Journal of the Royal Statistical Society
SERIES B Statistical Methodology

J. R. Statist. Soc. B (2009)
71, *Part* 1, *pp.* 97–125

Multiscale methods for data on graphs and irregular multidimensional situations

Maarten Jansen,

Katholieke Universiteit Leuven, Belgium

Guy P. Nason

University of Bristol, UK

and B. W. Silverman

St Peter's College, Oxford, UK

[Received October 2007. Revised May 2008]

Summary. For regularly spaced one-dimensional data, wavelet shrinkage has proven to be a compelling method for non-parametric function estimation. We create three new multiscale methods that provide wavelet-like transforms both for 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 with several other methods for spatial regression for both real and simulated data. Although we concentrate 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.

Keywords: Graph; Irregular data; Lifting; Wavelets; Wavelet shrinkage

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 and 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) and Abramovich *et al.* (2000) for reviews. Nearly all work in the statistical area has been based on the fast discrete wavelet transform that was 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 that ε_i is independent and identically distributed Gaussian with mean 0 and constant variance σ^2. Let $t_i = i/n$. We observe

Address for correspondence: Guy P. Nason, Department of Mathematics, University of Bristol, University Walk, Bristol, BS8 1TW, UK.
E-mail: g.p.nason@bristol.ac.uk

 1369–7412/09/71097

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

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

(a) The number of observations, n, is a power of 2, say $n = 2^J$ for some $J \in \mathbb{N}$. This restriction is not too difficult to overcome even when using fast wavelet transforms.
(b) 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, e.g., 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). 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 that commonly arise 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. Adaptations of lifting to curve estimation problems in one dimension are discussed in Vanraes *et al.* (2002) and Delouille *et al.* (2004). For lifting half-regular designs (tensor products 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 paper both develop and build on Jansen *et al.* (2001).
(c) The error distribution is independent and identically distributed Gaussian with zero mean and constant variance. Various researchers have weakened these assumptions. For example, see Johnstone and Silverman (1997) for correlated noise and 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 for those with discontinuities or other inhomogeneities (even when, *a priori*, it is not explicitly known whether the function is smooth or not) and fast computational speed.

1.2. Our main contributions

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

(a) a wavelet-like transform for data on a graph,
(b) wavelet-like transforms for irregularly spaced data in two- or higher dimensional space and
(c) 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 methods, 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 with 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 indepen-

dent and identically distributed noise but several of the techniques that were 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, i.e., 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 the smallest or largest point that is greater or less than the given point respectively. 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 analyse 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 or wavelet transforms for function estimation. For function estimation there is an enormous range of alternatives developed across a huge range of disciplines including many in statistics. Those which we have considered, and compared with 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 (1993)) and kriging (see Cressie (1993)). The last two sets of comparisons are to be found in Heaton and Silverman (2008); the others in Section 7. There are many more possibilities, e.g. 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 (see Jolliffe (2002) and, for graphs and network kriging, see Chua *et al.* (2006)).

Although our methods compare favourably with the first group 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 that were 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.

(a) 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, for example, Fortune (1987)). It is not always easy to discover the computational complexity of some of the methods that were listed above. However, empirical orthogonal functions are based on eigenvector determination ($\mathcal{O}(n^3)$), LOESS is quadratic in storage and some of the above algorithms rely on variants of Markov chain Monte Carlo sampling which do not scale well to large problems.
(b) Our methods are rotationally invariant. Some of the above methods are not.
(c) 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).

(d) On a range of real and simulated examples that are reported in Section 7, our methods work well. The examples include both discontinuous and smooth functions. It is reassuring that a method that was developed to allow for possible discontinuities also works 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, but some theoretical remarks are addressed in Section 5.

1.4. Krill intensity estimation example

We first consider an example that existing wavelet techniques would find difficult to solve and other statistical techniques, such as kriging, might find challenging. Goss and Everson (1996) described an experiment that was designed to quantify the amount and distribution of krill in the south Atlantic ocean around South Georgia. Fig. 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 we might wish to take into account when performing spatial regression. For example, in some applications we might be interested in regression on the transect itself, or 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. an 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 paper

Section 2 first reviews lifting and then introduces our variation on the theme 'lifting one coefficient at a time', then describes our scheme for irregular spatial data and graphs, and finally

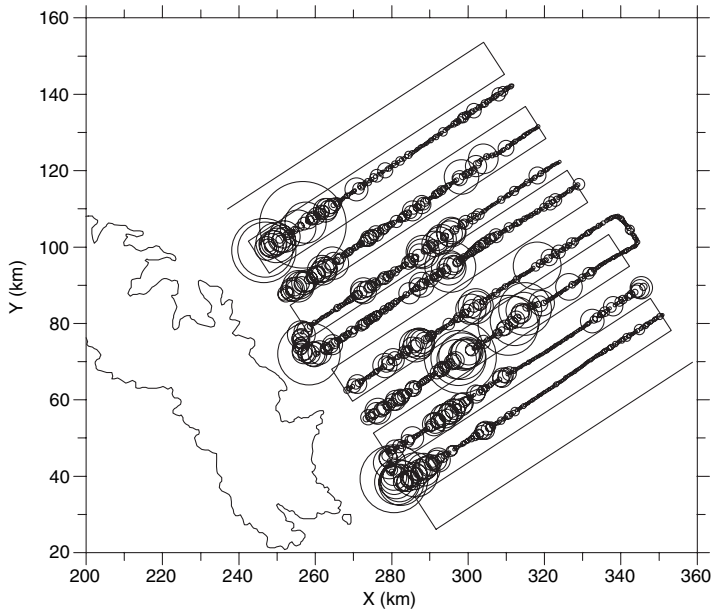

Fig. 1. Example krill sampling scheme: the island of South Georgia is shown at the bottom left-hand side; each sample is indicated by a circle and the diameter, which is proportional to the density of krill detected at that location (the figure was kindly supplied by Alistair Murray, British Antarctic Survey)

describes an efficient computational approximation for the variance of our lifting coefficients. Section 3 describes our version of lifting to be applied to a function on a graph (a network). Such a network might be constructed from, for example, irregularly spaced data in Euclidean space or the data themselves might naturally arise in the form of a network. For example, in a rail transportation network we might think of stations either as irregularly spaced points in two-dimensional space or we might think of them as nodes in a network where the edges are railway lines. For irregular data in Euclidean space Section 4 uses a Dirichlet tessellation to define neighbourhoods and constructs a lifting transform by using those neighbourhoods. Successful wavelet shrinkage depends on good compression abilities of the underlying wavelet transform. Section 5 explores the theoretical basis for our work and describes some compression studies. 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 discrete wavelet transform no longer exist in our work. We describe several empirical Bayes methods that were designed to work with the more general concept of scale. Section 7 contains a real life example and summarizes several simulation studies. The real example considers regression of the krill data where co-ordinate information is used. A further real life example, which is concerned with denoising of train delay data on a rail network, can be found in Jansen *et al.* (2008). 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 lifting as it has been considered previously. Given a vector x of data, we divide the indices of x into two subsets, which are 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 lifting step works as follows.

(a) *Predict*—use x^J to yield an appropriate predictor $\tilde{x}^I$ of x^I, and the residual is $(x^I)^* = x^I - \tilde{x}^I$.
(b) *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 that 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 $\tilde{x}_{2m-1} = x_{2m}$ 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 'scale' coefficients at the next level, a smoothed version of the original data.

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

$$x^I := x^I - x^J \quad \text{followed by} \quad x^J := x^J + \tfrac{1}{2}x^I. \tag{2}$$

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 multi-resolution

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 that 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 lifting is that the inverse transform can be constructed mechanically. Step (2) is inverted by reversing the assignment order, and changing the signs, to give

$$x^J := x^J - \tfrac{1}{2}x^I \quad \text{followed by} \quad x^I := x^I + x^J. \tag{3}$$

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 that we adopt will be as follows.

The first step is to construct an order $i_n, \ldots, i_{l+1}$ 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 l coefficients will remain, corresponding to the scaling coefficients at level l.

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

$$\textit{predict}, \; x_i := x_i - a' x^J, \quad \text{followed by} \quad \textit{update}, \; x^J := x^J + x_i b. \tag{4}$$

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:

$$x^J := x^J - x_i b \quad \text{followed by} \quad x_i := x_i + a' x^J. \tag{5}$$

For computational purposes, it is convenient to specify and store the transform in a standard format, as a ragged array with $n - l$ rows. We call this the *lifting coefficient array*. The sth 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 lifting 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 shall consider ways of constructing the lifting coefficient array, with particular attention paid to the case of spatial irregular data. Even the Haar trans-

form 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 that are 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 that are not on a lattice can be found in Daubechies *et al.* (1999). For data on a lattice see 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 that are relevant to lifting transforms for spatial irregular data are considered, but the discussion has wider validity for methods that are 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 ϕ_{nk} are scaling functions such that

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

Here δ_{ik} is the Kronecker delta, at least approximately. If the scaling functions satisfy condition (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 ϕ_{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

$$f(\mathbf{t}) = \sum_{l \in \mathcal{D}_r} d_l\, \psi_l(\mathbf{t}) + \sum_{k \in \mathcal{S}_r} c_{rk}\, \phi_{rk}(\mathbf{t}) \tag{8}$$

where the ψ_l are wavelet functions with zero integral, and the ϕ_{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 equation (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 that we adopt and will be discussed in subsequent sections of the paper. We calculate the coefficient d_{i_r} in the way that is specified in expression (4), setting

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

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 0, so we conduct the predict step with a set of weights satisfying $\Sigma a_j^r = 1$. 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 that 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 equation (8), we have $c_{r-1,j} = 1$ and all other $c_{r-1,s}$, $s \neq j$, and d_s equal to 0 for $s = i_r, \ldots, i_n$. Hence, inverting the lifting steps, $c_{rj} = 1$, from equation (10), and $c_{ri_r} = a_j$ from equation (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 equation (11) to obtain

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

For j in $\mathcal{S}_{r-1}$ that are not members of J_r, the same argument with $a_j^r = 0$ 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 0. From equation (10) we then have $c_{rj} = -b_j^r$ for j in J_r. Equation (9) then gives $c_{ri_r} = 1 - \Sigma_{j \in J_r} a_j^r b_j^r$. 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 expression (11).

The weights b_j^r are found from the requirement that the integral of the wavelet is 0. By integrating equation (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 by using expression (12). For reasons of numerical stability, we use the minimum norm solution of equation (14), setting

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

Within the process it is not necessary to express the wavelets or scaling functions explicitly, but the integrals of the scaling functions choose the coefficient i_r and specify the weight vector b^r. Therefore, to initiate the process, the integrals I_{nj} of the original scaling functions need to be specified. 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 MSTs.

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 ψ_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 α_{i_r}. In the natural neighbour method that is described later, α_{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 $a_j \geqslant 0$ we have

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

and so the scales α_i are a monotonic function of the index r and the order in which lifting determines the coefficients.

2.4. The dual basis functions

The lifting procedure can be thought of in two ways. On the one hand, if we have a function f of the form (6), then expansion (8) gives an expression of f in terms of a multi-resolution basis, where effects of different scales are captured by different wavelet coefficients. On the other hand, consider lifting as a linear tranformation of a vector of *values* x, yielding a coefficient vector $\tilde{x}$, say, whose elements have a multi-resolution interpretation. In either case the relationship between the original function or data, and the derived coefficients, can be elucidated by investigating the dual basis functions or vectors. More can be found in Section 2.4 of Jansen *et al.* (2008).

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 lifting. Of course, because lifting 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 $x_i^* = x_i - \Sigma_{j\in J}\, a_j x_j$, we have

$$\begin{aligned} \operatorname{var}(x_i^*) &= V_i + \sum_{j\in J} a_j^2 V_j, \\ \operatorname{cov}(x_i^*, x_j) &= -a_j V_j. \end{aligned} \tag{16}$$

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

$$\operatorname{var}(x_j^*) = V_j + b_j^2 \operatorname{var}(x_i^*) + 2b_j \operatorname{cov}(x_i^*, x_j) = (1 - 2a_j b_j)\, V_j + b_j^2 \operatorname{var}(x_i^*). \tag{17}$$

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

$$\begin{aligned} V_i^* &= V_i + \sum_{j\in J} a_j^2 V_j, \\ V_j^* &= (1 - 2a_j b_j) V_j + b_j^2 V_i^* \qquad \text{for } j \in J. \end{aligned} \tag{18}$$

The approximation that we use is to neglect any correlations between the coefficients that are obtained at the next stage, but simply to iterate the calculations (18). 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 that is 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 that prior or estimated information on the covariance structure can be fed into the calculation of the coefficients' variance along the lines of methods that are used for regular wavelet shrinkage such as Kovac and Silverman (2000).

3. Lifting 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—e.g. a transportation network or communications network. The other way is that data are supplied in a form that can be converted into a network, e.g. irregularly spaced data in K-dimensional space on which a graph can be induced by calculating interpoint distances and constructing, say, an MST.

3.1. Minimal spanning trees and other tree-based approaches

For data sets in two dimensions, approaches that are based on Voronoi cells in Section 4 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.

Here, we consider an alternative lifting approach that is based on trees. 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. If the original data sites $\mathbf{t}_i$ lie in a K-dimensional Euclidean space, a natural approach is to use MSTs (see for example Krzanowski and Marriott (1995)), which are easily computed. Other types of tree might be useful for particular applications, and these would be a possible topic for future work.

Some data sets naturally live on a tree rather than in some Euclidean space. For example, the data collection transects for the krill data that are depicted in Fig. 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 functions on a graph our methods 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 network function (similar to a single scale level of wavelet coefficients acting as a band-pass filter). See Smola and Kondor (2003) and Belkin *et al.* (2004) for other work on regularization of functions on graphs.

3.2. General aspects of tree-based lifting

The first step in the lifting scheme as set out in Section 2.3 was to specify the initial scaling functions ϕ_{nk} and to find their integrals. In the tree context, we define the scaling function ϕ_{ni} to be 1 at the node i and 0 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 $\Sigma_i w_i f(i)$. 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 must 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 has been defined, we need to define the prediction weight vector a. For reasons that are explained below, we mostly use *inverse distance prediction weights*, setting $a_{ij} = c\delta_{ij}^{-1}$, where δ_{ij} is the distance from point i to point j, and c is chosen so that 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 have *bona fide* Euclidean co-ordinates, in which case the tree can be used to define the neighbours but the co-ordinates are used by least squares to form prediction weights. To distinguish between these two variants we refer to them either as a 'tree with inverse distances weights' or a 'tree with least squares co-ordinate weights'. As an example of these two algorithms in action see Fig. 2 in Section 6.3.

Having defined the weight vector a, we can update the integrals by using equation (12) and calculate the update weights b_j by using 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 MST of the points that are 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 Fig. 2 (right) in Jansen *et al.* (2008) with the ship track in Fig. 2 (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 $2(1-1/n)$ 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 the set of neighbours J, there is less opportunity for 'mixed scales' to occur. Alternatively, we may wish to include higher order neighbours in J, to obtain better predictions. If we used higher order neighbours, we could either use neighbours up to a given order, or we could increase the order of the neighbours until the size of J reached a certain size.

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 in Jansen *et al.* (2008).

3.3. Why use inverse distance prediction weights?

We now explore a correspondence between inverse distance prediction weights and local linear prediction. Suppose that we are working on a tree, that we are predicting the value at point i

and that $J = \{j_1, j_2, \ldots, j_r\}$ for some $r \geqslant 2$. Also, the tree is defined only by its linkage structure and the lengths δ_{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 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 then have $\Sigma_{j\in J}\,\mathbf{u}_j = 0$. Now place 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 places 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 that is obtained by inverse distance weighting the values y_j. We now have, setting c such that $c\,\Sigma_j\,\delta_{ij}^{-1} = 1$,

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

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 the inverse distance weighted average y^*.

4. Lifting based on Voronoi polygons

In this section we consider lifting 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 that are 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 Ω be a suitable region in the plane containing all the sites under consideration. The region Ω may, for example, be the whole plane, or a suitable rectangle or the convex hull of the sites. Comments about the precise choice of Ω will be made later. The *Voronoi cell* of any particular site is the set of points in Ω that are 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 Ω into these polygons. See Fig. 2 of Jansen *et al.* (2008) for an example. Two sites are neighbours if their Voronoi cells have a boundary in common, and the joins of all pairs of neighbours form 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 lifting stage, 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 methods could be based on higher order neighbours.

The paradigm that was set out in Section 2.3 requires two more ingredients: the integrals of the initial scaling functions ϕ_{nk} and a method of specifying the prediction weights a^r at each stage. Provided that Ω is a finite region, a natural definition of the initial scaling function ϕ_{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.

4.2. Natural neighbour interpolation

If site i is removed and the Dirichlet tessellation recomputed, the Voronoi cell of that site will be divided between its neighbours. Assume that the region Ω is finite. Let A_i be the cell corresponding to site i and let A_{ij} be the part of the cell that is 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.

Lifting using natural neighbour interpolation works by setting $a_j = |A_{ij}|/|A_i|$ for each neighbour j of i, where $|\cdot|$ denotes area. Provided that the cell A_i does not intersect the boundary of Ω, the prediction weights that are 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 0 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 that is 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 is used to interpolate at the site $\mathbf{t}_i$. This scheme has the property that, if the function f is linear over the site $\mathbf{t}_i$ and its neighbours, then the wavelet coefficient is 0. 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 0. 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 they have 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 give rise to relationships between sites that are a long way apart on the scale that is 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 far from equilateral are 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 Ω is a considerable distance from the data boundary. This can be seen in the right-hand plot of Fig. 2 from Jansen *et al.* (2008).

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 redefining Ω 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°, say. In any event, the approach may need some modification at the corners of the configuration, where the approach that was described may leave sites with a single neighbour, and in this case it may be appropriate to reintroduce narrow triangles.

A related matter is the treatment of sites lying some distance from the remainder of the configuration, so that the angle that is 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 that is 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 that a site is removed. If the natural neighbour method is used, then the Dirichlet tessellation within the region Ω 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 Ω 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. Aspects of theory

Wavelet shrinkage is based on three properties of wavelet decompositions. The first is smoothness of the wavelet basis, including numerical convergence of the refinement scheme. The second is numerical stability and the third is sparsity of a wavelet decomposition. In classical dyadical or translation invariant transforms, the analyses of the three properties largely coincide and reduce to the solution of the two-scale equation

$$\varphi(x) = \sum_{k=-\infty}^{\infty} \sqrt{2} h_k \, \varphi(2x - k).$$

If, for given scaling coefficients h_k, the scaling function $\varphi(x)$ can be found by a numerically converging iteration, then the framework of multi-resolution analysis guarantees stable decompositions and reconstructions (see for instance Mallat (1998), chapter VII). Sparsity is determined by the support of $\varphi(x)$ and by the linear approximation power of the scheme for smooth functions. The approximation power depends on the maximum degree p of polynomials that can be represented exactly by an expansion

$$x^p = \sum_{k=-\infty}^{\infty} c_k \, \varphi(x-k),$$

where $p+1$ is the number of dual vanishing moments of the wavelet transform. Numerical stability, on the other hand, is ensured by the concept of Riesz bases, which is encapsulated in the definition of a multi-resolution analysis. A Riesz basis is 'almost' orthogonal in the sense that the norm equivalence (known as Plancherel's or Parseval's equality) holds within finite bounds. The norm equivalence is important in data processing, as it guarantees control on the effect of processing coefficients after inverse transform: small operations on wavelet coefficients result in small effects on the data.

Both stability and sparsity thus depend on the solution of the two-scale equation. In the settings that are described in this paper, the two-scale equation itself is scale dependent: both geometry and configurations are different in every step. As a consequence, the basis functions are no longer dilations and translations of a single father scaling function and the convergence analysis of the refinement process, leading to the basis functions, becomes difficult, if at all possible. One of the few exceptions is the convergence analysis of the cubic polynomial prediction refinement of Daubechies *et al.* (2001), which is based on a commutation principle for divided differences. The remainder of this section establishes results on sparsity and stability that do not make use of the scaling function $\varphi(x)$.

5.1. Sparsity

The dual number of vanishing moments is controlled by the prediction step in the lifting. The vanishing moments condition leads to perfect representations of polynomials. Smooth functions that are well approximated by polynomials should also be well approximated in the scaling bases, such that their wavelet coefficients are small and large coefficients correspond to singularities only.

The approximation power of a wavelet decomposition for smooth functions is formalized by the concept of Lipschitz regularity.

Definition 1. A function $f(x)$ is Lipschitz ν in a point x_0 if and only if there is a polynomial $p_{x_0}(x)$ of degree $r = \lceil \nu \rceil - 1$ and a finite number C such that $|f(x) - p_{x_0}(x)| \leqslant C|x - x_0|^{\nu - r}$. A function is uniformly Lipschitz ν on an interval I if it is Lipschitz ν in all points $x \in I$ with constant C independent of x.

It is well known (Jaffard, 1991) and straightforward to prove that the coefficients of a uniformly Lipschitz-ν-function f in an orthogonal or biorthogonal equidistant wavelet decomposition with L_2-normalized basis functions satisfy $|w_{j,k}| \leqslant 2^{-j(\nu+1/2)}C$. This result follows easily from the expression $w_{j,k} = \langle \psi^*_{j,k}, f \rangle$ with $\psi^*_{j,k}$ the L_2-normalized dual wavelet function at scale j, location k and $\langle \cdot, \cdot \rangle$ the usual L_2 inner product. Extension to two dimensions is straightforward. Unless an explicit rescaling takes place, lifting works with unnormalized basis functions, i.e., applied to one-dimensional regularly spaced data, the primal wavelet basis functions would be $\psi(2^j x - k)$ and the corresponding duals are then $2^j \psi^*(2^j x - k)$, leading to coefficients satisfying $|w_{j,k}| \leqslant 2^{-j\nu}C$. On irregular data points, in the absence of strong convergence results, similar bounds on wavelet coefficients can be established, assuming a weak statement of convergence. The following result is stated for lifting with linear least squares prediction or natural neighbour interpolating prediction. Similar results hold for other schemes.

Proposition 1. Let $x_i = x(\mathbf{t}_i)$ be n observations from a Lipschitz ν-function $x : \mathbb{R}^2 \to \mathbb{R}$. Consider lifting with linear least squares prediction or natural neighbour interpolating prediction.

Let C_r be the $(n-r)\times n$ matrix that maps the observed data vector onto the scaling coefficients c_{rk} after r lifting steps. Let D be the matrix that maps the observed data vector onto the vector of wavelet coefficients d. Assume that $\|C_r\|_\infty$ is bounded for all r and for $n\to\infty$. Then, for some constant C, independent from n and r, $|d_{i_r}|\leqslant Ch_{i_r}^{\nu^*}$, with $\nu^*=\min(\nu,2)$ and where $h_{i_r}=\max\{\|\mathbf{t}_p-\mathbf{t}_q\|,\, p,q\in\mathcal{I}_{i_r}\}$ with $\mathcal{I}_{i_r}=\{p\in\{1,\ldots,n\},\, D_{i_r,p}\neq 0\}$, i.e. the scale h_r of coefficient d_{i_r} is the maximum distance between points $\mathbf{t}_p$ with a non-zero contribution in the calculation of d_{i_r}.

Proof. We assume here that $1<\nu\leqslant 2$. The proof can be repeated with slight modifications if $\nu\leqslant 1$ or $\nu>2$. Define $\tilde{x}(\mathbf{t})=x(\mathbf{t}_{i_r})+\nabla x(\mathbf{t}_{i_r})'(\mathbf{t}-\mathbf{t}_{i_r})$. It can be verified that $\varepsilon(\mathbf{t})=\tilde{x}(\mathbf{t})-x(\mathbf{t})$ satisfies $|\varepsilon(\mathbf{t})|\leqslant C\|\mathbf{t}-\mathbf{t}_{i_r}\|^\nu$. Both local least squares prediction and natural neighbour interpolation schemes have the linear reproducing property (this is the local co-ordinate property of Sibson (1980) in the case of natural interpolation). As a consequence, when the wavelet transform is applied to $\tilde{x}(\mathbf{t})$, then all intermediate detail coefficients are 0, i.e. $\tilde{d}_{i_r}=0$, thereby annihilating *a priori* all update step effects. Thus, the scaling coefficients of $\tilde{x}(\mathbf{t})$ satisfy $\tilde{c}_{rj}=\tilde{x}_j$. The wavelet analysis is said to have two dual vanishing moments.

Denote by $d_{i_r}^\varepsilon$ the wavelet coefficients of $\varepsilon(\mathbf{t})$; then $|d_{i_r}|=|\tilde{d}_{i_r}+d_{i_r}^\varepsilon|=|d_{i_r}^\varepsilon|\leqslant\|C_{r-1}\|_\infty\cdot\|a^r\|_1 h_{i_r}^\nu$. If $\|C_r\|_\infty$ is bounded, then $\|C_{r-1}\|_\infty\cdot\|a^r\|_1$ can be bounded by a constant C.

Remark 1. The assumption that $\|C_r\|_\infty$ is bounded can be seen as a partial stability condition. It depends on the choice of update steps and on the homogeneity of the data points $\mathbf{t}$. Scattered data that are distributed in an inhomogeneous way may cause large coefficients. A formal definition of homogeneity is given in Section 5.2.

Remark 2. Checking that $\|C_r\|_\infty$ is bounded may proceed through the *adjoint lifting transform* (Jansen, 2007). The adjoint transform for lifting in equation (4) is an update-first scheme and reads as

$$\textit{update},\ x^J:=x^J+ax_i,\quad \text{and then}\quad \textit{predict},\ x_i:=x_i-b'x^J.$$

The adjoint transform switches the roles of primal and dual basis functions, by using the prediction coefficients in an update step and vice versa. The inverse adjoint transform starting from $d_{i_r}=0$ and $c_{rl}=0$, except for one index $l=k$, reveals the kth row of C_r.

On the basis of this result for smooth functions, we can construct a space of 'nearly' smooth functions, in a similar way as Besov or Triebel spaces. The smoothness space is defined in terms of the second-generation wavelet coefficients. Large coefficients are allowed, if they do not dominate the global decay, i.e. we define the second-generation Besov sequence norm as

$$\|f(\mathbf{t})\|_{b_{p,p}^\nu}=\sum_{r=1}^{n} h_{i_r}^{-\nu p}|d_{i_r}|^p\|\psi_{i_r}\|_{L_p}^p. \tag{19}$$

This can further be extended to a three-parameter norm $\|f(\mathbf{t})\|_{b_{p,q}^\nu}$, with $p\neq q$. The parameter p measures the sparsity within scale whereas the parameter q controls the decay rate across scales. This definition of Besov sequence spaces corresponds to a discretization from a continuous timescale analysis (Donoho *et al.*, 1998). As the second-generation wavelet transform is no longer a discretization of a continuous wavelet transform, there are several possible ways for further extension of the definition of Besov spaces to the case where $p\neq q$.

5.2. Stability

For proper processing, the set of wavelet functions should constitute a Riesz basis or stable basis in the infinitely dimensional space $L_2(\mathbb{R}^2)$. A Riesz basis is a basis that is almost orthogonal, in

the sense that the angles between any two vectors that are spanned by disjoint subsets of the set of basis functions are bounded from below. The formal definition is as follows.

Definition 2. Let $\{\phi_j, j=1,\ldots,\infty\}$ be a Schauder basis of the Hilbert space $\mathcal{H}$; then this is a Riesz basis if

(a) it is almost normalized, i.e. there are positive constants a and A so that $a \leqslant \|\phi_j\|^2_{\mathcal{H}} \leqslant A, \forall j \in \mathbb{N}$, and
(b) it is unconditional, i.e. there are positive constants c and C, so that, if $f = \Sigma_{j=1}^{\infty} w_j \phi_j$, then $c\|\mathbf{w}\|_2^2 \leqslant \|f\|^2_{\mathcal{H}} \leqslant C\|\mathbf{w}\|_2^2$.

The Riesz basis condition is difficult to check in the context of irregularly spaced data, since it depends on the subdivision of the irregular locations up to infinitely fine grids.

A necessary condition is that the multiscale grid is not arbitrarily inhomogeneous. Homogeneity in two dimensions is defined by the minimum angle in a triangulation. Let θ_{Δ_j} be the minimum angle in the triangulation at scale j; then we assume that, for all $j=1,\ldots,n$ and for $n \to \infty$, $\theta_{\Delta_j} > \theta^*_{\Delta}$, for some positive number θ^*_{Δ}.

A necessary, but far from sufficient (Jansen and Oonincx (2005), pages 88–89) condition for Riesz stability is that the one-level transforms (4) are uniformly bounded and boundedly invertible. We have the following result.

Proposition 2. Given a homogeneity constant θ^*_{Δ}, the one-level transforms of a lifting transform with local least squares prediction, inverse distance prediction or natural neighbour interpolation and with minimum norm update are uniformly bounded and boundedly invertible in j.

Proof. Thanks to the specific structure of lifting and its immediate invertibility property, it is sufficient that prediction and update vectors a and b are uniformly bounded. The number of non-zero entries in these vectors is bounded by $2\pi/\theta^*_{\Delta}$. Any norm of the vectors is bounded if all non-zero entries are bounded. The prediction coefficients a lie between 0 and 1 for the inverse distance prediction and for the natural neighbour prediction. It can be elaborated that they are bounded for the least squares prediction, thanks to the lower bound on θ_{Δ_j}.

For the update coefficients, we have the following result.

Lemma 1. If all $0 \leqslant a^r_j \leqslant 1$ and if i_r is the value of k that minimizes I_{rk}, then the update coefficients b^r_j, as defined in equation (15), satisfy $0 \leqslant b^r_j \leqslant \frac{1}{2}$. Similar bounds exist if a^r_j are bounded by values different from 0 and 1.

Proof. $b^r_j = I_{ri_r} I_{r-1,j} / \Sigma_{k \in J_r} I^2_{r-1,k}$ with $I_{r-1,j} = I_{rj} + a^r_j I_{ri_r}$. Clearly, $b^r_j > 0$, and

$$\|b^r\|_{\infty} = I_{ri_r} \frac{\max\limits_{k \in J_r}(I_{r-1,k})}{\sum\limits_{k \in J_r} I^2_{r-1,k}} = \frac{\max\limits_{k \in J_r}(I_{rj}/I_{ri_r} + a^r_j)}{\sum\limits_{k \in J_r}(I_{rj}/I_{ri_r} + a^r_j)^2}.$$

Note that linear reproduction (two vanishing moments) implies that $\Sigma_{j \in J_r} a^r_j = 1$. For $n_r = 1$, we have $a^1_j = 1$ and thus

$$b^1_r = \frac{1}{I_{r1}/I_{ri_r} + 1} \leqslant \frac{1}{2}.$$

For $n_r > 1$, setting $u = \max_{k \in J_r}(I_{rj}/I_{ri_r} + a^r_j)$, $\|b^r\|_{\infty}$ takes the shape of $f(u) = u/(u^2 + R^2)$ for $u > 1$ and with $R > 1$. It is straightforward to verify that $f(u) \leqslant \frac{1}{2}$.

5.3. Compression

Wavelet shrinkage relies on the ability of the underlying representation to compress functions into sparse representations. Section 5 of Jansen *et al.* (2008) exhibits simulations that show that our lifting methods (especially Voronoi) have compression abilities roughly in line with regular wavelets.

6. Bayesian shrinkage

Now consider the following model of observations subject to noise: $Z_i = f(\mathbf{t}_i) + \varepsilon_i$, where the noise ε_i is 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 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 that we have a parameter θ and an observation $Z \sim N(\theta, 1)$. In the wavelet smoothing case, θ would be an individual coefficient rescaled so that the empirical coefficient had unit variance. Following references such as Clyde *et al.* (1998), Abramovich *et al.* (1998) and Johnstone and Silverman (2004) the assumption that θ is a coefficient from an economical expansion is modelled by using a mixture prior for θ of the form

$$\theta \sim (1-\pi)\delta_0 + \pi\gamma \tag{20}$$

where γ is a symmetric density.

Johnstone and Silverman (2004) explored the advantages of using a heavy-tailed density for γ, such as the density

$$\gamma(u) = (2\pi)^{-1/2}\{1 - |u|\tilde{\Phi}(|u|)/\phi(u)\} \tag{21}$$

where $\tilde{\Phi}(u)$ is the upper tail probability of the standard normal distribution. This density has tails that decay as u^{-2}, which is the same weight as those of the Cauchy distribution. For this reason we refer to density (21) as the *quasi-Cauchy* density.

Suppose that $\theta \sim (1-\pi)\delta_0 + \pi\gamma$ 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 $f(Z) = \int\{(1-\pi)\,\delta_0(u) + \pi\,\gamma(u)\}\,\phi(z-u)\,\mathrm{d}u$.

6.2. Bayesian decision rule: posterior median

Once we have the expression for the posterior density $f_{\theta|Z}$, we have various choices of point estimates of θ. The posterior mean is popular, 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 be 0. 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 γ, this leads to a tractable expression for $\tilde{\theta}(z)$ in terms of the standard normal distribution and its inverse. See Johnstone and Silverman (2005a) for details and implementation.

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

6.3. Estimating the parameters (maximum likelihood estimation)

Suppose that we have a sequence θ_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 θ_i have independent prior distributions (20) all with the same value of π, and that the observations Z_i are themselves independent conditional on the θ_i. Let g be the convolution of γ 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) explored attractive features of a marginal maximum likelihood (ML) approach to the choice of π, chosen to maximize the log-likelihood $l(\pi) = \Sigma_i \log\{(1-\pi)\,\phi(z_i) + \pi\, g(z_i)\}$. This procedure is an empirical Bayes approach. First of all, the whole data set is used to estimate the parameter π. The estimated value is then used as a prior probability in model (20) and the inference is carried out for each coefficient separately. For theoretical and practical reasons, the maximization is usually carried out over a range of π 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 π to be constant within levels but to be allowed to vary between levels. For this, each level of the transform is treated separately by the marginal ML method, and an estimated parameter π_j is obtained for each level j. Typically, the parameter decreases as the resolution increases. At the levels corresponding to fine scale effects, the prior probability π_j is small and an observed coefficient must pass a high threshold in order not to yield an estimate of 0. At the coarser scale levels, a smaller threshold will usually be appropriate.

In the lifting case, the division into 'dyadic' levels is no longer appropriate, and instead some other possible approaches can be pursued. Overall, it can be assumed that the prior that is used for coefficient θ_i has probability π_i of being non-zero. The criterion for choosing the π_i is still the maximization of the marginal log-likelihood $l(\pi_1, \ldots, \pi_n) = \Sigma_i \log\{(1-\pi_i)\,\phi(z_i) + \pi_i\, g(z_i)\}$, but subject to appropriate constraints on the parameters π_i. Some possibilities are as follows.

(a) *Parametric dependence*: the coefficients are constrained to belong to a particular low dimensional parametric family. For example, for lifting we might constrain π_i to be proportional to the scale α_i, or perhaps to some power α_i^{λ}. This accords with the notion that there are singularities of some sort in the underlying function. If the singularities are points, α_i is proportional to the probability that the wavelet will encounter 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.

(b) *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 a quarter, an eighth, and so on, of the total number of coefficients in the order that is defined by the lifting scheme. An alternative is to group the coefficients by taking account of the values of their pseudoscales. For example, if α_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 $j \geqslant 1$, with the highest level consisting of all those coefficients with scales up to and including α_0.

(c) *Parametric dependence within artificial levels*: the simplest approach using artificial levels is to constrain π_i to be constant within levels. An alternative is to allow a parametric dependence, e.g. π_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 to interpolate the estimated π_i.

(d) *Monotone dependence*: conceptually the simplest constraint on the π_i would be to require only that π_i increases as the individual scale α_i increases. Because of the convexity properties of the log-likelihood function, estimation of π_i subject to this constraint can be carried out by 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 π_i are also piecewise constant functions of the scales α_i, and so this method indirectly splits the coefficients into levels, with constant π_i within each level. Further details are available from Johnstone and Silverman (2005b). See Fig. 2(d) for an example of using such an algorithm.

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

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

Then, by simple calculus, we have $\partial l/\partial \pi_i = \beta(z_i)\{1 + \pi_i\, \beta(z_i)\}^{-1}$, which is a decreasing function of π_i. Obviously, we always constrain $\pi_i \leqslant 1$. In addition, to avoid excessively high thresholds, and in line with the theory that was developed in Johnstone and Silverman (2004), we impose a lower limit on π_i corresponding approximately to a threshold value equal to the universal threshold $\sqrt{\{2\log(n)\}}$. For simplicity, we choose the lower limit π_{lo} to satisfy the condition $P[\theta_i = 0 | z_i = \sqrt{\{2\log(n)\}}] = \frac{1}{2}$, which is equivalent to setting $\pi_{\mathrm{lo}}^{-1} = 1 + (n-1)/2\log(n)$.

Details of the algorithms that were used to make the constrained ML choice of the π_i for the parametric and monotone dependence cases are set out in Johnstone and Silverman (2005a).

6.4. Parametric dependence within artificial levels

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

6.4.1. General set-up

Suppose that 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. To enforce the constraints $\pi_{\mathrm{lo}} \leqslant \pi_i \leqslant 1$ we refine this to

$$\pi_i(\zeta) = \mathrm{median}\{\pi_{\mathrm{lo}}, c_i\zeta, 1\}. \tag{22}$$

Letting g be the convolution of γ with ϕ, the marginal log-likelihood function is then given by

$$l(\zeta) = \sum_i \log[\{1 - \pi_i(\zeta)\}\, \phi(z_i) + \pi_i(\zeta)\, g(z_i)]. \tag{23}$$

By the definition of π_i there is no loss of generality in considering ζ only over the interval $[\zeta_{\mathrm{lo}}, \zeta_{\mathrm{hi}}]$, say, where $\zeta_{\mathrm{lo}} = \pi_{\mathrm{lo}}\ \max(c_i)^{-1}$, and $\zeta_{\mathrm{hi}} = \min(c_i)^{-1}$. If $\zeta < \zeta_{\mathrm{lo}}$ then all π_i will be π_{lo} and if $\zeta > \zeta_{\mathrm{hi}}$ then all π_i will be 1, regardless of how far outside the interval ζ lies.

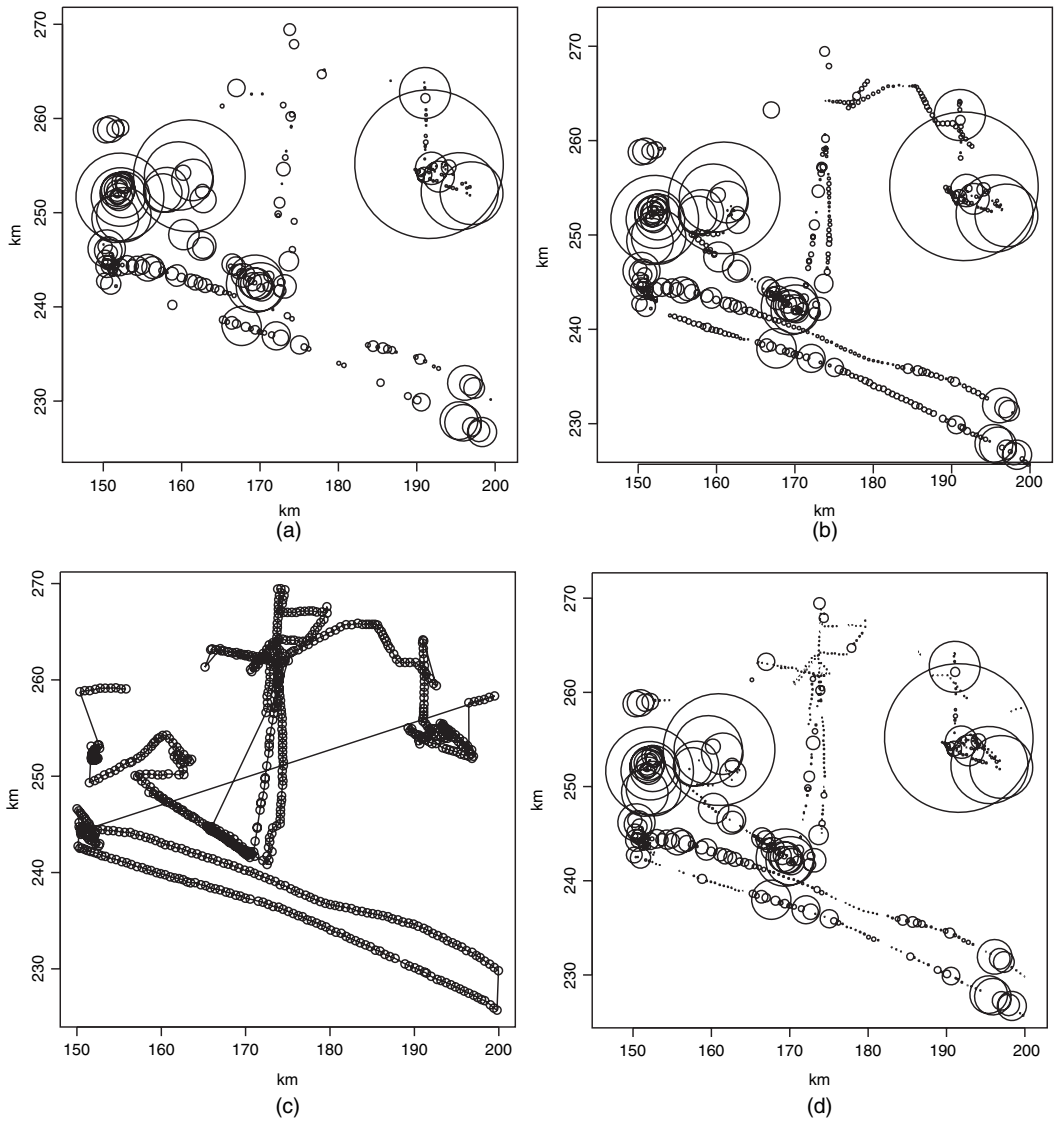

Fig. 2. Analyses of a selected portion of the krill data set (the radius of the circles, except in (c), encodes the square root of the krill density estimate in grams per square metre; the largest value is 14981 g m^{-2}): (a) krill density supplied by the British Antarctic Survey; (b) MST lifted estimate with least squares co-ordinate weights and eBayesThresh applied to lifting coefficients at all scales; (c) krill sample locations (○) and tree determined by the ship transect (———); (d) ship-determined transect tree-lifted estimate by using inverse distance weights and eBayesThresh with monotone dependence of π_j

6.4.2. For artifical levels

All the artificial levels cases reduce to the same general form. Within a particular level $\mathcal{L}$, we have equation (22), where c_i are known constants such as 1 or $\alpha_i^{1/2}$, and ζ is a parameter to be estimated. The likelihood $l_{\mathcal{L}}$ for the level $\mathcal{L}$ is now equation (23) but where the sum is now over $i \in \mathcal{L}$. In the straightforward artificial levels case, all the $c_i = 1$, and $l_{\mathcal{L}}$ is a concave function of ζ in $[\pi_{\mathrm{lo}}, 1]$. We have $l'_{\mathcal{L}}(\zeta) = \Sigma_{i \in \mathcal{L}} \beta(z_i)/\{1 + \zeta\, \beta(z_i)\}$, a decreasing function of ζ. By checking the signs of $l'_{\mathcal{L}}(\zeta)$ at the ends of the range it can be discovered whether $l_{\mathcal{L}}(\zeta)$ has its maximum

at one end or the other; if not, a binary search on the decreasing function $l'_{\mathcal{L}}(\zeta)$ will find the ML 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

7.1.1. Background

Goss and Everson (1996) reported 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. Goss and Everson (1996) stated that krill biomass determination is important because they are a basic part of the 'food web'. Krill are consumed by large numbers of birds, mammals and fish but are also increasingly being harvested for both human and animal consumption. As well as potential overfishing krill stocks are also under pressure from a variety of other sources such as sea temperature rise or increased ultraviolet penetration of seawater.

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 the sampling stations. Fig. 1 shows a selection from the transects and the sampled krill values along it. Fig. 2 shows a different portion of the krill data subjected to regression analyses using lifting with trees by using both least squares co-ordinate and inverse distance weights. Fig. 3 shows estimates that were obtained using Voronoi lifting.

7.1.2. Fitting

For all the regression estimates a small proportion of small negative values were replaced by 0. In all estimates many of the original zero data values have been replaced by very small intensity values. In Fig. 2 it is interesting to note the differences between the two estimates around the [175 km, 262 km] location. The estimate that is based on the MST estimates some 'lumps'

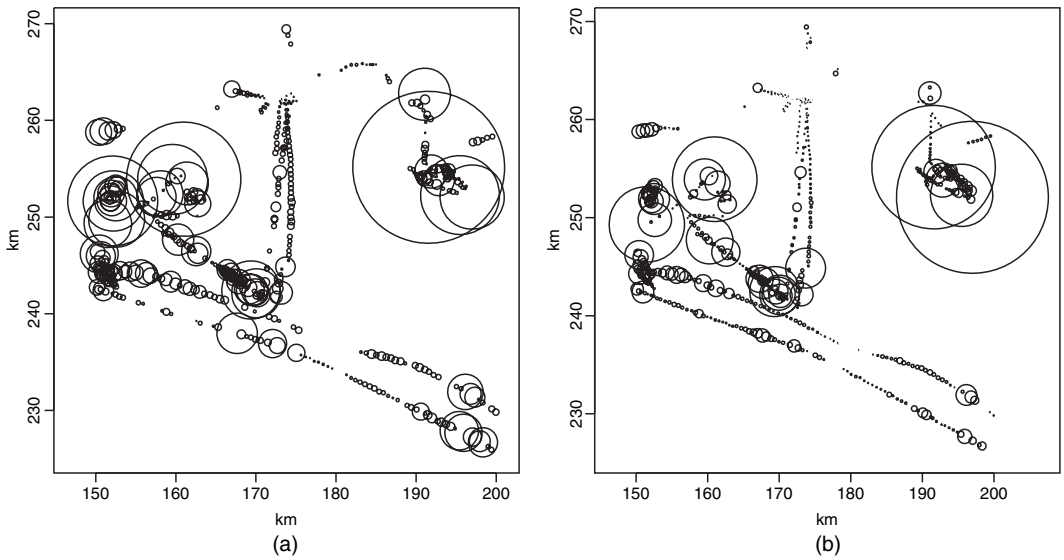

Fig. 3. Krill density estimates computed by using Voronoi least squares lifting with regular eBayesThresh: (a) estimate on the raw data; (b) estimate on log-transformed data

of intensity, whereas the estimate that is based on the ship's track estimates small intensities following the ship's path. There are at least two reasons for these differences:

(a) the ship's track only uses neighbours from the previous and next sample in the track whereas the MST algorithm will use nearest neighbours irrespectively of the track;
(b) the total time that the ship takes to cover points in the region (within a 25 km^2 box centred on [175 km, 262 km]) is approximately 12 h and the ship crosses near to the centre about five times and the actual krill density over this time may change.

With regard 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 estimation might be more appropriate.

7.1.3. Model verification

Let us take the MST lifted by using least squares co-ordinate weights analysis further. The estimate from this procedure is shown in Fig. 2(b). 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).

7.1.4. Comparisons

Our results directly contrast with those generated by LOESS and the MATLAB 'triogram' function. Neither of these methods dealt with the 'clumpiness' of the krill data at all well. Both methods smoothed out some features and missed others completely. Hence, their residuals also did not look satisfactory either. These results concur with our simulations in Section 7.2 below.

7.1.5. Physical interpretation

The likelihood maximization that was described in Section 6.3 results in piecewise constant thresholds (over scale), which are derived from the piecewise constant weight estimates π_i arising from the monotone dependence constraints. The thresholds are plotted in Fig. 5 of Jansen *et al.* (2008). The piecewise constant functions implicitly divide the scale space into a number of *data-defined resolution levels*. (For those who are familiar with regular wavelet methods, this 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 4.6×10^{-9} for the coarsest 345 coefficients. This means that wavelet coefficients that are in scale ranges from 0.8 km and up are essentially not thresholded. Another way of interpreting this, which is familiar to wavelet shrinkage researchers, is to say that 0.8 km is the 'primary resolution'. Finer scales than this receive monotonically higher thresholds in bands $[0.71, 0.8)$, $[0.58, 0.71)$, $[0.09, 0.58)$ and less than 0.09. The thresholds statistically indicate that there is little or no variation in the 'true' intensity pattern at less than 100 m and there is reduced variation at less than 600 m. 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. Fig. 3 shows two more estimates by 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.

Section 7.2 in Jansen *et al.* (2008) describes another example that is concerned with shrinkage of delays on part of the UK rail network via tree-based lifting.

7.2. Comparisons

7.2.1. Comparing our lifting methods with themselves and LOESS

We carried out a large simulation study with our new methods and compared them with LOESS by using R (see Cleveland and Devlin (1988) for more information on LOESS; see R Development Core Team (2008) for R). We evaluated these methods on two-dimensional analogues of the *Blocks*, *Bumps*, *Heavisine* and *Doppler* test functions that were introduced by Donoho and Johnstone (1994) and the piecewise linear function mfc. Pictures of the test functions appear in Fig. 4. 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 16×16 grid and adding independent and identically distributed Gaussian noise, varying amounts of jitter (distributed as Unif$[-\eta, \eta]$ for $\eta = 0.1, 0.01, 0.001$, and varying signal-to-noise ratios. Sensitivity to 'primary resolution' (the number of points that are removed in the lifting transform) was also explored. We also explored the performance of our different ways of carrying out our ML estimation as described in Section 6.3.

Fig. 4. Two-dimensional analogues of Donoho and Johnstone (1994) test functions: (a) Doppler; (b) Heavisine; (c) Blocks; (d) Bumps; (e) mfc (not an analogue)

Table 1. Medians (and median absolute deviations in parentheses) of 100 simulated sums-of-squares error values for LOESS, tree-based lifting using co-ordinate information and Voronoi-based lifting†

Signal	*Results for the following procedures:*		
	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)

†Jitter $\eta = 0.01$; signal-to-noise ratio 5; $n_g = 16^2$; monotone dependence EBayesThresh ($\times 1000$).

Table 1 shows a selection of results from Nason *et al.* (2004). We 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 median absolute deviation values); this is because of the fewer neighbours that it uses in constructing neighbours. The excellent performance of the Voronoi-based lifting is seen throughout all simulations. Primary resolution does not appear to influence performance dramatically but small differences appear, especially with the tree-based lifting. Likewise, among all the methods for carrying out ML estimation (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.

7.2.2. 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 with triograms by using the `quantreg` package.

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

$$\mathrm{gf}(x, y, \mathrm{horizon}) = (2x + y)\, \mathbb{I}\{\mathrm{horizon}(x, y) \leqslant 0\} + (10 - x)\, \mathbb{I}\{\mathrm{horizon}(x, y) > 0\}, \tag{24}$$

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

$$\begin{aligned} \mathrm{horizon_A}(x, y) &= 3x - y - 1, \\ \mathrm{horizon_B}(x, y) &= (x - \tfrac{1}{2})^2 + (y - \tfrac{1}{2})^2 - 1/16, \end{aligned} \tag{25}$$

and then our test functions are $\mathrm{mfa}(x, y) = \mathrm{gf}\{x, y, \mathrm{horizon_A}(x, y)\}$ and $\mathrm{mfb}(x, y)$ by replacing $\mathrm{horizon_A}$ by $\mathrm{horizon_B}$.

For each simulation in this section we generated 1000 (x, y) locations from a two-dimensional uniform density on $[0, 1] \times [0, 1]$. We then generated noisy observations by adding Gaussian noise with two signal-to-noise ratios of 18 dB and 15 dB. 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 signal-to-noise ratios*. Further experiments show that for very low signal-to-noise ratios triogram methods do better.

7.2.3. Comparing Voronoi lifting with thin plate splines and kriging

Heaton and Silverman (2008) compared our Voronoi lifting methodology, additionally equipped

Table 2. Mean averaged squared errors from 50 simulations for denoising of functions mfa and mfb by triogram and Voronoi lifting methods

Method	*Results for the following functions and levels of noise:*			
	mfa, 15 dB	*mfa, 18 dB*	*mfb, 15 dB*	*mfb, 18 dB*
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)

with an imputation method with both thin plate spline and kriging methodology, and showed that Voronoi lifting is competitive; see Section 8 for further information.

8. Conclusions and future possibilities

This paper 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 co-ordinate information in two dimensions, lifting with trees and graphs using co-ordinate information and lifting with graphs using interpoint 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. Some theoretical aspects have been discussed. We have also demonstrated the utility of our techniques both on the krill data (where ships' track information can optionally be used) and simulated data.

A further innovation would be to choose from among 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 built on Jansen *et al.* (2001) and preprint versions of this paper by permitting a choice of regression order (linear, quadratic or cubic) and/or number of neighbours that are involved in prediction. Nunes *et al.* (2006) provided a full literature review of adaptive lifting and a comprehensive simulation study, which shows that one-dimensional adaptive lifting one coefficient at a time produces extremely good compression and non-parametric regression results when compared with *locfit* (Loader, 1997, 1999), the `smooth.spline()` function in R and the irregular wavelet shrinkage algorithm by Kovac and Silverman (2000). Our methods can be developed further to cope with heteroscedastic variance by using ideas that are similar to those proposed by Kovac and Silverman (2000) as demonstrated in one dimension 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.

As well as estimating true values from a noisy function (either irregularly spaced or on a network) on a given set of points we might also wish to estimate the function at a new set of points. Heaton and Silverman (2008) described a method that imputes the value of the function at a set of sites given information from another set of sites by using the Bayesian lifting model that we present above using the Gibbs sampler. They demonstrated their method successfully both with regular wavelet shrinkage and also on simulated and real data using our two-dimensional Voronoi lifting. 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 detailed simulations and comparisons need to be performed to explore thoroughly the utility of these methods. Other questions along these lines remain—e.g. how to deal with locations that disappear when we are modelling data structures through time.

Another important possibility would be to model more accurately the variance and correlation between lifting coefficients, ideally in a computationally efficient way. Such a possibility could be incorporated in the empirical Bayes paradigm, but issues of computational efficiency would have to be dealt with. This leads on to the 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) by using our lifting techniques.

Acknowledgements

The authors thank the following people: Cathy Goss and Inigo Everson of the British Antarctic Survey for supplying them with the krill data, for helpful conversations and advice concerning the purposes of the study; 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); Roger Koenker for supplying the MATLAB version of his `quantreg` package; Matt Nunes for translating the Voronoi lifting code from MATLAB to R. GPN was partially supported by Engineering and Physical Sciences Research Council Advanced Research Fellowship AF/001664 and grant GR/D005221/1. All the authors were supported by Engineering and Physical Sciences Research Council research grant M10229. Three R packages (`NetTree`, `Liftvor` and `PicTree`) that carry out the three different kinds of lifting that were 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. C. and Sapatinas, T. (2000) Wavelet analysis and its statistical applications. *Statistician*, **49**, 1–29.

Abramovich, F., Sapatinas, T. and Silverman, B. W. (1998) Wavelet thresholding via a Bayesian approach. *J. R. 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., Gregoire, G. and Vial, P. (1997) Random design wavelet curve smoothing. *Statist. Probab. 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 Comput. Sci.*, **3120**, 624–638.

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

Chua, D., Kolaczyk, E. D. and Crovella, M. (2006) Network kriging. *IEEE J. Selectd Areas Communs*, **24**, 2263–2272.

Claypoole, R. L., Baraniuk, R. G. and Nowak, R. D. (2003) Nonlinear wavelet transforms for image coding via lifting. *IEEE Trans. Image Process.*, **12**, 1513–1516.

Cleveland, W. S. and Devlin, S. (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. (1993) *Statistics for Spatial Data*. New York: Wiley.

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

Daubechies, I., Guskov, I. and Sweldens, W. (2001) Commutation for irregular subdivision. *Constr. Approximn*, **17**, 479–514.

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

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

Delouille, V., Jansen, M. and von Sachs, R. (2003) Second generation wavelet methods for denoising of irregularly spaced data in two dimensions. *Signal Process.*, **86**, 1435–1450.

Delouille, V. and von Sachs, R. (2002) Smooth design-adapted wavelets for half-regular designs in 2D. *Technical Report 0226*. Institut de Statistique, Université catholique de Louvain, Louvain-la-Neuve.

Delouille, V., Simoens, J. and von Sachs, R. (2004) Smooth design-adapted wavelets for nonparametric stochastic regression. *J. Am. Statist. Ass.*, **99**, 643–658.
Denison, D. G. T., Holmes, C. C., Mallick, B. and Smith, A. (2002) *Bayesian Methods for Nonlinear Classification and Regression*. London: Wiley.
Donoho, D. L. and Johnstone, I. M. (1994) Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, **81**, 425–455.
Donoho, D. L., Vetterli, M., DeVore, R. A. and Daubechies, I. (1998) Data compression and harmonic analysis. *IEEE Trans. Inform. Theory*, **44**, 2435–2476.
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. Comput. Graph. Statist.*, **13**, 621–638.
Goss, C. and Everson, I. (1996) An acoustic survey of Antarctic krill on the South Georgia shelf. *Scientific Abstract WG–EMM–96/42*. Commission for the Conservation of Antarctic Marine Living Resources, Hobart.
Green, P. J. and Silverman, B. W. (1993) *Nonparametric Regression and Generalized Linear Models*. London: Chapman and Hall.
Hall, P. and Turlach, B. (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. Ass.*, **93**, 101–119.
Heaton, T. J. and Silverman, B. W. (2008) A wavelet- or lifting-scheme-based imputation method. *J. R. Statist. Soc.* B, **70**, 567–587.
Herrick, D. R. M. (2000) Wavelet methods for curve estimation. *PhD Thesis*. University of Bristol, Bristol.
Herrmann, E., Wand, M. P., Engel, J. and Gasser, T. (1995) A bandwidth selector for bivariate kernel regression. *J. R. Statist. Soc.* B, **57**, 171–180.
Jaffard, S. (1991) Pointwise smoothness, two-microlocalisation and wavelet coefficients. *Publ. Mat.*, **35**, 155–168.
Jansen, M. (2006) Multiscale Poisson data smoothing. *J. R. Statist. Soc.* B, **68**, 27–48.
Jansen, M. (2007) Refinement independent wavelets for use in adaptive multiresolution schemes. To be published.
Jansen, M., Nason, G. P. and Silverman, B. W. (2001) Scattered data smoothing by empirical Bayesian shrinkage of second generation wavelet coefficients. *Proc. SPIE*, **4478**, 87–97.
Jansen, M., Nason, G. and Silverman, B. W. (2008) Multiscale methods for data on graphs and irregular multidimensional situations. *Technical Report 08:07*. Department of Mathematics, University of Bristol, Bristol.
Jansen, M. and Oonincx, P. (2005) *Second Generation Wavelets and Applications*. Berlin: Springer.
Johnstone, I. M., Kerkyacharian, G., Picard, D. and Raimondo, M. (2004) Wavelet deconvolution in a periodic setting (with discussion). *J. R. Statist. Soc.* B, **66**, 547–573.
Johnstone, I. M. and Silverman, B. W. (1997) Wavelet threshold estimators for data with correlated noise. *J. R. 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. Statist. Softwr.*, **12**, 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*. New York: Springer.
Koenker, R. and Mizera, I. (2004) Penalized triograms: total variation regularization for bivariate smoothing. *J. R. Statist. Soc.* B, **66**, 145–163.
Kohler, M. (2003) Nonlinear orthogonal series estimation for random design regression. *J. Statist. Planng Inf.*, **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. Image Process.*, **9**, 480–496.
Krzanowski, W. J. and Marriott, F. H. C. (1995) *Multivariate Analysis*, part 2, *Classification, Covariance Structures and Repeated Measurements*. London: Arnold.
Loader, C. (1997) Locfit: an introduction. *Statist. Comput. Graph. News*, **8**, 11–17.
Loader, C. (1999) *Local Regression and Likelihood*. New York: Springer.
Mallat, S. G. (1989) A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattn Anal. Mach. Intell.*, **11**, 674–693.
Mallat, S. G. (1998) *A Wavelet Tour of Signal Processing*. San Diego: Academic Press.
Nason, G. P. (2002) Choice of wavelet smoothness, primary resolution and threshold in wavelet shrinkage. *Statist. Comput.*, **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, Bristol.
Nason, G. P., von Sachs, R. and Kroisandt, G. (2000) Wavelet processes and adaptive estimation of the evolutionary wavelet spectrum. *J. R. Statist. Soc.* B, **62**, 271–292.

Neumann, M. and von Sachs, R. (1995) Wavelet thresholding: beyond the Gaussian iid situation. *Lect. Notes Statist.*, **103**.
Nunes, M., Knight, M. and Nason, G. P. (2006) Adaptive lifting for nonparametric regression. *Statist. Comput.*, **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. Probab.*, **24**, 1903–1925.
Penrose, M. D. and Yukich, J. E. (2003) Weak laws of large numbers in geometric probability. *Ann. Appl. Probab.*, **13**, 277–303.
Pensky, M. and Vidakovic, B. (2001) On non-equally spaced wavelet regression. *Ann. Inst. Statist. Math.*, **53**, 681–690.
Percival, D. B. and Walden, A. T. (2000) *Wavelet Methods for Time Series Analysis*. Cambridge: Cambridge University Press.
R Development Core Team (2008) *R: a Language and Environment for Statistical Computing*. Vienna: R Foundation for Statistical Computing.
Rue, H. and Held, L. (2005) *Gaussian Markov Random Fields*. London: Chapman and Hall.
Sardy, S., Percival, D. B., Bruce, A. G., Gao, H.-Y. and Stuetzle, W. (1999) Wavelet shrinkage for unequally spaced data. *Statist. Comput.*, **9**, 65–75.
Sibson, R. (1980) A vector identity for Dirichlet tessellation. *Math. Proc. Camb. Phil. Soc.*, **87**, 151–155.
Sibson, R. (1981) A brief description of natural neighbour interpolation. In *Interpreting Multivariate Data* (ed. V. Barnet), pp. 21–36. Chichester: Wiley.
Silverman, B. and Vassilicos, J. (2000) *Wavelets: the Key to Intermittent Information*. Oxford: Oxford University Press.
Smola, A. J. and Kondor, R. (2003) Kernels and regularization on graphs. *Lect. Notes Artif. Intell.*, **2777**, 144–158.
Sweldens, W. (1996) Wavelets and the lifting scheme: a 5 minute tour. *Z. Angew. Math. Mech.*, **76**, 41–44.
Uytterhoeven, G. and Bultheel, A. (1997) The red-black wavelet transform. *Technical Report TW 271*. Department of Computer Science, Katholieke Universiteit Leuven, Leuven.
Vanraes, S., Jansen, M. and Bultheel, A. (2002) Stabilised wavelet transform for non-equispaced data smoothing. *Signal. Process.*, **82**, 1979–1990.
Vidakovic, B. (1999) *Statistical Modeling by Wavelets*. New York: Wiley.
Wahba, G. (1990) *Spline Models for Observational Data*. Philadelphia: Society for Industrial and Applied Mathematics.