WAVELET TRANSFORMS THAT MAP INTEGERS TO INTEGERS

A. R. CALDERBANK, INGRID DAUBECHIES, WIM SWELDENS, AND BOON-LOCK YEO

August 1996

ABSTRACT. Invertible wavelet transforms that map integers to integers have important applications in lossless coding. In this paper we present two approaches to build integer to integer wavelet transforms. The first approach is to adapt the precoder of Laroia et al., which is used in information transmission; we combine it with expansion factors for the high and low pass band in subband filtering. The second approach builds upon the idea of factoring wavelet transforms into so-called lifting steps. This allows the construction of an integer version of every wavelet transform. Finally, we use these approaches in a lossless image coder and compare the results to the literature.

1. INTRODUCTION

Wavelets, wavelet packets, and local cosine transforms are used in a variety of applications, including image compression [2, 29, 39, 10, 25]. In most cases, the filters that are used have floating point coefficients. For instance, if one prefers to use orthonormal filters with an assigned number N ($N \geqslant 2$) of vanishing moments and minimal filter length, then the resulting filter coefficients are real numbers which can be computed with high precision, but for which we do not even have a closed form expression if $N > 3$ [8]. When the input data consist of sequences of integers (as is the case for images), the resulting filtered outputs no longer consist of integers. Yet, for lossless coding it would be of interest to be able to characterize the output completely again with integers.

In the case of the Fourier and cosine transforms, this problem has been solved in [17]. An integer version of the orthogonal Haar transform has been known for some time as the S-transform [24, 16]. In [10] a redundant integer version of the Haar transform is introduced which uses rounded integer arithmetic and which yields near optimal approximation in certain Besov spaces. Relaxing the constraint of orthonormality of the wavelets makes it possible to obtain filter coefficients that are dyadic rationals [6, 14, 37, 31]; up to scaling these filters can be viewed as mapping integers to integers. This rescaling amplifies the dynamic range of the

---

1991 *Mathematics Subject Classification.* 42C15, 94A29.

Key words and phrases. wavelet, lossless compression, integer transforms, precoding, lifting, source coding.

data considerably, however, so that this does not lead to reasonable lossless compression. In the special case of binary valued images it is possible to define binary valued wavelet transforms (and thus no bit growth) [18, 31].

In this paper, we present two approaches that lead to wavelet transforms that map integers to integers, and which can be used for lossless coding. The two approaches are independent of each other, and the reader can explore the corresponding two parts of this paper (Section 2 and Section 3) in either order. The second approach looks more promising for lossless image coding. Note that in both cases we are not building *integer arithmetic* wavelet transforms. Thus the computations are still done with floating point numbers, but the result is guaranteed to be integer and invertibility is preserved. In software applications, this should not affect speed, as in many of today's microprocessors floating point and integer computations are virtually equally fast.

The first approach, in Section 2, is inspired by the precoding invented by Laroia, Tretter, and Farvardin (LTF) [21], which is used for transmission over channels subject to intersymbol interference and Gaussian noise. When integer data undergo causal filtering with a leading filter coefficient equal to 1, this precoding makes small (non-integer) adjustments to the input data resulting in integer outputs after filtering. In Section 2.1, we try to adapt this method to the subband filtering inherent to a wavelet transform. Even though this approach cannot be made to work for other wavelet bases than the Haar basis, it leads to a reformulation of the original problem, by including the possibility of an expansion coefficient. Section 2.2 shows how the specific structure of wavelet filters can be exploited to make this expansion coefficient work. In Section 2.3, we generalize the expansion idea to propose different expansion factors for the low and the high pass filter: for the high pass filter, we can afford a larger expansion factor because the dynamic range of its output tends to be smaller anyway. In most cases, the product of the expansion factors for high and low pass is larger than 1, which turns out to be bad for lossless compression.

A different approach (although there are points of contact) is taken in Section 3. An example in [39] of a wavelet transform that maps integers to integers is seen to be an instance of a large family of similar transforms, obtained by combining the lifting constructions proposed in [32, 33] with rounding-off in a reversible way. In Section 3.1 we review the S transform, and in Section 3.2 two generalizations, the TS transform and the S+P transform. In Section 3.3 we briefly review lifting, and in Section 3.4 we show how a decomposition in lifting steps can be turned into an integer transform. In general, this transform exhibits some expansion factors as well. Unlike to what happened in Section 2 the product of the low and high pass expansion factors is now always equal to 1. Section 3.5 gives a short discussion of how this approach can be linked to the one in Section 2. In Section 4, we show many examples.

Finally, in Section 5 we show applications of the two approaches to lossless image coding, and we compare our results with other results in the literature.

Except where specified otherwise, our notations conform with the standard notations for filters associated with orthonormal or biorthogonal wavelet bases, as found in e.g., [8].

2. Expansion factors and integer to integer transforms

2.1. **Precoding for subband filtering.** We start by reviewing briefly the LTF precoder [21]. Suppose we have to filter a sequence of integers $(a_n)_{n\in\mathbf{Z}}$, with a causal filter (i.e., $h_k = 0$ for $k < 0$) and leading coefficient 1 (i.e., $h_0 = 1$). If the h_k are not themselves integers, then the filtered output $b_n = a_n + \sum_{k=1}^{\infty} h_k a_{n-k}$ will in general not consist of integers either. The LTF precoder replaces the integers a_n by non-integers a'_n by introducing small shifts r_n, in such a way that the resulting b'_n are integers. More concretely, define

$$
\begin{aligned}
a'_n &= a_n - r_n \\
r_n &= \left\{ \sum_{k=1}^{\infty} h_k a'_{n-k} \right\} = \left\{ \sum_{k=1}^{\infty} h_k (a_{n-k} - r_{n-k}) \right\} ,
\end{aligned}
$$

where the symbol $\{x\} = x - \lfloor x \rfloor$ stands for the fractional part of x, and $\lfloor x \rfloor$ for the largest integer not exceeding x. In practice, both the a_n and the h_k are non-zero for only finite ranges of the indices n or k, so that the infinite recursion is not a problem: if $a_n = 0$ for $n < 0$, then so are a'_n and r_n and the recursion can be started. If we now compute $b'_n = a'_n + \sum_{k=1}^{\infty} h_k a'_{n-k}$, then we immediately see that

$$
b'_n = a_n + \left\lfloor \sum_{k=1}^{\infty} h_k a'_{n-k} \right\rfloor
$$

is an integer. To recover the original a_n from the b'_n, one first applies the inverse filter $1/H(z) = (1 + \sum_{k=1}^{\infty} h_k\, z^k)^{-1}$ (assuming this filter is stable), which gives the a'_n from which

$$
a_n = a'_n + \left\{ \sum_{k=1}^{\infty} h_k\, a'_{n-k} \right\}
$$

can be computed immediately.

This approach can also be used if $h_0 \neq 1$. In that case, if we have

$$
\begin{aligned}
a'_n &= a_n - r_n \\
r_n &= \left\{ (h_0 - 1)a'_n + \sum_{k=1}^{\infty} h_k a'_{n-k} \right\} \\
&= \left\{ (h_0 - 1)(a_n - r_n) + \sum_{k=1}^{\infty} h_k a'_{n-k} \right\} ,
\end{aligned}
\tag{2.1}
$$

then the

$$
\begin{aligned}
b'_n = \sum_{k=0}^{\infty} h_k a'_{n-k} \; &= \; a'_n + \left\lfloor (h_0 - 1)a'_n + \sum_{k=1}^{\infty} h_k a'_{n-k} \right\rfloor + r_n \\
&= a_n + \left\lfloor (h_0 - 1)a'_n + \sum_{k=1}^{\infty} h_k a'_{n-k} \right\rfloor
\end{aligned}
$$

are integers again. The situation is now slightly more complicated, because the r_n are defined implicitly. Since both a_n and the a'_{n-k} for $k \geqslant 1$ are known when r_n needs to be determined, we can reformulate (2.1) as

$$r_n = \{(1 - h_0)r_n + y_n\}$$

with $y_n = \left\{(h_0 - 1)a_n + \sum_{k=1}^{\infty} h_k a'_{n-k}\right\}$. Whether or not the equation

$$r = \{\alpha r + y\} \tag{2.2}$$

has a solution for arbitrary $y \in [0, 1)$ depends on the size and sign of α. As shown in Figure 1, a solution fails to exist for some choices of y if $0 < \alpha < 2$; for $\alpha \leqslant 0$ or $\alpha \geqslant 2$ there always exists at least one solution. This range for α where (2.2) always has a solution corresponds to $|h_0| \geqslant 1$. When faced with having to use the precoder for cases where $|h_0| < 1$, a simple expedient is to consider a renormalized filter, with $\widetilde{h}_n = \alpha h_n$, so that $|\widetilde{h}_0| \geqslant 1$. (For instance, $\alpha = h_0^{-1}$ will do.)

In the case of the wavelet transform, we work with several filters, followed by decimation. In this paper, we shall always work with 2-channel perfect reconstruction filter banks; in this section, unless specified explicitly, we assume the filter pair is orthonormal.

Assume that the two filters H and G are FIR, with $h_n = g_n = 0$ if $n < 0$ or $n \geqslant 2N$. Then we have

$$
\begin{aligned}
s_{1,n} &= \sum_k h_{2n-k} s_{0,k} \\
d_{1,n} &= \sum_k g_{2n-k} s_{0,k} ,
\end{aligned}
\tag{2.3}
$$

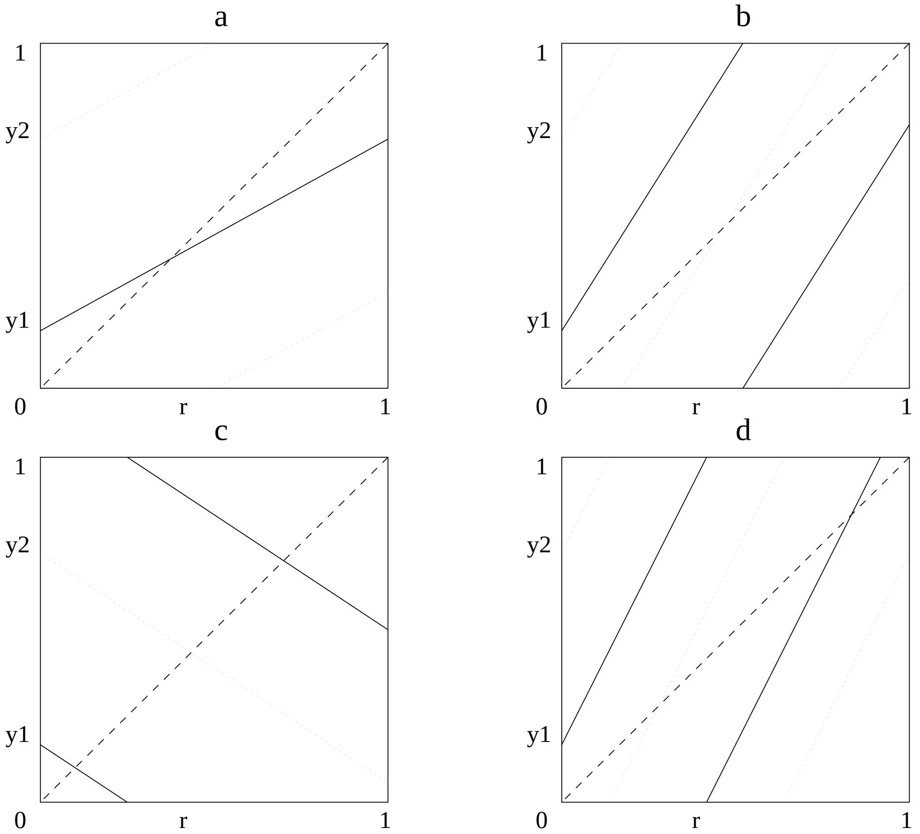

FIGURE 1. *Solving $r = \{\alpha r + y\}$:*

a: *For $0 < \alpha \leqslant 1$, the equation $r = \{\alpha r + y\}$ has solutions for $y = y_1$ (full) but not for $y = y_2$ (dotted), since $\{\alpha r + y_1\}$ intersects the diagonal (dashed), but $\{\alpha r + y_2\}$ does not.*

b: *For $1 \leqslant \alpha < 2$, the situation is reversed.*

c: *For $\alpha \leqslant 0$ we always have at least one solution.*

d: *Likewise for $\alpha \geqslant 2$.*

which can be rewritten as

$$\begin{pmatrix} s_{1,n} \\ d_{1,n} \end{pmatrix} = \sum_k \begin{pmatrix} h_{2(n-k)} & h_{2(n-k)+1} \\ g_{2(n-k)} & g_{2(n-k)+1} \end{pmatrix} \begin{pmatrix} s_{0,2k} \\ s_{0,2k-1} \end{pmatrix}$$
$$= \sum_{k=0}^{N} \begin{pmatrix} h_{2k} & h_{2k+1} \\ g_{2k} & g_{2k+1} \end{pmatrix} \begin{pmatrix} s_{0,2(n-k)} \\ s_{0,2(n-k)-1} \end{pmatrix} \tag{2.4}$$

(This corresponds to a polyphase decomposition of the filters; see e.g., [30, 35, 38].) The subscripts 0, 1 correspond to the “level” of the filtered sequences; level 0 are the unprocessed data, while level 1 corresponds to one application of the H, G filters. More generally, $s_{j,n}$ and $d_{j,n}$ are obtained by applying H, G to the $s_{j-1,k}$, so that they are the results of j layers of filters.

If we define

$$H_k = \begin{pmatrix} h_{2k} & h_{2k+1} \\ g_{2k} & g_{2k+1} \end{pmatrix} \text{ with } 0 \leqslant k \leqslant N-1\,,$$

then (2.4) can be read as a 2D analog to the situation above; now each a_n or b_n is a 2-vector, with $a_n = (s_{0,2n}\, s_{0,2n-1})^t$, and $b_n = (s_{1,n}\, d_{1,n})^t$. For an arbitrary 2-vector $a = (a_1\, a_2)^t$, we introduce the notations

$$\{a\} = \begin{pmatrix} \{a_1\} \\ \{a_2\} \end{pmatrix}, \text{ and } \lfloor a \rfloor = \begin{pmatrix} \lfloor a_1 \rfloor \\ \lfloor a_2 \rfloor \end{pmatrix}.$$

We can now generalize the LTF precoder by defining

$$\begin{aligned} a'_n &= a_n - r_n \\ r_n &= \left\{ (H_0 - \mathrm{I})a'_n + \sum_{k=1}^{N-1} H_k a'_{n-k} \right\} = \{(\mathrm{I} - H_0)r_n + y_n\}\,, \end{aligned} \tag{2.5}$$

with

$$y_n = \left\{ (H_0 - \mathrm{I})a_n + \sum_{k=1}^{N-1} H_k a'_{n-k} \right\}.$$

If the equations (2.5) for r_n can indeed be solved, then we find, as before, that the

$$b'_n = \sum_{k=0}^{n-1} H_k a'_{n-k}$$

consist of only integer entries. It is not clear, however, that the equation

$$r = \{(\mathrm{I} - H_0)r + y\} \tag{2.6}$$

always has solutions for any $y \in [0,1) \times [0,1)$. Indeed, for the Haar case, where

$$H_0 = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$$

one checks that (2.6) has no solution if $y_1 = 1/4, y_2 = 1/2$.

The simple expedient of "renormalizing" H_0 to αH_0 works here as well. Let us replace the orthonormal filtering (2.3) and its inverse

$$s_{0,k} = \sum_n \left[h_{2n-k} s_{1,n} + g_{2n-k} d_{1,n} \right]$$

by scaled versions:

$$\widetilde{s}_{1,n} = \alpha \sum_k h_{2n-k} s_{0,k}, \quad \widetilde{d}_{1,n} = \beta \sum_k g_{2n-k} s_{0,k}$$

$$s_{0,k} = \sum_n \left[\alpha^{-1} h_{2n-k} \widetilde{s}_{1,n} + \beta^{-1} g_{2n-k} \widetilde{d}_{1,n}\right].$$

The polyphase regrouping then uses the matrices

$$\widetilde{H}_k = \begin{pmatrix} \alpha h_{2k} & \alpha h_{2k+1} \\ \beta g_{2k} & \beta g_{2k+1} \end{pmatrix} = \begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix} H_k\,;$$

the corresponding equations for r_1, r_2 become

$$\begin{aligned} r_1 &= \{(1 - \alpha h_0) r_1 + (y_1 - \alpha h_1 r_2)\} \\ r_2 &= \{(1 - \beta g_1) r_2 + (y_2 - \beta g_0 r_1)\}\,. \end{aligned} \tag{2.7}$$

For the special case where the wavelet basis is the Haar basis, i.e., $h_0 = h_1 = \frac{1}{\sqrt{2}} = g_1 = -g_0$, with all other $h_n, g_n = 0$, we can choose $\alpha = \beta = \sqrt{2}$. The system for r_1, r_2 reduces then to

$$\begin{aligned} r_1 &= \{y_1 - r_2\} \\ r_2 &= \{y_2 + r_1\} \end{aligned}$$

which has the easy solution $r_1 = \{(y_1 - y_2)/2\}\,, r_2 = \{(y_1 + y_2)/2\}$ for arbitrary $y_1, y_2 \in [0, 1)$. Unfortunately, things are not as easy for orthonormal wavelet filters of higher order. When $N > 1$, orthonormality of the wavelet filters implies

$$\begin{aligned} h_{2N-2} g_0 + h_{2N-1} g_1 &= 0 \\ h_{2N-2} h_0 + h_{2N-1} h_1 &= 0\,, \end{aligned}$$

where $h_{2N-1} \neq 0$. Hence $h_0\, g_1 = -h_{2N-2}\, g_0\, h_0 / h_{2N-1} = h_1\, g_0$ or $\det H_0 = 0$.

This implies that

$$r = \{(\mathrm{I} - \alpha H_0) r + y\} \tag{2.8}$$

cannot be solved for arbitrary $y \in [0, 1)^2$, regardless of the choice of α, as shown by the following argument. Suppose that for a given y in the interior of $[0, 1)^2$, r solves (2.8). Then there exists $n \in \mathbf{Z}^2$ so that

$$r + n = (\mathrm{I} - \alpha H_0) r + y\,,$$

or $\alpha H_0 r + n = y$. Since both r and y are in $[0, 1)^2$, it follows that $\|n\|$ is bounded by some $C > 0$, uniformly in y. Since $\det H_0 = 0$, there exists a 2-vector e in $\mathbf{R}^2$ so that $\alpha H_0 \mathbf{R}^2 \subset \{\lambda e; \lambda \in \mathbf{R}\}$.

Take now $f \perp e$, and consider $y' = y + \mu f$. Since y was chosen in the interior of $[0,1)^2$, y' still lies within $[0,1)^2$ for sufficiently small μ. Suppose r' is the corresponding solution of (2.8),

$$\alpha H_0 r' + n' = y' = y + \mu f \,.$$

Taking inner products of both sides with f leads to

$$\langle n', f\rangle = \langle y, f\rangle + \mu \|f\|^2 \,,$$

or

$$\mu = \langle n' - y, f\rangle / \|f\|^2 \,.$$

Since the right hand side can take on only a finite set of values (since $n' \in \mathbf{Z}^2$, $\|n'\| \leqslant C$), this can only hold for finitely many values of μ, and not for the interval around 0 for which $y' = y + \mu f$ is within $[0,1)^2$. (Note that the same no-go argument still works if we replace α by a diagonal matrix A, corresponding to a non-uniform expansion; in that case it suffices to choose $f \perp AH_0\mathbf{R}^2$.)

There exists a way around this problem, which exploits that the y_n are not really independent of the a_n or r_n, as we have implicitly assumed here. This is the subject of the next subsection.

2.2. **Uniform expansion factors.** For the Haar filter, we were looking in the last subsection at the equation

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} n_1 \\ n_2 \end{pmatrix} + \alpha \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$$

For $\alpha = 1$, $n_1, n_2 \in \mathbf{Z}$ and $r_1, r_2 \in [0,1)$, the possible values taken on by the right hand side range over the union of tilted squares shown in Figure 2a, which clearly do not cover the square $[0,1)^2$. For $\alpha = \sqrt{2}$, the tilted squares are blown up so that their union does not leave any gaps.

In the Haar case, the different pairs $(s_{0,2k}, s_{0,2k-1})$ remain nicely decoupled in the subband filtering process, which enabled us to reduce the whole analysis to a 2D argument. As we saw above, the most naive way to reduce to 2D does not work for longer filters; a solution to the problem at the end of the last section is possible only by taking into account to some extent the coupling between the pairs $(s_{0,2k}, s_{0,2k-1})$ in the filtering process. To do this, we shall first consider the full problem, for the complete data sequence, and try to carry out a reduction to fewer dimensions at a later stage. Imagine that the initial data constitute a finite sequence (of length L) of integers, $a \in \mathbf{Z}^L$. The subband filtering operator $\mathbf{H}$ can now be viewed as an

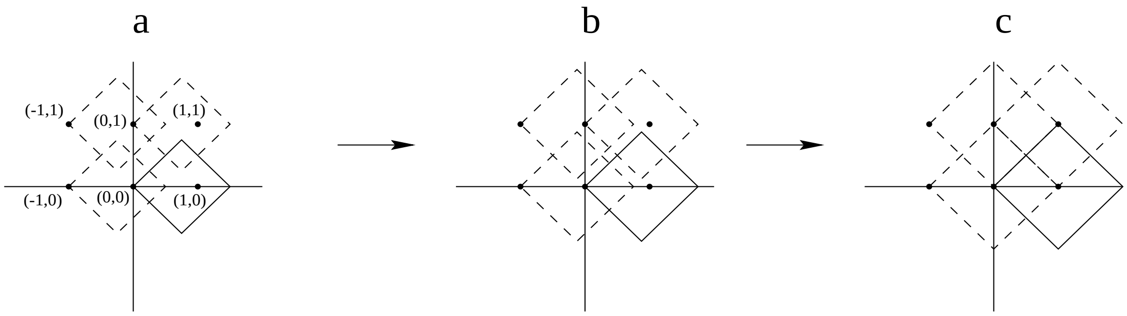

FIGURE 2.

a. *For* $H_0 = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$, *the union* $\bigcup_{n\in\mathbf{Z}^2}(H_0[0,1)^2+n)$ *does not cover* $[0,1)^2$.

b. and c. *As* $\alpha \geqslant 1$ *increases, the set* $[0,1)^2 \setminus \bigcup_{n\in\mathbf{Z}^2}(\alpha H_0[0,1)^2+n)$ *becomes smaller; the gap closes for* $\alpha = \sqrt{2}$.

$L \times L$-matrix, which decomposes into 2×2 blocks (see Section 2.1).

$$\mathbf{H} = \begin{pmatrix} H_0 & H_1 & H_2 & \cdots & H_{N-3} & H_{N-2} & H_{N-1} & 0 & 0 & \cdots & 0 \\ 0 & H_0 & H_1 & \cdots & H_{N-4} & H_{N-3} & H_{N-2} & H_{N-1} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ H_1 & H_2 & H_3 & \cdots & H_{N-2} & H_{N-1} & 0 & 0 & 0 & \cdots & H_0 \end{pmatrix} \tag{2.9}$$

(We have here given H a circulant-structure to deal with the finite length of the data sequence, which amounts to periodizing the data. Other ways of extending the data, such as reflections or interpolation techniques, can also be used. This would change only the first and last few rows and columns of the matrix, and it would not affect our argument significantly, although we would, of course, have to take those changes into account near the start and end of the signal. We have also implicitly assumed that L is even here, which is usually the case; odd L can be handled as well, but are slightly more tricky.)

In general, $\mathbf{H}a \notin \mathbf{Z}^L$ if $a \in \mathbf{Z}^L$. An appropriate shift of $\mathbf{H}a$ will bring it back to $\mathbf{Z}^L$. More precisely, if Λ is a fundamental region in $\mathbf{R}^L$ for the lattice $\mathbf{Z}^L$, then we can find $f = f(a) \in \Lambda$ so that $\mathbf{H}a + f \in \mathbf{Z}^L$. In order for the transform $a \longrightarrow \mathbf{H}a + f(a)$ to be invertible, we must require that if $a \neq a'$, then $\mathbf{H}a + f(a) \neq \mathbf{H}a' + f(a')$, or since $\mathbf{H}$ itself is invertible, $\mathbf{H}^{-1}f(a) \neq \mathbf{H}^{-1}f(a') + (a' - a)$. This will be satisfied if $\mathbf{H}^{-1}\Lambda \bigcap (\mathbf{H}^{-1}\Lambda + n) = \emptyset$ for $n \in \mathbf{Z}^L$, $n \neq 0$. The Haar example shows us already that it may not be possible to find such a fundamental region Λ without introducing an expansion factor. (For Haar, the problem reduces to considering H_0 only in 2D; since $H_0^{-1} = H_0^t$, we can look at Figure 2a again, which shows the regions $H_0[0,1)^2 + n$; their mirror images are the $H_0^t[0,1]^2 + m$. For $\alpha = 1$, they do intersect non-trivially.) If we do use an expansion factor α, then we are dealing with the transform $a \longrightarrow \alpha\mathbf{H}a + f$ instead, and the sufficient condition for invertibility becomes

$$\mathbf{H}^{-1}(\alpha^{-1}\Lambda) \bigcap \mathbf{H}^{-1}(\alpha^{-1}\Lambda + n) = \emptyset \text{ for all } n \in \mathbf{Z}^L \backslash \emptyset\,. \tag{2.10}$$

(In this section we consider uniform dilation factors only; non-uniform dilations, with different expansion factors for high and low pass parts, will be considered in the next subsection.) Given any bounded fundamental region Λ, (2.10) will always be satisfied for sufficiently large α; the challenge is, for a given H, to find a "reasonable" Λ for which α is as small as possible. (Since one goal of this approach is to develop a practical lossless coding scheme with wavelets, the region Λ must be tractable for numerical computations. In the Haar case of last section, we had $\Lambda = [0,1)^L$, and $\alpha = \sqrt{2}$.) It will be useful below, when we try to construct explicit Λ, to relax the condition that Λ be exactly a fundamental region and to require instead that Λ contain a fundamental region. Renaming $\alpha^{-1}\Lambda = \Gamma$, we have therefore the following challenge:
Find $\Gamma \subset \mathbf{R}^N$ so that

- $\alpha\Gamma$ contains a fundamental region for $\mathbf{Z}^L$ (with $\alpha \geqslant 1$ as small as we can get it)
- for all $n \in \mathbf{Z}^L, n \neq 0$.

$$\mathbf{H}^{-1}\Gamma \bigcap (\mathbf{H}^{-1}\Gamma + n) = \emptyset\,. \tag{2.11}$$

In order to reduce the problem to a smaller size and make the search for a suitable Γ more tractable, we shall here restrict ourselves to special Γ of the form

$$\Gamma = \Omega \times \Omega \times \ldots \times \Omega\,,$$

where $\Omega \subset \mathbf{R}^2$ so that $\alpha\Omega$ contains a fundamental region for $\mathbf{Z}^2$. This still enables us to exploit the 2×2-block structure of $\mathbf{H}$. Moreover, this will also make the implementation easier. Instead of working with the large matrix $\mathbf{H}$ and its inverse, for which we would have to wait to acquire all the data, restricting ourselves to product Γ makes a "running" implementation possible.

Let us look at this problem for the simplest example beyond the Haar case, which uses the 4-tap filters

$$\begin{array}{llll} h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}\,, & h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}\,, & h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}}\,, & h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}}\,, \\ g_0 = -h_3\,, & g_1 = h_2\,, & g_2 = -h_1\,, & h_3 = h_0\,. \end{array}$$

Introducing the unitary matrices

$$
\begin{aligned}
U = &\ \tfrac{1}{2\sqrt{2}} \begin{pmatrix} 1+\sqrt{3} & 1-\sqrt{3} \\ \sqrt{3}-1 & 1+\sqrt{3} \end{pmatrix} \\
V = &\ \tfrac{1}{2} \begin{pmatrix} 1 & \sqrt{3} \\ -\sqrt{3} & 1 \end{pmatrix},
\end{aligned}
$$

we can rewrite the matrices H_0 and H_1 (the only non-zero H_k) in this case as

$$
H_0 = U \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} V = U\,\widetilde{H}_0\,V, \quad H_1 = U \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} V = U\,\widetilde{H}_1\,V\,.
$$

(This is in fact a different way of writing the factorization of [36] for this particular case.)

It follows that $\mathbf{H} = \mathbf{U}\widetilde{\mathbf{H}}\mathbf{V}$, where $\mathbf{U}$ and $\mathbf{V}$ are matrices with the 2×2 blocks U resp. V, on the diagonal, and zeros elsewhere, and where

$$
\widetilde{\mathbf{H}} = \begin{pmatrix} \widetilde{H}_0 & \widetilde{H}_1 & 0 & \cdots & 0 \\ 0 & \widetilde{H}_0 & \widetilde{H}_1 & \cdots & 0 \\ \vdots & & & & \vdots \\ \widetilde{H}_1 & 0 & 0 & \cdots & \widetilde{H}_0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & \cdots & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \cdots & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 \end{pmatrix}.
$$

Note that $\widetilde{\mathbf{H}}f$ (and therefore $\widetilde{\mathbf{H}}^{-1}f$ as well) is just a permutation of the even entries of f. (For longer filters H this will no longer be true; we shall see below how to extend this approach to that case.) It follows that if $\widetilde{\Gamma}$ is a product of 2D regions, $\widetilde{\Gamma} = \widetilde{\Omega}\times\ldots\times\widetilde{\Omega}$, with $\widetilde{\Omega} \subset [a_1, a_2)\times[b_1, b_2) = R$ then $\widetilde{\mathbf{H}}\widetilde{\Gamma} \subset R\times\ldots\times R$. In particular, any $\widetilde{\Gamma}$ that is already of the form $R\times\ldots\times R$ is invariant under $\widetilde{\mathbf{H}}$ or $\widetilde{\mathbf{H}}^{-1}$. This suggests the following strategy for finding a suitable Ω that would satisfy all our requirements. Take Ω so that $U^t\Omega$ is a rectangle with its sides parallel to the axes; then $\mathbf{U}^t\Gamma = U^t\Omega\times\ldots\times U^t\Omega$ will be left invariant by $\widetilde{\mathbf{H}}$, so that $\mathbf{H}^{-1}\Gamma = (V^tU^t\Omega\times\ldots\times V^tU^t\Omega)$, and our sufficient condition reduces to a set of 2D requirements. In particular, the last part (2.11) of the sufficient condition reduces to

$$
V^tU^t\Omega \bigcap (V^tU^t\Omega + n) = \{0\} \text{ for } n\in\mathbf{Z}^2\backslash\{0\}\,.
$$

Since $U^t\Omega = [a_1, a_2)\times[b_1, b_2) = R$, this can be rewritten as

$$
R \bigcap (R + Vn) = \{0\} \text{ for } n\in\mathbf{Z}^2\backslash\{0\}
$$

or

$$R - R \bigcap V\mathbf{Z}^2 = \{0\}\,,$$

where we have used the notation $A - B = \{x - y;\, x \in A, y \in B\}$; in our case both A and B equal R, and

$$\begin{aligned} R - R \quad &= \{x - y;\, x, y \in R\} \\ &= \{z = (u, v);\, |u| < a_2 - a_1, |v| < b_2 - b_1\}\,. \end{aligned}$$

Note that this set is symmetric around the origin, even if R is not. We can therefore assume, without loss of generality, that R is centered around the origin, so that the condition reduces to

$$2R^\circ \bigcap V\mathbf{Z}^2 = \{0\} \text{ or } 2(V^tR)^\circ \bigcap \mathbf{Z}^2 = \{0\}\,,$$

where B° denotes the interior of the set B.

In summary, we are thus looking for a rectangle $R_{a,b} = [-a, a) \times [-b, b)$ so that

- $\alpha U R_{a,b}$ contains a fundamental region for $\mathbf{Z}^2$
- $2(V^tR_{a,b})^\circ \bigcap \mathbf{Z}^2 = \{0\}$.

We would like to find a, b so that α is as small as possible. We start by constructing one candidate for $R_{a,b}$, and computing the corresponding α. The following technical lemma will be useful in our construction:

Lemma 2.1. *Start with a parallelogram P that is a fundamental region for $\mathbf{Z}^2$ in $\mathbf{R}^2$. Form a hexagon $\Delta \subset P$ by picking six points on the boundary of P so that opposite vertices are congruent modulo $\mathbf{Z}^2$ and so that no three of the six points are collinear. Then any parallelogram $\widetilde{P}$ that contains Δ also contains the interior of a fundamental region of $\mathbf{Z}^2$.*

Proof. See Appendix A. $\square$

(Note that as a fundamental region for $\mathbf{Z}^2$, P cannot be open nor closed. Likewise, $\widetilde{P}$ can fail to contain a true fundamental region if $\widetilde{P}$ contains too little of $\partial\widetilde{P}$.)

Corollary 2.2. *Any parallelogram symmetric around 0 that contains the points $(\frac{1}{2}, 0), (0, \frac{1}{2})$ and either $(\frac{1}{2}, \frac{1}{2})$ or $(\frac{1}{2}, -\frac{1}{2})$ contains the interior of a fundamental region for $\mathbf{Z}^2$.*

Let us apply this to our problem. We start with the second requirement, that

$$R^\circ_{a,b} \bigcap \frac{1}{2}V\mathbf{Z}^2 = \{0\}\,.$$

Since V is an orthogonal matrix, $1/2V\mathbf{Z}^2$ is a (tilted) square lattice. It therefore makes sense (because of symmetry considerations) to choose $R_{a,b}$ itself to be a square, i.e., $a = b$. The largest possible value for a is then $\sqrt{3}/4$. (It suffices to check the points $(1, 0), (0, 1)$ and $(1, \pm 1)$ in $\mathbf{Z}^2$.)

Next, we check the other requirement. We have

$$\alpha U R_{a,b} = \alpha U R_{\frac{\sqrt{3}}{4},\frac{\sqrt{3}}{4}}$$
$$= \{(x,y) \in \mathbf{R}^2; |(1+\sqrt{3})x + (\sqrt{3}-1)y| \leqslant \tfrac{\sqrt{3}}{\sqrt{2}}\alpha, |(1-\sqrt{3})x + (1+\sqrt{3})y| \leqslant \tfrac{\sqrt{3}}{\sqrt{2}}\alpha\}\,.$$

A straightforward application of Corollary 2.2 shows that this contains a fundamental region for $\mathbf{Z}^2$ if $\alpha \geqslant \sqrt{2}$.

In order to implement this integer to integer transform we would then do the following:

- Given the integers $a_n = s_{0,n}$, $n = 1\ldots L$, compute b_n, $n = 1\ldots L$ by $b = \sqrt{2}\,\mathbf{H}\,a$.
- For each b_n, determine $b'_n \in \mathbf{Z}$ so that $b'_n - 1/2 \leqslant b_n \leqslant b_n + 1/2$.

To invert, we compute $c = 1/\sqrt{2}\,\mathbf{H}^{-1}\,b'$, and we find the unique $a \in \mathbf{Z}^L$ so that $V\,[(a_{2n,a_{2n+1}})^t - (c_{2n}, c_{2n+1})^t] \in R_{\sqrt{3}/4}$.

Remark: There exist other solutions, in which $R_{a,b}$ is not a square. In fact, we could just as well have replaced our choice $b = \frac{\sqrt{3}}{4}$ with the smaller value $b = \frac{1+\sqrt{3}}{8}$, and obtained the same value for α. This change of b corresponds to trimming the $R_{a,b}$ slightly in a non-essential way, which reduces the overlap of the $\alpha U R_{a,b}$ in the direction where we had room to spare. One can also find completely different $R_{a,b}$, with very different aspect ratios, that avoid $\frac{1}{2}V\mathbf{Z}^2$. These lead to worse α, however.

This approach can also be used for longer filters, and even for biorthogonal pairs. For the general biorthogonal case, the $L \times L$ wavelet matrix $\mathbf{H}$ is of the type

$$\mathbf{H} = \begin{pmatrix} h_0 & h_1 & \cdots & \cdots & h_{2N-2} & h_{2N-1} & 0 & 0 & \cdots & \cdots \\ g_0 & g_1 & \cdots & \cdots & g_{2N-2} & g_{2N-1} & 0 & 0 & \cdots & \cdots \\ 0 & 0 & h_0 & h_1 & \cdots & \cdots & h_{2N-2} & h_{2N-1} & 0 & \cdots \\ 0 & 0 & g_0 & g_1 & \cdots & \cdots & g_{2N-2} & g_{2N-1} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \end{pmatrix}$$

with

$$\mathbf{H}^{-1} = \begin{pmatrix} \widetilde{h}_0 & \widetilde{g}_0 & 0 & 0 & \cdots & \cdots \\ \widetilde{h}_1 & \widetilde{g}_1 & 0 & 0 & \cdots & \cdots \\ \widetilde{h}_2 & \widetilde{g}_2 & \widetilde{h}_0 & \widetilde{g}_0 & 0 & \cdots \\ \widetilde{h}_3 & \widetilde{g}_3 & \widetilde{h}_1 & \widetilde{g}_1 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

where $\sum_n h_n \widetilde{h}_{n+2k} = \delta_{k,0}$, $\widetilde{g}_n = (-1)^n h_{-n-1+2N}$, $g_n = (-1)^n \widetilde{h}_{-n-1+2N}$. We assume that h_0 and h_1 are not both zero (same for $\widetilde{h}_0, \widetilde{h}_1$), but we place no such restrictions on the final filter taps, so that our notation can accommodate biorthogonal filters of unequal length (which is always the case when the filter lengths are odd, as in the popular 9-7 filter pair [6]).

Leaving the circulant nature of $\mathbf{H}^{-1}$ aside, $\mathbf{H}^{-1}$ consists of repeats of the two rows

$$\begin{array}{cccccccccccccc} \cdots & 0 & 0 & \widetilde{h}_{2N-2} & \widetilde{g}_{2N-2} & \widetilde{h}_{2N-4} & \widetilde{g}_{2N-4} & \cdots & \widetilde{h}_0 & \widetilde{g}_0 & 0 & 0 & \cdots \\ \cdots & 0 & 0 & \widetilde{h}_{2N-1} & \widetilde{g}_{2N-1} & \widetilde{h}_{2N-3} & \widetilde{g}_{2N-3} & \cdots & \widetilde{h}_1 & \widetilde{g}_1 & 0 & 0 & \cdots \end{array}$$

with offsets of 2 at each repeat.

Since $h_0\widetilde{h}_{2N-2} + h_1\widetilde{h}_{2N-1} = 0$ and $h_{2N-2}\widetilde{h}_0 + h_{2N-1}\widetilde{h}_1 = 0$, the two 2×2 matrices at the start and at the end of this repeating block are again singular. Define ratios λ and μ so that

$$\lambda h_1 = \widetilde{h}_{2N-2}, \lambda h_0 = -\widetilde{h}_{2N-1}, \text{ and } \mu\widetilde{h}_1 = h_{2N-2}, \mu\widetilde{h}_0 = -h_{2N-1}\,,$$

and define the matrices

$$A = \begin{pmatrix} h_1 & \widetilde{h}_0 \\ -h_0 & \widetilde{h}_1 \end{pmatrix} \quad B = \begin{pmatrix} \lambda & 1 \\ 1 & -\mu \end{pmatrix}.$$

Then

$$\begin{pmatrix} \widetilde{h}_{2N-2} & \widetilde{g}_{2N-2} \\ \widetilde{h}_{2N-1} & \widetilde{g}_{2N-1} \end{pmatrix} = \begin{pmatrix} \lambda h_1 & h_1 \\ -\lambda h_0 & -h_0 \end{pmatrix} = A \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} B\,,$$

and

$$\begin{pmatrix} \widetilde{h}_0 & \widetilde{g}_0 \\ \widetilde{h}_1 & \widetilde{g}_1 \end{pmatrix} = \begin{pmatrix} \widetilde{h}_0 & -\mu\widetilde{h}_0 \\ \widetilde{h}_1 & -\mu\widetilde{h}_1 \end{pmatrix} = A \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} B\,.$$

It follows that $\mathbf{H}^{-1}$ has a structure reminiscent of (2.11),

$$\mathbf{H}^{-1} = \mathbf{A} \begin{pmatrix} & \vdots & & \vdots & & \vdots & \\ \cdots & 0\ \ 0\ \ 1\ \ 0 & & C_1 \ldots C_{N-2} & & 0\ \ 0\ \ 0\ \ 0 & \cdots \\ \cdots & 0\ \ 0\ \ 0\ \ 0 & & & & 0\ \ 1\ \ 0\ \ 0 & \cdots \\ & \vdots & & \vdots & & \vdots & \end{pmatrix} \mathbf{B}\,, \tag{2.12}$$

where $\mathbf{A}$ and $\mathbf{B}$ have the 2×2 blocks A, respectively B, on the diagonal, and zero entries elsewhere. The 2×2 matrices C_k are defined by

$$C_k = A^{-1} \begin{pmatrix} \widetilde{h}_{2N-2-2k} & \widetilde{g}_{2N-2-2k} \\ \widetilde{h}_{2N-1-2k} & \widetilde{g}_{2N-1-2k} \end{pmatrix} B^{-1}.$$

Note that since $\mathbf{H}^{-1}$ is invertible, both $\mathbf{A}$ and $\mathbf{B}$, and therefore also A and B must be invertible. (In the orthogonal case, this can be seen immediately because $\det A = h_0^2 + h_1^2, \det B = -1 - \lambda^2$.) We can now follow the same strategy as before: we try to find $\Omega \subset \mathbf{R}^2$ so that

- $\alpha\Omega$ contains a fundamental region for $\mathbf{Z}^2$
- for all $n \in \mathbf{Z}^L$, $n \neq 0$

$$\mathbf{H}^{-1}(\Omega \times \ldots \times \Omega) \bigcap \left[\mathbf{H}^{-1}(\Omega \times \ldots \times \Omega) + n\right] = 0 \tag{2.13}$$

We shall take the ansatz $\Omega = B^{-1}R_{a,b}$. On the other hand, we define $\widetilde{\Omega} = AR_{c,d}$; for appropriate c, d, we then have

$$\mathbf{H}^{-1}(\Omega \times \ldots \times \Omega) \subset \widetilde{\Omega} \times \ldots \times \widetilde{\Omega}\,.$$

Indeed, it suffices that

$$R_{a,b} + C_1 R_{a,b} + \cdots + C_{N-2} R_{a,b} \subset R_{c,d}\,. \tag{2.14}$$

This, in turn, is assured if

$$\begin{pmatrix} 1 + \sum_{\ell=1}^{N-2} |C_{\ell;1,1}| & \sum_{\ell=1}^{N-2} |C_{\ell;1,2}| \\ \sum_{\ell=1}^{N-2} |C_{\ell;2,1}| & 1 + \sum_{\ell=1}^{N-2} |C_{\ell;2,2}| \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} c \\ d \end{pmatrix}. \tag{2.15}$$

(Of course (2.14) may be satisfied for choices of a, b, c, d where (2.15) is not, but in practice, for the filter pairs we considered, not much is lost.) Finally, if $2AR^\circ_{c,d} \bigcap \mathbf{Z}^2 = \{0\}$ or $R^\circ_{c,d} \bigcap \frac{1}{2}A^{-1}\mathbf{Z}^2 = \{0\}$, then the second part of (2.13) is satisfied.

Let us see how this works on the example of the 6-tap orthogonal filter with three vanishing moments,

$$\begin{array}{lll} h_0 = .332671 & h_1 = .806891 & h_2 = .459878 \\ h_3 = -.135011 & h_4 = -.085441 & h_5 = .035226\,. \end{array}$$

Because of orthogonality, λ equals μ with $\lambda = \mu = \frac{h_\mu}{h_1} \simeq -.1059$; moreover both A and B are orthogonal matrices (up to an overall constant each),

$$A = (h_0^2 + h_1^2)^{\frac{1}{2}} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \quad B = (1+\lambda^2)^{\frac{1}{2}} \begin{pmatrix} \cos\varphi & \sin\varphi \\ \sin\varphi & -\cos\varphi \end{pmatrix},$$

with $\theta = \arctan\frac{h_0}{h_1}, \varphi = \arctan\frac{1}{\lambda}$. We now want to determine c, d so that $R^\circ_{c,d} \bigcap \frac{1}{2}A^{-1}\mathbf{Z}^2 = \{0\}$. Since A consists of a rotation combined with an overall scaling, $\frac{1}{2}A^{-1}\mathbf{Z}^2$ is a square lattice, and it makes sense to choose $c = d$, as in the 4-tap case. The largest possible value for $c = d$ is then $\frac{h_1}{2(h_0^2+h_1^2)} \simeq .5296$.

On the other hand, because the filters are so short, we have only one matrix C_1,

$$C_1 = \frac{1}{(h_0^2+h_1^2)(1+\lambda^2)} \begin{pmatrix} h_1 & -h_0 \\ h_0 & h_1 \end{pmatrix} \begin{pmatrix} h_2 & h_3 \\ h_3 & -h_2 \end{pmatrix} \begin{pmatrix} \lambda & 1 \\ 1 & -\lambda \end{pmatrix} = \begin{pmatrix} 0 & .5461 \\ -.5461 & 0 \end{pmatrix}$$

According to (2.15), we then have

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 & .5461 \\ .5461 & 1 \end{pmatrix}^{-1} \begin{pmatrix} c \\ d \end{pmatrix} \simeq \begin{pmatrix} .3425 \\ .3425 \end{pmatrix}.$$

Finally, it remains to determine α so that $\alpha\Omega = \alpha B^{-1}R_{a,b}$ contains a fundamental region for $\mathbf{Z}^2$. $\alpha B^{-1}R_{a,b}$ is the tilted square delimited by $\lambda x + y = \pm a\alpha(\lambda^2+1), x - \lambda y = \pm b\alpha(\lambda^2+1)$; by Corollary 2.2 this contains a fundamental region if $(\frac{1}{2}, 0), (0, \frac{1}{2})$ and $(-\frac{1}{2}, \frac{1}{2})$ are included, leading to $d \geqslant \frac{|\lambda|+1}{2a(1+\lambda^2)} \simeq 1.5965$. This expansion factor is significantly worse than the factor $\sqrt{2}$ obtained for the 4-tap orthonormal filter earlier. In fact, applying this same strategy to the family of all 6-tap orthonormal filters with at least one vanishing moment, leads to $\alpha \geqslant \sqrt{2}$ in all cases, with the minimum $\alpha = \sqrt{2}$ attained only for the 4-tap filter. In the next subsection, we shall see that better results for orthonormal filters can be obtained via non-uniform expansion factors.

The same strategy can also be used for biorthogonal filters. For the simple filter pair

$$\begin{aligned} &h_0 = -\frac{1}{8} = h_4, h_1 = \frac{1}{4} = h_3, h_2 = \frac{3}{4}, \text{ other } h_n = 0 \\ &\widetilde{h}_1 = \widetilde{h}_3 = \frac{1}{2}, \widetilde{h}_2 = 1, \text{ other } \widetilde{h}_n = 0\,, \end{aligned} \tag{2.16}$$

the matrices A, B, and C are

$$A = \begin{pmatrix} \frac{1}{4} & 0 \\ \frac{1}{8} & \frac{1}{2} \end{pmatrix} \quad B = \begin{pmatrix} 0 & 1 \\ 1 & \frac{1}{4} \end{pmatrix} \quad C = \begin{pmatrix} 0 & 4 \\ -\frac{7}{4} & 0 \end{pmatrix}\,,$$

leading to the choices $c = 2, d = 1$, and $a = \frac{1}{3}, b = \frac{5}{12}$. Then $\alpha B^{-1} R_{a,b}$ contains $(\frac{1}{2}, 0), (0, \frac{1}{2})$ and $(-\frac{1}{2}, \frac{1}{2})$ (and therefore a fundamental region for $\mathbf{Z}^2$) if $\alpha \geqslant \frac{3}{2}$. In this case it is not clear, however, what a uniform expansion factor really means, since the two filters h and $\widetilde{h}$ need not have the same normalization in a biorthogonal scheme—as indeed they do not in this example, where $\sum_n h_n = 1, \sum_n \widetilde{h}_n = 2$. One could explore "renormalized" versions, replacing h_n with δh_n and $\widetilde{h}_n$ with $\delta^{-1}\widetilde{h}_n$, and find the corresponding expansion factors. This amounts to the same as introducing nonuniform expansion factors, the subject of the next subsection.

2.3. **Different expansion factors for the high and low pass channels.** Nothing forces us, when we introduce our expansion factor, to choose the *same* expansion for the high pass channel as for the low pass channel; allowing for different factors may lead to better results. In practice, the dynamic ranges of the two channels on (natural) images are very different, and this might be another reason to consider different expansion factors.

The analysis is similar to what was done in the previous subsection. We replace the uniform factor α by a pair (α_L, α_H), and we denote by $\boldsymbol{\alpha}$ the operation which multiplies the odd-indexed entries a_{2k+1} of $\boldsymbol{\alpha} \in \mathbf{Z}^L$ with α_L, the even-indexed entries a_{2k} by α_H. We again seek a subset $\Gamma \subset \mathbf{R}^L$ that satisfies (2.11) and so that $\boldsymbol{\alpha}\Gamma$ contains a fundamental region for $\mathbf{Z}^L$. We can now seek to minimize α_L (which leads to the least increase in dynamical range as we keep iterating the wavelet filtering) or the product $\alpha_L \alpha_H$ (to minimize the impact on the high-low, low-high channels that detect horizontal and vertical edges in $2D$-images).

As a warm-up, let us look at the Haar case again. We can confine our analysis to $\mathbf{R}^2$ and $\mathbf{Z}^2$. A pair of expansion factors (α_L, α_H) will work if we can find a set Ω in $\mathbf{R}^2$ so that

- Ω is a fundamental region for $\mathbf{Z}^2$
- $H_0^{-1}\alpha^{-1}\Omega \bigcap (H_0^{-1}\alpha^{-1}\Omega + n) = 0$ for $n \in \mathbf{Z}^2, n \neq 0$,

where α denotes now the 2×2 matrix,

$$\alpha = \begin{pmatrix} \alpha_L & 0 \\ 0 & \alpha_H \end{pmatrix}.$$

Let us take $\Omega = [-\frac{1}{2}, \frac{1}{2})^2$, so that Ω° is symmetric with respect to the origin, consistent with our analysis of longer filters in Section 2.2; this clearly satisfies the first requirement. Then $\alpha^{-1}\Omega$ is the rectangle $[-\frac{\alpha_L^{-1}}{2}, \frac{\alpha_L^{-1}}{2}) \times [-\frac{\alpha_H^{-1}}{2}, \frac{\alpha_H^{-1}}{2})$. The matrix H_0^{-1} rotates this by $45°$, to a tilted rectangle bounded by $y = x \pm \frac{1}{\sqrt{2}}\alpha_H^{-1}, y = -x \pm \frac{1}{\sqrt{2}}\alpha_L^{-1}$ (see Figure 3). In order to satisfy the second requirement, this tilted rectangle can have area at most 1, or $\alpha_H\alpha_L \geqslant 1$. Can we choose α_L, α_H so as to achieve the extremum $\alpha_H\alpha_L = 1$? In that case $H_0^{-1}\alpha^{-1}\Omega$ must be a fundamental region itself. For this, it is sufficient that $(0, \frac{1}{2}), (\frac{1}{2}, 0)$ and $(\frac{1}{2}, \frac{1}{2}) \in H_0^{-1}\alpha^{-1}\Omega$, or $\frac{1}{2} \leqslant \frac{1}{\sqrt{2}}\alpha_H^{-1}, 1 \leqslant \frac{1}{\sqrt{2}}\alpha_L^{-1}$. This leads to the choices $\alpha_H = \sqrt{2}, \alpha_L = \frac{1}{\sqrt{2}}$, which do indeed satisfy $\alpha_L\alpha_H = 1$. For the Haar case, we can therefore find a non-uniform expansion matrix corresponding with a global expansion factor of 1 (i.e., no net expansion factor), a significant gain over the uniform case. Note that the total transform αH_0 is then given by

$$\begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} \frac{x+y}{2} \\ x - y \end{pmatrix}, \tag{2.17}$$

a (non-orthonormal) form of the Haar transform often used in practice, and one which has been known to lead to a map of integers to integers, called the S-transform [39]. We shall come back to this in Section 3.1.

After this warm-up, let us consider longer filters. We can re-use much of what was done in the previous section. The factorization (2.12) becomes now

$$\mathbf{H}^{-1}\boldsymbol{\alpha}^{-1} = \mathbf{A} \begin{pmatrix} & \vdots & & \vdots & & \vdots & \\ \cdots & 0\ 0\ 1\ 0 & & & & 0\ 0\ 0\ 0 & \cdots \\ & & & C_1 \ldots C_{N-2} & & & \\ \cdots & 0\ 0\ 0\ 0 & & & & 0\ 1\ 0\ 0 & \cdots \\ & \vdots & & \vdots & & \vdots & \end{pmatrix} \mathbf{B}\boldsymbol{\alpha}^{-1},$$

which suggests that we seek $\Omega = \alpha B^{-1}R_{a,b}$ so that $(\boldsymbol{\alpha}\mathbf{H})^{-1}(\Omega \times \ldots \times \Omega) \subset (AR_{c,d} \times \ldots \times AR_{c,d})$ where $R_{c,d}^\circ \bigcap \frac{1}{2}A^{-1}\mathbf{Z}^2 = \{0\}$. Let us check what this leads to in a few examples. We first take the 4-tap orthonormal filter. In that case, there are no C-matrices to worry about, and we have

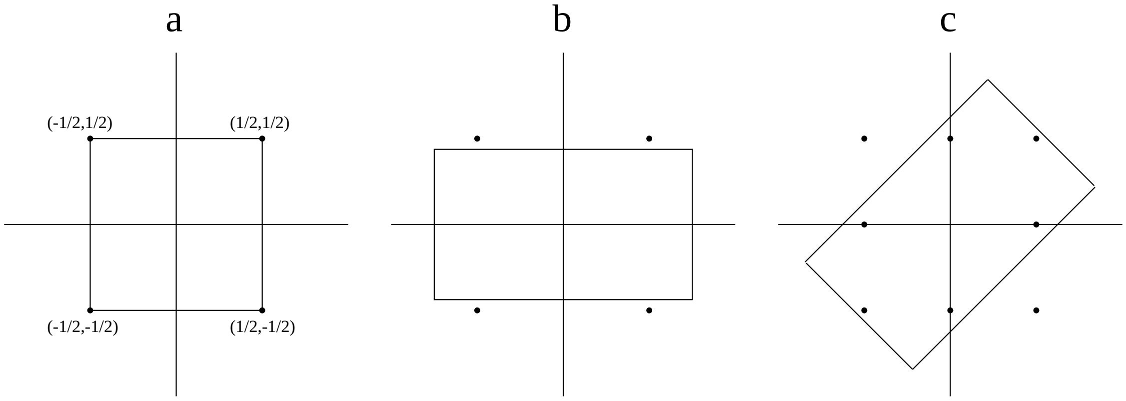

FIGURE 3.

a. *The square* $\Omega = \left[-\frac{1}{2}, \frac{1}{2}\right)^2$

b. *Applying* α^{-1} *leads to* $\left[-\frac{\alpha_L^{-1}}{2}, \frac{\alpha_L^{-1}}{2}\right) \times \left[-\frac{\alpha_H^{-1}}{2}, \frac{\alpha_H^{-1}}{2}\right)$

c. *Applying next* H_0^{-1} *leads to the rectangle given by* $|y-x| \leqslant \frac{\alpha_H^{-1}}{\sqrt{2}}, |y+x| \leqslant \frac{\alpha_L^{-1}}{\sqrt{2}}$. *This contains* $(\frac{1}{2}, 0), (0, \frac{1}{2})$ *and* $(\frac{1}{2}, \frac{1}{2})$ *if* $\alpha_H \geqslant \sqrt{2}, \alpha_L \geqslant \frac{1}{\sqrt{2}}$.

$c = a, d = b$. The matrix A $(= V^t$ in the notations of the first part of Section 2.2) is still the same rotation, so we still choose $R_{c,d}$ to be the square given by $c = d = \frac{\sqrt{3}}{4}$. Because $\alpha_L \neq \alpha_H$, the set $\alpha B^{-1} R_{a,b}$ will no longer be a square; it is the parallelogram

$$\Omega = \{(x,y); |(1+\sqrt{3})\frac{x}{\alpha_L} + (1-\sqrt{3})\frac{y}{\alpha_H}| \leqslant \frac{\sqrt{6}}{2} \text{ and } |(\sqrt{3}-1)\frac{x}{\alpha_L} + (\sqrt{3}+1)\frac{y}{\alpha_H}| \leqslant \frac{\sqrt{6}}{2}\}.$$

If we assume $\alpha_L \leqslant \alpha_H$, then the condition $\alpha_L \geqslant \frac{1+\sqrt{3}}{\sqrt{6}}$ already ensures that $(\frac{1}{2}, 0)$ and $(0, \frac{1}{2}) \in \Omega$. If we pick the minimal $\alpha_L = \frac{1+\sqrt{3}}{\sqrt{6}} \simeq 1.1$, does there exist a solution for α_H and how large does it need to be? With the notation $\alpha_H = \alpha_L \cdot \delta$, with $\delta \geqslant 1$, the parallelogram Ω contains the point $(\frac{1}{2}, \frac{1}{2})$ (and therefore also a fundamental region for $\mathbf{Z}^2$) if $\left|1 + \frac{1-\sqrt{3}}{1+\sqrt{3}}\frac{1}{\delta}\right| \leqslant 1$, or $\delta \geqslant \frac{\sqrt{3}+1}{2}$, leading to the solution $\alpha_H = \frac{2+\sqrt{3}}{\sqrt{6}} \simeq 1.6$. The product $\alpha_L \alpha_H$ is then $\frac{5+3\sqrt{3}}{6} \simeq 1.7$, an improvement over $\alpha^2 = 2$ in the uniform expansion case, although it is not as spectacular as in the Haar case.

Our next example is the 6-tap orthonormal filter also considered in Section 2.2. We keep again the same $R_{c,d}$ as before, with $c = d \simeq .5296$. The square $R_{a,b}$ is also not changed, with $a = b \simeq .3425$. It then remains to determine α_L, α_H (minimizing either α_L or $\alpha_L \alpha_H$) so that

$\alpha B^{-1}R_{a,b}$ contains a fundamental region. We have

$$\alpha B^{-1}R_{a,b}=\{(x,y);\left|\lambda\frac{x}{\alpha_L}+\frac{y}{\alpha_H}\right|\leqslant a(\lambda^2+1)\,,\left|\frac{x}{\alpha_L}-\lambda\frac{y}{\alpha_H}\right|\leqslant b(\lambda^2+1)\}\,.$$

This parallelogram contains $(\frac{1}{2},0),(0,\frac{1}{2})$ and $(-\frac{1}{2},\frac{1}{2})$, and therefore a fundamental region for $\mathbf{Z}^2$, if $\alpha_L=\frac{1}{2a(\lambda^2+1)}\simeq 1.4437, \alpha_H=\frac{1}{2a(1-|\lambda|)(1+\lambda^2)}\simeq 1.6147$. Again the product $\alpha_L\alpha_H\simeq 2.3310$ is smaller than the corresponding $(1.5965)^2\simeq 2.5490$ in the uniform expansion case, but the gain is not as dramatic as in the Haar case: we still have $\alpha_L\alpha_H>1$; worse, even α_L is still larger than $\sqrt{2}$.

Finally, let us revisit the biorthogonal example at the end of Section 2.2. Again, the values of c,d, and a,b can be carried over, and we consider

$$\alpha B^{-1}R_{a,b}=\left\{(x,y);\left|\frac{y}{\alpha_H}\right|\leqslant\frac{1}{3},\left|\frac{x}{\alpha_L}+\frac{y}{4\alpha_H}\right|\leqslant\frac{5}{12}\right\}\,.$$

This contains $(\frac{1}{2},0),(0,\frac{1}{2})$ and $(-\frac{1}{2},\frac{1}{2})$ if $\alpha_L=\frac{6}{5},\alpha_H=\frac{3}{2}$, a gain over uniform expansion, but still a rather large price to pay, when compared to the Haar case (where we obtained $\alpha_L<1$!).

In the next section, a different approach is explained, which goes beyond the reduction to $2D$ used so far, and which enables us to reduce *any* wavelet filtering to a map from integers to integers without global expansion. In fact, this amounts to constructing appropriate sets $\Gamma_{\mathbf{n}}$ without using products of $2D$ regions by using a different representation of the filtering operations.

3. The lifting scheme and integer to integer transforms

3.1. **The S transform.** We return again to the Haar transform, which we write in its unnormalized version (2.17) involving simply pairwise averages and differences:

$$\begin{aligned} s_{1,l} &= \frac{s_{0,2l}+s_{0,2l+1}}{2} \\ d_{1,l} &= s_{0,2l+1}-s_{0,2l}\,. \end{aligned} \tag{3.1}$$

Its inverse is given by

$$\begin{aligned} s_{0,2l} &= s_{1,l}+d_{1,l}/2 \\ s_{0,2l+1} &= s_{1,l}-d_{1,l}/2\,. \end{aligned} \tag{3.2}$$

Because of the division by two, this is not an integer transform. Obviously we can build an integer version by simply omitting the division by two in the formula for $s_{1,l}$ and calculating the sum instead of the average; this is effectively what was proposed in Section 2.1. But a more efficient construction, known as the S transform (**S**equential), is possible [24, 16]. Several definitions of the S transform exist, differing only in implementation details. We consider the following case:

$$\begin{aligned} s_{1,l} &= \lfloor (s_{0,2l} + s_{0,2l+1})/2 \rfloor \\ d_{1,l} &= s_{0,2l+1} - s_{0,2l}\,. \end{aligned} \tag{3.3}$$

At first sight, the rounding-off in this definition of $s_{1,l}$ seems to discard some information. However, the sum and the difference of two integers are either both even or both odd. We can thus safely omit the last bit of the sum, since it is equal to the last bit of the difference. The S-transform thus is invertible and the inverse is given by

$$\begin{aligned} s_{0,2l} &= s_{1,l} - \lfloor d_{1,l}/2 \rfloor \\ s_{0,2l+1} &= s_{1,l} + \lfloor (d_{1,l}+1)/2 \rfloor\,. \end{aligned} \tag{3.4}$$

As long as the S-transform is viewed as a way to exploit that the sum and difference of two integers have the same parity, it is not clear how to generalize this approach to other wavelet filters. A different way of writing the Haar transform, using "lifting" steps[1], leads to a natural generalization. Lifting is a flexible technique that has been used in several different settings, for an easy construction and implementation of "traditional" wavelets [32], and of "second generation" wavelets [33], such as spherical wavelets [26]. Lifting is also closely related to several other techniques [13, 22, 37, 34, 20, 4, 15, 7, 3, 19, 28]. Rather than giving the general structure of lifting at this point, we show how to rewrite the Haar and S transforms using lifting.

We rewrite (3.1) in two steps which need to be executed sequentially. First compute the difference and then use the difference in the second step to compute the average:

$$\begin{aligned} d_{1,l} &= s_{0,2l+1} - s_{0,2l} \\ s_{1,l} &= s_{0,2l} + d_{1,l}/2\,. \end{aligned} \tag{3.5}$$

This is the same as the Haar transform (3.1) because $s_{0,2l} + d_{1,l}/2 = s_{0,2l} + s_{0,2l+1}/2 - s_{0,2l}/2 = s_{0,2l}/2 + s_{0,2l+1}/2$. It is now immediately clear how to compute the inverse transform again in two sequential steps. First recover the even sample from the average and difference, and later recover the odd sample using the even and the difference. The equations can be found by reversing the order and changing the signs of the forward transform:

$$\begin{aligned} s_{0,2l} &= s_{1,l} - d_{1,l}/2 \\ s_{0,2l+1} &= d_{1,l} + s_{0,2l}\,. \end{aligned}$$

[1] The name "lifting" was coined in [32], where it was developed to painlessly increase ("lift") the number of vanishing moments of a wavelet.

As long as the transform is written using lifting, the inverse transform can be found immediately. We will show below how this works for longer filters or even for non-linear transforms.

We can change (3.5) into an integer transform by truncating the division in the second step:

$$\begin{aligned} d_{1,l} &= s_{0,2l+1} - s_{0,2l} \\ s_{1,l} &= s_{0,2l} + \lfloor d_{1,l}/2 \rfloor \,. \end{aligned}$$

This is the same as the S transform (3.3) because $s_{0,2l} + \lfloor d_{1,l}/2 \rfloor = s_{0,2l} + \lfloor s_{0,2l+1}/2 - s_{0,2l}/2 \rfloor = \lfloor s_{0,2l}/2 + s_{0,2l+1}/2 \rfloor$.

Even though the transform now is non-linear, lifting allows us to immediately find the inverse. Again the equations follow from reversing the order and changing the signs of the forward transform:

$$\begin{aligned} s_{0,2l} &= s_{1,l} - \lfloor d_{1,l}/2 \rfloor \\ s_{0,2l+1} &= d_{1,l} + s_{0,2l} \,. \end{aligned}$$

Substituting and using that $d - \lfloor d/2 \rfloor = \lfloor (d+1)/2 \rfloor$, we see that this is the same as the inverse S transform (3.4).

This example shows that lifting allows us to obtain an integer transform using simply truncation and without losing invertibility.

3.2. **Beyond the S transform.** Two generalizations of the S transform have been proposed: the TS transform [39] and the S+P transform [25]. The idea underlying both is the same: add a third stage in which a prediction of the difference is computed based on the average values; the new high pass coefficient is now the difference of this prediction and the actual difference. This can be thought of as another lifting step and therefore immediately results in invertible integer wavelet transforms.

We first consider the TS transform. The idea here is to build an integer version of the $(3,1)$ biorthogonal wavelet transform of Cohen-Daubechies-Feauveau [6]. We present it here immediately using sequential lifting steps. The original, non-truncated transform goes in three steps. The first two are simply Haar using a preliminary high pass coefficient $d_{1,l}^{(1)}$. In the last step the real high pass coefficient $d_{1,l}$ is found as $d_{1,l}^{(1)}$ minus its prediction:

$$\begin{aligned} d_{1,l}^{(1)} &= s_{0,2l+1} - s_{0,2l} \\ s_{1,l} &= s_{0,2l} + d_{1,l}^{(1)}/2 \\ d_{1,l} &= d_{1,l}^{(1)} + s_{1,l-1}/4 - s_{1,l+1}/4 \,. \end{aligned} \tag{3.6}$$

The prediction for the difference thus is $s_{1,l+1}/4 - s_{1,l}/4$. The coefficients of the prediction are chosen so that in case the original sequence was a second degree polynomial in l, then the new wavelet coefficient $d_{1,l}$ is exactly zero. This assures that the dual wavelet has three vanishing moments. This prediction idea also connects to average-interpolation as introduced by Donoho in [12]. Substituting we see that the high pass filter is $\{1/8, 1/8, -1, 1, -1/8, -1/8\}$ while the low pass filter is $\{1/2, 1/2\}$, and we thus indeed calculate the (3,1) transform from [6]. Here the numbers 3 and 1 stand respectively for the number of vanishing moments of the analyzing and synthesizing high pass filter. We shall follow this convention throughout.

The integer version or TS-transform now can be obtained by truncating the non-integers coming from the divisions in the last two steps:

$$\begin{aligned} d^{(1)}_{1,l} &= s_{0,2l+1} - s_{0,2l} \\ s_{1,l} &= s_{0,2l} + \lfloor d^{(1)}_{1,l}/2 \rfloor \\ d_{1,l} &= d^{(1)}_{1,l} + \lfloor s_{1,l-1}/4 - s_{1,l+1}/4 + 1/2 \rfloor \, . \end{aligned} \tag{3.7}$$

In the last step we add a term 1/2 before truncating to avoid bias; in the second step this is not needed because the division is only by two. As the transform is already decomposed into lifting steps, the inverse again can be found immediately by reversing the equations and flipping the signs:

$$\begin{aligned} d^{(1)}_{1,l} &= d_{1,l} - \lfloor s_{1,l-1}/4 - s_{1,l+1}/4 + 1/2 \rfloor \\ s_{0,2l} &= s_{1,l} - \lfloor d^{(1)}_{1,l}/2 \rfloor \\ s_{0,2l+1} &= d^{(1)}_{1,l} + s_{0,2l} \, . \end{aligned} \tag{3.8}$$

The S+P transform (**S** transform + **P**rediction) goes even further. Here the predictor for $d^{(1)}_{1,l}$ does not only involve $s_{1,k}$ values but also a previously calculated $d^{(1)}_{1,l+1}$ value. The general form of the transform is

$$\begin{aligned} d^{(1)}_{1,l} &= s_{0,2l+1} - s_{0,2l} \\ s_{1,l} &= s_{0,2l} + \lfloor d^{(1)}_{1,l}/2 \rfloor \\ d_{1,l} &= d^{(1)}_{1,l} + \lfloor \alpha_{-1}\,(s_{1,l-2} - s_{1,l-1}) + \alpha_0\,(s_{1,l-1} - s_{1,l}) + \alpha_1\,(s_{1,l} - s_{1,l+1}) - \beta_1\, d^{(1)}_{1,l+1} \rfloor \end{aligned} \tag{3.9}$$

Note that the TS-transform is a special case, namely when $\alpha_{-1} = \beta_1 = 0$ and $\alpha_0 = \alpha_1 = 1/4$. Said and Pearlman examine several choices for (α_w, β_1) and in the case of natural images suggest $\alpha_{-1} = 0, \alpha_0 = 2/8, \alpha_1 = 3/8$ and $\beta_1 = -2/8$. It is interesting to note that, even though this was not their motivation, this choice without truncation yields a high pass analyzing filter with 2 vanishing moments.

3.3. **The lifting scheme.** We mentioned that the S, TS, and S+P transform can all be seen as special cases of the lifting scheme. In this section, we describe the lifting scheme in general [32, 33]. There are two ways of looking at lifting: either from the basis function point of view or from the transform point of view. We consider the transform point of view.

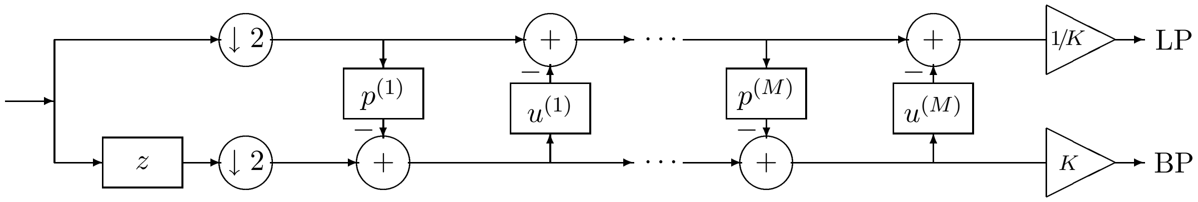

FIGURE 4. *The forward wavelet transform using lifting: First the Lazy wavelet, then alternating dual lifting and lifting steps, and finally a scaling.*

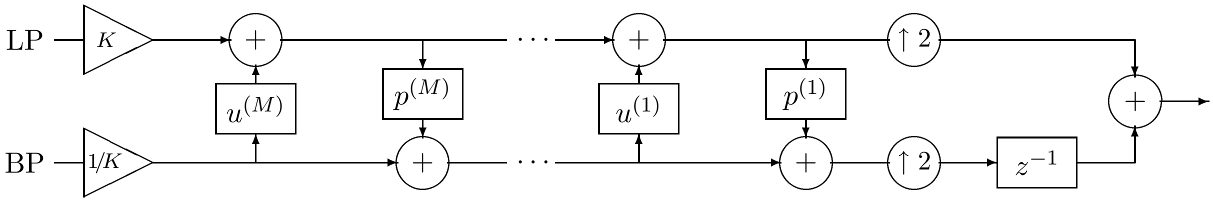

FIGURE 5. *The inverse wavelet transform using lifting: First a scaling, then alternating lifting and dual lifting steps, and finally the inverse Lazy transform. The inverse transform can immediately be derived from the forward by running the scheme backwards and flipping the signs.*

Computing the wavelet transform using lifting steps consists of several stages. The idea is to first compute a trivial wavelet transform (the Lazy wavelet or polyphase transform) and then improve its properties using alternating lifting and dual lifting steps, see Figure 4. The Lazy wavelet only splits the signal into its even and odd indexed samples:

$$\begin{aligned} s_{1,l}^{(0)} &= s_{1,2l} \\ d_{1,l}^{(0)} &= s_{1,2l+1} \,. \end{aligned}$$

A dual lifting step consists of applying a filter to the even samples and subtracting the result from the odd ones:

$$d_{1,l}^{(i)} = d_{1,l}^{(i-1)} - \sum_k p_k^{(i)} \, s_{1,l-k}^{(i-1)} \,.$$

A lifting step does the opposite: applying a filter to the odd samples and subtracting the result from the even samples:

$$s_{1,l}^{(i)} = s_{1,l}^{(i-1)} - \sum_k u_k^{(i)}\, d_{1,l-k}^{(i)}\,.$$

Eventually, after, say M pairs of dual and primal lifting steps, the even samples will become the low pass coefficients while the odd samples become the high pass coefficients, up to a scaling factor K:

$$s_{1,l} = s_{1,l}^{(M)}/K \quad \text{and} \quad d_{1,l} = K\, d_{1,l}^{(M)}\,.$$

As always, we can find the inverse transform by reversing the operations and flipping the signs, see Figure 5. We thus first compute

$$s_{1,l}^{(M)} = K\, s_{1,l} \quad \text{and} \quad d_{1,l}^{(M)} = d_{1,l}/K\,.$$

Then undo the M alternating lifting steps and dual lifting steps:

$$s_{1,l}^{(i-1)} = s_{1,l}^{(i)} + \sum_k u_k^{(i)}\, d_{1,l-k}^{(i)}\,,$$

and

$$d_{1,l}^{(i-1)} = d_{1,l}^{(i)} + \sum_k p_k^{(i)}\, s_{1,l-k}^{(i-1)}\,.$$

Finally retrieve the even and odd samples as

$$\begin{aligned} s_{1,2l} &= s_{1,l}^{(0)} \\ s_{1,2l+1} &= d_{1,l}^{(0)}\,. \end{aligned}$$

The following theorem holds:

Theorem 3.1. *Every wavelet or subband transform with finite filters can be obtained as the Lazy wavelet followed by a finite number of primal and dual lifting steps and a scaling.*

This theorem follows from a factorization well known to both electrical engineers (in linear system theory) as well as algebraists (in algebraic K-theory); [9] gives a self-contained proof as well as many examples coming from wavelet filter banks.. The proof is constructive; for a given wavelet transform, the filters used in the lifting steps can be found using the Euclidean algorithm. The number of lifting steps is bounded by the length of the original filters. It is also shown in [9] how, with at most 3 extra lifting steps, K can always be set to 1.

It is important to point out that the lifting factorization is not unique. For a given wavelet transform, one can often find many quite different factorization. Depending on the application one then chooses the factorization with the smallest M, or the one with K closest to one, or the one which preserves symmetry.

3.4. **Lifting and integer wavelet transforms.** Since we can write every wavelet transform using lifting, it follows that we can build an integer version of *every* wavelet transform. Like in the examples, one can, in each lifting step round-off the result of the filter right before the adding or subtracting. An integer dual lifting step thus becomes:

$$d_{1,l}^{(i)} = d_{1,l}^{(i-1)} - \lfloor \sum_k p_k^{(i)} \, s_{1,l-k}^{(i-1)} + 1/2 \rfloor \,,$$

while an integer primal lifting step is given by

$$s_{1,l}^{(i)} = s_{1,l}^{(i-1)} - \lfloor \sum_k u_k^{(i)} \, d_{1,l-k}^{(i)} + 1/2 \rfloor \,.$$

This obviously results in an integer to integer transform. Because it is written using lifting steps, it is invertible and the inverse again immediately follows by flipping the signs and reversing the operations.

The issue of the scaling factor K remains. Two solutions are possible:

1. Omit the scaling factor K, and keeping in mind that one actually calculates the low pass times K and the band pass divided by K. This approach is similar to the non uniform expansion factor of Section 2.3, with the difference that now the product of the factor for the low pass (K) and the factor for the high pass ($1/K$) is always one. In this case it is preferable to let K be as close to 1 as possible. This can be done using the non-uniqueness of the lifting factorization [9].
2. With three extra lifting steps K can be made equal to one. One can then build the integer version of those extra lifting steps. This comes down to shifting one coefficient b bits to the left and the other b bits to the right. The b bits that get lost in the shift to the right end up in the b locations freed up by the shift to the left.

This leads to the following pseudo-code implementation of an invertible integer wavelet transform using lifting:

$s_{1,l}^{(0)} := s_{0,2l}$

$d_{1,l}^{(0)} := s_{0,2l+1}$

`for i = 1:(1):M`

$\quad \forall l : d_{1,l}^{(i)} := d_{1,l}^{(i-1)} - \lfloor \sum_k p_k^{(i)} \, s_{1,l-k}^{(i-1)} + 1/2 \rfloor$

$\quad \forall l : s_{1,l}^{(i)} := s_{1,l}^{(i-1)} - \lfloor \sum_k u_k^{(i)} \, d_{1,l-k}^{(i)} + 1/2 \rfloor$

```
end
```

The inverse transform is given by:

for i = M:(-1):1

$\forall l : s_{1,l}^{(i-1)} := s_{1,l}^{(i)} + \lfloor \sum_k u_k^{(i)}\, d_{1,l-k}^{(i)} + 1/2 \rfloor$

$\forall l : d_{1,l}^{(i-1)} := d_{1,l}^{(i)} + \lfloor \sum_k p_k^{(i)}\, s_{1,l-k}^{(i-1)} + 1/2 \rfloor$

end

$s_{0,2l+1} := d_{1,l}^{(0)}$

$s_{0,2l} := s_{1,l}^{(0)}$

3.5. **Links with the approach in Section 2.** There are two ways in which the scheme from Section 3.4 can be linked to the approach in Section 2. First of all, by writing the filter bank as a succession of lifting steps, we have in fact reduced everything to a situation similar to LTF: every simple lifting step consists in "correcting" one entry ($d_{1,l}^{(i)}$ or $s_{1,l}^{(i)}$) by adding to it a linear combination of previously encountered entries. In LTF, the entry to be corrected is nudged away from its integer value by subtracting from it the fractional part of what will be added by the filter, but the filtering itself is then untouched. We choose to keep the integer value of the entry to be corrected, but to change the filter step, adding only the integer part of the correction. This amounts of course to the same thing.

A second link is the realization that we have in fact constructed in Section 3.4 particular sets in $\mathbf{R}^{2L}$ analogous to the $\mathbf{H}^{-1}\,\Gamma + n$ considered in Section 2. To see this, note that if the lifting scheme, with the truncation to integers of the "lifted" part (but not the first term) at every step, is applied to real entries instead of integers, the final outcome will also consist of real numbers ($s_{1,l}^{(M)}$ or $d_{1,l}^{(M)}$). For each choice $\mathbf{n} = (n_{s,l}, n_{d,l})_{l=1,\dots L} \in \mathbf{Z}^{2L}$ we can define the region $\Sigma_{\mathbf{n}}$ for which $\lfloor s_{1,l}^{(M)} \rfloor = n_{s,l}$, $\lfloor d_{1,l}^{(M)} \rfloor = n_{d,l}$. These regions correspond to the $\mathbf{H}^{-1}\,\Gamma + \mathbf{n}$ that we considered in Section 2.

The regions $\Sigma_{\mathbf{n}}$ are quite complex, however, and do not have the structure proposed in Section 2.2 or 2.3. Applied to the same data, they will also give different results. The different $\Sigma_{\mathbf{n}}$ need not be even translates of $\Sigma_{\mathbf{0}}$. To illustrate this, let us look at the filters for the integer TS transform as in (3.7). For simplicity, we take wraparound at the edges. Then the $2L$ entries $s_{0,0}, \dots, s_{0,2L-1}$ get transformed into $s_{1,l}, d_{1,l}$ with $l = 0, \dots, L-1$, as given by (3.7), with the boundary cases given by

$$d_{1,0} = d_{1,0}^{(1)} + \lfloor 1/4\,(s_{1,L-1} - s_{1,1}) + 1/2 \rfloor$$

$$d_{1,L-1} = d_{1,2L-2}^{(1)} + \lfloor 1/4\,(s_{1,L-2} - s_{1,0}) + 1/2 \rfloor\,.$$

The region $\Sigma_{\mathbf{0}}$ is now

$$\Sigma_{\mathbf{0}} = \{(s_{0,0}, \dots, s_{0,2L-1})\,;\, 0 \leqslant s_{1,l} < 1,\, 0 \leqslant d_{1,l} < 1,\, l = 0, \dots, L-1\}\,.$$

Since $\lfloor 1/4(a-b) + 1/2 \rfloor = 0$ if $\lfloor a \rfloor = \lfloor b \rfloor = 0$, we find that $\Sigma_{\mathbf{0}}$ is exactly

$$\Sigma_{\mathbf{0}} = \{(s_{0,0}, \dots, s_{0,2L-1})\,;\, 0 \leqslant s_{0,2l+1} - s_{0,2l} < 1,\, 0 \leqslant s_{0,2l+1} + s_{0,2l} < 2, l = 0, \dots, L-1\},$$

or a product of L tiles identical (up to a shift) to those we found for the non-uniform expansion Haar case in Section 2.3. Define now $2_k = (s_{1,l},\, d_{1,l})_{l=1,\dots,L-1}$, with $d_{1,l} = 0$ for all l, $s_{1,l} = 2\,\delta_{l,k}$. Then

$$\begin{aligned}\Sigma_{2_k} =& \{(s_{0,0}, \dots, s_{0,2L-1})\,;\, 2 \leqslant s_{1,k} < 3\\ & 0 \leqslant s_{1,l} < 1,\, l = 0, \dots, L-1,\, l \neq k,\\ & 0 \leqslant d_{1,l} < 1,\, l = 0, \dots, L-1\}\,.\end{aligned}$$

One easily checks that the integer in Σ_{2_k} is $\mathbf{m}_k = (0, \dots, 0, 2, 2, 1, 0, \dots, 0)$, i.e., $s_{0,2k} = s_{0,2k+1} = 2$, $s_{0,2k+2} = 1$, all other $s_{0,l} = 0$. Take now $s_0 = (0, \dots, 0, 2.9, 2.9, 1, 0, \dots, 0) \in \Sigma_{\mathbf{0}} + \mathbf{m}_k$. Then $d^{(1)}_{1,l} = -\delta_{l,k+1}$, $s_{1,l} = 2.9\,\delta_{l,k}$, $d_{1,l} = -\delta_{l,k-1}$, so that $s_0 \notin \Sigma_{2_k}$.

Note that in this particular case $\Sigma_{\mathbf{0}}$ happens to be a fundamental region for $\mathbf{Z}^{2L}$; in general (and in particular, in some of the more complex examples that we implemented) even this is not necessarily true. In most cases, especially when even more truncation steps are considered, $\Sigma_{\mathbf{0}}$ can also not be written as a product of two-dimensional regions.

4. Examples

4.1. **Interpolating transforms.** The first group of examples are instances of a family of symmetric, biorthogonal wavelet transforms built from the interpolating Deslauriers-Dubuc scaling functions [32]. We present examples of the form $(N, \tilde{N})$ where N is the number of vanishing moments of the analyzing high pass filter, while $\tilde{N}$ is the number of vanishing moments of the synthesizing high pass filter. Using this notation the S-transform (3.3) can be referred to as a $(1,1)$ transform. For all the interpolating examples we take $K = 1$.

- (2,2) interpolating transform

$$\begin{aligned} d_{1,l} &= s_{0,2l+1} - \lfloor 1/2\,(s_{0,2l} + s_{0,2l+2}) + 1/2 \rfloor\\ s_{1,l} &= s_{0,2l} + \lfloor 1/4\,(d_{1,l-1} + d_{1,l}) + 1/2 \rfloor\,. \end{aligned} \tag{4.1}$$

  Note that, up to a shift where $s_{1,l}$ here corresponds to $s_{1,l+1}$ before, this is exactly the filter pair given in (2.16).

- (4,2) interpolating transform

$$\begin{aligned} d_{1,l} &= s_{0,2l+1} - \lfloor 9/16\,(s_{0,2l} + s_{0,2l+2}) - 1/16\,(s_{0,2l-2} + s_{0,2l+4}) + 1/2 \rfloor \\ s_{1,l} &= s_{0,2l} + \lfloor 1/4\,(d_{1,l-1} + d_{1,l}) + 1/2 \rfloor\,. \end{aligned} \tag{4.2}$$

- (2,4) interpolating transform

$$\begin{aligned} d_{1,l} &= s_{0,2l+1} - \lfloor 1/2\,(s_{0,2l} + s_{0,2l+2}) + 1/2 \rfloor \\ s_{1,l} &= s_{0,2l} + \lfloor 19/64\,(d_{1,l-1} + d_{1,l}) - 3/64\,(d_{1,l-2} + d_{1,l+1}) + 1/2 \rfloor\,. \end{aligned} \tag{4.3}$$

- (6,2) interpolating transform

$$\begin{aligned} d_{1,l} &= s_{0,2l+1} - \lfloor 75/128\,(s_{0,2l} + s_{0,2l+2}) - 25/256\,(s_{0,2l-2} + s_{0,2l+4}) \\ &\qquad + 3/256\,(s_{0,2l-4} + s_{0,2l+6}) + 1/2 \rfloor \\ s_{1,l} &= s_{0,2l} + \lfloor 1/4\,(d_{1,l-1} + d_{1,l}) + 1/2 \rfloor\,. \end{aligned} \tag{4.4}$$

- (4,4) interpolating transform

$$\begin{aligned} d_{1,l} &= s_{0,2l+1} - \lfloor 9/16\,(s_{0,2l} + s_{0,2l+2}) - 1/16\,(s_{0,2l-2} + s_{0,2l+4}) + 1/2 \rfloor \\ s_{1,l} &= s_{0,2l} + \lfloor 9/32\,(d_{1,l-1} + d_{1,l}) - 1/32\,(d_{1,l-2} + d_{1,l+1}) + 1/2 \rfloor\,. \end{aligned} \tag{4.5}$$

4.2. **(2+2,2) transform.** Inspired by the S+P transform, we use one extra lifting step to build the earlier $(2,2)$ into a transform with 4 vanishing moments of the high pass analyzing filter. The resulting transform is different from the earlier $(4,2)$ transform and therefore is called the $(2+2,2)$ transform. The idea is to first compute a $(2,2)$ yielding low pass samples $s_{1,l}$ and preliminary detail or high pass samples $d^{(1)}_{1,l}$, and then use the $s_{1,l}$ combined with $d_{1,l+n}$ $(n > 0)$ to compute $d'_{1,l}$ as a prediction for $d^{(1)}_{1,l}$. The final detail sample then is $d^{(1)}_{1,l} - d'_{1,l}$. We suggest a scheme of the form:

$$\begin{aligned} d^{(1)}_{1,l} &= s_{0,2l+1} - \lfloor 1/2(s_{0,2l} + s_{0,2l+2}) + 1/2 \rfloor \\ s_{1,l} &= s_{0,2l} + \lfloor 1/4(d^{(1)}_{1,l-1} + d^{(1)}_{1,l}) + 1/2 \rfloor \\ d_{1,l} &= d^{(1)}_{1,l} - \lfloor \alpha\,(-1/2\,s_{1,l-1} + s_{1,l} - 1/2\,s_{1,l+1}) \\ &\qquad + \beta\,(-1/2\,s_{1,l} + s_{1,l+1} - 1/2\,s_{1,l+2}) + \gamma\, d^{(1)}_{1,l+1} + 1/2 \rfloor\,. \end{aligned} \tag{4.6}$$

Without truncation, we want the scheme to have 4 vanishing moments. This leads to the following conditions:

$$\begin{aligned} 8\,\beta + 3\,\gamma &= 1 \\ 4\,\alpha + 4\,\beta + \gamma &= 1\,. \end{aligned}$$

Special cases are: (1) $\alpha = 1/6, \beta = 0, \gamma = 1/3$, (2) $\alpha = 1/8, \beta = 1/8, \gamma = 0$, and (3) $\alpha = 1/4, \beta = -1/4, \gamma = 1$. In our experiments we found that (2) works considerably better than (1) and (3), and this is the case we use when we refer to $(2+2,2)$ in Section 5.

4.3. **D4 orthogonal transform.** The transform follows from the lifting factorization in [9].

$$\begin{aligned} d_l^{(1)} &= s_{0,2l+1} - \lfloor \sqrt{3}\, s_{0,2l} + 1/2 \rfloor \\ s_{1,l} &= s_{0,2l} + \lfloor \sqrt{3}/4\, d_{1,l}^{(1)} + (\sqrt{3}-2)/4\, d_{1,l-1}^{(1)} + 1/2 \rfloor \\ d_{1,l} &= d_{1,l}^{(1)} + s_{1,l+1}\,. \end{aligned}$$

Here $K = (\sqrt{3}+1)/\sqrt{2} \approx 1.577$.

4.4. $(9-7)$ **symmetric biorthogonal transform.** We consider the popular $(9-7)$ filter pair. The analysis low pass filter has 9 coefficients, while the analysis high pass filter has 7 coefficients. Both analysis and synthesis high pass filters have 4 vanishing moments [8]. The integer transform again follows from the lifting factorization in [9].

$$\begin{aligned} d_{1,l}^{(1)} &= s_{0,2l+1} + \lfloor \alpha\,(s_{1,2l} + s_{1,2l+2}) + 1/2 \rfloor \\ s_{1,l}^{(1)} &= s_{0,2l} + \lfloor \beta\,(d_{1,l}^{(1)} + d_{1,l-1}^{(1)}) + 1/2 \rfloor \\ d_{1,l} &= d_{1,l}^{(1)} + \lfloor \gamma\,(s_{1,l}^{(1)} + s_{1,l+1}^{(1)}) + 1/2 \rfloor \\ s_{1,l} &= s_{1,l}^{(1)} + \lfloor \delta\,(d_{1,l} + d_{1,l-1}) + 1/2 \rfloor\,. \end{aligned}$$

The constants are given by

$$\begin{aligned} \alpha &\approx -1.586134342 \\ \beta &\approx -0.05298011854 \\ \gamma &\approx 0.8829110762 \\ \delta &\approx 0.4435068522 \\ K &\approx 1.149604398\,. \end{aligned}$$

5. Evaluation of integer to integer wavelet transforms

In this section, we evaluate the effectiveness of the various wavelet filters described in the previous section for lossless compression of digital images. For the evaluation, we use selected images from the set of standard ISO test images [1]. In this set of test images, there are natural images, computer-generated images, compound images (mixed texts and natural images, e.g., “cmpnd1” and “cmpnd2”; “us” is an ultrasound image with text on it) and different types of medical images ("x_ray", "cr", "ct", and "mri"). Separable two dimensional wavelet transforms are taken of the images.

The effectiveness for lossless compression is measured using the entropy $H(X)$, given by

$$H(X) = -\sum_i P(s_i) \log P(s_i), \tag{5.1}$$

where X is a discrete random variable taking on values s_i with probability $P(s_i)$. For an image $X(,)$ of size $N \times M$, we take the probability $P(s_i)$ to be the normalized count of value s_i, i.e.,

$$P(s_i) = \frac{|(l,m) : X(l,m) = s_i|}{NM}. \tag{5.2}$$

We further take into account the fact that the statistics in different quadrants of a wavelet-transformed image are different, and compute the weighted mean of the entropies in each quadrant of the transformed image. For example, in the two-scale decomposition shown in Figure 6, the weighted entropy (in bits/pixel) is computed as

$$\frac{1}{16}\left(H(C_2) + H(D_{2,v}) + H(D_{2,h}) + H(D_{2,d})\right) + \frac{1}{4}\left(H(D_{1,v}) + H(D_{1,h}) + H(D_{1,d})\right).$$

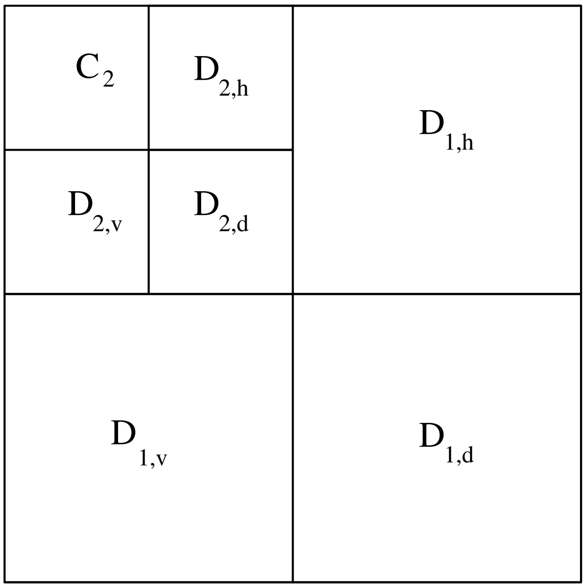

FIGURE 6. *A two-scale wavelet decomposition.*

In the evaluation, we decompose each image into a maximum of five scales, the length and width permitting. The weighted entropies are tabulated in Table 1.

Several observations can be made:

Images	(1,1)	(3,1)	(2,2)	S+P	(4,2)	(2+2,2)	$D4$	9-7
air2	5.13	4.91	4.82	4.81	4.77	**4.76**	5.09	4.98
bike	4.36	4.28	4.19	**4.18**	4.20	4.21	4.37	4.31
cafe	5.69	5.54	**5.41**	5.42	**5.41**	**5.41**	5.63	5.51
cats	3.69	3.53	3.47	3.43	**3.42**	**3.42**	3.60	3.47
cmpnd1	**2.25**	2.84	2.79	2.97	3.31	3.36	3.04	3.45
cmpnd2	**2.41**	2.96	2.79	3.01	3.28	3.33	3.12	3.42
cr	5.40	5.25	**5.20**	5.24	5.22	5.22	5.28	5.22
ct	5.54	4.63	4.50	4.30	**4.15**	4.16	4.96	4.36
faxballs	1.61	1.31	**1.08**	1.41	1.36	1.17	1.54	1.97
finger	6.24	5.71	5.49	5.48	**5.35**	**5.35**	5.85	5.45
gold	4.27	4.10	4.05	4.08	4.04	**4.03**	4.19	4.14
graphic	3.18	2.82	2.60	2.67	**2.56**	**2.56**	3.08	3.00
hotel	4.30	4.18	**4.03**	4.10	4.06	4.04	4.25	4.18
mri	6.59	6.16	6.02	**5.90**	5.91	5.91	6.26	5.97
tools	5.84	5.80	**5.69**	5.73	5.73	5.72	5.88	5.81
us	**3.64**	3.79	3.69	3.79	3.87	3.85	3.95	4.26
water	2.46	2.45	**2.42**	2.47	2.45	2.44	2.46	2.50
woman	4.87	4.67	4.57	4.54	**4.53**	4.54	4.78	4.64
x_ray	6.42	6.13	**6.06**	6.09	**6.06**	**6.06**	6.18	6.08

TABLE 1. *Weighted entropies of transformed images.*

1. There is no filter that consistently performs better than all the other filters on all the test images.
2. Wavelet filters with more analyzing vanishing moments generally perform well with natural and smooth images and not so with images with a lot of edges and high frequency components, such as with compound images.
3. On the other hand, a low order filter like (1,1) (S-transform) generally performs the worst, especially with natural images. It does a poor job in decorrelating the images. However, it performs significantly better on compound images (e.g., "cmpnd1" and "cmpnd2") than other higher order filters.

4. Wavelet filters (4,2) and (2+2,2) have similar performances and generally perform better than other filters evaluated. For images "finger" and "ct", they perform significantly better than other filters.
5. The filter (3,1) generally performs worse than filters with comparable number of analyzing vanishing moments.
6. It is interesting to note that even though the S+P has 2 analyzing vanishing moments, it performs better than the (3,1) which has 3 and has comparable performances with those with 4. This suggests that there are other factors besides the number of analyzing vanishing moments which affect compression efficiency.
7. The $(9-7)$, which is most popularly used for lossy compression of images and which has 4 analyzing vanishing moments, generally does not perform as well as the (4,2) and (2+2,2), which have the same number of analyzing vanishing moments.

We further evaluate the wavelet filters by attaching an entropy coder to compute the *actual* bit rate. We use the entropy coder of Said and Pearlman [25]. In each quadrant, the wavelet coefficients are visited in a scanline order; for each coefficient, a context C_j is derived based on the values of its adjacent coefficients which have already been visited and its parent coefficient. A total of 25 different contexts are used. The statistics of the context model is adaptively updated as more pixels are visited and the model is reset after each quadrant is visited. An arithmetic coder is then used to code the coefficients according to their contexts. The resulting bit rate will be lower bounded by the conditional entropy,

$$H(X|C) = -\sum_{i,j} P(s_i, c_j) \log P(s_i|c_j), \tag{5.3}$$

where $P(s_i, c_j)$ is the joint probability of value s_i and context c_j and $P(s_i|c_j)$ is the conditional probability of value s_i given context c_j. The results are tabulated in Table 2. The numbers are derived from compressed file sizes. Given that $H(X|C) \leq H(X)$, conditioning allows us to achieve actual bit rate lower than the entropy $H(X)$ computed in Table 1.

The relative strength of each filter over the others in actual coding is similar to that computed using the entropies. As previously observed, wavelet filters (4,2) and (2+2,2) generally outperforms other filters for natural and smooth images. Also, filters with more analyzing vanishing moments perform poorly with compound images. As can be seen from the two tables, numbers computed from entropies provide a good indication of the actual performance of a wavelet filter. Alternatively, with a context model in mind, one can compute the conditional entropies for a better estimate of the actual bit rate.

Images	(3,1)	(2,2)	S+P	(4,2)	(2+2,2)	$D4$	9-7
air2	4.34	4.29	4.28	**4.24**	**4.24**	4.60	4.52
bike	3.83	3.78	**3.76**	3.78	3.79	3.94	3.88
cafe	5.07	4.98	4.98	**4.95**	4.98	5.17	5.07
cats	2.60	2.56	2.53	**2.52**	**2.52**	2.67	2.57
cmpnd1	2.34	**2.31**	2.38	2.59	2.62	2.40	2.64
cmpnd2	2.46	**2.33**	2.45	2.58	2.61	2.49	2.64
cr	5.21	**5.17**	5.22	5.19	5.19	5.25	5.20
ct	3.96	3.92	3.70	**3.61**	3.65	4.30	3.80
faxballs	1.12	**1.02**	1.20	1.17	1.07	1.51	1.59
finger	5.73	5.56	5.51	**5.41**	5.42	5.90	5.50
gold	3.99	3.94	3.99	3.95	**3.94**	4.11	4.09
graphic	3.05	2.91	2.93	**2.87**	2.89	3.34	3.29
hotel	3.97	**3.90**	3.95	3.91	**3.90**	4.10	4.07
mri	5.99	5.90	**5.77**	5.80	5.81	6.13	5.89
tools	5.59	**5.49**	5.53	5.52	5.51	5.65	5.60
water	1.83	**1.79**	1.85	1.81	1.80	1.85	1.88
woman	4.24	4.17	4.15	**4.14**	4.15	4.37	4.28
us	3.17	**3.13**	3.16	3.23	3.22	3.38	3.53
x_ray	5.99	**5.92**	5.94	**5.92**	**5.92**	6.05	5.95

TABLE 2. *Exact bit rate of test images.*

We also observe that the use of simple context modeling of the wavelet coefficients as in [25] cannot provide a significant enough gain in actual coding when the performance of the wavelet filter is poor in the first place. This points to the importance of using "good" wavelet filters and necessitates the search for such filters.

The classical approach to lossless compression is decomposed into two steps: spatial decorrelation and entropy coding of the decorrelated signals. The decorrelating steps have always been performed in the spatial domains and they involve some form of non adaptive or adaptive prediction of a pixel values based on previously visited pixels (see [23] and the references therein for comparisons of state-of-the-art spatial-domain predication techniques). In fact, we can view wavelet transforms that use two lifting steps such as those built from Deslauriers-Dubuc scaling functions as decorrelation steps. The high pass coefficients are the decorrelated components.

The lifting step computes the difference between a true coefficient and its prediction and has the following form:

$$d_{j+1,l} = s_{j,2l+1} - \lfloor f(s_{j,2l-2L} \cdots, s_{j,2l-2}, s_{j,2l}, s_{j,2l+2}, s_{j,2l+4}, \cdots, s_{j,2l+2L}) \rfloor. \tag{5.4}$$

We are thus predicting a pixel at an odd location $2l+1$ using pixels at even locations from both sides of $2l+1$. This is in contrast to the spatial domain approach to prediction, which has the following one-sided form:

$$d_j = s_j - \lfloor g(s_{j-J}, \cdots, s_{j-3}, s_{j-2}, s_{j-1}) \rfloor. \tag{5.5}$$

The price to pay for prediction based on (5.4) is to retain knowledge of the remaining pixels at the even locations. Similar prediction steps can be performed on these remaining pixels, but because the distances between adjacent pixels are larger than before, correlations between adjacent pixels tend to be lower. The dual lifting step in generating the $s_{j+1,l}$ smoothes this set of remaining pixels before the lifting step on the coarser level is performed.

The big advantage of using wavelet transform to represent images is multiresolution representation, which spatial-domain prediction techniques based on (5.5) cannot offer. Using wavelet transforms that map integers to integers permits lossless representation of the image pixels and easily allows the transmission of lower resolution versions first, followed by transmissions of successive details. Such a mode of transmission is especially valuable in scenarios where bandwidth is limited, image sizes are large and lossy compression is not desirable. Examples are transmission of 2-D and 3-D medical images for tele-medicine applications and transmission of satellite images down to earth. Each of these images is typically several megabytes in size.

Valuable lessons could also be learned from the recent advances in spatial-domain prediction techniques [23]. Adaptation of prediction methods is made based on local statistics of the pixels. In the wavelet transform approach to decorrelating an image, we could also use an adaptive scheme in deciding the use of wavelet filters on a pixel by pixel basis. As we have seen, there is not a wavelet filter that performs uniformly better than the others. Thus, the activities in some small neighborhood should be used in determining the type of filters to use. The use of adaptive wavelet filters in lossless and multiresolution representation of images warrants further investigation.

Acknowledgements. Ingrid Daubechies would like to thank NSF (grant DMS-9401785), AFOSR (grant F49620-95-1-0290), ONR (grant N00014-96-1-0367) as well as Lucent Technologies, Bell Laboratories for partial support while conducting the research for this paper. I. D. and A. R. Calderbank would like to thank the Institut des Hautes Etudes Scientifiques in Bures-sur-Yvette, France, for its hospitality and support in June '95 when part of this work was done.

Wim Sweldens is on leave as Senior Research Assistant of the Belgian Fund of Scientific Research (NFWO). Boon-Lock Yeo thanks the support of the IBM Graduate Fellowship in spring, AT&T Bell Laboratories in summer, and the Wallace Memorial Fellowship in Engineering from Princeton University in fall 1995.

Note. After finishing this work, we learned that a construction similar to the one presented in Section 3 was obtained independently by Dewitte-Cornelis and Chao-Fischer [11, 5].

The Chao-Fischer approach has the extra feature that it uses modular arithmetic to eliminate all increase in dynamic range; thus if the original image uses 8 bits per pixel, all the wavelet coefficients uses 8 bits as well. The disadvantage is that large wavelet coefficients can become small due to the modular arithmetic, and vice versa. In our experiments, this wraparound disadvantage tends to cancel out the advantage of no bit growth; nevertheless it is useful for certain implementations.

References

[1] ISO/IEC JTC 1/SC29/WG1. Call for contributions - lossless compression of continuous-tone still pictures. ISO Working Document ISO/IEC JTC1/SC29/WG1 N41, March 1994.

[2] C. M. Brislawn, J. N. Bradley, R. J. Onyshczak, and T. Hopper. The FBI compression standard for digitized fingerprint images. In A. G. Tescher, editor, *Applications of Digital Image Processing XIX*. Proc. SPIE 2847, 1996.

[3] A. A. M. L. Bruekens and A. W. M. van den Enden. New networks for perfect inversion and perfect reconstruction. *IEEE J. Selected Areas Commun.*, 10(1), 1992.

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

[5] H. Chao and P. Fisher. An approach of fast integer reversible wavelet transform for image compression. Preprint.

[6] A. Cohen, I. Daubechies, and J. Feauveau. Bi-orthogonal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, 45:485–560, 1992.

[7] W. Dahmen and C. A. Micchelli. Banded matrices with banded inverses II: Locally finite decompositions of spline spaces. *Constr. Approx.*, 9(2-3):263–281, 1993.

[8] I. Daubechies. *Ten Lectures on Wavelets*. CBMS-NSF Regional Conf. Series in Appl. Math., Vol. 61. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992.

[9] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. Technical report, Bell Laboratories, Lucent Technologies, 1996.

[10] R. A. DeVore, B. Jawerth, and B. J. Lucier. Image compression through wavelet transform coding. *IEEE Trans. Inform. Theory*, 38(2):719–746, 1992.

[11] S. Dewitte and J. Cornelis. Lossless integer wavelet transform. Technical Report IRIS-TR-0041, Royal Meteorological Institute Belgium, 1996.

[12] D. L. Donoho. Smooth wavelet decompositions with blocky coefficient kernels. In [27], pages 259–308.

[13] D. L. Donoho. Interpolating wavelet transforms. Preprint, Department of Statistics, Stanford University, 1992.

[14] J. Gros. Binary friendly wavelets. Preprint, Department of Electrical Engineering and Computer Sciences, UC Berkeley, 1995.

[15] A. Harten. Multiresolution representation of data: A general framework. *SIAM J. Numer. Anal.*, 33(3):1205–1256, 1996.

[16] V. K. Heer and H-E Reinfelder. A comparison of reversible methods for data compresssion. In *Medical Imaging IV*, pages 354–365. Proc. SPIE 1233, 1990.

[17] J. Hong. *Discrete Fourier, Hartley and cosine transforms in signal processing*. PhD thesis, Columbia Univerisity, Department of Electrical Engineering, 1993.

[18] C. P. Johnston. The lifting scheme and finite-precision-error-free filter banks. In M. Unser, A. Aldroubi, and A. F. Laine, editors, *Wavelet Applications in Signal and Image Processing IV*, pages 307–316. Proc. SPIE 2825, 1996.

[19] T. A. C. M. Kalker and I. Shah. Ladder Structures for multidimensional linear phase perfect reconstruction filter banks and wavelets. In *Proceedings of the SPIE Conference on Visual Communications and Image Processing* (Boston), pages 12–20, 1992.

[20] C. W. Kim, R. Ansari, and A. E. Cetin. A class of linear-phase regular biorthogonal wavelets. In *Proc. IEEE Int. Conference on Acoustics, Speech, Signal Processing*, pages 673–676, 1992.

[21] R. Laroia, S. A. Tretter, and N. Farvardin. A simple and effective precoding scheme for noise whitening on intersymbol interference channels. *IEEE Trans. Commun.*, 41:460–463, 1993.

[22] M. Lounsbery, T. D. DeRose, and J. Warren. Multiresolution surfaces of arbitrary topological type. Department of Computer Science and Engineering 93-10-05, University of Washington, October 1993. Updated version available as 93-10-05b, January, 1994.

[23] N. Memon, V. Sippy, and X. Wu. A comparison of prediction schemes proposed for a new lossless image compression standard. In *IEEE International Symposium on Circuits and Systems*, volume 2, pages 309–312, May 1995.

[24] M. Rabbini and P. W. Jones. *Digital Image Compression Techniques*. SPIE Opt. Eng. Press, Bellingham WA, 1991.

[25] A. Said and W. A. Pearlman. An image multiresolution representation for lossless and lossy image compression. *IEEE Trans. Image Process.*, 5(9):1303–1310, 1996.

[26] P. Schröder and W. Sweldens. Spherical wavelets: Efficiently representing functions on the sphere. *Computer Graphics Proceedings, (SIGGRAPH 95)*, pages 161–172, 1995.

[27] L. L. Schumaker and G. Webb, editors. *Recent Advances in Wavelet Analysis*. Academic Press, New York, 1993.

[28] I. Shah and T. A. C. M. Kalker. On Ladder Structures and Linear Phase Conditions for Bi-Orthogonal Filter Banks. In *Proceedings of ICASSP-94*, volume 3, pages 181–184, 1994.

[29] J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. Signal Process.*, 41(12):3445–3462, 1993.

[30] G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley, Cambridge, 1996.

[31] M. D. Swanson and A. H. Tewfix. A binary wavelet decomposition of binary images. *IEEE Trans. Image Process.*, 5(12):1637–1650, 1996.

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

[33] W. Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.*, to appear, 1997.

[34] L. M. G. M. Tolhuizen, I. A. Shah, and T. A. C. M. Kalker. On constructing regular filter banks from domain bounded polynomials. *IEEE Trans. Signal Process.*, 42(2):451–456, 1994.

[35] P. P. Vaidyanathan. *Multirate Systems and Filter Banks.* Prentice Hall, Englewood Cliffs, NJ, 1992.

[36] P. P. Vaidyanathan and P.-Q. Hoang. Lattice structures for optimal design and robust implementation of two-band perfect reconstruction QMF banks. *IEEE Trans. Acoust. Speech Signal Process.*, 36:81–94, 1988.

[37] M. Vetterli and C. Herley. Wavelets and filter banks: Theory and design. *IEEE Trans. Acoust. Speech Signal Process.*, 40(9):2207–2232, 1992.

[38] M. Vetterli and J. Kovačević. *Wavelets and Subband Coding.* Prentice Hall, Englewood Cliffs, NJ, 1995.

[39] A. Zandi, M. Boliek, E. L. Schwartz, and M. J. Gormish. CREW lossless/lossy medical image compression. Technical Report CRC-TR-9526, RICOH California Research Center, 1995.

Appendix A. Proof of Lemma 2.1

We start by reformulating Lemma 2.1 in a slightly different way, introducing explicit notations for all the points involved.

Lemma 2.1. *Let P be a parallelogram that is a fundamental region for $\mathbf{Z}^2$, and let A_1, A_2, A_3 be three points on the boundary of P such that*

- *A'_j, the mirror image of A_j with respect to the center of P, is congruent with A_j modulo $\mathbf{Z}^2$ ($j = 1, 2,$ or 3)*
- *no three points among $A_1, A_2, A_3, A'_1, A'_2, A'_3$ are collinear.*

Let Ω be the hexagon with vertices $A_1, A_2, A_3, A'_1, A'_2, A'_3$. Then any closed parallelogram Q that contains Ω also contains a fundamental region for $\mathbf{Z}^2$.

(Note that Ω is a strict subset of a fundamental region, and is therefore *not* a fundamental region itself.)

Proof. Without loss of generality, we can assume that P is centered around the origin.

Because of the prohibition of collinearity, the six points must be distributed among the four sides of P, with two points each on two opposite sides, and one point each on the remaining two sides. If none of the points A_j, A'_j lies on a vertex of P, then we can assume (by renumbering if necessary) that $A_1, A_2,$ and A_3 are three consecutive corners of the hexagon Ω, lying on three different sides of P; see Figure 7a. (If one of the A_j, A'_j lies on a vertex, as in Figure 7b, then the rest of this proof needs to be changed in a straightforward way; we leave this as an exercise for the reader.)

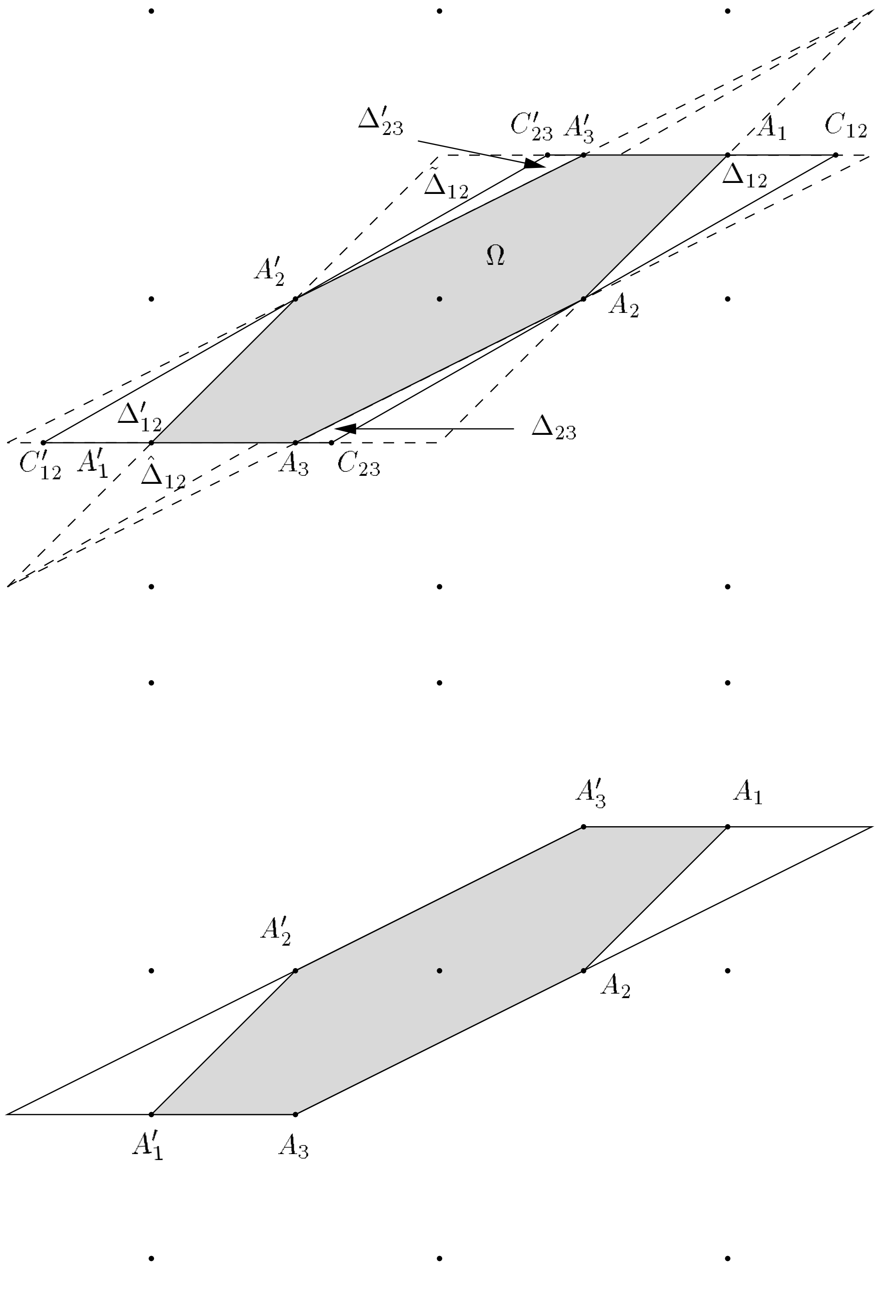

FIGURE 7.

a. *The parallelogram P (with vertices $C_{12}, C_{23}, C'_{12},$ and C'_{23}) and the inscribed hexagon Ω, with associated triangles $\triangle_{12}, \triangle_{23}, \triangle'_{12}$ and $\triangle'_{23}$. All the $\widetilde{\ }$ and $\widehat{\ }$ versions of the triangles are drawn, but only $\widetilde{\triangle}_{12}$ and $\widehat{\triangle}_{12}$ have been labelled.*

b. *A special case where some vertices of Ω coincide with vertices of P.*

Denote by C_{12} the vertex of P that separates A_1 from A_2, and similarly by C_{23} the vertex that separates A_2 from A_3. We denote their opposites, which separate A_1' from A_2', A_2' from A_3', respectively, by C_{12}' and C_{23}'.

The original fundamental region P is the union of the hexagon Ω and four triangles $\triangle_{12}$ (with vertices A_1, A_2, C_{12}), $\triangle_{23}$ (vertices A_2, A_3, C_{23}), $\triangle_{12}'$ (vertices A_1', A_2', C_{12}'), and $\triangle_{13}'$ (vertices A_1', A_3', C_{13}'). For each of the triangles $\triangle_{ij}$ we define the two triangles $\widehat{\triangle}_{ij}$ and $\widetilde{\triangle}_{ij}$ by $\widehat{\triangle}_{ij} = \triangle_{ij} - A_i + A_i'$, $\widetilde{\triangle}_{ij} = \triangle_{ij} - A_j + A_j'$; $\widehat{\triangle}_{ij}'$ and $\widetilde{\triangle}_{ij}'$ are defined analogously. Every one of the four original triangles is congruent $\mathrm{mod}\,\mathbf{Z}^2$ with its $\widehat{\ }$ and $\widetilde{\ }$ versions.

Suppose there exists a parallelogram Q that contains Ω and that does not contain a fundamental region for $\mathbf{Z}^2$. If the parallel lines that bound Q do not touch Ω, we may reduce their separation until they do. Let R be the resulting reduced parallelogram, bounded by the parallel lines L, L' and M, M', each of which is incident with Ω. Let x be a point in P so that R does not contain any point congruent to x mod $\mathbf{Z}^2$. Then $x \in \triangle_{ij}$ or $\triangle_{ij}'$ for some i, j, and there exist points $\widehat{x}, \widetilde{x}$ in the corresponding $\widehat{\triangle}, \widetilde{\triangle}$ that are congruent to x mod $\mathbf{Z}^2$. Observe that if L, L' are parallel lines through opposite vertices j, j' then exactly two of the points $x, \widetilde{x}, \widehat{x}$ lie between L and L'. Similarly for M, M'. Hence one of the points $x, \widetilde{x}, \widehat{x}$ lies in the parallelogram R bounded by L, L' and M, M', which is a contradiction.

□

AT&T-Labs Research, Rm. 2C-363, 600 Mountain Avenue, Murray Hill NJ 07974
rc@research.att.com

Program for Applied and Computational Mathematics, Princeton University, Princeton NJ 08544
ingrid@math.princeton.edu

Lucent Technologies, Bell Laboratories, Rm. 2C-376, 600 Mountain Avenue, Murray Hill NJ 07974
wim@bell-labs.com

IBM Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598
yeo@watson.ibm.com