines and kriging.

In recent work Heaton and Silverman (2006) compare our Voronoi lifting methodology additionally equipped with an imputation method with both thin-plate spline and kriging methodology and show that Voronoi lifting is competitive when compared to those methods, see Section 8 for further information.

## 8 Conclusions and future possibilities

This article has described a variation on the lifting theme: "lifting one coefficient at a time" and specified a new multiscale methodology for non-parametric regression in two or more dimensions. Three types of lifting methodology are developed: lifting with the Dirichlet tessellation using coordinate information in two-dimensions, lifting with trees and graphs using coordinate information and lifting with trees and graphs using inter-point distance information. With these algorithms "scale" naturally arises as a continuous concept and various empirical Bayes methods have been invented that make use of the continuous scale knowledge in a consistent way. The compression abilities of our techniques have been investigated and compare well to the standard 2D wavelet transform. We have also demonstrated the utility of our techniques both on the krill data (where ships track information can optionally be used) and also to an example where the underlying neighbourhood network is prespecified (the rail delay example). An aim of this article is to provide a solid framework for developing lifting "one coefficient at a time" algorithms. Subsequently we described three main methods for applying to spatial data with and without a predefined graph structure.

Clearly though, there is room for imaginative alternatives: new ways of defining and using neighbourhood structures, new integral definitions and new ways of predicting points that get left out.

A further innovation would be to choose from amongst different types of predict and/or update steps as each coefficient is generated. In generic lifting this is known as 'adaptive lifting', see Claypoole *et al.* (2003). For lifting one coefficient at a time adaptive lifting has been described in

one dimension by Nunes *et al.* (2006) who build on Jansen *et al.* (2001) and preprint versions of this article by permitting a choice of regression order (linear, quadratic or cubic) and/or number of neighbours involved in prediction. Nunes *et al.* (2006) provide a full literature review of adaptive lifting. They also present the results of a comprehensive simulation study which shows that one-dimensional adaptive lifting one coefficient at a time produces extremely good compression and nonparametric regression results on irregular data when compared to *Locfit* (Loader, 1997, 1999) the smoothing spline function in S-Plus (`smooth.spline()` and the regular wavelet algorithm for irregular data introduced by Kovac and Silverman (2000). Our methods can be developed further to cope with heteroscedastic variance using ideas similar to those proposed by Kovac and Silverman (2000). This has already been done for the 1D implementation of our lifting one-coefficient-at-a-time by Nunes *et al.* (2006). The techniques of Kovac and Silverman (2000) could also be used to cope with correlated errors: essentially an estimate of the correlation structure would be fed into the variance estimation stage as described in section 2.5. Naturally, there are several other ideas that might be tried.

As well as estimating values of a function (either on irregularly spaced spatial data or on a network) in the presence of noise from a given set of points one might also wish to estimate the function at a new set of points. For example, in a wireless network over time network nodes may enter and leave a network for a variety of reasons such as going in and out of radio range or as a result of power saving considerations. Heaton and Silverman (2006) describe a method that imputes the value of the function at a set of sites given information from another set of sites using the Bayesian lifting model that we present above using the Gibbs sampler. The demonstrate their method successfully both on regularly spaced data using the classical wavelet transform and also on simulated and real data using our two-dimensional Voronoi lifting that we describe above. In particular, they exhibit good results for rainfall prediction at 'new' sites in the US using data from the National Atmospheric Deposition Program (see `http://nadp.sws.uiuc.edu`). For both simulated and real data their results are competitve with both kriging and thin-plate spline methods and in one of the three cases for the rainfall data the lifting imputation method is significantly better. More in-depth simulations and comparisons need to be performed to thoroughly expore the utility of these methods. Other questions along these lines remain — for example, how to deal with locations that disappear when one is modelling data structures through time.

Another important possibility would be to more accurately model the variance and correlation between lifting coefficients ideally in a computationally efficient way. Such a possibility could be incorporated into the empirical Bayes paradigm but issues of computational efficiency would have to be dealt with. This train of thought also leads onto the fascinating possibility of defining stochastic processes on the lifting coefficients themselves, and additionally, defining a process for the locations $\mathbf{t}_i$. For example, one might envisage developing a similar kind of model to locally stationary wavelet processes as introduced by Nason *et al.* (2000) using our lifting techiques, or, defining a Markov random field model on the coefficients rather than in the data domain. Our main aim in this paper is to introduce a new multiscale tool to domains where wavelets are hard or impossible to use and show some examples of its use. However, there are many other situations that might benefit from these new general tools.

This article introduces a new methodology which we believe to be useful not only to statistics but more widely to situations involving data on graphs and irregular spatial data. However, we have not, as yet, discussed any theoretical considerations. In contrast to the cornucopia of theoretical properties of regular wavelet estimators (e.g. near-optimal risk bounds over wide function classes,

oracle inequalities) there are several challenges to developing similar theory for lifting, even for dyadic lifting. Some obstacles to statistical advancement can be traced back to well-known mathematical difficulties in determining the smoothness of functions constructed by lifting, see, e.g., Daubechies *et al.* (1999), that the basis functions are now no longer dilation and translations of a single function (like in regular wavelets, see Sweldens (1997)), and that the bases are not guaranteed to be Riesz basis Additional statistical difficulties would be caused by the fact that the order of points to be removed depends on the locations $\mathbf{t}_i$ (and for adaptive lifting the order additionally depends on the function values themselves, see Nunes *et al.* (2006)). In summary, it would be a considerable challenge to develop a deep theoretical understanding of our methods it is, of course, an interesting topic for future research.

## 9 Acknowledgements

The authors would like to thank Cathy Goss and Inigo Everson of the British Antarctic Survey for supplying them with the krill data and for helpful conversations and advice concerning the purposes of the study. The authors would also like to thank (i) Alistair Murray who initially provided us with krill data and inspiration. The krill study was funded by the Government of South Georgia and the South Sandwich islands; (ii) Roger Koenker for supplying the Matlab version of his `quantreg` package; (iii) Matt Nunes for translating the Voronoi lifting code from Matlab to R.

Estimated arrival times of UK trains can be found at the `www.livedepartureboards.co.uk` website. GPN was partially supported by EPSRC Advanced Research Fellowship AF/001664. All three authors were supported by EPSRC Research Grant M10229.

Three R packages (`NetTree`, `Liftvor` and `PicTree`) that carry out the three different kinds of lifting scheme described in this paper are available from Nason. The (original) version of `LiftVor` coded in Matlab is available from Jansen.

## References

Abramovich, F., Bailey, T. and Sapatinas, T. (2000) Wavelet analysis and its statistical applications. *J. Roy. Statist. Soc.* D, **49**, 1–29.

Abramovich, F., Sapatinas, T. and Silverman, B.W. (1998) Wavelet thresholding via a Bayesian approach. *J. Roy. Statist. Soc.* B, **60**, 725–749.

Allard, D. and Fraley, C. (1997) Nonparametric maximum likelihood estimation of features in spatial point processes using Voronoi tessellation. *J. Am. Statist. Ass.*, **92**, 1485–1493.

Antoniadis, A. and Fan, J. (2001) Regularization of wavelet approximations. *J. Am. Statist. Ass.*, **96**, 939–967.

Antoniadis, A., Grégoire, G. and Vial, P. (1997) Random design wavelet curve smoothing. *Stat. Prob. Lett.*, **35**, 225–232.

Averkamp, R. and Houdré, C. (2003) Wavelet thresholding for non-necessarily Gaussian noise: idealism. *Ann. Statist.*, **31**, 110–151.

Belkin, M., Matveeva, I. and Niyogi, P. (2004) Regularization and semi-supervised learning on large graphs. *Lect. Notes Comp. Sci.*, **3120**, 624–638.

Butler, K. (1999) *Rail mileages in Britain.* `www.mscs.dal.ca/~butler/railmile.htm`

Cai, T. and Brown, L.D.. (1999) Wavelet estimation for samples with random uniform design. *Stat. Prob. Lett.*, **42**, 313–321.

Chua, D.B., Kolaczyk, E.D. and Crovella, M. (2006) Network Kriging *IEEE Journal of Selected Areas in Communications*, (to appear).

Claypoole, R.L., Davis, G.M., Sweldens, W. and Baraniuk, R.G. (2003) Nonlinear wavelet transforms for image coding via lifting. *IEEE Trans. Im. Proc.*, **12**, 1449–1459.

Cleveland, W.S. and Devlin, S.J. (1988) Locally weighted regression: an approach to regression analysis by local fitting. *J. Am. Statist. Ass.*, **83**, 596–610.

Clyde, M., Parmigiani, G. and Vidakovic, B. (1998) Multiple shrinkage and subset selection in wavelets. *Biometrika*, **85**, 391–402.

Cressie, N.A.C. (1993) *Statistics for spatial data.* Wiley: New York.

Daubechies, I. and Sweldens, W. (1998) Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.*, **4**, 247–269.

Daubechies, I., Guskov, I., Schröder, P., and Sweldens, W. (1999) Wavelets on irregular point sets. *Phil. Trans. R. Soc. Lond.* A, **357**, 2397–2413.

Delouille, V., Simoens, J. and von Sachs, R. (2001) Smooth design-adapted wavelets for nonparametric stochastic regressiob. *Discussion Paper 0117*, Institut de Statistique, Université Catholique de Louvain, Belgium.

Delouille, V. (2002) *Nonparametric stochastic regression using design-adapted wavelets.* PhD Thesis, Université Catholique de Louvain, Belgium.

Delouille, V. and von Sachs, R. (2002) Smooth design-adapted wavelets for half-regular designs in two dimensions. *Discussion Paper 0226*, Institut de Statistique, Université Catholique de Louvain, Belgium.

Delouille, V., Jansen, M. and von Sachs, R. (2003) Second generation wavelet methods for denoising of irregularly spaced data in two dimensions. *Discussion Paper 0305*, Institut de Statistique, Université Catholique de Louvain, Belgium.

Denison, D.G.T., Holmes, C.C., Mallick, B.K. and Smith, A.F.M. (2002) *Bayesian Methods for Nonlinear Classification and Regression.* John Wiley: London.

Donoho, D.L. and Johnstone, I.M. (1994) Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, **81**, 425–455.

Fortune, S. (1987) A sweepline algorithm for Voronoi diagrams. *Algorithmica*, **2**, 153–174..

Fryzlewicz, P. and Nason, G.P. (2004) A Haar-Fisz algorithm for Poisson intensity estimation. *J. Comp. Graph. Stat.*, **13**, 621–638.

Green, P.J. and Silverman, B.W. (1994) *Nonparametric regression and generalized linear models.* Chapman and Hall: London.

Goss, C. and Everson, I. (1996) An acoustic survey of Antarctic krill on the South Georgia shelf. *CCAMLR Scientific Abstract*, WG-EMM-96/42.

Hall, P. and Turlach, B.A. (1997) Interpolation methods for nonlinear wavelet regression with irregularly spaced design. *Ann. Statist.*, **25**, 1912–1925.

Hansen, M., Kooperberg, C. and Sardy, S. (1998) Triogram Models *J. Am. Statist. Soc.*, **93**, 101–119.

Herrmann, E., Engel, J., Wang, M.P. and Gasser, T. A bandwidth selector for bivariate kernel regression. *J. Roy. Statist. Soc.* B, **57**, 171–180.

Herrick, D.R.M. (2000) *Wavelet methods for curve and surface estimation.* PhD Thesis, University of Bristol.

Heaton, T. and Silverman, B.W. (2006) A wavelet/lifting scheme based imputation method. (Submitted for publication). `www.stats.ox.ac.uk/~silverma/pdf/heatonsilverman.pdf`

Jansen, M. (2006) Multiscale Poisson data smoothing. *J. Roy. Statist. Soc.* B, **68**, 27–48.

Jansen, M., Nason, G.P. and Silverman, B.W. (2001) Scattered data smoothing by empirical Bayesian shrinkage of second generation wavelet coefficients. In Unser, M. and Aldoubi, A. (eds) *Wavelet applications in signal and image processing*, Proceedings of SPIE, **4478**, 87–97.

Johnstone, I.M., Kerkyacharian, G., Picard, D. and Raimondo, M. (2004) Wavelet deconvolution in a periodic setting (with discussion). *J. Roy. Statist. Soc.* B, **66**, (to appear).

Johnstone, I.M. and Silverman, B.W. (1997) Wavelet threshold estimators for data with correlated noise. *J. Roy. Statist. Soc.* B, **59**, 319–351.

Johnstone, I.M. and Silverman, B.W. (2004) Needles and hay in haystacks: Empirical Bayes estimates of possibly sparse sequences. *Ann. Statist.*, **32**, 1594–1649.

Johnstone, I.M. and Silverman, B.W. (2005a) EBayesThresh: R programs for Empirical Bayes thresholding. *J. Stat. Software*, **12.8**, 1–38.

Johnstone, I.M. and Silverman, B.W. (2005b) Empirical Bayes selection of wavelet thresholds. *Ann. Statist.*, **33**, 1700–1752.

Jolliffe, I.T. (2002) *Principal component analysis.* Springer: New York.

Koenker, R. and Mizera, I. (2004) Penalized triograms: total variation regularization for bivariate smoothing. *J. Roy. Statist. Soc.* B, **66**, 145–163.

Kohler, M. (2003) Nonlinear orthogonal series estimation for random design regression. *J. Stat. Plan. Infer.*, **115**, 491–520.

Kovac, A. and Silverman, B.W. (2000) Extending the scope of wavelet regression methods by coefficient-dependent thresholding. *J. Am. Statist. Ass.*, **95**, 172–183.

Kovačević, J. and Sweldens, W. (2000) Wavelet families of increasing order in arbitrary dimensions. *IEEE Trans. Im. Proc.*, **9**, 480–496.

Krzanowski, W.J. and Marriott, F.H.C. (1995) *Multivariate Analysis. Part 2. Classification, covariance structures and repeated measurements.* Arnold: London.

Loader, C. (1997) Locfit: an introduction. *Stat. Comput. Graph. News.*, **8**, 11–17.

Loader, C. (1999) *Local regression and likelihood.* Springer: New York.

Mallat, S.G. (1989) A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattn. Anal. Mach. Intell.*, **11**, 674–693.

Nason, G.P. (2002) Choice of wavelet smoothness, primary resolution and threshold in wavelet shrinkage. *Statistics and Computing*, **12**, 219–227.

Nason, G.P., Jansen, M. and Silverman, B.W. (2004) Simulations and examples for multivariate nonparametric regression using lifting. *Technical Report* 04:18, Department of Mathematics, University of Bristol.

Nason, G.P., von Sachs, R. and Kroisandt, G. (2000) Wavelet processes and adaptive estimation of the evolutionary wavelet spectrum. *J. Roy. Statist. Soc.* B, **62**, 271–292.

Neumann, M. and von Sachs, R. (1995) Wavelet thresholding: beyond the Gaussian I.I.D. situation. *Lect. Notes Stat.*, **103**, 301–329.

Nunes, M., Knight, M. and Nason, G.P. (2006) Adaptive lifting for nonparametric regression. *Statistics and Computing*, **16**, 143–159.

Okabe, A., Boots, B. and Sugihara, K. (1992) Spatial Tessellations – Concepts and Applications of Voronoi Diagrams. Chichester: Wiley.

Penrose, M.D. (1996) The random minimal spanning tree in high dimensions. *Ann. Prob.*, **24**, 1903–1925.

Penrose, M.D. and Yukich, J.E. (2003) Weak laws of large numbers in geometric probability. *Ann. App. Prob.*, **13**, 277–303.

Pensky, M. and Vidakovic, B. (2001) On non-equally spaced wavelet regression. *Ann. Inst. Stat. Math.*, **53**, 681–690.

Percival, D.B. and Walden, A.T. (2000) *Wavelet methods for time series analysis.* Cambridge University Press: Cambridge.

R Development Core Team (2005) *R: A language and environment for statistical computing.* R Foundation for Statistical Computing. `http://www.R-project.org` Vienna: Austria. ISBN 3-900051-07-0.

Rue, H. and Held, L. (2005) *Gaussian Markov Random Fields.* Chapman and Hall: London.

Sardy, S., Percival, D.B., Bruce, A.G., Gao, H.Y. and Stuetzle, W. (1999) Wavelet shrinkage for unequally spaced data. *Statistics and Computing*, **9**, 65–75.

Sibson, R. (1981) A brief description of natural neighbour interpolation. In Barnet, V. (ed) *Interpreting Multivariate Data*, 21–36. Wiley: Chichester.

Silverman, B.W. and Vassilicos, J.C. (2000) *Wavelets: the key to intermittent information.* Oxford University Press: Oxford.

Smola, A.J. and Kondor, R. (2003) Kernels and regularization on graphs. *Lect. Notes Art. Intell.*, **2777**, 144–158.

Starck, J.L., Candès, E.J. and Donoho, D.L. (2000) The Curvelet Transform for Image Denoising. *IEEE Trans. Im. Proc.*, **11**, 670–684.

Sweldens, W. (1996) Wavelets and the lifting scheme: A 5 minute tour. *Z. Angew. Math. Mech.*, **76**, 41–44.

Sweldens, W. (1997) The lifting scheme: a construction of second generation wavelets. *SIAM J. Math. Anal.*, **29**, 511–546.

Uytterhoeven, G. and Bultheel, A. (1997) The red-black wavelet transform. *Technical Report* TW 271, Department of Computer Science, Katholieke Universiteit Leuven, Belgium.

Vanraes, S., Jansen, M. and Bultheel, A. (2002) Stabilised wavelet transform for non-equispaced data smoothing. *Signal Processing*, **82**, 1979–1990.

Vidakovic, B. (1999) *Statistical modeling by wavelets.* Wiley: New York.

Wahba, G. (1990) *Spline models for observational data.* SIAM: Philadelphia.