Representations of two-dimensional acoustic velocity fields with Haar wavelets

Saulo P. Oliveira, Emerson L. Gogola, and Jorge V. R. Bonato, Federal University of Paraná, Brazil

Wilson M. Figueiró, Federal University of Bahia, Brazil

Abstract

We present three algorithms for the representation of two-dimensional velocity fields with Haar wavelets and compare their efficiency in data compression. These approaches are: 1) to recast the 2D field as a 1D signal, compute its Haar expansion, and map the 1D expansion back to the 2D domain; 2) to perform the tensor product of 1D wavelet bases in the x- and z-directions and to compute the expansion of the 2D field with respect to this tensorial basis; and 3) to build a multi-resolution analysis from the two-dimensional Haar scaling function and to find the orthogonal complements between adjacent levels. We test these approaches on a synthetic acoustic velocity field.

Introduction

Wavelet theory has long been served as an analysis tool in geophysics and efficient discrete transform algorithms are available (Beylkin et al, 1991; Mallat, 1989), as well as classical Fourier analysis has been benefited from fast Fourier transform algorithms. In this paper we revisit these algorithms with the aim of seeking a more compact representation of seismic velocity fields, benefiting from the multiscale features of wavelet functions.

In this preliminary study, we focus on two-dimensional acoustic velocity fields and consider the classical Haar wavelet. We review the basics of multi-resolution analysis (see, e.g., Blatter, 1998) and proceed to the construction of the one- and two-dimensional algorithms. The algorithm referred herein as the third approach is the classical approach of multi-dimensional fast wavelet transform (Beylkin et al, 1991). We contrast this algorithm with two other intuitive strategies that have not been thoroughly documented in the literature.

Theory

A simple way to (approximately) represent signals at a prescribed resolution is to expand them into piecewise functions:

$$f(t) = \sum_{n=-\infty}^{\infty} f_{j,n}\phi_{j,n}(t), \quad \phi_{j,n}(t) := 2^{-j/2}\phi\left(\frac{t}{2^j} - n\right), \qquad (1)$$

where $\phi(t) = 1$ for $t \in [0,1)$ and $\phi(t) = 0$ otherwise. Similarly to Fourier series, we have

$$f_{j,n} = \langle f, \phi_{j,n} \rangle = \int_{-\infty}^{\infty} f(t)\overline{\phi_{j,n}(t)}\,dt. \qquad (2)$$

The resolution is driven by the scaling parameter j. In general, a signal may present variation in several scales, so it would be convenient to consider *multiple resolutions* in the representation of the signal:

$$f(t) = \sum_{j=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} f_{j,n}\phi_{j,n}(t). \qquad (3)$$

The drawback of this representation is that $f_{j,n}$ are no longer Fourier coefficients, since the basis functions with scaling parameter j depend on the basis functions of the level $j-1$, thus are not orthogonal. For instance,

$$\phi(t) = \phi_{0,0}(t) = \frac{\phi_{-1,0}(t) + \phi_{-1,1}(t)}{\sqrt{2}}. \qquad (4)$$

However, a simple modification of the basis functions,

$$\psi_{j,n}(t) := 2^{-j/2}\psi\left(\frac{t}{2^j} - n\right), \; \psi(t) = \frac{\phi_{-1,0}(t) - \phi_{-1,1}(t)}{\sqrt{2}}, \qquad (5)$$

leads us back to an orthogonal system (Walter, 1994), and the following representation holds:

$$f(t) = \sum_{j=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} f_{j,n}\psi_{j,n}(t), \quad f_{j,n} = \langle f, \phi_{j,n} \rangle. \qquad (6)$$

The functions ϕ and ψ are known as the Haar *scaling function* and the *mother wavelet*, respectively. In wavelet theory, the vector spaces V_j sppaned by $\{\phi_{j,n}(t)\}_{n\in\mathbb{Z}}$ satisfy

$$... \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset ... \subset V_j \subset V_{j-1} \subset ..., \qquad (7)$$

constituting a *multi-resolution analysis*, whereas the vector spaces W_J sppaned by $\{\psi_{j,n}(t)\}_{n\in\mathbb{Z}}$ serve as orthogonal complements between consecutive levels $j-1$ and j, i.e.,

$$V_{j-1} = V_j \oplus W_j, \quad j \in \mathbb{Z}. \qquad (8)$$

In particular, we can derive from (8) the relation $V_0 = W_1 \oplus W_2 \oplus \ldots \oplus W_J \oplus V_J$, in order that can approximate f in V_0 as

$$\tilde{f}(t) = \langle f, \phi_{J,0} \rangle \phi_{J,0}(t) + \sum_{j=1}^{J}\sum_{n=-\infty}^{+\infty} \langle f, \psi_{j,n} \rangle \psi_{j,n}(t). \qquad (9)$$

If $f(t) \neq 0$ only in a bounded domain, the number of nonzero coefficients $\langle f, \phi_{j,n} \rangle$ is finite (Blatter, 1998). In particular, if $f(t) = 0$ if $t \notin [0, 2^J)$, then expansion (9) reduces to

$$\tilde{f}(t) = \bar{f} + \sum_{j=1}^{J}\sum_{n=0}^{2^{J-j}-1} d_{j,n}\psi_{j,n}(t), \quad d_{j,n} = \langle f, \psi_{j,n} \rangle, \qquad (10)$$

and $\bar{f}$ is the mean of f. We employ a standard recursive algorithm to compute the Fourier coeffients $d_{j,n}$, known as *Mallat's algorithm*. Let us define the auxiliary coefficients $s_{j,n} = \langle f, \phi_{j,n} \rangle$. Extending relations (4)-(5) to $\phi_{j,n}(t)$ and $\psi_{j,n}(t)$ and replacing them into $s_{j,n}$ and $_{j,n}$, we find

$$s_{j,n} = \frac{s_{j-1,2n} + s_{j-1,2n+1}}{\sqrt{2}}, \quad d_{j,n} = \frac{s_{j-1,2n} - s_{j-1,2n+1}}{\sqrt{2}}. \quad (11)$$

These equations allow us to retrieve $d_{j,n}$ from the initial data $s_{0,n} = \langle f, \phi_{0,n} \rangle$. In particular, we know discrete samples of the signal f, i.e., $f = [f_0, \ldots f_{2^J-1}]$ we begin with $s_{0,n} = f_n$ (Blatter, 1998).

Representation of 2D data

We now proceed to the representation of a two-dimensional velocity field in the form

$$v_{i,j}, \quad 0 \le i, j \le N, \quad N = 2^J - 1. \quad (12)$$

First approach: 1D Haar wavelets

The first strategy considered herein is to recast the velocity field (12) as the signal

$$f = [v_{0,0}, v_{0,1}, \ldots, v_{0,N}, v_{1,0}, \ldots, v_{1,N}, \ldots, v_{N,N}], \quad (13)$$

compute the Haar expansion (10) with the desired number of terms, and compute the approximate field

$$\tilde{v}_{i,j} = \tilde{f}(N\,i + j), \quad 0 \le i, j \le N. \quad (14)$$

In summary, we have the following algorithm:

$\tilde{J} \leftarrow 2J$;
$s_{0,Ni+j} \leftarrow v_{i,j}$ $(0 \le i, j \le N)$;
for $j = 1, \ldots, \tilde{J}$ **do**
  for $n = 0, \ldots, 2^{\tilde{J}-j} - 1$ **do**
    $s_{j,n} \leftarrow (s_{j-1,2n} + s_{j-1,2n+1})\sqrt{2}$;
    $d_{j,n} \leftarrow (s_{j-1,2n} - s_{j-1,2n+1})\sqrt{2}$;
  end for
end for
Compute $\tilde{f}$ in (10);
$\tilde{v}_{i,j} \leftarrow \tilde{f}(N\,i + j)$, $(0 \le i, j \le N)$;

Second approach: a tensor product basis

One natural step to improve of the above strategy is to use the Haar approximation in each spatial coordinate. In the following we describe how to implement this.

We have used the following basis functions in the representation (10) of a signal $f(t)$:

$$S_t = \{\{\phi_{J,0}(t)\}, \{\{\psi_{j,n}(t)\}_{n=0}^{2^{J-j}-1}\}_{j=1}^{J}\}. \quad (15)$$

Let us build the set $S_{xz} = \{u(x)v(z),\, u, v \in S_t\}$ from the tensor product of functions in S_t in the $x-$ and $z-$directions, i.e.,

$$\begin{aligned} S_{xz} = \; & \{\{\phi_{J,0}(x)\phi_{J,0}(z)\}\,,\, \{\{\phi_{J,0}(x)\psi_{j,n}(z)\}_{n=0}^{2^{J-j}-1}\}_{j=1}^{J}, \\ & \{\{\psi_{j,n}(x)\phi_{J,0}(z)\}_{n=0}^{2^{J-j}-1}\}_{j=1}^{J}, \qquad (16) \\ & \{\{\{\{\psi_{j_x,n_x}(x)\psi_{j_z,n_z}(z)\}_{n_x=0}^{2^{J-j}-1}\}_{j_x=1}^{J}\}_{n_z=0}^{2^{J-j}-1}\}_{j_z=1}^{J}\}. \end{aligned}$$

Considering the 2D inner product

$$\langle f, g \rangle = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,z)\overline{g(x,z)}\,dx\,dz, \quad (17)$$

we have that the functions in S_{xz} inherit the orthogonality from S_t in the sense that, it $f = u_1 v_1$ and $g = u_2 v_2$ with $u_1, u_2, v_1, v_2 \in S_t$,

$$\langle f, g \rangle = \begin{cases} 1, & (u_1, v_1) = (u_2, v_2), \\ 0, & (u_1, v_1) \ne (u_2, v_2). \end{cases} \quad (18)$$

Thus, the coefficients in the 2D expansion

$$\begin{aligned} \tilde{f}(x,z) &= \tilde{d}^0 \phi_{J,0}(x)\phi_{J,0}(z) + \sum_{j=1}^{J}\sum_{n=0}^{2^{J-j}-1} \tilde{d}^z_{j,n}\phi_{J,0}(x)\psi_{j,n}(z) \\ &+ \sum_{j=1}^{J}\sum_{n=0}^{2^{J-j}-1} \tilde{d}^x_{j,n}\psi_{j,n}(x)\phi_{J,0}(z) \qquad (19) \\ &+ \sum_{j_x=1}^{J}\sum_{n_x=0}^{2^{J-j}-1}\sum_{j_z=1}^{J}\sum_{n_z=0}^{2^{J-j}-1} d^{xz}_{n_x,j_x,n_z,j_z}\psi_{j_x,n_x}(x)\psi_{j_z,n_z}(z) \end{aligned}$$

satisfy

$$\tilde{d}^0 = \langle f, \phi_{J,0}(x)\phi_{J,0}(z) \rangle, \quad \tilde{d}^x_{j,n} = \langle f, \psi_{j,n}(x)\phi_{J,0}(z) \rangle, \quad (20)$$

$$\tilde{d}^z_{j,n} = \langle f, \phi_{J,0}(x)\psi_{j,n}(z) \rangle, \quad d^{xz}_{j_x,n_x,j_z,n_z} = \langle f, \psi_{j_x,n_x}(x)\psi_{j_z,n_z}(z) \rangle.$$

Since $\phi_{J,0}(t) = 2^{-J/2}$ for $t \in [0, 2^J)$, we find

$$\begin{aligned} \tilde{f}(x,z) &= \bar{f} + \sum_{j=1}^{J}\sum_{n=0}^{2^{J-j}-1} d^z_{j,n}\psi_{j,n}(z) + d^x_{j,n}\psi_{j,n}(x) \qquad (21) \\ &+ \sum_{j_x=1}^{J}\sum_{n_x=0}^{2^{J-j}-1}\sum_{j_z=1}^{J}\sum_{n_z=0}^{2^{J-j}-1} d^{xz}_{j_x,n_x,j_z,n_z}\psi_{j_x,n_x}(x)\psi_{j_z,n_z}(z) \end{aligned}$$

for $(x,z) \in [0,2^J) \times [0,2^J)$, where $\bar{f}$ is the mean of f over $[0,2^J) \times [0,2^J)$ and

$$d^x_{j,n} = 2^{-J}\langle f, \psi_{j,n}(x) \rangle, \quad d^z_{j,n} = 2^{-J}\langle f, \psi_{j,n}(z) \rangle. \quad (22)$$

Let us define the auxiliary coefficients

$$s^x_{j,n} = 2^{-J}\langle f, \phi_{j,n}(x) \rangle, \quad s^z_{j,n} = 2^{-J}\langle f, \phi_{j,n}(z) \rangle. \quad (23)$$

Analogously to (11), we find

$$s^x_{j,n} = \frac{s^x_{j-1,2n} + s^x_{j-1,2n+1}}{\sqrt{2}}, \quad d^x_{j,n} = \frac{s^x_{j-1,2n} - s^x_{j-1,2n+1}}{\sqrt{2}}, \quad (24)$$

$$s^z_{j,n} = \frac{s^z_{j-1,2n} + s^z_{j-1,2n+1}}{\sqrt{2}}, \quad d^z_{j,n} = \frac{s^z_{j-1,2n} - s^z_{j-1,2n+1}}{\sqrt{2}}. \quad (25)$$

On the other hand, the computation of d^{xz} requires two families of auxiliary coefficients:

$$s_{j_x,n_x,j_z,n_z} = \langle f, \phi_{j_x,n_x}(x)\phi_{j_z,n_z}(z) \rangle, \quad (26)$$

$$\tilde{s}_{j_x,n_x,j_z,n_z} = \langle f, \phi_{j_x,n_x}(x)\psi_{j_z,n_z}(z) \rangle. \quad (27)$$

The recursion (11) in the $x-$directions yields

$$\tilde{s}_{j_x,n_x,j_z,n_z} = \frac{\tilde{s}_{j_x-1,2n_x,j_z,n_z} + \tilde{s}_{j_x-1,2n_x+1,j_z,n_z}}{\sqrt{2}}, \tag{28}$$

$$d^{xz}_{j_x,n_x,j_z,n_z} = \frac{\tilde{s}_{j_x-1,2n_x,j_z,n_z} - \tilde{s}_{j_x-1,2n_x+1,j_z,n_z}}{\sqrt{2}}. \tag{29}$$

By repeating the recursive steps (28)-(29), d_{j_x,n_x,j_z,n_z} is eventually written with respect to $\tilde{s}_{0,n_x,j_z,n_z}$. These coefficients in turn are determined by a recursion in the $z-$direction:

$$\tilde{s}_{0,n_x,j_z,n_z} = \frac{s_{0,n_x,j_z-1,2n_z} - s_{0,n_x,j_z-1,2n_z+1}}{\sqrt{2}}, \tag{30}$$

where s_{0,n_x,j_z,n_z} are computed in a similar fashion:

$$s_{0,n_x,j_z,n_z} = \frac{s_{0,n_x,j_z-1,2n_z} + s_{0,n_x,j_z-1,2n_z+1}}{\sqrt{2}}. \tag{31}$$

It follows from (24)-(31) that all coefficients in (21) can be determined from s^x_{0,n_x}, s^z_{0,n_z}, and $s_{0,n_x,0,n_z}$ for $n_x, n_z = 0,\ldots N$. However, we can also verify that

$$s^x_{0,n_x} = \sum_{k=0}^{N} s_{0,n_x,0,k}, \quad s^z_{0,n_z} = \sum_{k=0}^{N} s_{0,k,0,n_z}, \tag{32}$$

in order that just the coefficients $s_{0,n_x,0,n_z}$ are sufficient to build the expansion (21). As in the 1D case, we set $s_{0,n_x,0,n_z} = v_{n_x,n_z}$ $(n_x, n_z = 0,\ldots N)$. The algorithm for this approach is then given as follows:

$s_{0,n_x,0,n_z} \leftarrow v_{n_x,n_z}$ $(0 \le n_x, n_z \le N)$;
$s^x_{0,n_x} \leftarrow \sum_{k=0}^{N} s_{0,n_x,0,k}$;
$s^z_{0,n_z} \leftarrow \sum_{k=0}^{N} s_{0,k,0,n_z}$;
for $j = 1,\ldots,J$ **do**
  for $n = 0,\ldots,2^{J-j}-1$ **do**
    $s^x_{j,n} \leftarrow (s^x_{j-1,2n} + s^x_{j-1,2n+1})/\sqrt{2}$;
    $d^x_{j,n} \leftarrow (s^x_{j-1,2n} - s^x_{j-1,2n+1})/\sqrt{2}$;
    $s^z_{j,n} \leftarrow (s^z_{j-1,2n} + s^z_{j-1,2n+1})/\sqrt{2}$;
    $d^z_{j,n} \leftarrow (s^z_{j-1,2n} - s^z_{j-1,2n+1})/\sqrt{2}$;
  end for
end for
for $n_x = 0,\ldots,2^J-1$ **do**
  for $j_z = 1,\ldots,J$ **do**
    for $n_z = 0,\ldots,2^{J-j_z}-1$ **do**
      $s_{0,n_x,j_z,n_z} \leftarrow (s_{0,n_x,j_z-1,2n_z} + s_{0,n_x,j_z-1,2n_z+1})/\sqrt{2}$;
      $\tilde{s}_{0,n_x,j_z,n_z} \leftarrow (s_{0,n_x,j_z-1,2n_z} - s_{0,n_x,j_z-1,2n_z+1})/\sqrt{2}$;
    end for
  end for
end for
for $j_x = 1,\ldots,J$ **do**
  for $n_x = 0,\ldots,2^{J-j_x}-1$ **do**
    for $j_z = 1,\ldots,J$ **do**
      for $n_z = 0,\ldots,2^{J-j_z}-1$ **do**
        $\tilde{s}_{j_x,n_x,j_z,n_z} \leftarrow (\tilde{s}_{j_x-1,2n_x,j_z,n_z} + \tilde{s}_{j_x-1,2n_x+1,j_z,n_z})/\sqrt{2}$;
        $d^{xz}_{j_x,n_x,j_z,n_z} \leftarrow (\tilde{s}_{j_x-1,2n_x,j_z,n_z} - \tilde{s}_{j_x-1,2n_x+1,j_z,n_z})/\sqrt{2}$;
      end for
    end for
  end for
end for
Compute $\tilde{f}$ in (21);

Third approach: the non-standard form

Beylkin (1993) refers to tensor basis (16) as the *standard form* and denominates *non-standard form* the basis constructed from an orthogonal decomposition of the 2D multi-resolution analysis $\ldots V_1 \subset V_0 \subset V_{-1} \subset \ldots$ formed by the tensor-product spaces

$$V_j = V^x_j \otimes V^z_j, \quad \begin{cases} V^x_j = \text{span}\{\phi_{j,n}(x)\}_{n\in\mathbb{Z}}, \\ V^z_j = \text{span}\{\phi_{j,n}(z)\}_{n\in\mathbb{Z}}, \end{cases}$$

Following Mallat (1989), let us decompose V_{j-1} as a direct sum of V_j and its orthogonal complement W_j employing the orthogonal decompositions of the spaces V^x_{j-1} and V^y_{j-1}:

$$\begin{aligned} V_{j-1} &= V^x_{j-1} \otimes V^z_{j-1} = \left(V^x_j \oplus W^x_j\right) \otimes \left(V^z_j \oplus W^z_j\right) \\ &= (V^x_j \otimes V^z_j) \oplus (V^x_j \otimes W^z_j) \oplus (W^x_j \otimes V^z_j) \oplus (W^x_j \otimes W^z_j), \end{aligned}$$

i.e., $V_{j-1} = V_j \oplus W_j$, where W_j is defined as

$$W_j = (V^x_j \otimes W^z_j) \oplus (W^x_j \otimes V^z_j) \oplus (W^x_j \otimes W^z_j). \tag{33}$$

In particular for $j = 0$, we have that

$$V^x_0 = \text{span}\{\phi(x-n)\}_{n\in\mathbb{Z}},\ W^x_0 = \text{span}\{\psi(x-n)\}_{n\in\mathbb{Z}}, \tag{34}$$

$$V^z_0 = \text{span}\{\phi(z-n)\}_{n\in\mathbb{Z}},\ W^z_0 = \text{span}\{\psi(z-n)\}_{n\in\mathbb{Z}}, \tag{35}$$

thus

$$\begin{cases} V^x_0 \otimes W^z_0 &= \text{span}\{\phi(x-n_x)\psi(z-n_z)\}_{n_x,n_z\in\mathbb{Z}}, \\ W^x_0 \otimes V^z_0 &= \text{span}\{\psi(x-n_x)\phi(z-n_z)\}_{n_x,n_z\in\mathbb{Z}}, \\ W^x_0 \otimes W^z_0 &= \text{span}\{\psi(x-n_x)\psi(z-n_z)\}_{n_x,n_z\in\mathbb{Z}}. \end{cases} \tag{36}$$

In general, we have that

$$W_j = \text{span}\{\psi^a_{j,n_x,n_z}, \psi^b_{j,n_x,n_z}, \psi^c_{j,n_x,n_z}\}_{n_x,n_z\in\mathbb{Z}}, \tag{37}$$

$$\begin{cases} \psi^a_{j,n_x,n_z}(x,z) = \phi_{j,n_x}(x)\psi_{j,n_z}(z), \\ \psi^b_{j,n_x,n_z}(x,z) = \psi_{j,n_x}(x)\phi_{j,n_z}(z), \\ \psi^c_{j,n_x,n_z}(x,z) = \psi_{j,n_x}(x)\psi_{j,n_z}(z), \end{cases} \tag{38}$$

and the representation corresponding to (19) is given as

$$\begin{aligned} \tilde{f}(x,z) &= \bar{f} + \sum_{j=1}^{J} \sum_{n_x=0}^{2^{J-j}-1} \sum_{n_z=0}^{2^{J-j}-1} \Big(a_{j,n_x,n_z} \psi^a_{j,n_x}(x,z) \\ &+ b_{j,n_x,n_z} \psi^b_{j,n_x}(x,z) + c_{j,n_x,n_z} \psi^c_{j,n_x}(x,z) \Big), \end{aligned} \tag{39}$$

for $(x,z) \in [0,2^J) \times [0,2^J)$, where $a_{j,n_x,n_z} = \langle f, \psi^a_{j,n_x,n_z} \rangle$, $b_{j,n_x,n_z} = \langle f, \psi^b_{j,n_x,n_z} \rangle$, and $c_{j,n_x,n_z} = \langle f, \psi^c_{j,n_x,n_z} \rangle$. In order to recursively compute these coefficients, let us define

$$s_{j,n_x,n_z} = \left\langle f, \phi_{j,n_x}(x)\phi_{j,n_z}(z) \right\rangle, \tag{40}$$

which satisfies, as in (11),

$$\begin{aligned} s_{j,n_x,n_z} &= \frac{1}{2}(s_{j-1,2n_x,2n_z} + s_{j-1,2n_x,2n_z+1} \\ &+ s_{j-1,2n_x+1,2n_z} + s_{j-1,2n_x+1,2n_z+1}), \end{aligned} \tag{41}$$

and analogously,

$$\begin{aligned} a_{j,n_x,n_z} &= \frac{1}{2}(s_{j-1,2n_x,2n_z} - s_{j-1,2n_x,2n_z+1} \\ &+ s_{j-1,2n_x+1,2n_z} - s_{j-1,2n_x+1,2n_z+1}), \quad (42) \\ b_{j,n_x,n_z} &= \frac{1}{2}(s_{j-1,2n_x,2n_z} + s_{j-1,2n_x,2n_z+1} \\ &- s_{j-1,2n_x+1,2n_z} - s_{j-1,2n_x+1,2n_z+1}), \quad (43) \\ c_{j,n_x,n_z} &= \frac{1}{2}(s_{j-1,2n_x,2n_z} - s_{j-1,2n_x,2n_z+1} \\ &- s_{j-1,2n_x+1,2n_z} + s_{j-1,2n_x+1,2n_z+1}). \quad (44) \end{aligned}$$

Thus, for each $1 \le j \le J$, the coefficients a_{j,n_x,n_z}, b_{j,n_x,n_z} and c_{j,n_x,n_z} depend on $s_{0,n_x,n_z} = v_{n_x,n_z}$ for $n_x, n_z = 0, \dots N$. This leads to the following algorithm:

$s_{0,n_x,n_z} \leftarrow v_{n_x,n_z}$ $(0 \le n_x, n_z \le N)$;
for $j = 1, \dots, J$ **do**
  for $n_x = 0, \dots, 2^{J-j} - 1$ **do**
    for $n_z = 0, \dots, 2^{J-j} - 1$ **do**
      $s_{j,n_x,n_z} \leftarrow (s_{j-1,2n_x,2n_z} + s_{j-1,2n_x,2n_z+1} + s_{j-1,2n_x+1,2n_z} + s_{j-1,2n_x+1,2n_z+1})/2$;
      $a_{j,n_x,n_z} \leftarrow (s_{j-1,2n_x,2n_z} - s_{j-1,2n_x,2n_z+1} + s_{j-1,2n_x+1,2n_z} - s_{j-1,2n_x+1,2n_z+1})/2$;
      $b_{j,n_x,n_z} \leftarrow (s_{j-1,2n_x,2n_z} + s_{j-1,2n_x,2n_z+1} - s_{j-1,2n_x+1,2n_z} - s_{j-1,2n_x+1,2n_z+1})/2$;
      $c_{j,n_x,n_z} \leftarrow (s_{j-1,2n_x,2n_z} - s_{j-1,2n_x,2n_z+1} - s_{j-1,2n_x+1,2n_z} + s_{j-1,2n_x+1,2n_z+1})/2$;
    end for
  end for
end for
Compute $\tilde{f}$ in (39);

Examples

In order to illustrate the three algorithms outlined in the previous section, let us consider the synthetic velocity field composed of 32×32 blocks shown in Figure 1.

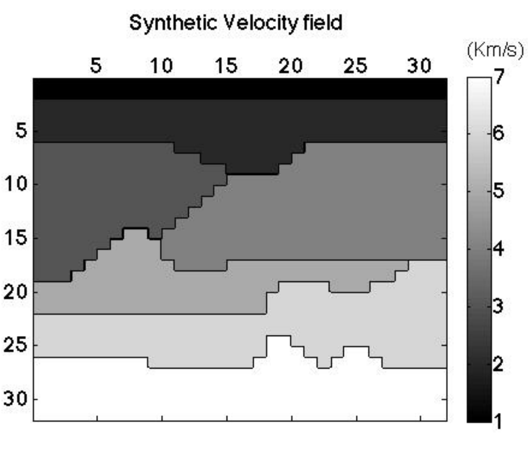

Figure 1: The synthetic velocity field.

The representations (10), (21) and (39) exactly reproduce the synthetic velocity field. These expansions have respectively 429, 324, and 211 nonzero coefficients, in contrast with the 2^{10} coefficients $\{v_{i,j}\}_{i,j=0}^{31}$ of the original data, incurring in memory savings as high as 80%.

For further comparison, we show in Figures 2 and 3 the truncated representations of the velocity field by the first and second approaches when the number of coefficients is close to 211, the number of nonzero coefficients of the third approach. For the first approach we dropped all coefficients whose magnitude was below 1.0, whereas in the second approach we dropped all coefficients whose magnitude was below 0.3.

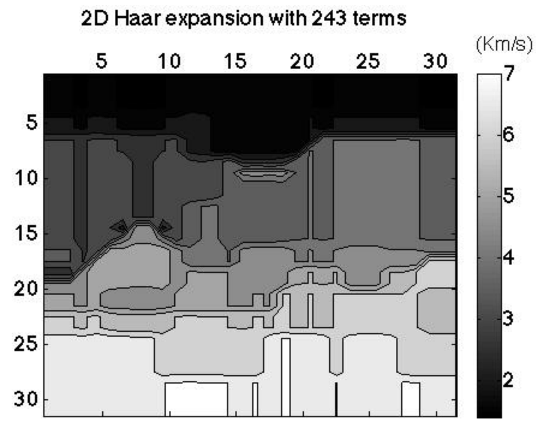

Figure 2: Truncated representation of the velocity field by the first approach (Haar coefficients of magnitude below 1.0 were dropped).

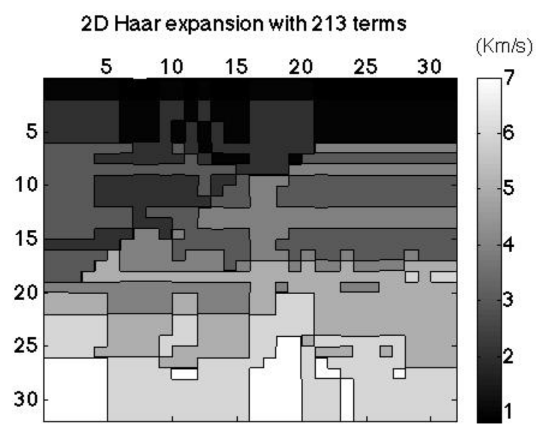

Figure 3: Truncated representation of the velocity field by the second approach (Haar coefficients of magnitude below 0.3 were dropped).

Conclusions

We have outlined three algorithms to efficiently represent two-dimensional velocity fields based on the Haar wavelet. Revisiting the classical wavelet theory, we illustrated how the use of linear algebra concepts such as direct sums leads to a multiscale Fourier-type representation of data to the design of an algorithm with 2D Haar wavelets that is simpler than the tensor-product algorithm.

The savings in the numerical example confirm the benefits of a multiscale representation over a finest-scale representation of data. Note that the third approach provided the best compression rate in this example.

References

Beylkin, G.; Coifman, R. and Rokhlin, V.; 1991. Fast wavelet transforms and numerical algorithms. I. Commun. Pure Appl. Math., v. 44, n. 2, p. 141-183.

Beylkin, G.; 1993. Wavelets and numerical algorithms (lecture notes). ArXiv:comp- gas/9304004v1.

Blatter, C.; 1998. Wavelets. A primer. Natick: A K Peters.

Mallat, S.; 1989. A theory for multiresolution signal decomposition: The wavelet representation. IEEE Trans. Pattern Anal. Mach. Intell., v. 11, n. 7, p. 674-693.

Walter, G.; 1994. Wavelets and other orthogonal systems with applications. Boca Raton: CRC Press.

Acknowledgments

This work was funded by the CENPES-PETROBRAS Applied Geophysics Network.

Appendix: further considerations on two-dimensional bases

Let us further compare the 2D bases employed in the latter two approaches presented in this paper. Note that the representation (39) has a total of

$$1+\sum_{j=1}^{J}\sum_{n_x=0}^{2^{J-j}-1}\sum_{n_z=0}^{2^{J-j}-1}(3)=1+3\sum_{j=1}^{J}2^{2(J-j)}=4^J$$

coefficients (taking also into account the zeroth order term $\bar{f}$), which is consistent with the

$$1+2\sum_{j=1}^{J}2^{J-j}+\left(\sum_{j=1}^{J}2^{J-j}\right)^2=4^J$$

terms in (19), and the $(2^J)^2$ terms in the vector space

$$V_0^J=\text{span}\{\{\phi_{0,n_x}(x)\phi_{0,n_z}(x)\}_{n_z=0}^{2^J-1}\}_{n_x=0}^{2^J-1}. \qquad (45)$$

The sets of functions (45), and the functions associated with the expansions (19) and (39), all serve as basis functions for the space V_0^J. Figures 4-6 illustrate these bases for $J=2$.

1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1

Figure 4: Basis functions (45) of the space V_0^2.

1	1	1	1	1	1	-1	-1	1	-1	0	0	0	0	1	-1
1	1	1	1	1	1	-1	-1	1	-1	0	0	0	0	1	-1
1	1	1	1	1	1	-1	-1	1	-1	0	0	0	0	1	-1
1	1	1	1	1	1	-1	-1	1	-1	0	0	0	0	1	-1

1	1	1	1	1	1	-1	-1	1	-1	0	0	0	0	1	-1
1	1	1	1	1	1	-1	-1	1	-1	0	0	0	0	1	-1
-1	-1	-1	-1	-1	-1	1	1	-1	1	0	0	0	0	-1	1
-1	-1	-1	-1	-1	-1	1	1	-1	1	0	0	0	0	-1	1

1	1	1	1	1	1	-1	-1	1	-1	0	0	0	0	1	-1
-1	-1	-1	-1	-1	-1	1	1	-1	1	0	0	0	0	-1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	-1	-1	1	-1	0	0	0	0	1	-1
-1	-1	-1	-1	-1	-1	1	1	-1	1	0	0	0	0	-1	1

Figure 5: Basis functions (19) of the space V_0^2.

1	1	1	1	1	1	-1	-1	1	1	1	1	1	1	-1	-1
1	1	1	1	1	1	-1	-1	1	1	1	1	1	1	-1	-1
1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	1	1
1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	1	1

1	-1	0	0	1	-1	0	0	1	1	0	0	0	0	1	-1
-1	1	0	0	1	-1	0	0	-1	-1	0	0	0	0	1	-1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	0	1	1	0	0	1	-1	0	0	0	0	0	0	0	0
0	0	-1	-1	0	0	-1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	-1	0	0	1	1	0	0
0	0	0	0	0	0	0	0	1	-1	0	0	-1	-1	0	0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	-1	0	0	0	0	1	-1	0	0	1	1	0	0	1	-1
-1	1	0	0	0	0	1	-1	0	0	-1	-1	0	0	-1	1

Figure 6: Basis functions (39) of the space V_0^2.