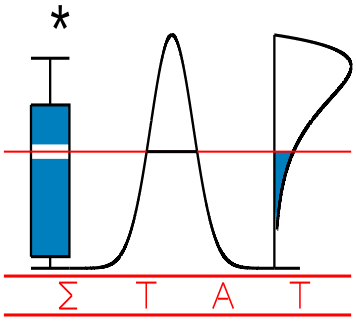
T E C H N I C A L
R E P O R T

0303

SECOND GENERATION WAVELET METHODS FOR DENOISING OF IRREGULARLY SPACED DATA IN TWO DIMENSIONS

V. DELOUILLE, M. JANSEN and R. von SACHS



I A P S T A T I S T I C S
N E T W O R K

INTERUNIVERSITY ATTRACTION POLE

http://www.stat.ucl.ac.be/IAP

Second generation wavelet methods for denoising of irregularly spaced data in two dimensions

V. Delouille[1,4] M. Jansen[2] R. von Sachs[3,4]

January 16, 2003

Abstract

We treat bivariate nonparametric regression, where the design of experiment can be arbitrarily irregular. Our method uses second-generation wavelets built with the lifting scheme: Starting from a simple initial transform, we propose to use some predictor operators based on a generalization in two dimensions of the Lagrange interpolating polynomial. These predictors are meant to provide a smooth reconstruction. Next, we include an update step which helps to reduce the correlation amongst the detail coefficients, and hence stabilizes the final estimator. We use a Bayesian thresholding algorithm to denoise the empirical coefficients, and we show the performance of the resulting estimator through a simulation study.

Keywords: biorthogonal wavelet transform, irregular two-dimensional design, lifting scheme, relaxation.

1 Introduction

Nonparametric curve estimation by wavelets has been widely used to treat the univariate model

$$Y_i = m(X_i) + \epsilon_i,\ i = 1, \ldots, n\ ,$$

where $\epsilon_1, \ldots, \epsilon_n$ are noise. In case the regressors X_i are equispaced and the sample size n is a power of two, classical wavelet methods can be applied, and the nonlinear thresholding schemes of Donoho and Johnstone [15, 12, 13, 14] have proved powerful to obtain estimators which perform well even in the presence of discontinuities (jumps or cusps) in the underlying function m. However, the simplicity of the standard wavelet algorithms is largely lost when, for example, the design on which the regression curve is observed is no longer fixed and equidistant, or the sample size is no longer a power of two. Consequently, a large number of

[1]Research Assistant of the National Fund for Scientific Research – Wallonia, Belgium (F.N.R.S.). Université catholique de Louvain, Institut de Statistique, Louvain-la-Neuve, Belgium

[2]Assistant professor, TU Eindhoven, Department of Mathematics and Computer Science, and Postdoctoral Researcher with the Fund of Scientific Research – Flanders, Belgium (F.W.O.), K.U.Leuven, Department of Computer Science

[3]Professor. Université catholique de Louvain, Institut de Statistique, Louvain-la-Neuve, Belgium

[4]Financial support from the contract 'Projet d'Actions de Recherche Concertées' nr. 98/03–217 from the Belgian government, and from the IAP research network nr. P5/24 of the Belgian State (Federal Office for Scientific, Technical and Cultural Affairs) is gratefully acknowledged.

papers have been devoted to finding some procedures that remove these restrictions. Most of these methods use some preliminary step to get back to the equidistant design situation and then apply a traditional orthogonal wavelet transform [3, 18, 24, 11, 1]. Note that these methods do not easily generalize beyond the one-dimensional setting.

In contrast, the algorithms presented in [9, 31] use some biorthogonal wavelet transforms that deal directly with the irregularity of the grid. The filters associated to these transforms are built with the *lifting scheme* [29, 4], a general tool for the construction of wavelet transforms on arbitrary domain. Such transforms are said to be of 'second-generation' [29]. The method presented in [9] easily generalizes to half-regular grids, which are built by taking the Cartesian product of two irregular one-dimensional grids [10].

In this paper, we consider the nonparametric regression problem where the design of experiment is bivariate and arbitrarily irregular. Our aim is to provide a good estimation even in the presence of discontinuities in the signal. To achieve this aim, we propose some wavelet-type estimators for the model

$$Z_k = m(X_k, Y_k) + \epsilon_k, \;\; k = 1, \ldots, n \;, \tag{1.1}$$

where the errors ϵ_k are independent of (X_k, Y_k) with $E(\epsilon_k) = 0, \mathrm{Var}(\epsilon_k) = \sigma_\epsilon^2$.

Similarly to the procedures proposed in [9, 31] in the irregular one-dimensional case, we use the lifting scheme to construct some multiscale transforms adapted to the grid. Indeed, the lifting scheme allows to build wavelet filters in the time-domain as follows. First, an initial, often trivial, transform is performed. Next, a *predictor* operator is chosen to produce better detail coefficients. Finally, the values of the scaling coefficients are *updated*. In [20], Jansen, Nason and Silverman have already treated the model (1.1) using a wavelet transform built with the lifting scheme. The transform proposed in [20] uses a linear least square prediction in order to compute the detail coefficients, and a so-called 'minimum norm' update.

The contribution of the present paper is twofold. First, we propose to use some predictor operators that are based on *relaxation*, a possible generalization in two dimensions of the Lagrange interpolating polynomial. Note that, in the context of data smoothing, the use of predictor operators that are based on a relaxation formula is new. Some stability problems may arise when using a predictor based on a relaxation with third order differences [16]. We propose here some ways to stabilize this prediction. Second, we use an update operator which performs a local semi-orthogonalization. Similarly to what was observed in [9], some simulation study shows that bringing the biorthogonal transform closer to a semi-orthogonal one with such an update improves the quality of the denoising procedure, and hence of the resulting estimator.

In order to obtain a wavelet estimator, we adopt the Bayesian thresholding algorithm proposed by Johnstone and Silverman in [22, 23] since it has already proved to be efficient to treat data coming from the model (1.1), see [20] for more details. The proposed wavelet transforms, coupled with this Bayesian denoising, produce several estimators for the regression function m in the model (1.1). We compare these estimators through a simulation study. In contrast to more traditional nonparametric regression techniques such as, e.g., locally weighted regression, wavelet methods offer a *multiscale representation* of the data. This means that the original data can be described by the scaling coefficients living at coarsest scale together with the detail coefficients that have survived the Bayesian thresholding.

The remaining part of this paper is organized as follows. In Section 2, we briefly recall the basic setting of a multiscale decomposition in the second generation context, and we introduce the lifting scheme in its general form. Section 3 explains the particular instance of the lifting scheme needed in our irregular bivariate setup, whereas in Section 4, we detail the

construction of our proposed predictor and update operators. Section 5 briefly recalls the Bayesian algorithm of Johnstone and Silverman [22, 23] and the different wavelet estimators produced. Finally, the simulation study of Section 6 compares the performance of these estimators.

2 Preliminaries

2.1 Multiscale decomposition

We first briefly recall the basic setting of a multiscale decomposition. More can be found in [5, 29]. Consider a *multiresolution analysis* (*MRA*) in a general $\mathbf{L}_2$ space, i.e. a strictly increasing and dense sequence $\mathcal{V} := \{V_j\}_{j \geq j_0}$ of closed subspaces of $\mathbf{L}_2$,

$$V_j \subset V_{j+1}\,,\ j \in \mathbb{Z}\,, \quad \text{and} \quad \operatorname{clos} \bigcup_{j=j_0}^{\infty} V_j = \mathbf{L}_2 \ \text{ for any } \ j_0 \ \in \ \mathbb{N}\,.$$

Between successive spaces in $\mathcal{V}$, construct algebraic complements W_j such that

$$V_{j+1} = V_j \oplus W_j\ ,$$

where '$\oplus$' denotes the inner sum of disjoint linear spaces. The complement space W_j is not necessarily orthogonal to V_j. A fine-resolution space V_j can then be written as a telescopic decomposition into a coarser-resolution space and intermediate complement spaces,

$$V_j = V_{j_0} \oplus \bigoplus_{i=j_0}^{j-1} W_i\ . \tag{2.1}$$

With the notational convention that $W_{j_0-1} := V_{j_0}$, we call the sequence $\{W_j\}_{j \geq j_0 - 1}$ a *multiscale decomposition* (*MSD*).

The spaces V_j and W_j are equipped with bases,

$$\begin{aligned} V_j &= \operatorname{clos}_{\mathbf{L}_2} \operatorname{span} \Phi_j, \quad \Phi_j := \{\, \varphi_{jk} \mid k \in \mathcal{K}_j \,\}\ , \\ W_j &= \operatorname{clos}_{\mathbf{L}_2} \operatorname{span} \Psi_j, \quad \Psi_j := \{\, \psi_{jm} \mid m \in \mathcal{M}_j \,\}\ , \end{aligned}$$

with $\mathcal{K}_j$ and $\mathcal{M}_j$ some index sets. Any basis Φ_j for the space V_j is called a set of *scaling functions*, and any basis Ψ_j for any type of complement space W_j, $j \geq 0$, is a set of *wavelets* at level j.

Due to the multiresolution structure of the spaces V_j and W_j, there exist refinement coefficients $\{h_{jlk}\}$ and $\{g_{jlm}\}$ such that

$$\varphi_{jk} = \sum_{l \in \mathcal{K}_{j+1}} h_{jlk}\, \varphi_{j+1,l}\ , \quad \psi_{jm} = \sum_{l \in \mathcal{K}_{j+1}} g_{jlm}\, \varphi_{j+1,l}\ . \tag{2.2}$$

It is convenient to rewrite the refinement relations (2.2) as matrix expressions

$$\Phi_j = \Phi_{j+1} H_j \quad \text{and} \quad \Psi_j = \Phi_{j+1} G_j\ , \tag{2.3}$$

where Φ_j is a row vector containing the functions $\{\varphi_{jk}, k \in \mathcal{K}_j\}$.

Biorthogonal bases generalize orthogonal bases in that different bases are used for analysis and for synthesis of the signal: a primal MSD, generated by the basis $\Phi_{j_0} \cup \bigcup_{j=j_0}^{\infty} \Psi_j$ is used to

reconstruct the signal, whereas a dual MSD, with spaces $\tilde{V}_{j_0}$ and $\tilde{W}_j$ spanned by $\tilde{\Phi}_{j_0}$ and $\tilde{\Psi}_j$, respectively, is used for decomposition. Both MSD's must be linked through biorthogonality relations, i.e.,

$$\begin{aligned} \left\langle \varphi_{jk} \, , \tilde{\varphi}_{jk'} \right\rangle &= \delta_{kk'} \; \forall \, j \; , & \left\langle \varphi_{jk} \, , \tilde{\psi}_{jm'} \right\rangle &= 0 \; \forall \, j, k, m' \; , \\ \left\langle \psi_{jm} \, , \tilde{\varphi}_{jk'} \right\rangle &= 0 \; \forall \, j, m, k' \; , & \left\langle \psi_{jm} \, , \tilde{\psi}_{jm'} \right\rangle &= \delta_{mm'} \; \forall \, j \; , \end{aligned} \tag{2.4}$$

where $\delta_{kk'}$ is the Kronecker symbol.

In the univariate case, if there exists a dual MSD satisfying (2.4), the resulting pair of biorthogonal wavelet bases allows the following decomposition of any function m.

$$m(x) = \sum_k s_{j_0,k} \varphi_{j_0,k}(x) + \sum_{j,k} \langle m \, , \tilde{\psi}_{jk} \rangle \, \psi_{jk}(x) \; , \tag{2.5}$$

where $s_{j_0,k} := \langle m \, , \tilde{\varphi}_{j_0,k} \rangle$ and $d_{j,k} := \left\langle m \, , \tilde{\psi}_{jk} \right\rangle$.

The dual refinement operators $\tilde{H}_j$ and $\tilde{G}_j$ are defined analogously to (2.3):

$$\tilde{\Phi}_j = \tilde{\Phi}_{j+1} \tilde{H}_j \quad \text{and} \quad \tilde{\Psi}_j = \tilde{\Phi}_{j+1} \tilde{G}_j \; . \tag{2.6}$$

Inserting the refinement relations in (2.4) gives the conditions to have *biorthogonal filters*

$$\begin{aligned} \tilde{H}_j^* H_j &= I \; , & \tilde{G}_j^* H_j &= 0 \; , \\ \tilde{H}_j^* G_j &= 0 \; , & \tilde{G}_j^* G_j &= I \; , \end{aligned} \tag{2.7}$$

with $\tilde{H}_j^*$ denoting the Hermitian conjugate of H_j^*. Let $s_j := \{ s_{j,k}, k \in \mathcal{K}_j \}$ and $d_j := \{ d_{j,k}, k \in \mathcal{M}_j \}$. The forward wavelet tranform is given by

$$s_j = \tilde{H}_j^* s_{j+1} \quad ; \quad d_j = \tilde{G}_j^* s_{j+1} \; ,$$

and the inverse transform is achieved as follows.

$$s_{j+1} = H_j s_j + G_j d_j \; .$$

The order of a MRA in the second-generation setting is defined as follows.

Definition 2.1. *The* (polynomial) order *of a univariate MRA is given by*

$$\tilde{N} := \max\{ \, n \mid \exists j_\star \text{ such that } \forall j \geq j_\star : \Pi_{n-1} \subset V_j \, \} \; ,$$

where Π_{n-1} *is the space of polynomials of degree at most* $n-1$.

When the order of a MRA is $\tilde{N}$ and an associated dual MSD exists, the analyzing wavelets $\tilde{\psi}_{jm}$ with $j \geq j_\star$ have $\tilde{N}$ vanishing moments, i.e., the space $\tilde{W}_j$ is orthogonal to $\Pi_{\tilde{N}-1}$ for $j \geq j_\star$. Symmetrically, the primal wavelets are said to have N vanishing moments if, for j greater than a given level j', W_j is orthogonal to Π_{N-1}.

2.2 Lifting scheme in general

Starting from two initial pairs of biorthogonal filter (H_j^o, G_j^o) and $(\tilde{H}_j^o, \tilde{G}_j^o)$, it is possible to gradually improve the properties of these filters using the lifting scheme [8, 28, 29]. Note that, often we have $H_j^o = \tilde{H}_j^o$ and $G_j^o = \tilde{G}_j^o$. The lifting scheme states that, for any operator P_j, new pairs of biorthogonal filters can be found as

$$(H_j := H_j^o + G_j^o P_j\,,\ G_j := G_j^o) \quad \text{and} \quad (\tilde{H}_j := \tilde{H}_j^o\ \,, \tilde{G}_j := \tilde{G}_j^o - \tilde{H}_j^o P_j^*)\,. \tag{2.8}$$

It suffices to check that the biorthogonality relations are still satisfied. In this operation, one obtains new primal refinement matrices H_j, while the dual refinement matrices $\tilde{H}_j^o$ remain unchanged.

Naturally, the roles of the primal and the dual side in (2.8) can be interchanged, giving for any operator U_j the new biorthogonal filter pairs

$$(H_j := H_j^o\,,\ G_j := G_j^o - H_j^o U_j) \quad \text{and} \quad (\tilde{H}_j := \tilde{H}_j^o + \tilde{G}_j^o U_j^*\,,\ \tilde{G}_j := \tilde{G}_j^o)\,. \tag{2.9}$$

The operations (2.8) and (2.9) are commonly called *prediction* and *update* steps, respectively.

In this paper, we propose a wavelet-type transform adapted to an irregular bidimensional grid. Coupled with a Bayesian thresholding scheme, this wavelet-type transform allows us to estimate the regression function $m(.)$ of the model (1.1) at the data points $\{(X_k, Y_k)\}_{k=1}^n$.

Our wavelet transform uses finite biorthogonal filters built with a particular instance of the lifting scheme. To present this transform, we define some compactly supported scaling functions $\{\Phi_J, \tilde{\Phi}_J\}$ at the finest level J (the level of the observations). Using refinements relations (2.3)- (2.6) and appropriate filters we obtain an expression for the scaling functions $\Phi_j, \tilde{\Phi}_j$ and for the wavelet functions $\Psi_j, \tilde{\Psi}_j$ at coarser levels $j = J-1, J-2, \ldots$. Since our filters are finite, all these functions are compactly supported.

Note that the lifting scheme itself does not assure that these scaling and wavelet functions form a biorthogonal basis that generates a proper function space. To prove that Φ_j is a proper basis for V_j, it is necessary to first show the existence of the limit of the subdivision scheme $a_{j+1} = H_j a_j, j \geq j_0$, with $a_{j_0} = \{a_{j_0,k}\} = \{\delta_{k,k_0}\}$ for each $k_0 \in \mathbb{N}$. Such a proof is beyond the scope of this paper. In the univariate case on irregular grids, Daubechies *et al.* analyze in [6, 7] the existence of this limit.

Throughout the paper, we denote by $\Phi_{j,I} = \{\varphi_{j,k} | k \in I\}$ the set of scaling functions whose location indices are in I. When dealing with filters, scaling or wavelet functions, the subscript j will always denote the level. The location index of the basis function is given by the index of the vertex on which this function is defined. For example on the grid χ_j at level j, $\varphi_{j,k}$ indicates the scaling functions associated to the vertex k. Note that, with this convention, a vertex keeps its location index across scales.

3 Lifting scheme for second generation wavelets in two dimensions

3.1 Neighborhood of a point and definition of the scales

In a two-dimensional arbitrary grid, the notion of neighborhood is no more straightforward. In order to define the connectivity between the vertices, we use Delaunay's triangulation on the irregular experimental design [2, 25]. The Delaunay triangulation of a set of vertices in the plane is a set of triangles connecting the vertices and satisfying an 'empty circle' property:

the circumcircle (that is, the circle that goes through the three vertices of a given triangle) of each triangle does not contain any points of the triangulation.

The Delaunay triangulation is the dual structure of a set of cells, called the Voronoi diagram. Each vertex of the plane is associated to a cell in the Voronoi diagram, which is built as follows: the Voronoi cell of the vertex v contains all locations that are closer to v than to any other vertices in the plane. Indeed, each edge of the Voronoi cell is the bisector of an edge from the Delaunay triangulation. This shows the connection between the Delaunay triangulation and the Voronoi diagram: the circle circumscribed about a Delaunay triangle has its center at the corner of a Voronoi polygon, see Figure 1.

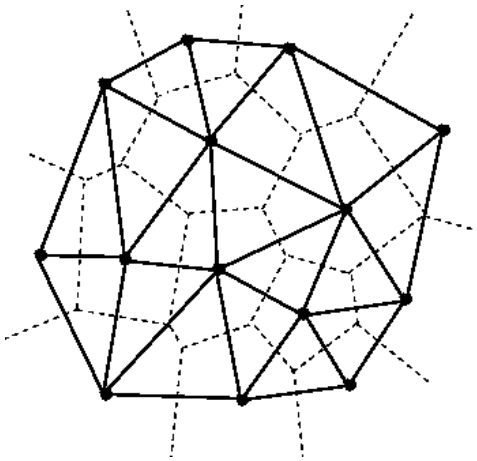


Figure 1: A Delaunay triangulation (plain line) with its dual structure, the Voronoi diagram (dashed line). The points of the grid are represented with a circle. Each edge of a Voronoi polygon is the bisector of an edge of the Delaunay triangulation, hence the center of the circle circumscribed about a Delaunay triangle is a corner of a Voronoi polygon. This representation is taken from [25].

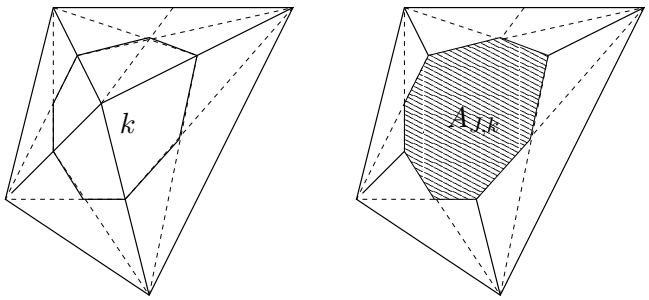
In data smoothing with wavelets, the finest level is given by the observations, that is the finest level grid χ_J is made of the design points.

In one dimension, starting from the finest level, it is always possible to begin the lifting scheme with the Lazy transform, that is, to split the data into 'even' and 'odd' points: if $\mathcal{K}_J := \{0, \ldots, n-1\}$ is the index set at finest level, $\mathcal{K}_{J-1} := \{2k, k = 0, 1, \ldots, \lceil n/2 \rceil - 1\}$ is the index set of the 'even' points and $\mathcal{M}_{J-1} = \{2k+1,\ k = 0, \ldots, \lfloor (n)/2 \rfloor - 1\}$ contains the indices of the 'odd' points. At level $J-2$, the set $\mathcal{K}_{J-2}$ is further split into two, and so on. By doing so, we impose a dyadic structure on the levels.

In two dimensions on irregular lattices such a dyadic structure is no longer trivial or even possible. Instead, we have to find a consistent way of splitting the data and of defining the different levels. We want to produce a multilevel grid, that is, a sequence of grids $(\chi_j)_{j=0}^J$ such that $\chi_j \subset \chi_{j+1}$.

At the level of the data, the scaling function $\varphi_{J,k}(x)$ corresponding to the vertex k is approximated by the indicator (or characteristic) function of a cell, centered at k and built as follows. Take one triangle τ for which k is a corner, and draw its three medians, which meet at the barycenter b. Consider the part of the median that starts at the middle of an edge to whom k belongs and stops at the barycenter. From this barycenter b, follow the part of this other median that starts at the middle of the second edge of τ to whom k belongs

and stops at the same barycenter b. Repeat this operation with all the neighboring triangles of the vertex k. Figure 2 shows an example of this construction of a cell associated to the scaling function $\varphi_{J,k}$. The area of this cell is easily computed: attribute one third of the area of each triangle (in the Delaunay triangulation) to each of its vertices. The sum of all contributions to a given vertex k gives the area of the cell associated to $\varphi_{J,k}(x)$, that is, it gives the integral of $\varphi_{J,k}(x)$. We denote this integral $A_{J,k}$. Note that, in the case of a regular, hexagonal lattice, this procedure produces cells corresponding to the Voronoi polygons.



(a) (b)

Figure 2: Determination of the support of the scaling function $\varphi_{J,k}$ associated to the vertex k. The medians of the triangles for which k is a corner are drawn. Consider the line which starts at the middle of an edge to whom k belongs and that ends at the barycenter of one (of the two) neighboring triangle. This line is an edge of the polygonal which delimitates the support of $\varphi_{J,k}$. In (b), the support of $\varphi_{J,k}$, which has an area $A_{J,k}$ is represented with a filled pattern.

The dual scaling functions $\tilde{\varphi}_{J,k}$ are the Dirac functions, and hence the scaling coefficients $\{s_{J,k}\}$ at the beginning of the transform are just the observations $\{Z_k\}$.

We now present the particular instance of lifting scheme that we use. This requires to define a type of hierarchy amongst the grids $\{\chi_j\}$.

3.2 Prediction of one scaling coefficient per scale

In the algorithms proposed in this paper, at a given scale j, only one scaling coefficient is being predicted before its immediate neighbors are updated. This means that $\chi_{j+1} \setminus \chi_j = \{v_0\}$, where v_0 is the vertex whose scaling function φ_{j+1,v_0} has the smallest support. Indeed, we want to process fine scale details first before proceeding to coarse scale features.

In this context, the initial filters of the lifting scheme are defined as follows.

$$\begin{aligned} G_j^\circ &= \tilde{G}_j^\circ = \{v_0\} \text{ in the sense that } \tilde{G}_j^{\circ *} s_j = s_{j+1,v_0} \\ H_j^\circ &= \tilde{H}_j^\circ = \{\mathcal{N}(v_0)\} \text{ in the sense that } \tilde{H}_j^{\circ *} s_j = s_{j+1,\mathcal{N}(v_0)} \end{aligned} \tag{3.1}$$

where $\mathcal{N}(v_0)$ represents the neighborhood of v_0. The nature of the predictor operator P_j determines the type of neighborhood $\mathcal{N}(v_0)$ chosen, see Section 4.1 below. In comparison, in one dimension on a dyadic grid, the effect of the initial filters would be to split the data into 'even' and 'odd' points, that is:

$$H_j^\circ = \tilde{H}_j^\circ = \{2k, k = 0, \ldots, 2^j - 1\}\ ; \quad G_j^\circ = \tilde{G}_j^\circ = \{2k+1,\ k = 0, \ldots, 2^j - 1\} \tag{3.2}$$

Now, using the refinement relations $\Phi_j = \Phi_{j+1} H_j$ and $\Psi_j = \Phi_{j+1} G_j$, the prediction step described in (2.8), and the initial filter (3.1), we obtain an expression for the new, lifted, primal scaling functions in two dimensions:

$$\Phi_{j,\mathcal{N}(v_0)} = \Phi_{j+1,\mathcal{N}(v_0)} + \varphi_{j+1,v_0} P_{j,\mathcal{N}(v_0)}, \tag{3.3}$$

where $P_{j,\mathcal{N}(v_0)}$ is a row vector of length $\#\mathcal{N}(v_0)$ containing the prediction weights. Integrating (3.3) gives, for each $k \in \mathcal{N}(v_0)$

$$A_{j,k} = A_{j+1,k} + P_{j,k} A_{j+1,v_0} \ . \tag{3.4}$$

The detail coefficient corresponding to the vertex v_0 is computed as :

$$\begin{aligned} d_{j,v_0} &= \tilde{G}_j^* s_{j+1} \\ &= \tilde{G}_j^{\circ *} s_{j+1} - P_j \tilde{H}_j^{\circ *} s_{j+1} \\ &= s_{j+1,v_0} - P_{j,\mathcal{N}(v_0)} s_{j+1,\mathcal{N}(v_0)} \ . \end{aligned} \tag{3.5}$$

The detail coefficient thus encodes the difference between the scaling coefficient and its predicted value, based on its neighbors. This illustrates the choice of the name 'predictor' or 'prediction' for the operator P_j. The primal high-pass filter G_j does not change during the prediction step. Hence, following the refinement equation (2.3) the wavelet corresponding to d_{j,v_0} is just the scaling function of the previous step.

The prediction step is followed by an update (or primal lifting) step, whose first aim is to keep $\sum_k s_{j,k} A_{j,k}$ constant across scales. By biorthogonality, this is equivalent to providing the primal wavelet ψ_{j,v_0} with one vanishing moment. This update step ensures the stability of the wavelet transform. The relations in (2.9) give:

$$\begin{aligned} s_{j,\mathcal{N}(v_0)} &= \tilde{H}_j^* s_{j+1} \\ &= \tilde{H}_j^{o*} s_{j+1} + U_j \tilde{G}_j^* s_{j+1} \\ &= s_{j+1,\mathcal{N}(v_0)} + U_j d_{j,v_0} \end{aligned} \tag{3.6}$$

and

$$\begin{aligned} \psi_{j,v_0} &= \varphi_{j+1,v_0} - \Phi_{j+1} H_j U_j \\ &= \varphi_{j+1,v_0} - \sum_{k \in \mathcal{N}(v_0)} \varphi_{j,k} u_{j,k} \ , \end{aligned} \tag{3.7}$$

where $\tilde{G}_j$ and H_j have been modified by the prediction step. If we want ψ_{j,v_0} to have a vanishing moment, that is, a zero-integral, we have to impose that

$$A_{j+1,v_0} = \sum_{k \in \mathcal{N}(v_0)} A_{j,k} u_{j,k} \ . \tag{3.8}$$

Since there are more than one update value $u_{j,k}$, we use the remaining degrees of freedom to further improve the stability of the transform. This is explained in more detail in Section 4.2.

Every time a scaling coefficient is being predicted and replaced by a detail coefficient, its vertex is taken away from the triangulation. This creates a new, slightly coarser grid χ_j and new scaling functions around v_0: the space previously occupied by φ_{j+1,v_0} is taken by the surrounding scaling functions $\Phi_{j+1,\mathcal{N}(v_0)}$. The new integrals are computed using expression (3.4).

As an illustration, we now give an example of operators P_j and U_j in the setting of an univariate regular grid. Given the initial filters in (3.2), we take $P_j = \mathbf{1}$, where $\mathbf{1}$ denotes the identity matrix. Equation (3.5) then reduces to

$$d_{j,k} = s_{j+1,2k+1} - s_{j+1,2k} \ , \tag{3.9}$$

that is, the detail coefficients represent the difference between two neighboring scaling coefficients. Next, we take $U_j = \mathbf{1}/2$. This yields the scaling coefficients $s_{j,k}$, where

$$s_{j,k} = s_{j+1,2k} - d_{j,k}/2 \tag{3.10}$$

that is, $s_{j,k}$ is the average between two scaling coefficients from the next finer scale. It is clear that the expressions (3.9) and (3.10) give the usual Haar transform, written in the form of the lifting scheme [30].

3.3 Order of predictor and update operators

In order to improve the properties of the filters H_j and $\tilde{G}_j$, the predictor operator P_j must be such that, if the scaling coefficients $\{s_{j+1,v_0}; s_{j+1,\mathcal{N}(v_0)}\}$ are lying on a polynomial curve of order q, the corresponding detail coefficient d_{j,v_0} is equal to zero. The predictor operator P_j is then said to be of order q.

Since $d_{j,v_0} = \left\langle m \, , \tilde{\psi}_{j,v_0} \right\rangle$, a predictor of order q provides $\tilde{\psi}_{j,v_0}$ with $\tilde{N} = q + 1$ dual vanishing moments. In words this means that $\tilde{\psi}_{j,v_0}$ 'is blind to' polynomial of order smaller or equal to q. This property allows to have a sparse representation of the data. We will investigate the properties of some predictor operators of order one and two by means of a simulation study. Section 4.1 explains how to obtain such predictors.

On the primal side, only one vanishing moment is needed for our regression purpose. In Section 4.2, we present two types of update operators.

3.4 Geometrical setting: description of an arbitrary grid

Before proceeding to the description of the multiscale transform, we need some representation of the geometry and of the topology of an arbitrary grid in two dimensions. Our notations are inspired by [16, 17, 27]. A triangulated mesh is denoted as a pair $(\mathcal{P}, \mathcal{K})$, where $\mathcal{P}$ describes the geometric aspect of the grid and $\mathcal{K}$ provides the topological, or connectivity information as follows. $\mathcal{P}$ is a set of $\overline{N}$ point positions, $\mathcal{P} = \left\{p_i \in \mathbb{R}^3 | 1 \leq i \leq \overline{N}\right\}$. In our functional setting, $p_i = (x_i, y_i, z_i)$ where x_i and y_i have independent coordinates. The value z_i is called the *function value* of the vertex i.

To describe the connectivity in the triangulation of the design of experiment, also called parameter plane, we use the set $\mathcal{K}$, which is composed of subsets of $\left\{1, \ldots, \overline{N}\right\}$. These subsets, called simplices, come in three types: vertices $v = \{i\} \in \mathcal{V}$, edges $e = \{i, j\} \in \mathcal{E}$ and faces (also called *triangles*) $\tau = \{i, j, k\} \in \mathcal{F}$. As said before, the triangulation $\mathcal{F}$ on the finest level is given here by Delaunay's algorithm. Consider a central vertex v_0 (in our transform it will be the point to be predicted). We describe its neighborhood using connectivity (or topological) information. Its $1-$ring neighbors form a set $\mathcal{V}_1(v_0) = \{j | \{v_0, j\} \in \mathcal{E}\}$, see Figure 3(a). $K_{v_0} := \#\mathcal{V}_1(v_0)$ is the degree of v_0. The edges from v_0 to its $1-$ring neighbors form a set $\mathcal{E}_1(v_0) = \{\{v_0, j\} | j \in \mathcal{V}_1(v_0)\}$. The flap corresponding to the edge $e = \{i, j\}$ is defined as the vertex k such that the two triangles $\{v_0, i, j\}, \{k, i, j\} \in \mathcal{F}$ have the edge e in common, see Figure 3(b), where a $1-$ring neighbors with flaps is represented. The vertices in this figure,

except the central vertex v_0, form a set $\mathcal{V}_2(v_0)$, and $\mathcal{E}_2(v_0)$ denotes the set of its interior edges. The neighborhood $w(e)$ of an edge is formed by the four vertices of its two incident triangles, see Figure 3(c).

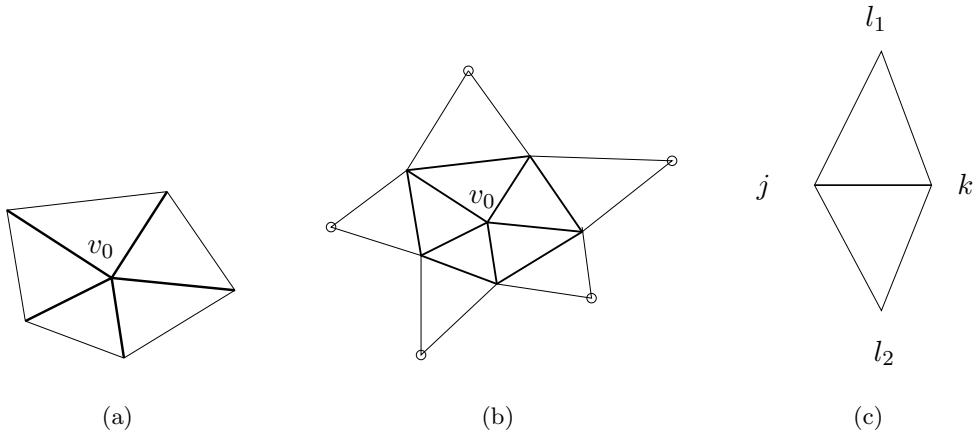


Figure 3: (a) 1−ring neighborhood. The vertices except the center one form $\mathcal{V}_1(v_0)$ and the bold edges form $\mathcal{E}_1(v_0)$. In (b), the 1−ring neighborhood with flaps (denoted by circles) is represented. The vertices except the center one form $\mathcal{V}_2(v_0)$ and the bold edges form $\mathcal{E}_2(v_0)$. (c) Neighborhood $w(e)$ of the edge $e = \{j, k\}$, formed by the four vertices $\{j, k, l_1, l_2\}$ of the two incident triangles.

4 Multiscale transform adapted to an irregular bidimensional grid

In order to fully describe our proposed multiscale transforms, three specifications have to be made: how to construct a predictor operator P_j at a given level j, how to find an update operator, and finally how to deal with the vertices that are lying at the boundary of the domain. These three topics are developed in Section 4.1, 4.2 and 4.3, respectively.

4.1 Predictor operators for scattered design points

4.1.1 Least squares prediction

In the setting of irregular bidimensional designs, a first type of prediction operator has been proposed in [20]. It consists of computing P_{j,v_0} as the least squares plane that goes through the scaling coefficients $s_{j+1,\mathcal{V}_1(v_0)}$. This predictor is of order one, that is, two dual vanishing moments are produced. An immediate extension is given by the *quadratic* least squares polynomial, which is defined on $\mathcal{V}_2(v_0)$. Let $x_{\mathcal{V}_2(v_0)}, y_{\mathcal{V}_2(v_0)}$ be the vectors of the neighbors' coordinates and (x_{v_0}, y_{v_0}) be the coordinates of the central point v_0 chosen at level j. In the interior of the domain, we always have $\#\mathcal{V}_2(v_0) \geq 6$, so we are able to estimate the parameters $\beta = [\beta_0\ \beta_x\ \beta_y\ \beta_{x^2}\ \beta_{xy}\ \beta_{y^2}]'$ of the model

$$z_t = \beta_0 + \beta_x x_t + \beta_y y_t + \beta_{x^2} x_t^2 + \beta_{xy} x_t y_t + \beta_{y^2} y_t^2 + e_t,\ e_t \text{ i.i.d, } E(e_t) = 0\ .$$

The design matrix of this least squares problem is

$$X = [\mathbf{1}\ x_{\mathcal{V}_2(v_0)}\ y_{\mathcal{V}_2(v_0)}\ x^2_{\mathcal{V}_2(v_0)}\ x_{\mathcal{V}_2(v_0)}y_{\mathcal{V}_2(v_0)}\ y^2_{\mathcal{V}_2(v_0)}],$$

where $\mathbf{1}$ is a $(\#\mathcal{V}_2(v_0) \times 1)$-vector of ones. The least squares estimator of β is given by $\hat{\beta} = (X'X)^{-1}X's_{j,\mathcal{N}(v_0)}$, and the predicted value for this scheme is obtained as

$$P_j s_{j,\mathcal{N}(v_0)} = [1\ x(v_0)\ y(v_0)\ x^2(v_0)\ x(v_0)\cdot y(v_0)\ y^2(v_0)] \cdot \hat{\beta}\ . \tag{4.1}$$

If all the values of the scaling coefficients $\{s_{j,\mathcal{N}(v_0)}, s_{j,v_0}\}$ are lying on a quadratic polynomial curve, then obviously the predicted value will be equal to s_{j,v_0}, hence $d_{j,v_0} = 0$. This means that $\tilde{\psi}_{j,v_0}$ has three vanishing moments, or, equivalently, that P_j defined in (4.1) is of order two.

4.1.2 Prediction based on relaxation: introduction

We now introduce another method of building a prediction operator, based on the subdivision schemes proposed by Guskov [16]. Consider a function $f : \chi \to \mathbb{R}$, where χ is a grid in $\mathbb{R}^2$, specified by a finite (or at most countable) set collection of vertices. We denote $f(v) := f(x_v, y_v)$ the function value at the vertex v. In the remaining part of Section 4.1, we consider the generic problem of finding a prediction of $f(v_0)$ using the function values of neighbor points of v_0. The obtained predictor operator can then be cast into the lifting scheme framework by replacing, for a given level j, χ by χ_j, the grid at level j, and $f(v)$ by $s_{j,v}$, the scaling coefficients at level j, where $v \in \chi_j$.

One contribution of this paper lies in the application of predictors that are based on the minimization of second and third order differences for wavelet denoising. This minimization provides a formula for prediction that is related to the multivariate Lagrange interpolation.

A $p-$th order difference is defined as the difference between two divided differences of order $p-1$. In the univariate case, divided differences of consecutive orders satisfy the following recursive relation [6, 16].

$$\frac{f^{[p+1]}(\{x_0, \ldots, x_{p+1}\})}{p+1} = \frac{f^{[p]}(\{x_1, \ldots, x_{p+1}\}) - f^{[p]}(\{x_0, \ldots, x_p\})}{x_{p+1} - x_0}\ .$$

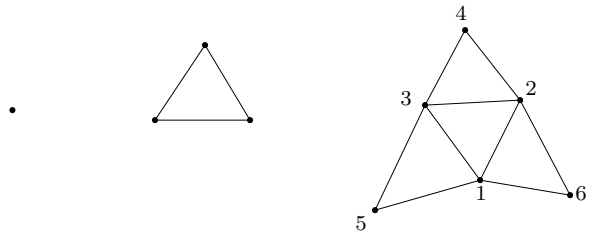
This recursion is initialized by taking $f^{[0]}(\{x_0\}) = f(x_0)$. The generalization to the multivariate setting however is not straightforward, and can be made in various ways. In this paper, we adopt the definitions proposed by Guskov in [16].

Divided differences of order p are defined on a stencil collection $\Omega^{[p]}$. Let $\mathcal{P}(\chi)$ denote the set of point positions of the vertices present in the grid χ. The zero-th order divided difference at the vertex v is the function sample value: $f^{[0]}_v = f(v)$. Hence the stencil collection for $f^{[0]}$ equals the set of vertices : $\Omega^{[0]} := \mathcal{P}(\chi)$. Following [16], we set $\Omega^{[1]} := \mathcal{F}$, the set of faces (or triangles). For second order divided differences, the stencil collection is made of one triangle together with his three neighbors:

$$\begin{aligned}\Omega^{[2]} \quad := \quad & (\{v_1, v_2, v_3, v_4, v_5, v_6\} : \{v_1, v_2, v_3\}, \{v_3, v_2, v_4\}, \{v_1, v_3, v_5\}, \\ & \{v_2, v_1, v_6\} \in \mathcal{F})\ ,\end{aligned}$$

see Figure 4.

Sections 4.1.3 and 4.1.4 introduce the formulas to compute the second and third order differences, respectively. Section 4.1.5 then presents how to combine these quantities to build a predictor operator.



(a)$f^{[0]}$ (b)$f^{[1]}$ (c)$f^{[2]}$

Figure 4: Stencils for divided differences.

4.1.3 First order divided differences and second order differences

In [16], the author provides a formula for the first divided difference $f_\tau^{[1]} = [f_x^{[1]}(\tau), f_y^{[1]}(\tau)]$, $\tau \in \mathcal{F}$ that behaves like a first order derivative, i.e. if f is constant on τ, the vector $f_\tau^{[1]}$ is zero. In other words, $f_\tau^{[1]}$ is the gradient of the linear spline that interpolates f on τ. Considering the normal $n_\tau = [-f_x^{[1]}(\tau), -f_y^{[1]}(\tau), 1]$, second order differences are defined as $\triangle_e^2 f = n_{\tau_2} - n_{\tau_1}$, where the two faces (τ_1, τ_2) have the edge e in common. Obviously, $\triangle_e^2 f$ is orthogonal to e, and is parallel to the parameter plane [17]. Hence only its signed length, noted $D_e^2 f$, counts. The expression for $D_e^2 f$ depends linearly on four function values at the vertices $w(e) = \{j, k, l_1, l_2\}$, see [16, 17]:

$$D_e^2 f = \sum_{l \in w(e)} c_{e,l} f_l \ . \tag{4.2}$$

The coefficients are given by

$$\begin{aligned} c_{e,l_1} &= \frac{L_e}{A_{[l_1,k,j]}}, \qquad & c_{e,l_2} &= \frac{L_e}{A_{[l_2,j,k]}} \\ c_{e,j} &= -\frac{L_e A_{[k,l_2,l_1]}}{A_{[l_1,k,j]} A_{[l_2,j,k]}}, \qquad & c_{e,k} &= -\frac{L_e A_{[j,l_1,l_2]}}{A_{[l_1,k,j]} A_{[l_2,j,k]}} \end{aligned} \tag{4.3}$$

where $A_{[k_1,k_2,k_3]}$ is the signed area of the triangle formed by (x_{k_1}, y_{k_1}), (x_{k_2}, y_{k_2}), (x_{k_3}, y_{k_3}) in the parameter plane, and L_e is the length of the segment between (x_k, y_k) and (x_j, y_j), as in Figure 3(c), see [16, 17]. Figure 5 illustrates that $D_e^2 f$ is the difference between two normals of f across a common edge e.

4.1.4 Second order divided differences and third order differences

On the stencil $\xi = \{1, 2, 3, 4, 5, 6\}$ of Figure 4(c), it is possible to compute the quadratic polynomial that interpolates the points $(x_i, y_i, f_i), i = 1, \ldots, 6$. This allows us to compute the second order divided difference $f^{[2]}(\xi) = [f_{xx}^{[2]}(\xi), f_{xy}^{[2]}(\xi), f_{yy}^{[2]}(\xi)]$ by solving the following system.

$$\begin{bmatrix} 1 & x_1 & y_1 & x_1^2/2 & x_1 y_1 & y_1^2/2 \\ 1 & x_2 & y_2 & x_2^2/2 & x_2 y_2 & y_2^2/2 \\ 1 & x_3 & y_3 & x_3^2/2 & x_3 y_3 & y_3^2/2 \\ 1 & x_4 & y_4 & x_4^2/2 & x_4 y_4 & y_4^2/2 \\ 1 & x_5 & y_5 & x_5^2/2 & x_5 y_5 & y_5^2/2 \\ 1 & x_6 & y_6 & x_6^2/2 & x_6 y_6 & y_6^2/2 \end{bmatrix} \begin{bmatrix} C \\ C_x \\ C_y \\ f_{xx}^{[2]}(\xi) \\ f_{xy}^{[2]}(\xi) \\ f_{yy}^{[2]}(\xi) \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} \tag{4.4}$$

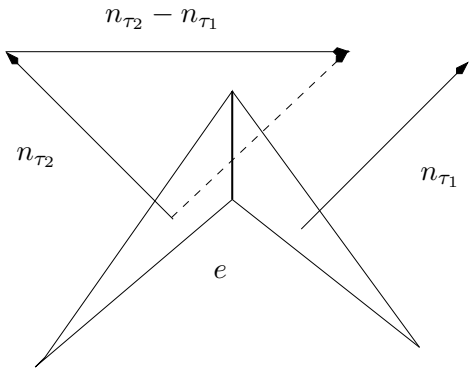


Figure 5: The normals n_{τ_1} and n_{τ_2} represent a measure of the gradient of the linear spline that interpolates f on τ_1 and τ_2, respectively. The faces τ_1 and τ_2 having the edge e in common, the second order difference at the edge e is defined as the difference between these two normals.

It is easy to see [16] that whenever (4.4) has a unique solution, the following system also has a unique solution:

$$\begin{bmatrix} D^2_{(\tau_{123},\tau_{234})}(x^2/2) & D^2_{(\tau_{123},\tau_{234})}(xy) & D^2_{(\tau_{123},\tau_{234})}(y^2/2) \\ D^2_{(\tau_{123},\tau_{135})}(x^2/2) & D^2_{(\tau_{123},\tau_{135})}(xy) & D^2_{(\tau_{123},\tau_{135})}(y^2/2) \\ D^2_{(\tau_{123},\tau_{126})}(x^2/2) & D^2_{(\tau_{123},\tau_{126})}(xy) & D^2_{(\tau_{123},\tau_{126})}(y^2/2) \end{bmatrix} \begin{bmatrix} f^{[2]}_{xx}(\xi) \\ f^{[2]}_{xy}(\xi) \\ f^{[2]}_{yy}(\xi) \end{bmatrix} = \begin{bmatrix} D^2_{(\tau_{123},\tau_{234})}f \\ D^2_{(\tau_{123},\tau_{135})}f \\ D^2_{(\tau_{123},\tau_{126})}f \end{bmatrix} \tag{4.5}$$

where $\tau_{123} = \{v_1, v_2, v_3\}$, $\tau_{234} = \{v_2, v_3, v_4\}$, $\tau_{135} = \{v_1, v_3, v_5\}$ and $\tau_{126} = \{v_1, v_2, v_6\}$. The notation $D^2_{\tau_{ijk},\tau_{jkl}}$ indicates the second order differences computed at the edge $\{j, k\}$ which is at the intersection of the two faces τ_{ijk} and τ_{jkl}. The system (4.5) supplies the relation needed to compute $f^{[2]}$ from second order differences only. This computation asks for the inversion of a matrix, and as such fails for the stencils which lead to a singular matrix in the system (4.5). More precisely, if all the points of a stencil $\xi \in \Omega^{[2]}$ lie on an algebraic curve of second order, the matrix will be singular and $f^{[2]}(\xi)$ does not exist [16]. A stencil for which $f^{[2]}$ does not exist is called nonadmissible. Figure 6 presents one admissible stencil and two nonadmissible stencils. Figure 6(d) presents a mesh fragment where no second order divided difference exists.

Third order differences are defined as the difference between two divided differences,

$$D^3_{\xi_1,\xi_2} f := f^{[2]}_{\xi_2} - f^{[2]}_{\xi_1} , \tag{4.6}$$

where the central triangles in the stencils ξ_1 and ξ_2 are adjacent, see Figure 7.

Formally, third-order differences are defined on the set of ordered pairs of stencils $\overline{\mathcal{E}}^{[3]}$, where

$$\overline{\mathcal{E}}^{[3]} = \{(\xi_1, \xi_2) : \xi_1, \xi_2 \in \Omega^{[2]}, \exists (\tau_1, \tau_2) \in \mathcal{F} \text{ with } \tau_1 \cap \tau_2 \in \mathcal{E} \text{ and } \xi_1 \cap \xi_2 = \tau_1 \cup \tau_2\} . \tag{4.7}$$

In words, third order differences represent the change in the second derivative as we go from the stencil ξ_1 to the neighboring stencil ξ_2.

From the above definitions (4.2) and (4.6), it is clear that the operators D^{p+1}_η, defined on adequate stencils η, annihilate polynomials of degree p. This property will be kept if we combine in an adequate way these differences in order to obtain a predictor operator of order p. The next section explains how to build such a predictor.

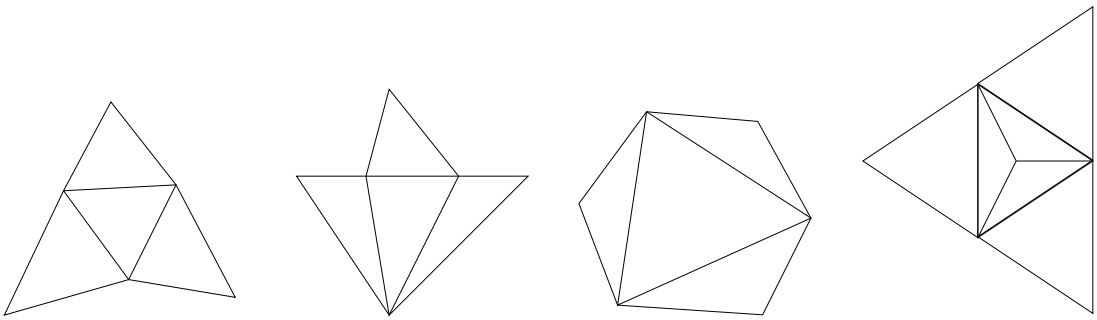


Figure 6: (a) One admissible stencil for $f^{[2]}$. (b) and (c) represent nonadmissible stencils. In (b), all the vertices lie in the union of two straight lines, in (c) they lie on a circle. (d) gives a combination of stencils of the type represented in (c). It is a mesh fragments where no $f^{[2]}$ exists.

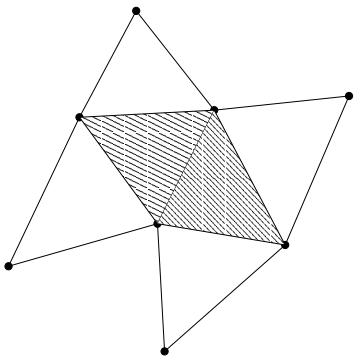


Figure 7: Stencil for third order differences, made of two adjacent stencils for second divided differences. The two central triangles have filled patterns.

4.1.5 Prediction based on a relaxation formula

In this section, we explain how to obtain a predictor of order $p, (p = 1, 2)$ based on a so-called *relaxation formula* [17].

Relaxation formula in general Beside the least squares prediction, it is possible to find a prediction operator based on the minimization of a quadratic energy defined as

$$\mathbf{E} = \sum_{\eta \in \overline{\mathcal{E}}^{[p+1]}} \left\| D_\eta^{p+1} f \right\|_2^2, \; p = 1, 2 \; , \tag{4.8}$$

where $\overline{\mathcal{E}}^{[2]} := \{(\tau_1, \tau_2) : \tau_1, \tau_2 \in \mathcal{F}, \tau_1 \cap \tau_2 \in \mathcal{E}\}$. The definition for $\overline{\mathcal{E}}^{[3]}$ is given in (4.7).

We now consider the energy $\mathbf{E}$ as a function of f_{v_0}. We are looking for the value $P_{f_{v_0}}$ that minimizes $\mathbf{E}(f_{v_0})$. Hence, f_{v_0} is treated here as a variable argument, and all the other quantities in $\mathbf{E}$ are considered as constant. For the minimization, only the stencils $\eta \in \overline{\mathcal{E}}^{[p+1]}$ which contain the vertex v_0 are important:

$$P_{f_{v_0}} = \arg\min \sum_{\eta \in \overline{\mathcal{E}}^{[p+1]} | v_0 \in \eta} \left\| D_\eta^{p+1} f \right\|_2^2 \; . \tag{4.9}$$

If, in $D_\eta^{p+1} f$, we separate the quantities that contain the variable f_{v_0} from the other quantities, we can write $D_\eta^{p+1} f = a_\eta + f_{v_0} b_\eta$. The expression for $P_{f_{v_0}}$ is then easily obtained by setting the partial derivative of $\mathbf{E}$ with respect to f_{v_0} equal to zero:

$$P_{f_{v_0}} = -\frac{\sum_{\eta \in \overline{\mathcal{E}}^{[p+1]} | v_0 \in \eta} \langle a_\eta \, , b_\eta \rangle}{\sum_{\eta \in \overline{\mathcal{E}}^{[p+1]} | v_0 \in \eta} \langle b_\eta \, , b_\eta \rangle} \; . \tag{4.10}$$

Note that, for the prediction of order one, the quantities a_η and b_η are scalars, since only the signed magnitude of $D_e^2 f$ matters, see equation (4.11) below. When $p = 2$, a_η and b_η are vectors of length three, since we use the system (4.5) to compute $f_\eta^{[2]}$ and hence $D_\eta^3 f$. This is further detailed below.

We now look at the set of vertices (also called *support*) used to compute the prediction and we develop the expression (4.10) for $p = 1$ and 2.

Relaxation with second order differences In case of second order differences, we consider in $\mathbf{E}(f_{v_0})$ the sum over all the edges e for whom v_0 belongs to the neighborhood $w(e)$, i.e. the set $\mathcal{E}_2(v_0)$. This means that the neighborhood $\mathcal{N}(v_0)$ used for prediction is $\mathcal{V}_2(v_0)$, see Figure 3(b). For $p = 1$, the formula (4.10) becomes:

$$P_{f_{v_0}} = \sum_{j \in \mathcal{V}_2(v_0)} w_{v_0,j} f_j, \quad w_{v_0,j} = -\frac{\sum_{\{e \in \mathcal{E}_2(v_0) | j \in w(e)\}} c_{e,v_0} c_{e,k}}{\sum_{e \in \mathcal{E}_2(v_0)} c_{e,v_0}^2} \; , \tag{4.11}$$

where the coefficients $c_{e,\cdot}$ are given by (4.3). Since the prediction operators based on a relaxation formula reconstruct exactly constant functions, we have that $\sum_j w_{v_0,j} = 1$.

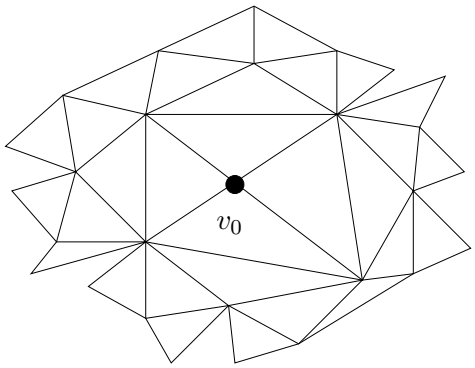


Figure 8: Complete support for prediction using third-order differences.

Relaxation with third order differences An example of a support for the prediction based on third order differences is represented in Figure 8. Such a support is rather large for a practical implementation. Hence, following Guskov [16], we use a reduced support, constituted of the set $\mathcal{V}_2(v_0)$, as in the case of second-order differences. Indeed, $\mathcal{V}_2(v_0)$ contains the stencils $\eta \in \overline{\mathcal{E}}^{[3]}$ that are closest to the central vertex v_0.

Since we take $\mathcal{V}_2(v_0)$ as support for the prediction, for each stencil η involved in the relaxation procedure, v_0 belongs to the intersection of the two central triangles represented in Figure 7. Consider now the second divided difference $f^{[2]}_{\xi_1}$, where a typical stencil ξ_1 is represented in Figure 4(c). Suppose that, in this Figure 4(c), the central vertex has the index 1: $v_0 = 1$. We define the three following edges: $e_1 = \{2,3\}, e_2 = \{1,3\}, e_3 = \{1,2\}$. The system (4.5) allows us to write

$$f^{[2]}_{\xi_1} = A^{-1} \begin{bmatrix} \sum_{i\in\{1,2,3,4\}} c_{e_1,i} f_i \\ \sum_{i\in\{1,2,3,5\}} c_{e_2,i} f_i \\ \sum_{i\in\{1,2,3,6\}} c_{e_3,i} f_i \end{bmatrix} , \tag{4.12}$$

where A is the matrix of the system of equations (4.5), and the coefficients $\{c_{e,i}\}$ are given in (4.3). Next, we separate in (4.12) the quantities that contain f_1 from the other quantities and obtain:

$$\begin{aligned} f^{[2]}_{\xi_1} &= A^{-1} \begin{bmatrix} \sum_{i\in\{2,3,4\}} c_{e_1,i} f_i \\ \sum_{i\in\{2,3,5\}} c_{e_2,i} f_i \\ \sum_{i\in\{2,3,6\}} c_{e_3,i} f_i \end{bmatrix} + f_1 A^{-1} \begin{bmatrix} c_{e_1,1} \\ c_{e_2,1} \\ c_{e_3,1} \end{bmatrix} \\ &= a_{\xi_1} + f_1 b_{\xi_1} \ . \end{aligned}$$

The above operation is repeated for the neighboring stencil of ξ_1, denoted ξ_2, see Figure 7 for a schematic representation. The union of ξ_1 and ξ_2 gives the stencil η for third order differences, and we have:

$$D^3_\eta f = (a_{\xi_2} - a_{\xi_1}) + f_1(b_{\xi_2} - b_{\xi_1}) = a_\eta + f_1 b_\eta = a_\eta + f_{v_0} b_\eta \ . \tag{4.13}$$

It remains to apply the formula (4.10) to obtain the relaxed value $P_{f_{v_0}}$.

4.1.6 Instabilities of the relaxation based on third-order differences

In this paper, one of our objective is to use the relaxation formula (4.10) to build predictor operators tailored to an irregular grid. However, we saw in Section 4.1.4 some restrictions to

the use of second order divided differences: $f_\xi^{[2]}$ does not exist if the stencil ξ is nonadmissible, that is, if all the vertices in the given stencil lie in the union of two straight lines or on a circle.

When dealing with irregular grids, there will always be some stencils that are close to a nonadmissible stencil. This will lead to a bad conditioning of the matrix in (4.5), and to instabilities in the reconstruction.

However, since we know for which class of stencils some problems arise, we can choose in advance the neighborhood $\mathcal{N}(v_0)$ used for prediction in such a way that most of the instabilities are removed. In other words, if necessary, we will not use the neighborhood $\mathcal{V}_2(v_0)$ given by the triangulation to compute $P_{f_{v_0}}$. Instead, we use a neighborhood which is as close as possible to $\mathcal{V}_2(v_0)$, but where all the second order divided differences are computed with a good precision (i.e. the matrix to be inverted is well-conditioned). Note that the lifting scheme allows us to choose freely the neighborhood used for prediction.

We now present the two main situations which lead to an ill-conditioned prediction operator, together with some solutions to stabilize the prediction.

1. A flap of an edge belonging to $\mathcal{E}_1(v_0)$ belongs at the same time to the $1-$ring neighbors $\mathcal{V}_1(v_0)$. This happens when a triangle, having v_0 as one of its vertex, is subdivided into three subtriangles. Figures 9(a) and 9(c) give two examples of such situations. The stencil to compute $f^{[2]}$ is close to the mesh fragment represented in Figure 6(d) and thus leads to a badly conditioned matrix.

   The solution proposed is the following. In the neighborhood $\mathcal{N}(v_0)$ used for the prediction, we do not include the inner vertex i that generates the three subtriangles. Next, a new flap has to be found in the given triangulation, as illustrated in Figures 9(b) and 9(d).

2. At least four vertices that belong to $\mathcal{V}_1(v_0)$ are approximately on a line. In this case, we are close to the situation of Figure 6(b). In order to detect such situations, for each quadruple of neighboring points in $\mathcal{V}_1(v_0)$, the least squares line that goes through these four points is computed. If the associated R^2-statistics (which represents the percentage of the variance of the data explained by the linear model) is larger than 90%, this means that a model where these four points are on a line is valid, and hence that we are close to the situation of Figure 6(b).

   The remedy to this problem is to take in $\mathcal{N}(v_0)$ only the first and last of these points that are nearly on a line. So a new edge (the edge $e = \{a, d\}$ in Figure 10(b)) is created, and a flap is attributed to this edge, see Figure 10(b). The new stencil is admissible, but, in order to construct it, we use another triangulation than the initial one.

These two modifications allow us to remove most of the instabilities that cause large bias in the reconstruction.

4.2 Update operator

The aim of the update step is to keep $\sum_k A_{jk} s_{jk}$ constant across scales. By biorthogonality relationships, this is equivalent to providing the primal wavelets with one vanishing moment. In order to obtain one primal vanishing moment, we must build a vector $U_j = [u_{j,1}, \ldots, u_{j,\#\mathcal{N}(v_0)}]$ that satisfies equation (3.8). The remaining degrees of freedom are used to improve the stability of the transform.

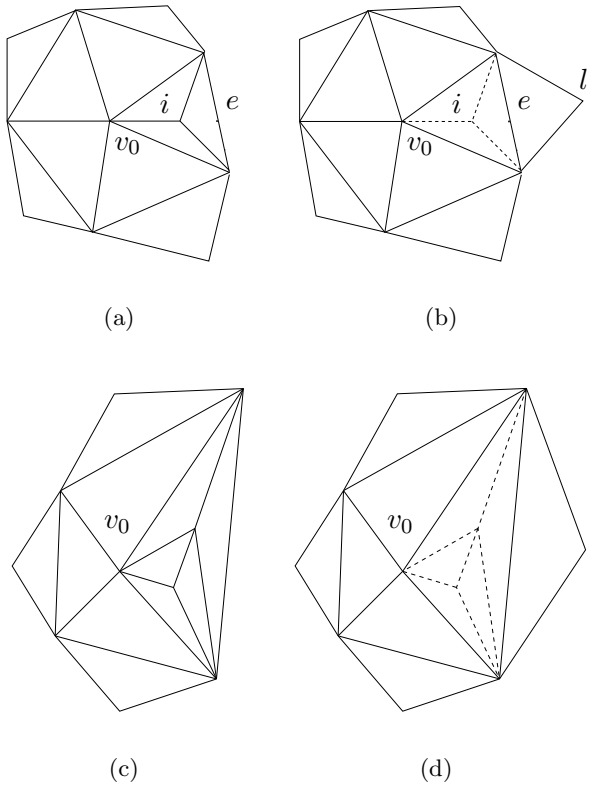
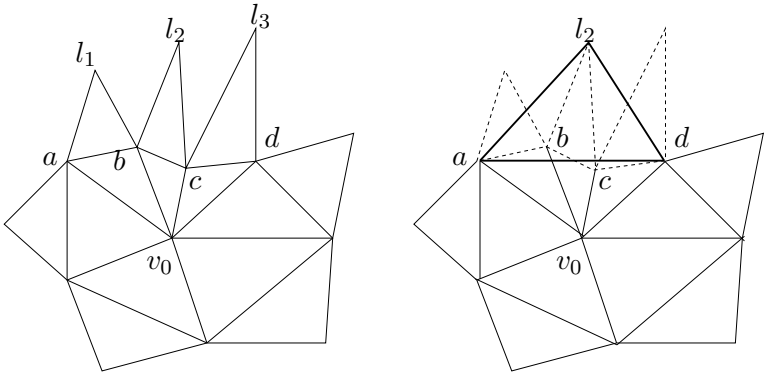


Figure 9: In (a), the three triangles generated by the vertex i will provoke some instabilities in the computation of the corresponding second divided difference $f^{[2]}$. The solution represented in (b) is to ignore the vertex i for the prediction of v_0 and to find in the triangulation a new flap l corresponding to the edge e. This makes the stencil stable. In(c), two triangles are imbricate and in (d) the corresponding stable stencil is represented in plain line.



(a) (b)

Figure 10: In (a), a R^2-statistics superior to 90% has been detected. When computing $f^{[2]}$ on the stencil belonging to $\overline{\Omega}^{[2]}$ and having $\{v_0, b, c\}$ as central triangle, the matrix to be inverted will be badly conditioned. In (b), we ignore for prediction the vertices b, c. Amongst the three flaps l_1, l_2, l_3, we choose to keep in (b) the one with the smallest index (l_2 in this case). This convention allows to always have a perfect reconstruction of a noisefree signal. In (b), the new neighborhood will produce a stable prediction. But the triangulation used is slightly different from the given Delaunay's triangulation (represented in dashed line).

A first possibility is to take the update values $\{u_{jk}\}$ that are of minimum l_2-norm, as Jansen *et al.* proposed in [20]:

$$u_{j,k} = \frac{A_{j,k} A_{j+1,v_0}}{\sum_{l \in \mathcal{N}(v_0)} A_{j,l}^2} \ . \tag{4.14}$$

We call this first update $U_j^{(1)} = \{u_{jk}\}$ the *minimum norm update operator*. To understand that this choice has a stabilizing effect, we remind that the update step modifies the primal wavelet basis functions according to (3.7). If the update coefficients have a large magnitude, the lifted wavelet function ψ_{j,v_0} will be close to the space spanned by the scaling functions $\varphi_{j,k}$ at the coarser level, hence φ_{jk} and ψ_{j,v_0} will be far from orthogonal. Minimizing the update coefficients norm is a straightforward and fast way to avoid such unstable situations.

Thinking of a stable transform as a close-to-orthogonal one, we can also use the remaining degrees of freedom to perform a local semi-orthogonalization of the transform. More precisely, we take first u_{jk} as in (4.14) to provide the wavelet ψ_{j,v_0} of equation (3.7) with one vanishing moment. Next we orthogonalize ψ_{j,v_0} with respect to the space $V_{j,\mathcal{N}(v_0)} := \text{span}\left\{\Phi_{j,\mathcal{N}(v_0)}\right\}$ by performing a second update with values

$$u_{j,\mathcal{N}(v_0)} = \left\langle \Phi_{j,\mathcal{N}(v_0)} \, , \Phi_{j,\mathcal{N}(v_0)} \right\rangle^{-1} \left\langle \psi_{j,v_0} \, , \Phi_{j,\mathcal{N}(v_0)} \right\rangle \ ,$$

as it is proposed in [26] for example. With this second step however, the vanishing moment is lost. To restore it, we multiply the elements of the update operator by an appropriate constant c. The whole update operator is:

$$U^{(2)} = c \left(U^{(1)} + \left\langle \Phi_{j,\mathcal{N}(v_0)} \, , \Phi_{j,\mathcal{N}(v_0)} \right\rangle^{-1} \left\langle \psi_{j,v_0} \, , \Phi_{j,\mathcal{N}(v_0)} \right\rangle \right) \tag{4.15}$$

We call this second update operator $U^{(2)}$ *minimum norm update operator followed by a local semi-orthogonalization.*

4.3 Treatment of the boundary points

The stencil used for a prediction at v_0 based on a relaxation procedure or on quadratic least squares is the second neighborhood $\mathcal{V}_2(v_0)$. However, when a vertex in $\mathcal{V}_1(v_0)$ lies on the boundary of the domain, or when v_0 itself lies on the boundary, the whole neighborhood $\mathcal{V}_2(v_0)$ cannot be defined, hence the transform must be adapted.

If one of the first neighbors $\mathcal{V}_1(v_0)$ is lying on the boundary, we always use the linear least squares prediction, even if, inside the domain, another predictor operator is utilized.

Now, if the vertex v_0 chosen to be predicted itself is on the boundary, the task of finding a good prediction becomes more delicate since no information is available 'on the other side' of the boundary. This means that, if we want to keep using linear least squares prediction, some kind of extrapolation (in contrast to interpolation) is inevitable.

In this paper, we use the treatment of boundary proposed by Jansen, Nason and Silverman in [21]. In this scheme, special precautions are taken in order to reduce instabilities. First, neighbors on long, narrow triangles at the boundary are not taken into account for prediction. As a matter of fact, these triangles are eliminated all together. Second, if only two neighbors are available, prediction becomes extrapolation. The method then adopts a plane which fits the two neighbors, and which is constant in the direction orthogonal to the line connecting them.

5 Nonlinear wavelet estimator

The previous section described some wavelet transforms that adapt to an irregular bidimensional grid. Consider again the model (1.1), where the errors are independent, of zero mean and variance σ_ϵ^2. The next section recalls the Bayesian thresholding scheme proposed in [20] and used in the simulation study of Section 6. Let $\tilde{d}_{jk}$ be the resulting thresholded detail coefficients. The nonlinear wavelet estimator is

$$\widehat{m}(x,y) = \sum_k \hat{s}_{j_0,k}\varphi_{j_0,k}(x,y) + \sum_{j=j_0}^{J-1}\sum_k \tilde{d}_{jk}\psi_{jk}(x,y).$$

Here, we restrict ourself to models with homoscedastic errors. However, it is possible to treat heteroscedastic errors (where $\epsilon_k = \sigma(X_k, Y_k)e_k$, and the e_k are i.i.d.) in a three-step procedure. First a pilot estimator $\hat{m}_0(x,y)$ is obtained using a linear denoising scheme. Second, the residuals $r_k := Z_k - \hat{m}_0(x,y)$ are computed and the variance function $\sigma^2(.)$ is estimated from the data set (X_k, Y_k, r_k^2) with a linear wavelet estimator. The estimation of $\sigma^2(x,y)$ allows us to estimate the variance of the detail coefficients, and we can then apply the Bayesian thresholding scheme.

5.1 Bayesian thresholding scheme

Thresholding is a common technique for wavelet coefficient selection. Classical data-adaptive procedures such as SURE [14] or Generalized Cross-Validation [19] for example, treat all detail coefficients equally: only the magnitude of the coefficient (normalized by its standard deviation) determines whether or not the coefficient will be selected. In reality, not all coefficients are equal: at coarse scales, for instance, detail coefficients usually convey important information. The optimal threshold for those coefficients is thus lower than at fine scales This calls for a scale-dependent threshold assessment. However, for such procedure to be efficient,

the number of coefficients, and the degree of sparsity within each scale should be sufficiently large.

In our setup of irregular bidimensional designs, scale is a continuous notion. Indeed, every basis function, as well as every data point, has its own scale. Therefore, a fully data dependent method not only requires a data dependent threshold choice, but also a different threshold for every single coefficient.

A way to determine these thresholds is to use an empirical Bayes method, see [22, 23]. For every detail coefficient, we estimate the prior probability $\tilde{p}$ that the corresponding basis function catches an important data feature, such as edges (singularities). This probability depends on the scale, which can be measured by the support of the basis function. The estimation of the probability $\tilde{p}$ is done in two steps. First, detail coefficients with similar scales are grouped into artificial resolution levels. From these data, we estimate the relative number of large coefficients. This relative number of important coefficients is a rough estimate for the requested prior probability. In a second step, the estimation is further corrected according to the exact support of every basis function. Besides a prior probability on two classes of coefficients (important and not important), we also need to specify a prior density on noise-free wavelet coefficients in both classes. We adopt the heavy tail model proposed by Johnstone and Silverman [22, 23] which has already proved to be effective in the framework of scattered data smoothing [20]. This model assumes non-important coefficients to be exactly zero, while important coefficients are distributed according to a continuous mixture of a normal by a Beta density, in such a way that the tail of the resulting density behaves like a Cauchy. If we call V the unknown noise-free wavelet coefficient, and W the corresponding observed value, we can write for the posterior density:

$$\begin{aligned} f_{V|W}(v|w) &= f_{V|W}(v|w, V=0)\, P(V=0|W=w) \\ &\quad + f_{V|W}(v|w, V\neq 0)\, P(V\neq 0|W=w). \end{aligned}$$

The posterior probability $P(V \neq 0|W = w)$ then follows from the prior $\tilde{p} = P(V \neq 0)$ and Bayes' rule:

$$p^* := P(V\neq 0|W=w) = \frac{P(V\neq 0) f_W(w|V\neq 0)}{f_W(w)}.$$

A second application of Bayes' rule is necessary to fill in the posterior density for V, given that V belongs to the class of important coefficients:

$$f_{V|W}(v|w, V\neq 0) = \frac{f_V(v|V\neq 0) f_{W|V}(w|v)}{f_W(w|V\neq 0)}.$$

In this equation $f_V(v|V \neq 0)$ is the heavy tail prior. The conditional noise model $f_{W|V}(w|v)$ is a normal density: $f_{W|V}(w|v) = \phi_{\sigma_\epsilon}(w - v)$, where ϕ_{σ_ϵ} is the zero-mean normal density with standard deviation σ_ϵ. Convolution of the heavy tail prior and the density of the errors provides the density $f_W(w|V \neq 0)$. Then, from $f_W(w|V \neq 0)$ and $\tilde{p} := P(V \neq 0)$, we easily obtain the noisy coefficient density f_W. For more details, we refer the reader to [20, 22, 23].

5.2 Recapitulation of the different estimators

In Section 4, we saw how to build several wavelet transforms for an irregular two-dimensional grid. These wavelet transforms, used in conjunction with the Bayesian denoising scheme described above, produce eight different wavelet estimators of the regression function in the model (1.1): four obtained with the minimum norm update and four with the minimum norm update followed by a local semi-orthogonalization.

The four predictor operators that we consider here are the following.

1. Prediction based on relaxation with second order differences (called 'linear relaxation' hereafter),
2. Prediction based on relaxation with third order differences ('quadratic relaxation'), with the stabilization procedure proposed in Section 4.1.6.
3. Linear least squares prediction on the $1-$ring neighbors.
4. Quadratic least squares prediction on the $1-$ring neighbors and flaps.

6 Simulation study

6.1 Parameters of the simulation study

We tested the proposed estimators on data sets of sample size $n = 700$. The irregular grid is generated by taking uniform regressors in the model (1.1): $X_k \sim U[0,1]$ and $Y_k \sim U[0,1]$ in the model (1.1).

We consider three test functions.

1. A linear function ('Plane')
$$m(x,y) = 4x + y \ . \tag{6.1}$$
2. A piecewise linear function ('Piece-linear')
$$m(x,y) = (2x+y)1_{\{3x-y<1\}} + (10-x)1_{\{3x-y\geq 1\}} \ . \tag{6.2}$$
3. A quadratic function ('Quad')
$$m(x,y) = 6 - 12x + 12x^2 + 3y - 5y^2 \ . \tag{6.3}$$

The three test functions are displayed on Figure 11. For each function, the decomposition is done until only $n_0 = 10$ scaling coefficients remain at the coarsest level grid χ_{j_0}.

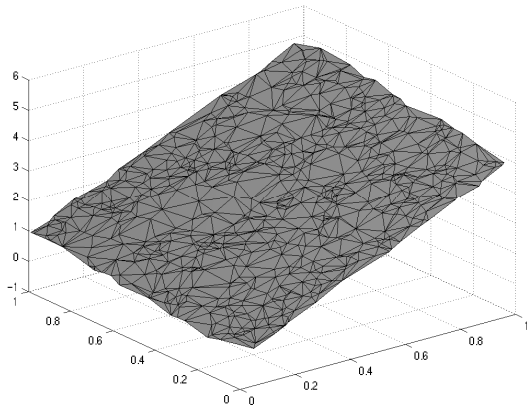
In this simulation study, we compare the performance of the four predictor operators and of the two updates described above. In order to specify the variance σ_ϵ^2 in the model (1.1), we define a signal-to-noise ratio as $\mathrm{SNR} := \mathrm{sd}(m)/\sigma_\epsilon$, where

$$\mathrm{sd}(m)^2 = \mathrm{Var}(m) = \frac{1}{n-1}\sum_{t=1}^{n}(m(x_t,y_t) - \overline{m})^2, \tag{6.4}$$

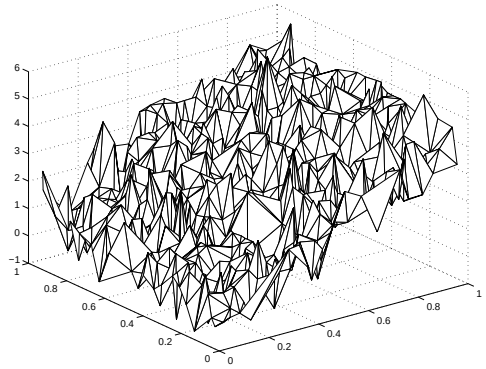
where the design $(x_t,y_t)_{t=1}^n$ is a realization of two independent Uniform random variables, $X \sim U[0,1], Y \sim U[0,1]$. The quantity $\overline{m}$ is equal to the mean value of the $m(x_t,y_t)$, $t = 1,\ldots,n$. We set the SNR equal to two, which allows us to specify σ_ϵ. In Figure 11, we show, for the three test functions, a typical realization of a data set generated with a $\mathrm{SNR} = 2$.

In the literature, the closeness-of-fit is often measured using a 'residual signal to noise ratio' (denoted rSNR), where the 'noise' is estimated with the residuals $\widehat{m} - m$:

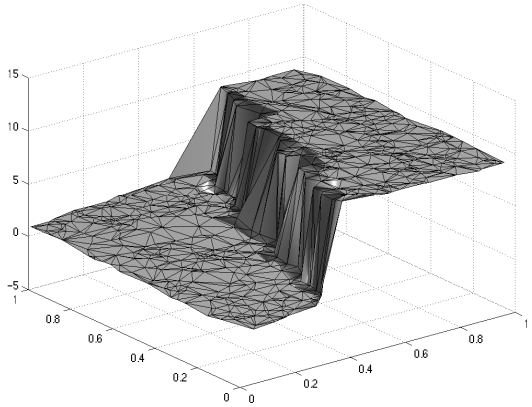
$$r\mathrm{SNR}(m,\hat{m}) = 10 * \log_{10}\left(\frac{\mathrm{Var}(m)}{\mathrm{Var}(\hat{m}-m)}\right) \ .$$



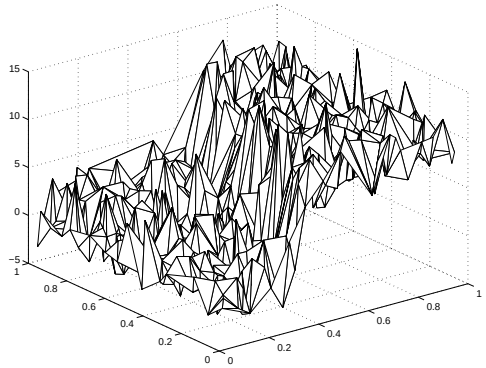
(a)



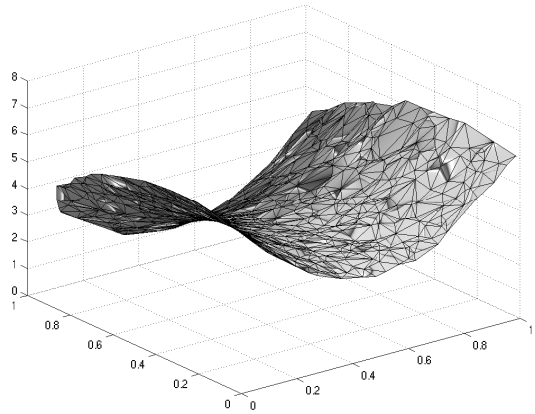
(b)



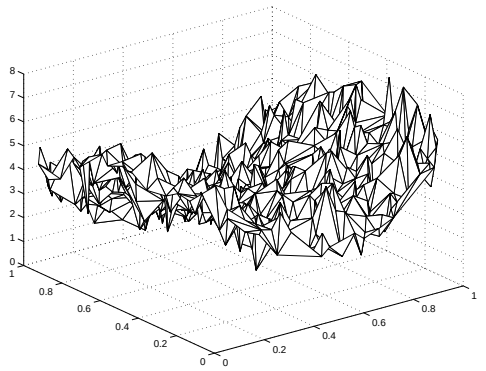
(c)



(d)



(e)



(f)

Figure 11: Test functions together with a typical realization of a noisy data set with a SNR = 2: (a) Linear function (Plane) (b) Noisy 'Plane' function (c) Piecewise linear function (Piece-linear) (d) Noisy 'Piece-linear' function (e) Quadratic smooth function (Quad) (f) Noisy 'Quad' function.

In this expression, $\mathrm{Var}(m) = n^{-1}(\sum_{k=1}^{n}(m(x_k, y_k) - \overline{m})^2)$, where $\overline{m}$ is the mean value of m. We can also compute a residual SNR for the initial observations $z = \{z_k\}_{k=1}^{n}$:

$$r\mathrm{SNR}(m, z) = 10 * \log_{10}\left(\frac{\mathrm{Var}(m)}{\mathrm{Var}(z - m)}\right) .$$

In our simulation study, we found it convenient to measure the performance of an estimator $\hat{m}$ by an efficiency ratio:

$$\mathrm{Eff}(\hat{m}) = \frac{r\mathrm{SNR}(m, \hat{m})}{r\mathrm{SNR}(m, z)} . \tag{6.5}$$

A good estimator will have a large efficiency $\mathrm{Eff} > 1$. On the opposite, if $\mathrm{Eff} \leq 1$, it means that the estimator does not do better than the raw data from the rSNR point of view. This may happen when some instabilities occur.

For $B = 50$ different data sets, we computed the efficiency using (6.5) and summarized the information by taking the median, first and third quartiles over these B efficiency values. Note that the different estimators were computed on the same data sets. Thus, although the number of runs is not large, a difference in the efficiency values reveal a different behaviour of the estimator.

6.2 Simulation results

We now present and discuss the results of the simulation studies. In the graphs, we represent realizations where the corresponding estimate has an efficiency close to the corresponding median efficiency reported in the table.

Table 1 gives for each test function, the efficiencies of the eight estimators obtained by combining the four predictors and the two updates. Two main features of the behaviour of these estimators emerge from this table.

1. The transforms with minimum norm update lead to estimators which behave similarly, in the sense that, for the four predictor operators, the inter-quartile intervals overlap each other. However, in terms of median efficiency, the linear least squares prediction performs the best. One explanation for this may be that the least squares prediction operation already includes some kind of smoothing, since it computes a regression plane (as opposed to a type of interpolation) on an appropriate set of neighbors. Hence, typically smaller absolute values of the noisy detail coefficients are observed when using a least squares prediction as compared to a prediction based on relaxation. Thus, the correction (3.6) of the update step is of smaller magnitude, and this improves the quality of the estimator.

2. For all the predictors, performing a local semi-orthogonalization in the update significantly improves the efficiency of the estimator, in the sense that the inter-quartile intervals obtained with $U_j^{(1)}$ and $U_j^{(2)}$ do not overlap. As already observed in [9], adding to the transform an update step which brings the transform closer to an orthogonal one makes the denoising more efficient. With this second update, the linear relaxation gives the best median efficiency for the Plane and Piece-linear function, whereas the quadratic relaxation provides the best median efficiency for the quadratic function (6.3).

Figure 12 represents the estimation of the 'Plane' function using linear least squares prediction and semi-orthogonalization in the update. The efficiency is larger than three, which means that the estimation is good. In the graphs, we represent the triangulation effectively used

Table 1: Results of the simulation study using Bayesian denoising scheme and eight different wavelet transforms, obtained by combining four predictor operators and two updates. One detail coefficient is produced at every scale. The four predictor operators are obtained as follows: 1. relaxation on second order differences (Lin relax), 2. relaxation on third order differences (Quad relax), 3. linear least squares (Lin LS), and 4. quadratic least squares prediction (Quad LS). The two update operators are the minimum norm update and the minimum norm update followed by a local semi-orthogonalization. For each function and each transform, the first line in the table gives the median efficiency Eff over the $B = 50$ runs. The numbers between brackets represent the inter-quartile intervals of the efficiency Eff. The sample size is $n = 700$ and the SNR used to generate the data sets is taken equal to two.

Plane				
Minimum norm update				
Predictor:	Lin relax	Quad relax	Lin LS	Quad LS
	2.636	2.805	3.029	2.903
	[2.509; 2.894]	[2.472; 2.967]	[2.758; 3.200]	[2.4827; 3.116]
Minimum norm update and semi-orthogonalization				
Predictor:	Lin relax	Quad relax	Lin LS	Quad LS
	3.287	3.217	3.273	3.249
	[2.970; 3.618]	[3.008; 3.474]	[2.981; 3.489]	[2.934; 3.488]
Piece-linear				
Minimum norm update				
Predictor:	Lin relax	Quad relax	Lin LS	Quad LS
	1.379	1.389	1.573	1.368
	[1.237; 1.499]	[1.236; 1.483]	[1.471; 1.634]	[1.231; 1.426]
Minimum norm update and semi-orthogonalization				
Predictor:	Lin relax	Quad relax	Lin LS	Quad LS
	1.817	1.776	1.702	1.723
	[1.711; 1.877]	[1.698; 1.851]	[1.618; 1.773]	[1.629; 1.777]
Quad				
Minimum norm update				
Predictor:	Lin relax	Quad relax	Lin LS	Quad LS
	2.108	2.156	2.384	2.071
	[1.930; 2.267]	[1.920; 2.280]	[2.183; 2.491]	[1.919; 2.264]
Minimum norm update and semi-orthogonalization				
Predictor:	Lin relax	Quad relax	Lin LS	Quad LS
	2.625	2.662	2.643	2.608
	[2.493; 2.747]	[2.440; 2.741]	[2.490; 2.768]	[2.383; 2.715]

during the wavelet transform, where the long, narrow triangles present at the boundary have been removed.

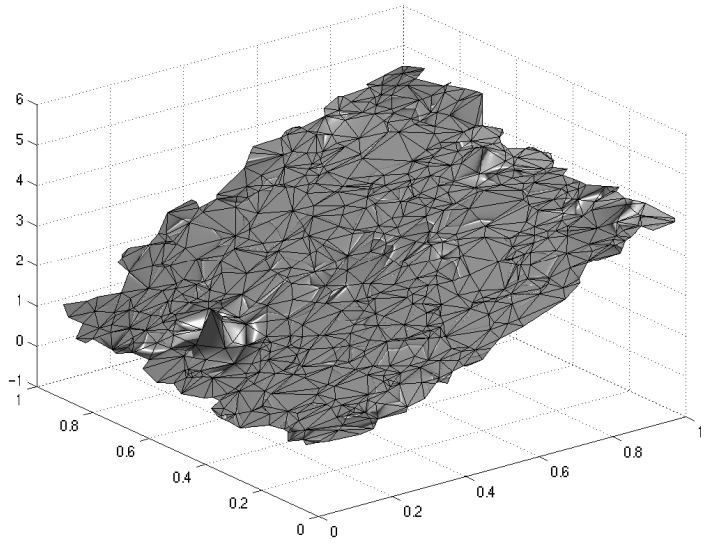
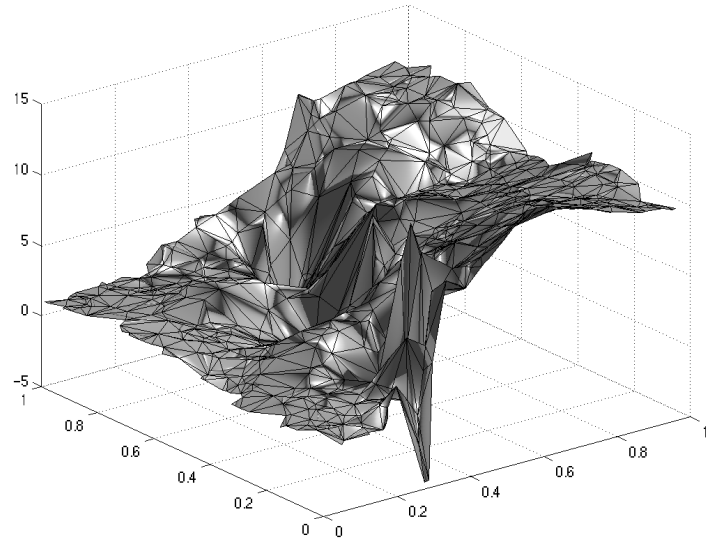


Figure 12: Estimation of the linear function (6.1) ($n = 700$, SNR $= 2$). The linear least squares predict operator is used, together with the minimum norm update followed by semi-orthogonalization. The efficiency Eff obtained for this estimate is equal to 3.273.

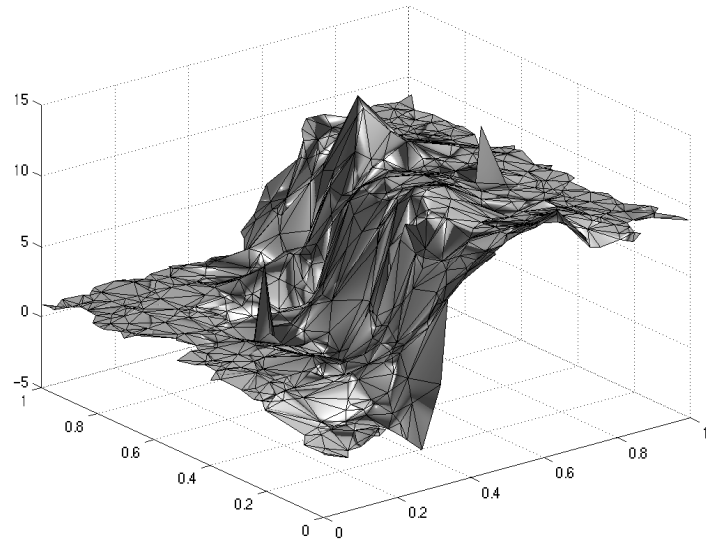
The estimation of the piecewise linear function (6.2) is not easy since a discontinuity is present in the underlying signal. Moreover, our data set has a small sample size of $n = 700$ and a low signal-to-noise ratio. This situation makes the task of any denoising scheme difficult. We represent in Figure 13 two estimators obtained with relaxation on second order differences. In Figure 13(a), the minimum norm update is used, whereas in Figure 13(b), a local semi-orthogonalization in the update has been performed. In both cases, the discontinuity line can be distinguished from the smooth parts of the function. In Figure 13(b), the smooth parts are better estimated than with the minimum norm update alone. In both cases however, some spikes still subsist, due to noisy coefficients which have survived the Bayesian shrinkage procedure.

Figure 14 gives some examples of the estimation of the quadratic function (6.3) when using as predictor the relaxation on third order differences (Figure 14(a)), or the quadratic least squares prediction (Figure 14(b)). The update performed is the minimum norm update with local semi-orthogonalization. Both predictor performs in a similar way. On this example, the quadratic relaxation seems to provide a smoother estimate than the quadratic least squares prediction.

Note that, since the filters used in the wavelet transform are of finite length, the computational load of all the methods is still $O(n)$ as in the classical case. However, in practice, the linear least squares prediction provides the fastest algorithm because there is no need to find the second neighbors $\mathcal{V}_2(v_0)$ with this method. On the opposite, the relaxation on third order differences is the slowest method since it requires the detection of the potentially unstable situations and the intensive inversion of matrices of size 3×3, see equation (4.12) and (4.13). (The methods based on prediction with relaxation on second order differences and with quadratic least squares have similar computational time.)

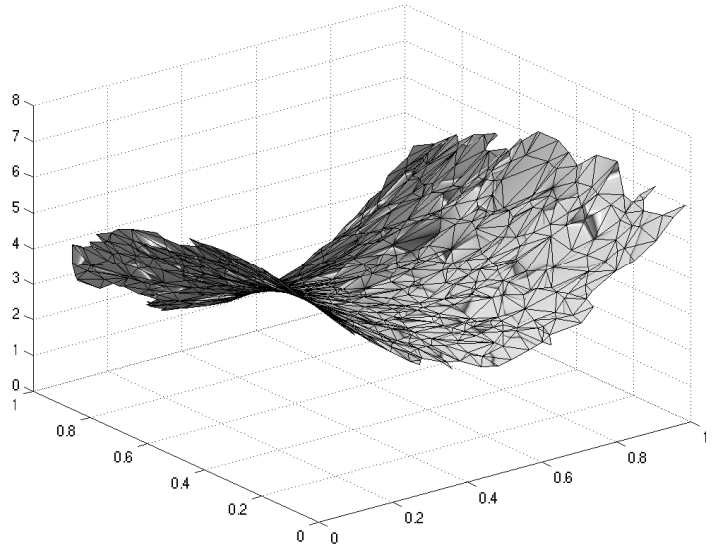


(a)

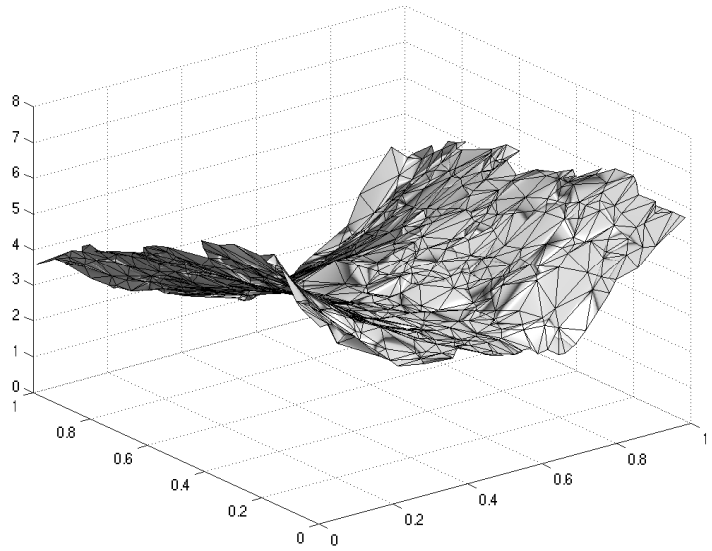


(b)

Figure 13: Estimation of the piecewise linear function (6.2) on two data sets, both having $n = 700$ observations and a SNR $= 2$. Relaxation on second order differences is used to predict the 'odd' scaling coefficients. In (a), only a minimum norm update has been implemented, whereas in (b), the minimum norm update was followed by a local semi-orthogonalization. The efficiency Eff is equal to (a) 1.379, (b) 1.817.



(a)



(b)

Figure 14: Estimation, on two data sets ($n = 700, \mathrm{SNR} = 2$), of the quadratic function (6.3). The update performed is a minimum norm update followed by local semi-orthogonalization. In (a), the stabilized relaxation with third order differences is used as prediction operator, whereas in (b) quadratic least squares prediction has been utilized. The efficiency Eff is equal to (a) 2.662, and (b) 2.608.

7 Conclusion

We proposed here different wavelet transforms suitable for the denoising of a regression function defined on an irregular bivariate grid. Combining four possible predictors with two types of update, we obtain eight wavelet transforms, and hence eight different estimators. The four predictors yield estimators which behave in a similar way. On the other hand, performing a local semi-orthogonalization during the update improves the quality of the smoothing. When using local semi-orthogonalization, the resulting estimators have proved to work well for smooth or piecewise smooth functions.

The vertices lying on the boundary must be treated with special care in order to obtain a good estimator. Here we use the stable method proposed by Jansen, Nason and Silverman in [21], which avoids the creation of artificial bias at the boundary.

The nonparametric estimators proposed here are of a wavelet type, that is, there are based on the idea of multiscale representation of the data. In other words, the denoised data can be represented in the wavelet domain by a few scaling coefficients present at the coarsest scale together with the detail coefficients that survived the Bayesian shrinkage procedure. The estimated function can be represented on a coarser grid than the original data, and in this sense some compression of the data is possible.

Acknowledgement

Financial support from the contract 'Projet d'Actions de Recherche Concertées' nr. 98/03–217 from the Belgian government, and from the IAP research network nr. P5/24 of the Belgian State (Federal Office for Scientific, Technical and Cultural Affairs) is gratefully acknowledged. Part of this work was carried out while V. Delouille was a visiting student at the Digital Signal Processing Group of Rice University, supported by a travel grant from the Belgian National Research Foundation (FNRS). Hospitality of Rice University is gratefully acknowledged.

References

[1] A. Antoniadis and D.T. Pham. Wavelet regression for random or irregular design. *Computational Statistics and Data Analysis*, 28:353–369, 1998.

[2] C.B Barber, D.P. Dobkin, and H.T. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, 1996.

[3] T. Cai and L.D. Brown. Wavelet shrinkage for nonequispaced samples. *Annals of Statistics*, 26(5):1783–1799, 1998.

[4] J.M. Carnicer, W. Dahmen, and J.M. Peña. Local decomposition of refinable spaces and wavelets. *Appl. Comput. Harmon. Anal.*, 3:127–153, 1996.

[5] W. Dahmen. Some remarks on multiscale transformations, stability and orthogonality. In P. J. Laurent, A. Le Méhauté, and L. L. Schumaker, editors, *Curves and Surfaces II*, pages 1–32. A. K. Peters, Boston, 1991.

[6] I. Daubechies, I. Guskov, and W. Sweldens. Regularity of irregular subdivision. *Constructive Approximation*, 15(3):381–426, 1999.

[7] I. Daubechies, I. Guskov, and W. Sweldens. Commutation for irregular subdivision. *Constructive Approximation*, 17(4):479–514, 2001.

[8] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.*, 4(3):245–267, 1998.

[9] V. Delouille, J. Simoens, and R. von Sachs. Smooth design-adapted wavelets for non-parametric stochastic regression. Discussion Paper 0117, Institut de Statistique, UCL, Belgium, 2001.

[10] V. Delouille and R. von Sachs. Smooth design-adapted wavelets for half-regular designs in two dimensions. Discussion Paper 0226, Institut de Statistique, UCL, Belgium, 2002.

[11] B. Delyon and A. Juditsky. On the computation of wavelet coefficients. *Journal of Approximation Theory*, 88:47–79, 1997.

[12] D.L. Donoho. De-noising via soft-thresholding. *IEEE Transactions on Information Theory*, 41:613–627, 1995.

[13] D.L. Donoho and I.M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81:425–455, 1994.

[14] D.L. Donoho and I.M. Johnstone. Adapting to unknown smoothness via wavelet shrinking. *J. Am. Statist. Assoc.*, 90:1200–1224, 1995.

[15] D.L. Donoho and I.M. Johnstone. Minimax estimation via wavelet shrinkage. *Annals of Statistics*, 26:879–921, 1998.

[16] I. Guskov. Multivariate subdivision scheme and divided differences. Technical report, Princeton University, 1998.

[17] I. Guskov, W. Sweldens, and P. Schröder. Multiresolution signal processing for meshes. *Computer Graphics Proceedings (SIGGRAPH 99)*, pages 325–334, 1999.

[18] P. Hall and B.A. Turlach. Interpolation methods for nonlinear wavelet regression with irregularly spaced design. *Annals of Statistics*, 25(5):1912–1925, 1997.

[19] M. Jansen, M. Malfait, and A. Bultheel. Generalized cross validation for wavelet thresholding. *Signal Processing*, 56:33–44, 1997.

[20] M. Jansen, G. Nason, and B. Silverman. Scattered data smoothing by empirical Bayesian shrinkage of second generation wavelet coefficients. In M. Unser and A. Aldroubi, editors, *Wavelet Applications in Signal and Image Processing*, volume 4478, pages 87–97. IX Proceedings of SPIE, 2001.

[21] M Jansen, G. Nason, and B. Silverman. Second generation wavelets and empirical Bayes estimates for scattered data smoothing. In preparation, 2002.

[22] I.M. Johnstone and B.W. Silverman. Empirical Bayes selection of wavelet thresholds. Stanford University and University of Bristol, 2002.

[23] I.M. Johnstone and B.W. Silverman. Finding needles and hay in haystacks: Risk bounds for empirical bayes estimates of possibly sparse sequences. Stanford University and University of Bristol, 2002.

[24] A. Kovac and B.W. Silverman. Extending the scope of wavelet regression methods by coefficient-dependent thresholding. *J. Am. Statist. Assoc.*, 95:172–183, 2000.

[25] J. Krämer. Delaunay triangulation in two and three dimensions. Master's thesis, Wilhelm Schickard Institute for Computer Science Graphical-Interactive Systems, University of Tübingen, 1995.

[26] J. Simoens and S. Vandewalle. A stabilized lifting construction of wavelets on irregular meshes on the interval. To appear in *SIAM J. Sci. Comput.*, 2002.

[27] E.H. Spanier. *Algebraic Topology.* McGraw-Hill, New York, 1996.

[28] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.*, 3(2):186–200, 1996.

[29] W. Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2):511–546, 1997.

[30] W. Sweldens and P. Schröder. Building your own wavelets at home. In *Wavelets in Computer Graphics*, pages 15–87. ACM SIGGRAPH Course notes, 1996.

[31] E. Vanraes, M. Jansen, and A. Bultheel. Stabilized wavelet transforms for non-equispaced data smoothing. *Signal Processing*, 82(12):1979–1990, 2002.