Centrum voor Wiskunde en Informatica

REPORT RAPPORT

PNA

Probability, Networks and Algorithms

Probability, Networks and Algorithms

Numerical methods for decomposition of 2D signals by rotation and wavelet techniques

P.M. de Zeeuw, R.A. Zuidwijk

Report PNA-R0124 December 31, 2001

CWI is the National Research Institute for Mathematics and Computer Science. It is sponsored by the Netherlands Organization for Scientific Research (NWO).
CWI is a founding member of ERCIM, the European Research Consortium for Informatics and Mathematics.

CWI's research has a theme-oriented structure and is grouped into four clusters. Listed below are the names of the clusters and in parentheses their acronyms.

Probability, Networks and Algorithms (PNA)

Software Engineering (SEN)

Modelling, Analysis and Simulation (MAS)

Information Systems (INS)

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-3711

Numerical Methods for Decomposition of 2D Signals by Rotation and Wavelet Techniques

P.M. de Zeeuw
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
Paul.de.Zeeuw@cwi.nl

R.A. Zuidwijk
EUR
Rotterdam School of Management
P.O. Box 1738, 3000 DR Rotterdam, The Netherlands
R.Zuidwijk@fbk.eur.nl

ABSTRACT
Segregation of desirable and undesirable components in a signal given by measurements is a broad subject with many applications of huge importance. We focus on the problem that the signal to be detected is superposed by polluting signals which are characterized by a large amplitude and a few dominant directions. Such problems occur for instance in the analysis of seismic signals. We devise numerical algorithms which combine rotation of the given data with one-dimensional and two-dimensional discrete wavelet decomposition techniques respectively. The numerical algorithms are tested on both real and synthetic datasets and are compared with more classical techniques based on Fourier transforms.

2000 Mathematics Subject Classification: 44A12, 65D05, 86A15
1998 ACM Computing Classification System: G.1.1, J.2
Keywords and Phrases: Discrete wavelets, finite element method, Fourier transform, ground-roll, interpolation, rotation, seismic signals, wavelets.
Note: This research was supported by Stichting Technische Wetenschappen (STW), projectnumber CWI44.3403

1. Introduction

We can think of many examples where signals are distorted: be it sound, visual images, or measurements by instruments. If put in this way, the problem of getting rid of the distortion is a very general one. The thing to do is to transform the signal such that the distortion becomes explicit. It follows that the "right" transform should be adapted to the kind of distortion. Finding the "right" transform is equivalent to pinpointing the distortion.

A transform is finding a new basis of representation. It is an art and science to transform a signal step by step in such a manner that the coefficients with respect to the new basis can be divided into a set corresponding to the desirable components and another set corresponding to the undesirable components. If successful, we then can simply mute the coefficients going with the undesirable components.

In this paper we predominantly consider the field of seismic applications. Nevertheless, our methods may prove valuable in other fields as well. For obvious reasons geophysicist are after geometric information on the stratification of impedance beneath the earth's surface. Acquisition of data is done by exciting a signal at the surface. Waves are reflected at interfaces (i.e. where the impedances changes rapidly). Seismometers at the surface record the groundmotion as the reflecting waves arrive.

As the deflections are measured in time and for an array of seismometers, we can represent the recorded signals as a two-dimensional gridfunction (i.e. a rectangular uniform grid where each gridpoint has been assigned a real number). The information of interest is then constituted by hyperbolic shaped events. Unfortunately, these events are overshadowed by ground-roll and surface waves directly stemming from the excitative source. Indeed, these waves often have dominant amplitude and completely blur the picture (compare: a concert of a violin and a drill). This is where our problem differs from denoising-problems in other fields of application. It is a major challenge to filter out the surface waves from the above seismic data. In this paper we investigate the potential of new methods to do so. We merely consider problems where seismic data are superposed by polluting signals which are characterized by a large amplitude and a few dominant directions. We try to explore cautiously the merits of new approaches, we do not claim to solve impressive practical problems at this point.

The methods we discuss can be characterized by the following template with seven subsequent steps:
Directional denoising template (DD-template):

step 1 Flatten data.

step 2 Rotate data.

step 3 Transform data.

step 4 Adapt (mute, limit) appropriate coefficients, i.e. filtering.

step 5 Apply backtransform.

step 6 Derotate data.

step 7 Deflatten data.

Apart from step 2 (and its counterpart step 6) this is quite a conventional framework. Most or all present methods fit in this framework where often 'fast Fourier' is used as transform (step 3). Step 1 (and its counterpart step 7) is a common method to compensate for the exponential decay in time of the signal. Step 2 (and its counterpart step 6) is unusual, and a main topic in this paper. For reasons to be explained later, we rotate the data such that they more or less align with either horizontal or vertical gridlines. The actual filtering takes place at step 4, its success depends on whether the transform succeeds in isolating the polluting signals. In a way, the combination of steps 1–3 can be seen as one big transform with the aim of making the undesirable components explicit.

Another main topic in this paper is the choice of the transform. We investigate the applicability of wavelet transforms as an alternative to the Fourier transform. Results of earlier research in this respect are reported in [1, 2, 3, 4, 5]. In [20] a preliminary sketch of our method can be found.

What's new in our method is determined by two elements: rotation/interpolation and the transform (of wavelet-type). We organize the paper as follows. Section 2 contains a description and analysis of the rotation/interpolation. Section 3 discusses application of the one-dimensional wavelet X-ray transform and the two-dimensional discrete wavelet transform respectively. We make a comparison with the fast Fourier transform. Associated with these respective transforms we use simple filtering techniques (step 4). In Section 4 some numerical results are reported for a small but realistic dataset. In the last section, Section 5, we end up with a few concluding remarks.

2. Rotation/Interpolation

2.1 Introduction

Why rotate the data? As with every transform in the context of seismic data processing the aim is to make a better separation between the unwanted components and the events of interest. For the two-dimensional wavelet-decomposition J.C. Cohen et al. [1] showed that if the angle between ground-roll and the gridlines is within well-defined bounds a bias exists for distribution of the energy towards either horizontal detail or vertical detail wavelet coefficients. These findings supported our research to investigate the effects of rotation. Figure 1 shows how a simple synthetic dataset on a uniform 300 × 300-grid (left-hand picture) is rotated / interpolated onto a uniform 600 × 600-grid (right-hand picture). The dataset consists of low-amplitude hyperbolic events on which a high-amplitude function g has been superimposed. The function g impersonates the ground-roll. After rotation, the function g needs to be (merely) approximately aligned with the grid. Figure 2 illustrates the improvement of separating capabilities by rotation for this example set. Horizontally we vary the frequency of the superimposed distortion (Figure 1 shows frequency 4). Vertically the energy is depicted of the horizontal, vertical and diagonal detailcomponents as found by a 2D-wavelet transform (biorthogonal, energy accumulated over all scales). We observe that rotation indeed improves the separation between horizontal and vertical details. This holds for both low and several (but not all) high frequencies (there appears to be a break-even point). The end of the horizontal scale corresponds to the Nyquist-frequency.

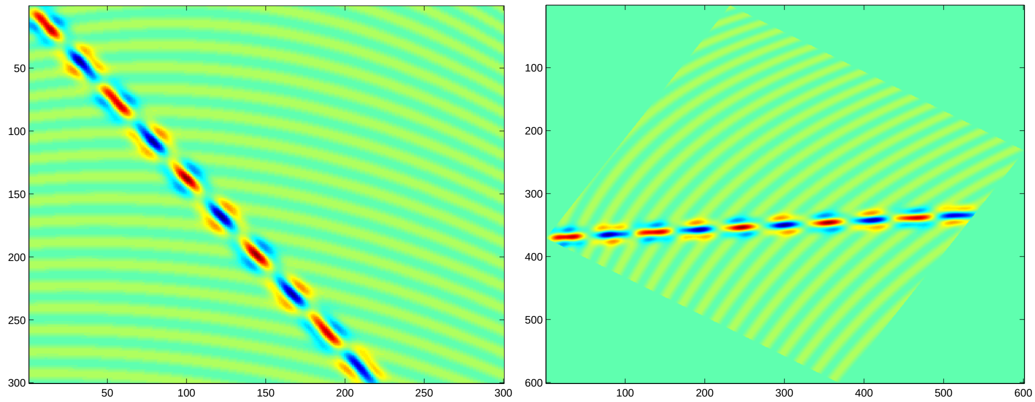


Figure 1: Idealized testproblem (frequency= 4)

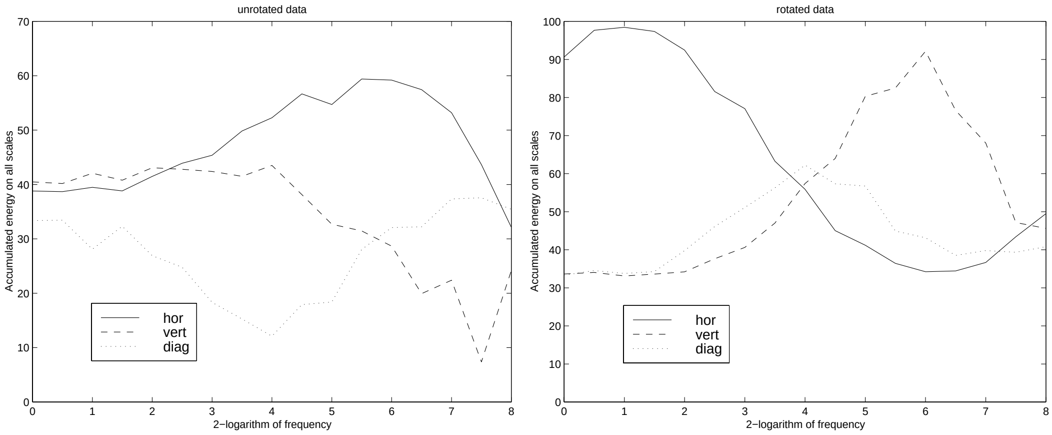


Figure 2: Energy distribution

Here, the "right" angle of rotation is chosen visually throughout. However, with the ground-roll characterized as a low frequency and high amplitude event, ways exist to determine the "right" angle numerically. Demands to accuracy in this respect are low. In the results to come, the quality of the corrections proved not too sensitive with respect to the angle and some deviation is tolerable (up to, say, 10 degrees).

We have the following matters of concern:

1. Accuracy.

2. Costs (complexity).

We have got an original rectangular domain R with origin o, unit vectors e_1, e_2 along the sides and cornerpoints o, $o + ae_1$, $o + be_2$, $o + ae_1 + be_2$ where a, $b > 0$ are the dimensions. With each rotation of angle θ there exists a rectangle R' with origin o' and unit vectors e'_1, e'_2 that envelopes rectangle R efficiently (see Figure 3, $\angle(e_1, e'_1) = \theta$). The original grid Ω_h with meshsize $h = (h_1, h_2) \in \mathbb{R}^2$ is defined as:

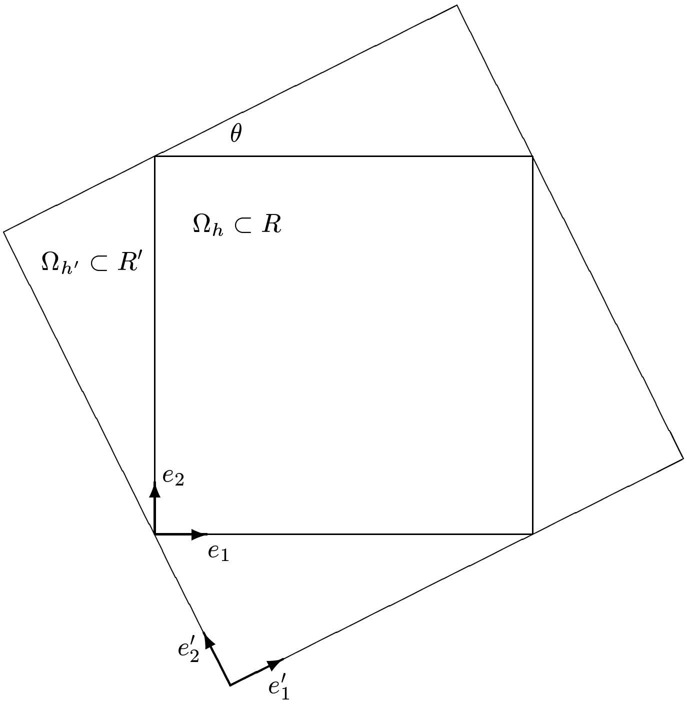


Figure 3: Rotated domain enveloping original domain efficiently.

$$\Omega_h \equiv \left\{ j_1 h_1 e_1 + j_2 h_2 e_2 \mid \mathbf{j} \in \mathbb{Z}^2 \right\} \cap R. \tag{2.1}$$

The rotated grid $\Omega_{h'}$ is defined by:

$$\Omega_{h'} \equiv \left\{ j_1 h'_1 e'_1 + j_2 h'_2 e'_2 \mid \mathbf{j} \in \mathbb{Z}^2 \right\} \cap R'. \tag{2.2}$$

A mapping $g_h \ : \ \Omega_h \to \mathbb{R}$ is called a gridfunction on Ω_h. For any given gridfunction g_h on Ω_h we can define an interpolating function $\widetilde{g_h} \ : \ R \to \mathbb{R}$ with

$$\widetilde{g_h} \equiv \sum_{\mathbf{i} \in I(\Omega_h)} g_{\mathbf{i}} \varphi_{\mathbf{i}}^h, \ g_{\mathbf{i}} \in \mathbb{R}, \ \widetilde{g_h} \in L^2(\mathbb{R}^2), \tag{2.3}$$

where

$$I(\Omega_h) \equiv \left\{ \mathbf{j} \in \mathbb{Z}^2 \mid j_1 h_1 e_1 + j_2 h_2 e_2 \in \Omega_h \right\}$$

and $\varphi_{\mathbf{i}}^h$ are standard nodal basisfunctions $\in L^2(\mathbb{R}^2)$. These basisfunctions can be e.g. piecewise constant functions. In the upcoming section we employ the spaces $\widetilde{B_h}$ and $\widetilde{B_{h'}}$ generated by the piecewise bilinear hatfunctions on the grids Ω_h and $\Omega_{h'}$ respectively.

2.2 Accuracy

Let Ω_h denote the original grid and $\Omega_{h'}$ denote the rotated grid with meshsizes h and h' respectively, see Figure 3 (in practice it suffices that both Ω_h and $\Omega_{h'}$ are curvilinear grids). Figure 4 shows a detail of both Ω_h and $\Omega_{h'}$ intersecting at an angle θ. Let $V(\Omega_h)$, denote the space of gridfunctions (vectors) on Ω_h:

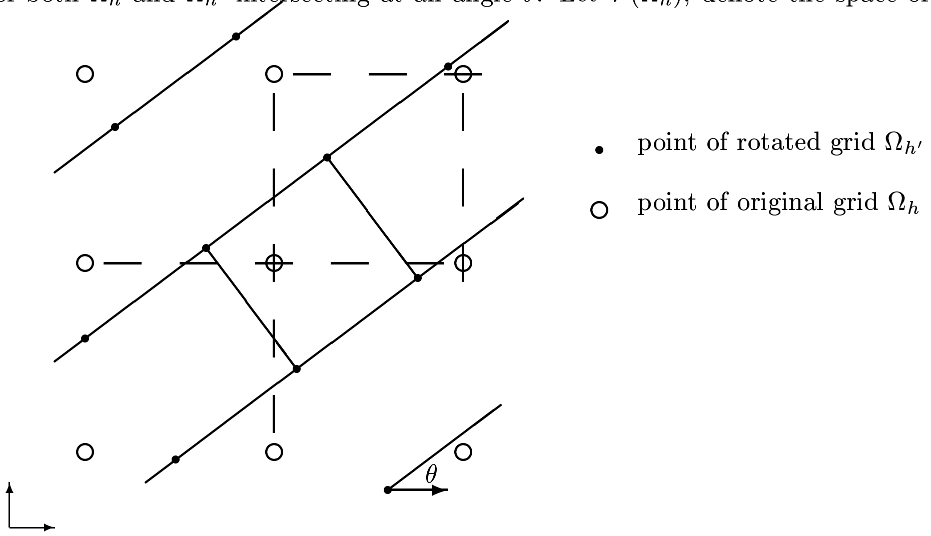

● point of rotated grid $\Omega_{h'}$
○ point of original grid Ω_h
θ

Figure 4: Rotation of gridfunction.

$$V(\Omega_h) \equiv \{g_h \mid g_h \ : \ \Omega_h \to \mathbb{R}\}. \tag{2.4}$$

Likewise $V(\Omega_{h'})$ is defined as the space of gridfunctions on $\Omega_{h'}$. We require that the rotation

$$P_{h'h} : \ V(\Omega_h) \to V(\Omega_{h'}) \tag{2.5}$$

is a linear transform. Similarly, we require that the backrotation

$$P_{hh'} : \ V(\Omega_{h'}) \to V(\Omega_h) \tag{2.6}$$

is a linear transform. (We have omitted θ from the notation.) Let I_h be the identity operator on $V(\Omega_h)$ and define:

$$S_h \quad : \quad V(\Omega_h) \to V(\Omega_h), \tag{2.7a}$$

$$S_h(h') \quad = \quad P_{hh'} P_{h'h} \tag{2.7b}$$

i.e. rotation followed by backrotation. (This operator applies to data that are not adapted by step 4 in the template.) Ideally, $S_h(h')$ should be identical to I_h but this is not feasible.

The template offers a way of countering the expected smoothing effect of $S_h(h')$ as we do not need to rotate and derotate the full set of seismic data. Instead, we only need to derotate the *correction* obtained at step 5 and to add this to the original data.

Adjoint interpolation Suppose that $P_{h'h}$ has already been defined. Then we can define the adjoint $P^{(a)}_{hh'}$ of $P_{h'h}$:

$$P^{(a)}_{hh'} : \ V(\Omega_{h'}) \to V(\Omega_h)$$

in the sense that it satisfies:

$$\int_R (\widetilde{P_{h'h} g_h}) \widetilde{f_{h'}} d\Omega = \int_R \widetilde{g}_h \Big(\widetilde{P^{(a)}_{hh'} f_h}\Big) d\Omega. \tag{2.8}$$

If $\varphi^h_{\mathbf{i}}$, $\mathbf{i} \in I(\Omega_h)$ and $\varphi^{h'}_{\mathbf{j}}$, $\mathbf{j} \in I(\Omega_{h'})$ are piecewise bilinear hatfunctions on Ω_h and $\Omega_{h'}$ respectively then this adjoint exists and the construction follows from:

$$\int_R (\widetilde{P_{h'h} \varphi^h_{\mathbf{i}}}) \widetilde{\varphi^{h'}_{\mathbf{j}}} d\Omega = \int_R \widetilde{\varphi^H_{\mathbf{i}}} \Big(\widetilde{P^{(a)}_{hh'} \varphi^{h'}_{\mathbf{j}}}\Big) d\Omega \tag{2.9}$$

for all $\mathbf{i} \in I(\Omega_h)$ and $\mathbf{j} \in I(\Omega_{h'})$.

Bilinear interpolation For convenience, we may perform the rotation by simple interpolation method:

$$P_{h'h} = P^{\mathrm{bi}}_{h'h} \tag{2.10}$$

bilinear interpolation between the values at the gridpoints of Ω_h. An appropriate choice for the backrotation would be the above (scaled) adjoint of $P_{h'h}$. Instead, for reasons of simplicity, we may take again bilinear interpolation (with respect to $\Omega_{h'}$)

$$P_{hh'} = P^{\mathrm{bi}}_{hh'}. \tag{2.11}$$

The definitions (2.10) and (2.11) make $S_h(h') = S_h(h')^{\mathrm{bi}}$ an operator which yields an undesirable smoothing effect. Nevertheless, we can get $S_h(h')$ arbitrarily close to I_h when $h' \downarrow 0$. In fact,

$$\|I_h - S_h(h')^{\mathrm{bi}}\|_\infty = \mathcal{O}(h'). \tag{2.12}$$

That is, rotation followed by backrotation, both performed by bilinear interpolation, yields first order accuracy with respect to the meshsize of the rotated grid. This can be seen as follows: let $\widetilde{f_h}$ be the (continuous!) piecewise bilinear interpolating function of a gridfunction f_h. At the points of $\Omega_{h'}$, $f_{h'} = P^{\mathrm{bi}}_{h'h} f_h$ finds the exact values of $\widetilde{f_h}$ at these points (injection). Because $\widetilde{f_h}$ has no continuous first partial derivatives, $P_{hh'} f_{h'} - f_h$ is merely first order accurate with respect to h'. Moreover, first order accuracy can only be observed if $h' < h$, which is also intuitively clear if we put the natural demand that the rotated gridfunction should not lose information with respect to the original.

Interpolation by finite elements The above bilinear interpolation has the advantage of simplicity but is rather crude. An alternative method is to use the finite element method. For given gridfunction g_h we search for a gridfunction $f_{h'}$ that satisfies $\widetilde{f_{h'}} = \widetilde{g_h}$, $\widetilde{f_{h'}} \in \widetilde{B_{h'}}$, $\widetilde{g_h} \in \widetilde{B_h}$. That is

$$\sum_{\mathbf{j} \in I(\Omega_{h'})} f_{\mathbf{j}} \, \varphi^{h'}_{\mathbf{j}} = \sum_{\mathbf{i} \in I(\Omega_h)} g_{\mathbf{i}} \, \varphi^h_{\mathbf{i}}, \ \varphi^{h'}_{\mathbf{j}} \in \widetilde{B_{h'}}, \varphi^h_{\mathbf{i}} \in \widetilde{B_h} \tag{2.13}$$

where $\varphi^{h'}_{\mathbf{j}}$ and $\varphi^h_{\mathbf{i}}$ are piecewise bilinear hatfunctions on $\Omega_{h'}$ and Ω_h respectively. Generally, this requirement is impossible to satisfy and we therefore approximate as follows. We introduce the bilinear form:

$$a \quad : \quad L^2(\mathbb{R}^2) \times L^2(\mathbb{R}^2) \to \mathbb{R}, \tag{2.14a}$$

$$a(\varphi, f) \quad \equiv \quad \int_{R'} \varphi f d\Omega. \tag{2.14b}$$

We state our approximation: find $f_{h'} \in \widetilde{B_{h'}}$ such that

$$a(\varphi_{\mathbf{k}}^{h'}, \widetilde{f_{h'}}) = \int_{R'} \varphi_{\mathbf{k}}^{h'} \widetilde{g}_h d\Omega \tag{2.15}$$

for the set of piecewise bilinear hatfunctions $\varphi_{\mathbf{k}}^{h'}$ that span $\widetilde{B_{h'}}$. That is:

$$\sum_{\mathbf{j} \in I(\Omega_{h'})} f_{\mathbf{j}} \int_R \varphi_{\mathbf{j}}^{h'} \varphi_{\mathbf{k}}^{h'} d\Omega = \sum_{\mathbf{i} \in I(\Omega_h)} g_{\mathbf{i}} \int_R \varphi_{\mathbf{i}}^{h} \varphi_{\mathbf{k}}^{h'} d\Omega \tag{2.16}$$

for all $\mathbf{k} \in I(\Omega_{h'})$. Indeed, (2.13) implies (2.16) but not vice versa. Because $\varphi_{\mathbf{j}}^{h'}$ and $\varphi_{\mathbf{i}}^{h}$ are piecewise bilinear hatfunctions on $\Omega_{h'}$ and Ω_h respectively, (2.16) results into a linear system of equations where the corresponding matrix $M_{h'}$ (on the left-hand side) is of special type. In the inner area of $\Omega_{h'}$ a row of the matrix $M_{h'}$ can be written in two-dimensional stencil notation as:

$$\begin{bmatrix} \frac{1}{36} & \frac{1}{9} & \frac{1}{36} \\ \frac{1}{9} & \frac{4}{9} & \frac{1}{9} \\ \frac{1}{36} & \frac{1}{9} & \frac{1}{36} \end{bmatrix}. \tag{2.17}$$

This stencil shows the non-zero values of the result of the action of $M_{h'}$ on a gridfunction which equals 1 at one point (given by $\mathbf{j}$) and 0 elsewhere. Note that both the upper and lower part of the stencil are multiples of the middle part. The particular system (2.16) is solvable and, moreover, in a direct manner so efficient that it takes a number of operations linearly proportional to the number of unknowns $f_{\mathbf{j}}$ (see [16, § 14.3]). However, automation of the computation of the right-hand side of (2.16) is a nontrivial task.

We denote the error in the approximation as:

$$e^{h'} \equiv \widetilde{f_{h'}} - \widetilde{g}_h. \tag{2.18}$$

We remark that $e^{h'} \in L^2(\mathbb{R}^2)$ but (generally) $e^{h'} \notin \widetilde{B_{h'}}$. It can be proven that:

1. $a(\varphi_{\mathbf{k}}^{h'}, e^{h'}) = 0$ for all $\varphi_{\mathbf{k}}^{h'} \in \widetilde{B_{h'}}$ (i.e. the error in the approximation is perpendicular to the bilinear hatfunctions),
2. $a(e^{h'}, e^{h'}) \leq a(\widetilde{b_{h'}} - \widetilde{g_h}, \widetilde{b_{h'}} - \widetilde{g_h})$ for all $\widetilde{b_{h'}} \in \widetilde{B_{h'}}$ (i.e. in $\widetilde{B_{h'}}$ there is no better approximation to $\widetilde{g_h}$ than $\widetilde{f_{h'}}$).

The proof is analogous to the one in Hughes [6, Ch. 4]. Because of statement 2 we say that $\widetilde{f_{h'}}$ is the *least squares approximation* of $\widetilde{g_h}$.

Let us consider this finite-element type interpolation in terms of linear algebra, that is (2.16) yields:

$$M_{h'} f_{h'} = C_{h'h} g_h \tag{2.19}$$

with $M_{h'}$, $C_{h'h}$ matrices. We recall that $M_{h'}$ is symmetric and can be inverted (true with the piecewise bilinear hatfunctions, see also (2.17)). From (2.19) it follows that

$$f_{h'} = P_{h'h} g_h$$

with $P_{h'h}$ defined as

$$P_{h'h} \equiv M_{h'}^{-1} C_{h'h}. \tag{2.20}$$

This leads to the definition of a possible backrotation via the transpose of $P_{h'h}$:

$$P_{h'h}^T = C_{h'h}^T M_{h'}^{-1}. \tag{2.21}$$

Alternatively, the backrotation could again be defined via a least squares approximation.

2.3 Costs

After rotation we obtain a rectangle that envelopes our original domain. Consider a $n \times m$ domain of gridpoints. We need to maintain the original resolution after rotation over an angle θ. This implies that the number of gridpoints on the rotated grid is λ times as large as on the original grid with

$$\lambda(\theta) = 1 + \frac{1}{2}\left(\frac{n}{m} + \frac{m}{n}\right)\sin(2\theta). \tag{2.22}$$

Though it makes a subsequent fast wavelet transform (FWT) λ times as expensive, the rotation does not affect the *order* of complexity. The additional work and storage capacity induced by rotation is linearly proportional to the number of gridpoints. Moreover, on the rotated grid many of the wavelet coefficients will be zero on the fine-grained scales, this may be taken advantage of, depending on the datastructure chosen.

3. Transforming and Filtering

3.1 Introduction

After a transform, be it Fourier, wavelet or other, the gridfunction remains the same but is expressed with respect to a different basis. The use of a transform is to find an alternative presentation by which the unwanted components are represented by much fewer coefficients and/or by coefficients which are more easy to identify (that is, clearly separated from coefficients of the wanted components). By the way, with this point of view rotation is just part of an overall transform. Three of such transforms are discussed below as candidates for Step 3 in the DD-template, namely the 1D-wavelet X-ray transform, the 2D discrete wavelet transform and the (classical) fast Fourier transform (Sections 3.2–3.6).

Step 4 of the DD-template means that the transform is followed by filtering techniques, i.e. ways of adapting the coefficients c resulting from the transform. By numerical experiments our ways of adapting naturally evolved from $\mathcal{A}$ to $\mathcal{C}$:

$\mathcal{A}$ $c := \max(\min(c, \text{upperbound}), \text{lowerbound});$

$\mathcal{B}$ **if** c satisfies mute-criterion **then** $c := 0$ (mute c);

$\mathcal{C}$ **if** c satisfies mute-criterion **then** replace c by interpolation between nearby coefficients that do not satisfy the mute-criterion.

The values 'lowerbound' and 'upperbound' are defined in a problem-dependent way. Often $\mathcal{A}$ turns out to be too crude: the ground-roll remains too large. This is improved by $\mathcal{B}$ which is more radical by muting coefficients completely. The mute-criterion in $\mathcal{B}$, $\mathcal{C}$ still needs to be filled in. It can be simple like:

$$\text{mute-criterion} = (c > \text{upperbound } \textbf{or } c < \text{lowerbound})$$

be it that the values 'lowerbound' and 'upperbound' still need to be determined. For a more subtle mute-criterion see Section 3.3. It shows that such a criterion can be sophisticated and complicated. However, all our mute-criterions used so far, rely heavily on the rationale that coefficients of the unwanted components are substantially larger than of the wanted components. An important difference of $\mathcal{C}$ with both $\mathcal{A}$ and $\mathcal{B}$ is that the coefficients which need to be adapted are replaced by interpolation between remaining nearby coefficients that (are supposed to) correspond to the wanted components. Section 3.5 contains more details. Finally, note that with Fourier transform we create only one set of coefficients that need to be filtered. In contrast, with wavelet transform we usually create many sets of coefficients. Fourier transform together with $\mathcal{B}$ works satisfactory if we are well able to isolate the coefficients of the unwanted components.

3.2 1D-wavelet X-ray transform

Consider a function f on a bounded and closed domain $\Omega \subset \mathbb{R}^n$. We assume that $f = 0$ outside Ω. An arbitrary line $\mathcal{L}$ in $\mathbb{R}^n$ can be parametrized as $x + t\theta$, with origin $x \in \mathbb{R}^n$, $\theta \perp x$ a vector $\in \mathbb{R}^n$ with length 1 designating direction and parameter $t \in \mathbb{R}$. Suppose that $\mathcal{L}$ intersects Ω. The well-known *X-ray transform* [10]

$$Pf(\theta, x) = \int_{\mathbb{R}} f(x + t\theta)\, dt \tag{3.1}$$

integrates f along the line $\mathcal{L}$. For an illustration see Figure 5. Often f represents a density in some medium. The transform is extensively used in tomography where it describes how radiation emitted in the direction θ is absorbed by living tissue. In this paper, f would correspond to seismic data, that is amplitude as function of time and space.

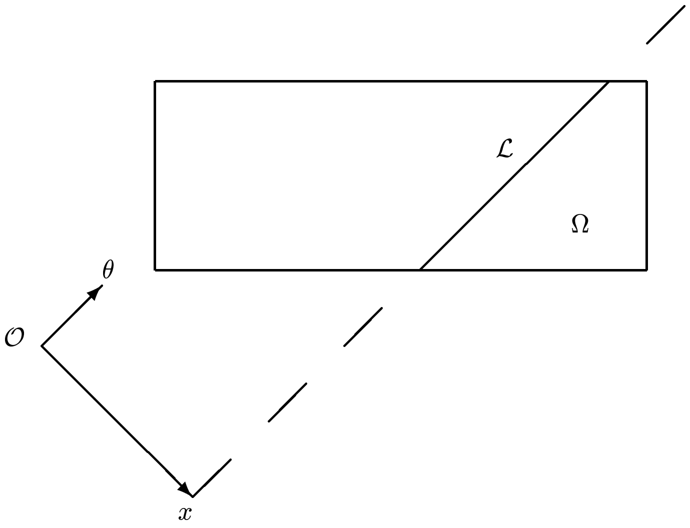


Figure 5: X-ray intersecting Ω.

If, as an extension of (3.1), the function f is weighted by a wavelet function ψ, this gives rise to the so-called *wavelet X-ray transform* [7, 12, 18, 19, 20] given by

$$P_\psi f(\theta, x, b, a) = \int_{\mathbb{R}} f(x + t\theta) \frac{1}{\sqrt{a}} \overline{\psi\left(\frac{t-b}{a}\right)}\, dt \tag{3.2}$$

The *translation, dilation* parameter pair (b, a) is taken from the open upper half plane

$$\mathbb{H} = \{(x, y) \in \mathbb{R}^2 \mid y > 0\}, \tag{3.3}$$

just as for the ordinary continuous 1-D wavelet transform. The wavelet ψ satisfies additional conditions so that the function f can be reconstructed from its wavelet X-ray coefficients $P_\psi f(\theta, x, b, a)$. The transform has been applied to detect events in radar images [13].

The analogue of the *discrete* 1-D wavelet transform is the *discrete* X-ray wavelet transform. Therefore $\mathbb{H}$ is replaced by a countable subset $K \subset \mathbb{H}$. We opt for the usual choice

$$K = \{(a, b) \mid a = 2^j, b = 2^j k, (j, k) \in \mathbb{Z}^2\}. \tag{3.4}$$

These dyadic scales and positions make the analysis far more efficient but without much loss of accuracy. Zuidwijk [17] gives a theoretical justification of the discrete X-ray wavelet transform together with sufficient conditions for a meaningful (convergent) reconstruction. The transforms make use of wavelet orthonormal bases or, more generally, of biorthogonal systems of wavelet Riesz bases.

Our given seismic data are discrete, that is, discrete data recorded in discrete time and discrete space. This implies that we need a *discretization* of (3.2) with respect to f, for f is not given as a continuous function. (We have to beware that the nomenclature concerning "discrete" can lead to confusion.) A first approach to handle this is the following. The underlying grid of the data is rectangular. We construct a piecewise bilinear function g that is the interpolating function of the seismic data. Figure 6 shows the squares where g is locally determined by the given values at the four cornerpoints $\circ$. Obviously, at the gridlines g is continuous but not differentiable. If we assume f to be this particular function ($f = g$) then definition (3.2) is applicable with respect to both the continuous (3.3) and discrete case (3.4). Note that $g|_{\mathcal{L}}$ (g restricted to $\mathcal{L}$) is a piecewise parabolic function. Note further that if we would use the Haar-wavelet for ψ in the wavelet X-ray transform (3.2) we can easily compute the integration exactly, without the need to use some approximation rule for numerical integration.

The discretization of the wavelet X-ray transform (3.2) implies the use of gridpoints at the X-ray $\mathcal{L}$ as well, see Figure 6. The gridpoints of the ray do not coincide with the ones of the seismic data.

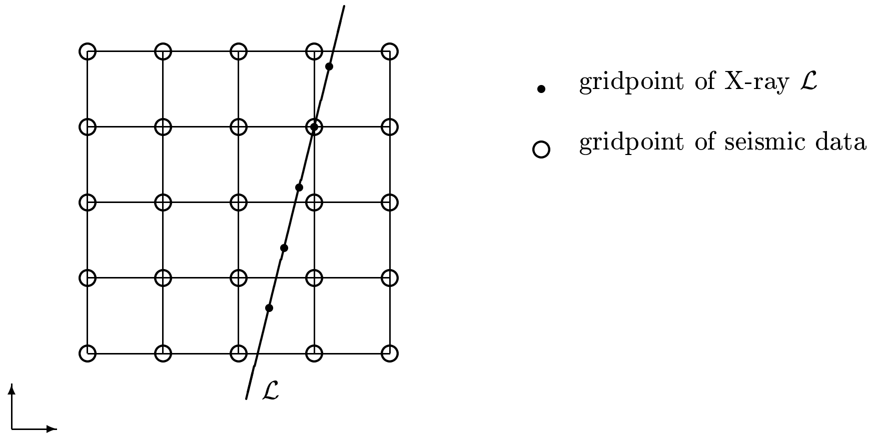

$\mathcal{L}$
● gridpoint of X-ray $\mathcal{L}$
○ gridpoint of seismic data

Figure 6: Gridpoints of data and X-ray.

At this point we may consider the *fast* wavelet X-ray transform, the analogue of the *fast wavelet transform* [8]. The complexity of this transform is $O(l)$ with l the number of gridpoints at the ray $\mathcal{L}$ (the amount of work is linearly proportional to the number of gridpoints l). If we would aim for an X-ray wavelet decomposition that is capable of a complete reconstruction of a $n \times m$ dataset, we would need rays through all nm points of the dataset leading to a complexity of $O(mnl)$. This is far too high for practical purposes (e.g. consider the case $m = n = l$).

A different approach for the implementation of the wavelet X-ray transform is to rotate the data beforehand as explained in Sections 1–2. This approach fits in the DD-template (Section 1) and it is one of which numerical results are reported on in this paper. The angle of rotation is chosen such that $\mathcal{L}$ aligns with either a horizontal of vertical gridline of the rotated grid. We then apply fast wavelet X-ray transform on all of the gridlines $\parallel \mathcal{L}$.

For the purpose of denoising (removal of the ground-roll) we choose $\mathcal{L}$ such that it lies approximately *along* respectively *across* the undesirable component (see the wake of g in Figure 1). Suppose $\mathcal{L}$ is along the ground-roll and the waveletdecomposition is performed for each gridline $\parallel \mathcal{L}$ of the rotated gridfunction. The ground-roll is then predominantly presented at the coarse-grained scales. Suppose $\mathcal{L}$ is across the ground-roll and the waveletdecomposition is performed for each gridline $\parallel \mathcal{L}$ of the rotated gridfunction. The ground-roll is now predominantly presented at the fine-grained scales. This yields two different approaches for removing the ground-roll. In this way, we employ (3.2) as a tool to investigate given seismic data, and do so literally from different angles. It is also possible to combine the approaches. We give numerical examples further on in this paper.

Orthogonalization of corrections Given a correction to the data, we devise here a means to improve this correction. If more than one correction is available, this device is generalized.

We obtained a correction c on the original dataset f_{orig} which results in the gridfunction

$$f_{\text{new}} = f_{\text{orig}} - c.$$

We may consider instead

$$f_{\text{opt}} = f_{\text{orig}} - \alpha c,$$

$\alpha \in \mathbb{R}$, where α is chosen such that the 2-norm of f_{opt} is minimized which is equivalent to demanding $f_{\text{opt}} \perp c$, from which it follows that

$$\alpha \equiv \frac{(f_{\text{orig}}, c)}{(c, c)}. \tag{3.5}$$

With the 2D wavelet approach to come, orthogonality of the corrections $c \perp f_{\text{new}}$ is immediate with filtering of type $\mathcal{B}$ (above) by definition of the wavelets used. The corrections obtained by the X-ray approach prove often not orthogonal to the original gridfunction. Indeed, in practice we observe that usually $|\alpha| > 1$, values of 2–4 are common.

Likewise, in case we have two linear independent corrections c_1 and c_2 available, we may consider

$$f_{\text{opt}} = f_{\text{orig}} - \alpha_1 c_1 - \alpha_2 c_2,$$

with α_1, $\alpha_2 \in \mathbb{R}$ chosen such that again the 2-norm of the resulting gridfunction is minimized, which is equivalent to $f_{\text{opt}} \perp c_k$ for $k = 1, 2$. The latter leads to the following system of equations:

$$\begin{pmatrix} (c_1, c_1) & (c_2, c_1) \\ (c_1, c_2) & (c_2, c_2) \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \begin{pmatrix} (f, c_1) \\ (f, c_2) \end{pmatrix}. \tag{3.6}$$

The generalization to more than two corrections is straightforward.

Numerical example of the Wavelet X-ray transform Figure 8 is the result when an original dataset shown by Figure 7 has been filtered by the X-ray transform across and along. The DD-template has been used (including rotation) and the correction has been orthogonalized.

3.3 2D discrete wavelet transform

In this section we describe a transform and filtering technique that differs drastically from the previous section: instead of two separate one-dimensional waveletdecompositions (horizontal and vertical) we now consider the two-dimensional wavelet decomposition. Within this class we confine ourselves to separable wavelets. We propose the following algorithm:

1. Choose a separable wavelet.
2. Choose the number of levels L.
3. Apply the fast wavelet transform to the gridfunction and store:

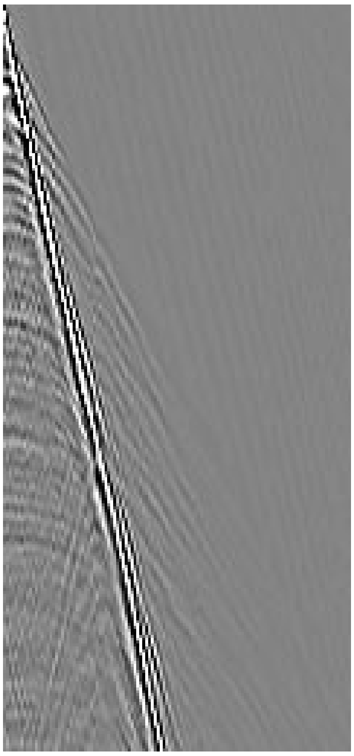


Figure 7: Original dataset.

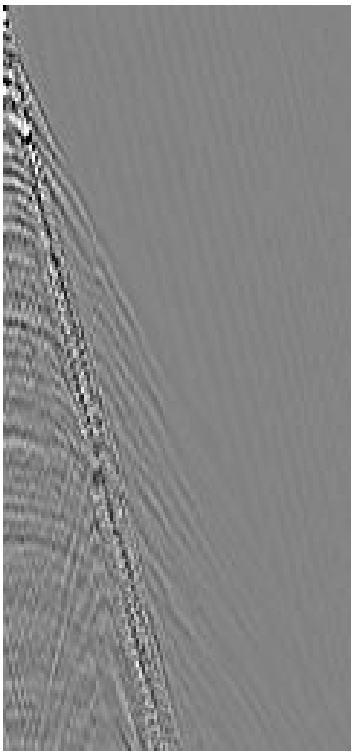


Figure 8: Filtered with the X-ray transform.

- the horizontal detail coefficients H_l, $l = 1, \cdots, L$;
- the vertical detail coefficients V_l, $l = 1, \cdots, L$;
- the diagonal detail coefficients D_l, $l = 1, \cdots, L$;
- the approximation coefficients A_L.

4. Apply a filter of type $\mathcal{C}$ (see above) to H_l, V_l, D_l, $l = 1, \cdots, L$ and A_L.

The construction of the wavelet coefficients is done by the well-known procedure of Mallat [8]. In 2D the number of detail coefficients decreases with a factor 4 for each subsequent scale l.

We do not know beforehand on which scale l the ground-roll is reproduced best. Therefore, in step 2 it is wise to take the maximum level allowed (the rule is the last level for which at least one coefficient is correct, see [9]). This maximum depends on the type of wavelet chosen in step 1. Some wavelets fit better with the data then others, e.g. the Haar-wavelet produces somewhat erratic results.

The algorithm entails that we have to adapt $3L + 1$ sets of wavelet coefficients, see Section 3.1. Obviously, for a substantial L it becomes prohibitive to do this by hand. We need mute-criterions that can be checked numerically and with as few parameters as possible. We experimented with a procedure to prescribe lower and upper bounds for the sets of detail coefficients that is clarified by Figure 9. It relates to the 2D wavelet decomposition, at a certain level $l < L$, of the rotated dataset of Figure 1. The vertical axis depicts the amplitude of the wavelet coefficients. The coefficients are sorted with respect to amplitude, so that we obtain monotonically increasing functions $g : \mathbb{N} \to \mathbb{R}$ (on a finite domain). We observe that many of the horizontal wavelet coefficients dominate as a consequence of rotating the data before application of the discrete wavelet transform. The sudden increases / decreases in the graph indicate which of the coefficients can be supposed to belong to the ground-roll. These locations are used to define the lower and upper bound beyond which coefficients

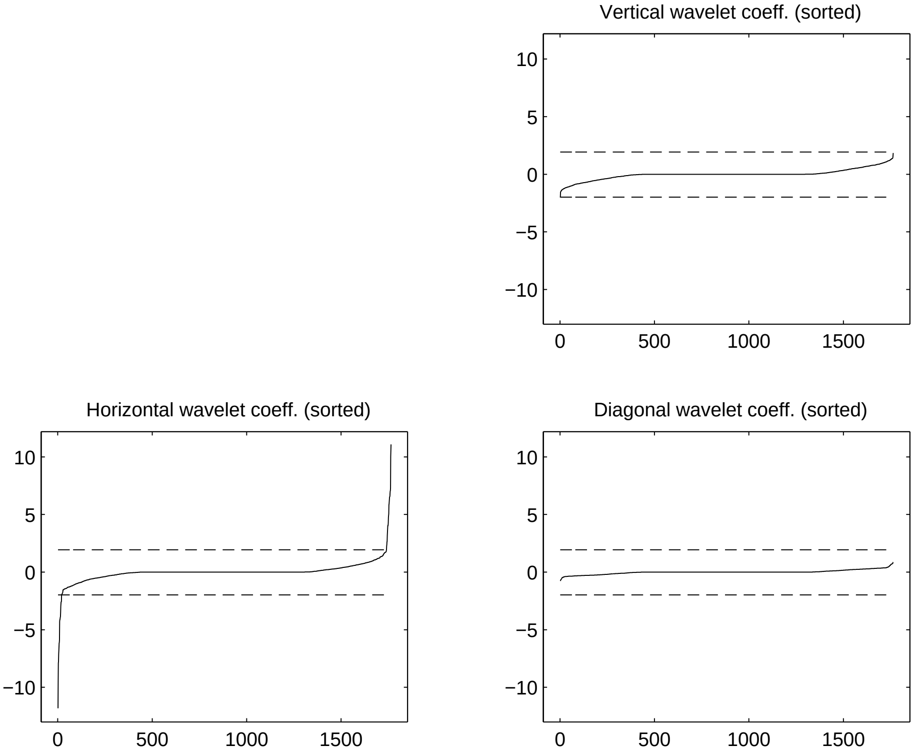


Figure 9: Sorted wavelet coefficients

are muted ($\mathcal{B}$, or adapted according $\mathcal{C}$). We call this mute-criterion the *derivative-jump criterion*. The implementation is a procedure which requires no user interference. It looks for strong curvature, that is, it establishes whether the standard deviation of the derivative of g is exceeded with a large factor. The bounds are indicated as dashed lines in Figure 9 (in this example they do not take effect for the diagonal and vertical detail coefficients). At the coarsest scale ($l = L$) we may prescribe the bounds also to the approximation coefficients A_L but as we take L as large as possible, we choose not do so.

Consider (again) the simple synthetic dataset of the idealized testproblem shown by Figure 1, where weak hyperbolic events (wanted) are superimposed by a high-amplitude function g (unwanted). We apply the above 2D-wavelet based algorithm to the rotated dataset. We choose a compactly supported biorthogonal spline wavelet (to be precise "bior2.2", see [9]) and $L = 6$. Figure 9 applies to level ($l = 4$) of the decomposition. We apply the mute-criterion sketched above and the filter of type $\mathcal{C}$. Intermediate results are demonstrated in Figure 10. From the finest-grained scale (level 1) towards the coarsest-grained scale (level 6) the rotated gridfunction is depicted with accumulated corrections. In this manner we can observe how the correction interferes (hopefully little) with the wanted components. Yet, it should be noted that the ranking order of the scales (levels) has no influence on the end result (the ranking order follows from the algorithm of the fast wavelet transform.) The end result is depicted by Figure 11, it shows the numerical segregation of the wanted and unwanted components, after derotation. We observe that the proposed algorithm handles the idealized problem rather well: the ground-roll has been removed and the (weak) hyperbolic events remain (almost) coherent.

We remark that we can run the algorithm also without rotation. But then the dis-balance between the energy in the horizontal and vertical details diminishes and the algorithm produces a segregation that is not as crisp.

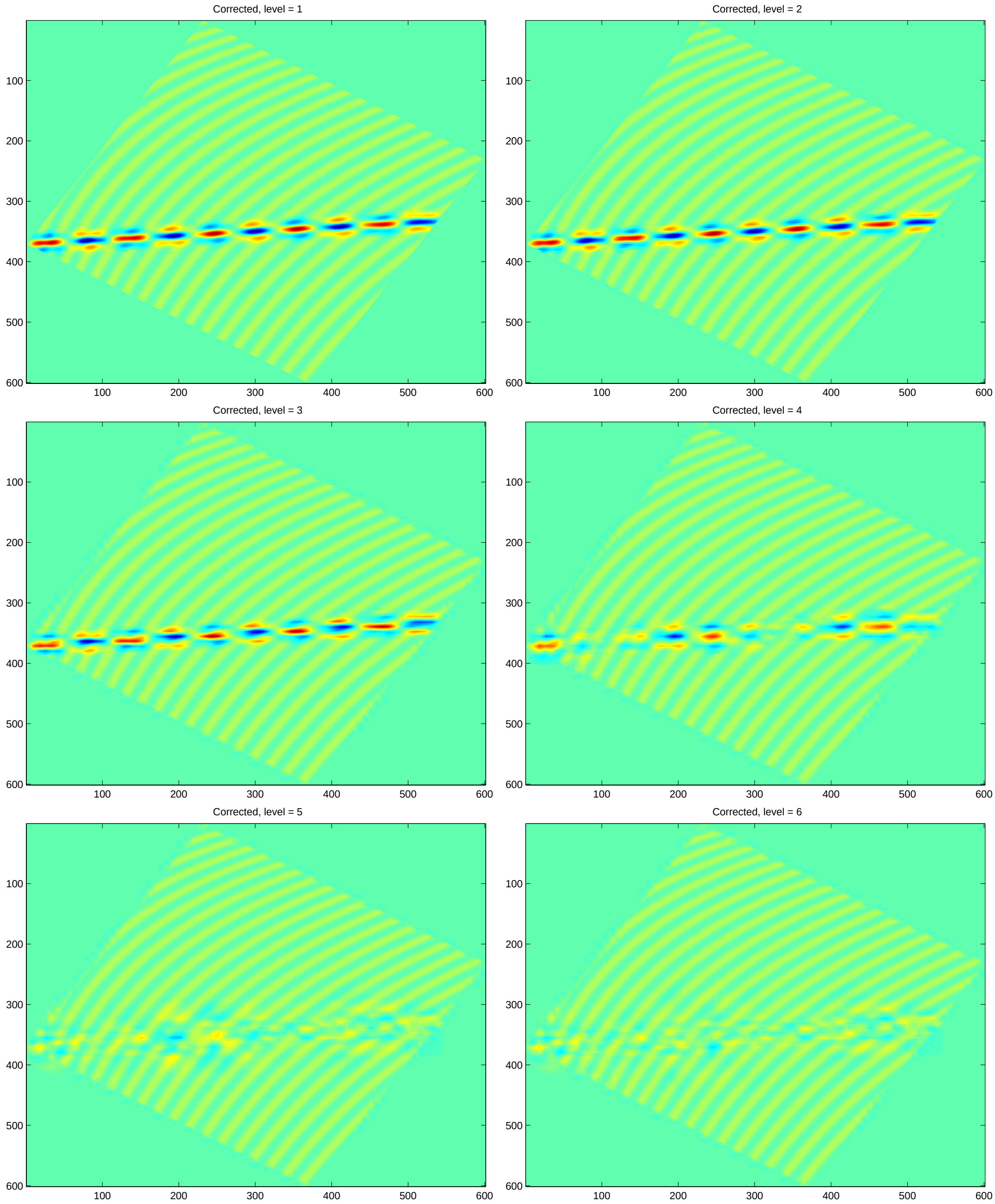


Figure 10: Rotated dataset with accumulated corrections, from level 1 towards level 6.

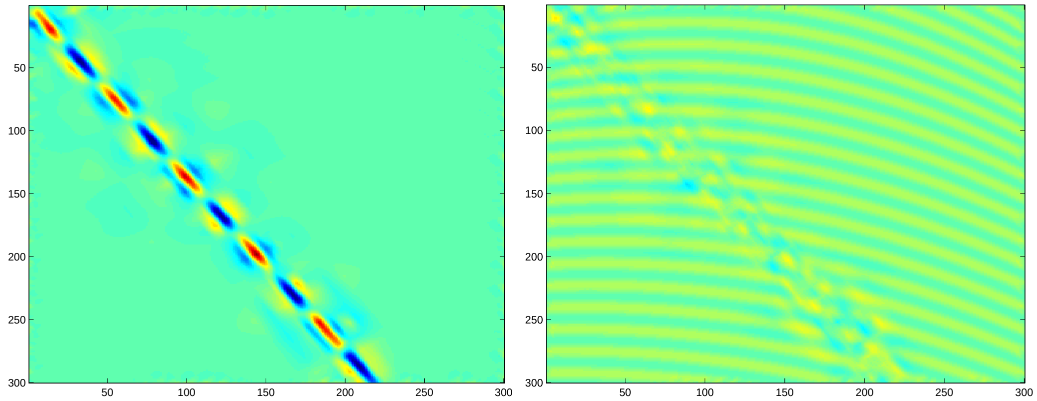
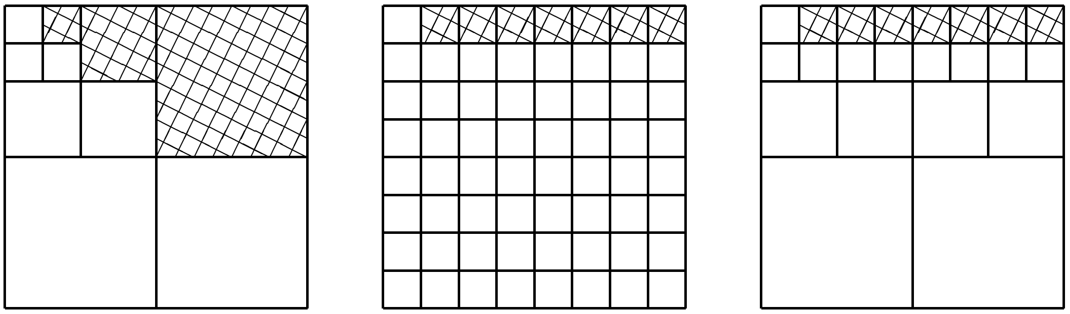


Figure 11: Numerical separation of wanted and unwanted components.

3.4 Wavelet packets

The 2D wavelet packets transform [14] provides an analysis much richer than the standard 2D discrete wavelet analysis. Figure 12 illustrates the difference (for a three-level wavelet decomposition). At one level of refinement in the wavelet decomposition procedure the subdomains of the wavelet coefficients are geometrically ordered as



standard DWT wavelet packets - complete tree wavelet packets - best tree

Figure 12: Structured domains of wavelet coefficients.

$$\begin{bmatrix} a & h \\ v & d \end{bmatrix},$$

where a stands for approximation, h for horizontal detail, v for vertical detail and d for diagonal detail coefficients. The left-hand part of Figure 12 corresponds to the standard discrete wavelet transform (in 2D). At each level only the approximation coefficients (a) are analysed further.

The middle part of Figure 12 shows the subdomains that arise if at each level of refinement all the four kinds (a, h, v, d) of wavelet coefficients are analysed further (yielding the complete quadtree).

The right-hand part of Figure 12 shows the subdomains that arise if at each level of refinement both the approximation and the horizontal detail coefficients (a, h) are analysed further. For the application in this paper, it appears the most suitable choice. The shaded areas indicate the coefficients that will be adapted at step 4 of the DD-template (see Section 4).

3.5 Interpolation of wavelet coefficients

We use wavelet coefficients defined at rectangular and equidistant grids. Hence, the real-valued sets of wavelet coefficients can be looked at as gridfunctions. Mere muting (see Section 3.1) annihilates part(s) of the gridfunction. Instead, we demand that bivariate polynomials of degree $p \geq 0$ are interpolated exactly (between neighbouring gridpoints). That is, we assume a smooth behaviour (in the above sense) of the wavelet coefficients of a signal from which the ground-roll has been removed. Therefore, we first reject the wavelet coefficients indicated by a certain criterion (e.g. the one of Section 3.3) and then we replace the rejected coefficients by values obtained by interpolation between coefficients that were not rejected. See Figure 13 for an illustrated example. In the shaded regions wavelet coefficients are not rejected (accepted). At the point with the cross (with a rejected coefficient) we first consider the square with the gridpoints $\circ$. As this initial square contains no gridpoints with accepted coefficients, we widen the square until it does: see the square with the gridpoints $\bullet$. We then interpolate by using the values at the gridpoints $\bullet$. More formally, we could determine $u_{\mathbf{i}}$ by interpolation as follows:

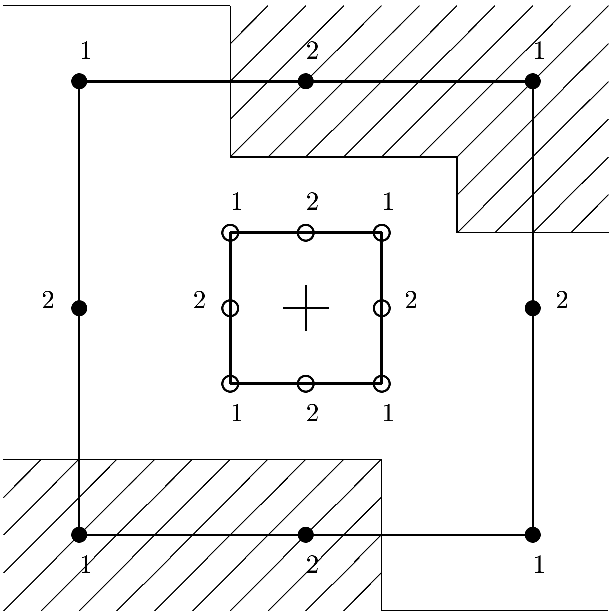


Figure 13: Interpolation of wavelet coefficients.

$$u_{\mathbf{i}} = \frac{\sum_{\mathbf{q}\in\mathbf{Q}} a_{\mathbf{i}+s\mathbf{q}} w_{\mathbf{q}} u_{\mathbf{i}+s\mathbf{q}}}{\sum_{\mathbf{q}\in\mathbf{Q}} a_{\mathbf{i}+s\mathbf{q}} w_{\mathbf{q}}}, \tag{3.7a}$$

with

$$s \in \mathbb{N}, \tag{3.7b}$$

$$\mathbf{i}, \mathbf{q} \in \mathbb{N}^2, \tag{3.7c}$$

$$\mathbf{Q} \equiv \{(-1,1),(0,1),(1,1),(-1,0),(1,0),(-1,-1),(0,-1),(1,-1)\} \subset \mathbb{N}^2, \tag{3.7d}$$

$$w_{\mathbf{q}} = ||q_1| - |q_2|| + 1, \tag{3.7e}$$

$$a_{\mathbf{j}} = \begin{cases} 1 & \text{if} \quad u_{\mathbf{j}} \text{ is accepted} \\ 0 & \text{if} \quad u_{\mathbf{j}} \text{ is rejected} \end{cases} \tag{3.7f}$$

Obviously, at least one $a_{\mathbf{j}}$ needs to be 1. If not, we increase s by steps until it does.

3.6 Fast Fourier transform with rotation

The classical approach of Fourier transform can take benefit from the rotation as well. We assume that ground-roll is characterized by components of low frequency *along* its propagation in time-space and by components of arbitrary (including high) frequency *across* this propagation. As before, we rotate such that the ground-roll gets (approximately) aligned with the horizontal gridlines. We apply the Fourier transform towards the (ω_1, ω_2) plane of harmonics. We proceed by limiting the coefficients along a narrow strip along the ω_2-axis as follows. The strip is defined by $|\omega_1| < b$, ω_2 arbitrary, for some (small) positive value of parameter b (to be tuned). Because of the rotation this is where the ground-roll is expected to be represented now. The coefficients within the strip could be put to zero (muting), instead we replace such coefficients by the value at the nearest gridpoint on the boundary of the strip (a simple interpolation approach). For a numerical example see the next section.

4. More Numerical Results

Already some numerical results were presented in the previous sections. Here we present additional results. As testproblem we consider a small but rather realistic dataset: hyperbolic events heavily polluted with ground-roll. Figure 14 shows the original dataset, We try to remove the ground-roll components by several of the respective means described before (Section 3):

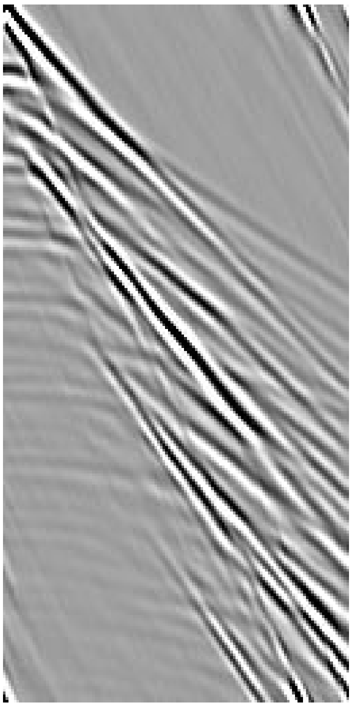


Figure 14: Original dataset.

FFT The fast Fourier transform (see Section 3.6, including interpolation): We apply the DD-template of Section 1 two subsequent times: firstly with an angle of rotation $\theta_1 = -49.6°$, secondly with an angle of rotation $\theta_2 = -38°$. Figure 15 and Figure 16 show the subsequent results.

DWT We apply the DD-template with the (standard) 2D discrete wavelet transform (Section 3.3). As wavelet we choose the Daubechies-8 (db8) wavelet, number of scales is 5, angle of rotation

$\theta = -49.6°$. The shaded area in the left-hand part of Figure 12) indicates which detail coefficients are adapted. We adapt the detail coefficients according to $\mathcal{C}$ (Section 3.1) with the derivative-jump criterion (Section 3.3). Figure 17 shows the result.

WP We apply the DD-template with the transform by wavelet packets (Section 3.4). As wavelet we choose the Daubechies-8 (db8) wavelet, number of levels is 5, angle of rotation $\theta = -49.6°$. The shaded area in the middle or right-hand part of Figure 12) indicates which detail coefficients are adapted. We adapt the detail coefficients according to $\mathcal{C}$ with the derivative-jump criterion. Figure 18 shows the result.

The problem proved not suitable for the 1D-wavelet X-ray transform (Section 3.2).

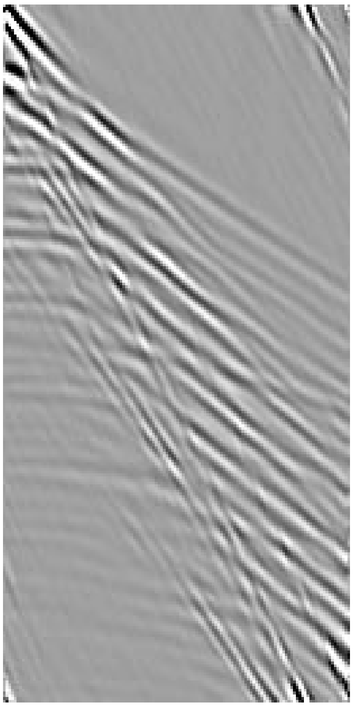


Figure 15: Filtered with **FFT**, θ_1.

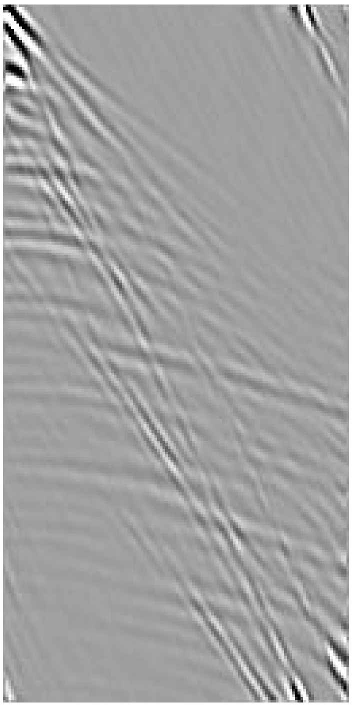


Figure 16: Filtered with **FFT**, θ_1 & θ_2.

5. Concluding Remarks

We have shown that by means of rotating seismic data first, we can more easily identify undesirable components after discrete 1D and 2D wavelet transform or Fourier transform. Some numerical data and background material can be viewed at [11, 15].

The wavelettransform is competing with the Fourier transform with respect to filtering results but so far offers no real improvement. An alternative approach could be the use of *directional* wavelets, this we have not investigated. Future research should aim at the design of wavelets with shapes that fit better with the distortion (ground-roll).

Theoretically, the wavelet transform combined with rotation has a computational complexity which is directly proportional to the number of gridpoints.

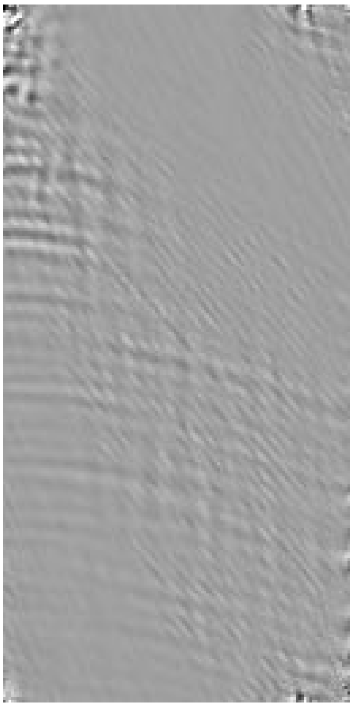


Figure 17: Filtered with **DWT**.

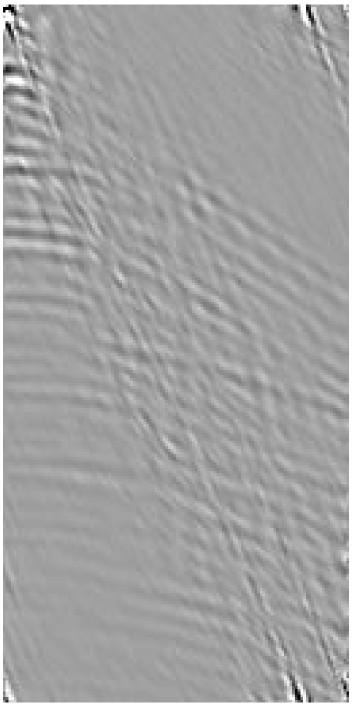


Figure 18: Filtered with **WP**.

References

1. J.C. Cohen and T. Chen, Fundamentals of the Discrete Wavelet Transform for Seismic Data Processing, Preprint, 1993, `www.mathsoft.com/wavelets.html`
2. Andrew J. Deighan, Doyle R. Watts, Ground-roll suppression using the wavelet transform, Geophysics, Vol. 62, No. 6, 1896–1903 (1997). `www.seg.org/publications/geoarchive/1997/nov-dec/deighan.html`
3. Ayon Dey, Larry Lines, Wavelet filtering of seismic data, *M.U.S.I.C. Annual Report 1995*, Memorial University of Newfoundland (1995).
4. Bingwen Du, Larry Lines, Wavelet filtering of tube waves in the Glenn Pool croswell seismic survey, *M.U.S.I.C. Annual report on wavelet application*, Memorial University of Newfoundland (1998).
5. Liu Faqi, M.M. Nurul Kabir and D.J. Verschuur, Seismic processing using the wavelet and the Radon transform, *J. of Seismic Exploration* 4, 375–390 (1995).
6. Th.J.R. Hughes, *The finite element method*, Prentice-Hall Inc., New Jersey (1987).
7. G. Kaiser, R.F. Streater, Windowed Radon transforms, analytic signals, and the wave equation, *Wavelets: A tutorial in theory and applications*, C.K. Chui (ed.), pp. 399–441, Academic Press (1992).
8. S. Mallat, A theory for multiresolution signal decomposition: the wavelet representation, in: *IEEE Pattern Anal. and Machine Intelligence*, vol. 11, no. 7, pp. 674–693 (1989).
9. M. Misiti, Y. Misiti, G. Oppenheim, J.-M. Poggi, *Wavelet Toolbox User's Guide (For Use*

with Matlab) The MathWorks Inc., 1996.

10. F. Natterer, *The mathematics of computerized tomography*, John Wiley & Sons, Chicester (1986).
11. P.J. Ooninck, N.M. Temme, P.M. de Zeeuw, R.A. Zuidwijk, Wavelets: Analysis of Seismic Signals,
    `www.cwi.nl/~pauldz/wvl/wavelets.html`
12. T. Takiguchi, On invertibility of the windowed Radon transform, *J. Math. Sci. Univ. Tokyo* 2: 621–636, 1995.
13. A.L. Warrick, P.A. Delaney, A wavelet localized Radon transform, in: *Wavelet Applications in Signal and Image Processing III*, SPIE proceedings 2569 (1995) 632–643.
14. M.V. Wickerhausen, INRIA lectures on wavelet packet algorithms, in: *Ondelettes et paquets d'ondes*, proceedings Rocquencourt, June 17–21, 1991. 31–99.
15. P.M. de Zeeuw, A benchmark problem representing a strongly simplified and synthetic shotrecord,
    `www.cwi.nl/ftp/pauldz/Demos/SimpleSet`
16. P.M. de Zeeuw, Chapter 14: multigrid and advection, in: C.B. Vreugdenhil and B. Koren, eds., *Numerical Methods for Advection-Diffusion Problems*, Notes on Numerical Fluid Mechanics 45 (Vieweg Verlag, Braunschweig, 1993) 335–351.
17. R.A. Zuidwijk, Directional and time-scale wavelet analysis, *SIAM J. Mathematical Analysis*, Vol. 31(2): 416–430 (2000).
18. R.A. Zuidwijk, The Wavelet X-Ray Transform, CWI Report PNA-R9703, Amsterdam, April 1997.
19. R.A. Zuidwijk, P.M. de Zeeuw, The fast Wavelet X-Ray Transform, CWI Report PNA-R9908, Amsterdam, 1999.
20. R.A. Zuidwijk, P.M. de Zeeuw, Fast algorithm for directional time-scale analysis using wavelets, in: *Wavelet Applications in Signal and Image Processing VI*, SPIE proceedings 3458 (1998) 222–231.