The Red-Black Wavelet Transform

Geert Uytterhoeven *Adhemar Bultheel*

Report TW 271, December 1997

Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

The Red-Black Wavelet Transform

Geert Uytterhoeven *Adhemar Bultheel*

Report TW 271, December 1997

Department of Computer Science, K.U.Leuven

Abstract

We present a new kind of second generation wavelets on a rectangular grid. These wavelets are constructed using a 2D lifting scheme which is based on a red-black blocking scheme. Compared to classical tensor product wavelets on the same grid, these new wavelets show less anisotropy. The performance of the new wavelets is compared to tensor product wavelets in an image denoising application.

Keywords : wavelets, lifting, image processing, integer transform.
AMS(MOS) Classification : 41A30, 68U10, 94A12, 65D10.

The Red-Black Wavelet Transform

Geert Uytterhoeven[*†]   Adhemar Bultheel[*]

Abstract

We present a new kind of second generation wavelets on a rectangular grid. These wavelets are constructed using a 2D lifting scheme which is based on a red-black blocking scheme. Compared to classical tensor product wavelets on the same grid, these new wavelets show less anisotropy. The performance of the new wavelets is compared to tensor product wavelets in an image denoising application.

1 Introduction

Classical one-dimensional wavelet transforms can be extended to more dimensions using tensor products, yielding a separable multi-dimensional transform. A disadvantage of this technique is the introduction of an anisotropy in the wavelet decomposition. In the two-dimensional case, a tensor product wavelet transform will favor horizontal, vertical and diagonal features of the original data. Other features are not that easily detected.

Non-separable wavelets can provide a solution to this. They allow for the construction of two-dimensional wavelets on e.g. lattices (e.g. on the so called 'quincunx' lattice [7]) or on hexagonal grids [10]. All these are heavily based on Fourier tranforms, just like the classical wavelets.

Second generation wavelets designed using the lifting scheme are another option. The lifting scheme is a generic method to create wavelets on intervals, irregular samplings, meshes, manifolds, ..., without relying on the Fourier transform. As a bonus the classical one-dimensional wavelets can be constructed using lifting, too.

In this paper, we present a new kind of second generation wavelets on a rectangular grid — more specifically, on a quincunx lattice — constructed using the lifting scheme. The performance of the new wavelets is compared to tensor product wavelets in an image denoising application.

2 The Lifting Scheme

2.1 Introduction

The lifting scheme is an algorithm to calculate wavelet transforms in an efficient way. It found its roots in a method to improve a given wavelet transform to obtain some specific properties. Later it was extended to a generic method to create so-called 'Second Generation' wavelets. In de 1D case, these are wavelets that need not be obtained as dilates and translates of one function. They are much

[*] Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Heverlee, Belgium

[†] `Geert.Uytterhoeven@cs.kuleuven.ac.be`, `http://www.cs.kuleuven.ac.be/~geert/`

more flexible and can be used to define a wavelet basis on an interval or on an irregular grid, or even on a sphere. The lifting scheme can also be used to introduce wavelets without using the concept of Fourier transform. Several introductions to the lifting scheme are available [13, 11, 9, 12].

Second generation wavelets are more general in the sense that all the classical wavelets can be generated by the lifting scheme. In fact, the decomposition of a classical wavelet filter into lifting steps can be easily obtained via the Euclidean algorithm [4].

In this paper, we design a new family of second generation wavelets on a rectangular grid.

2.2 Predict and Update

The wavelet transform of a 1D signal is a multiresolution representation of that signal where the wavelets are the basis functions which at each resolution level give a highly decorrelated representation.

Thus at each level, the signal is split into a high pass and a low pass part and the low pass part is split again etc. These high pass and low pass parts are obtained by applying corresponding wavelet filters. In general these filters are coupled if certain conditions are to be fulfilled, like for example perfect reconstruction.

The lifting scheme is an efficient implementation of these filtering operations at each level when computing a discrete wavelet transform. So suppose that the low resolution part of a signal at level $j+1$ is given and that it consists of a data set which we represent by λ_{j+1}. This set is transformed into two other sets at level j: the low resolution part λ_j and the high resolution part γ_j. This is obtained first by just splitting the data set λ_{j+1} into two separate data subsets λ_j and γ_j. Traditionally this is done by separating the set of even samples and the set of odd samples. Such a splitting is sometimes referred to as the *lazy wavelet transform*. Doing just this of course does not improve our representation of the signal. Therefore, the next step is to recombine these two sets in several subsequent lifting steps which decorrelate the two signals.

Lifting steps usually come in pairs of a primal and a dual lifting step. A *dual lifting* step can be seen as a prediction: the data γ_j are 'predicted' from the data in the subset λ_j. When the signals are still highly correlated, then such a prediction will usually be very good, and thus we do not have to keep this information in both signals. That is why we can keep γ_j and store only the part of γ_j that is not predictable (the prediction error). Thus γ_j is replaced by $\gamma_j - \mathcal{P}(\lambda_j)$ where $\mathcal{P}$ represents the prediction operator. This is the real decorrelating step.

However, the new representation has lost certain basic properties, which one usually wants to keep, like for example the mean value of the signal. To restore this property, one needs a *primal lifting* step, whereby the set λ_j is updated with data computed from the (new) subset γ_j. Thus λ_j is replaced by $\lambda_j + \mathcal{U}(\gamma_j)$ with $\mathcal{U}$ some updating operator.

In general, several such lifting steps can be applied in sequence to go from level $j+1$ to level j.

The principles that were just explained become very clear in the case of the simple CFD(2, 2) transform that we give below in Section 2.5. The odd samples are predicted by linear interpolation of the neighbouring even samples and are therefore replaced by the interpolation error. The even samples are updated to preserve the mean value of the signal.

To recapitulate, let us consider a simple lifting scheme with only one pair of lifting steps to go from level $j+1$ to level j.

Splitting (*lazy wavelet transform*) Partition the data set λ_{j+1} into two distinct data sets λ_j and γ_j.

Prediction (*dual lifting*) Predict the data in the set γ_j by the data set λ_j.

$$\gamma_j \leftarrow \gamma_j - \mathcal{P}(\lambda_j).$$

Update (*primal lifting*) Update the data in the set λ_j by the data in set γ_j.

$$\lambda_j \leftarrow \lambda_j + \mathcal{U}(\gamma_j).$$

These steps can be repeated by iteration on the λ_j, creating a multi-level transform or multi-resolution decomposition.

2.3 The Inverse Transform

One of the great advantages of the lifting scheme realization of a wavelet transform is that it decomposes the wavelet filters into extremely simple elementary steps, and each of these steps is very easily invertible. As a result, the inverse transform can always be obtained immediately from the forward transform. The inversion rules are obvious: revert the order of the operations, invert the signs in the lifting steps, and replace the splitting step by a merging step. Thus, inverting the three step procedure above results in

Inverse update

$$\lambda_j \leftarrow \lambda_j - \mathcal{U}(\gamma_j),$$

Inverse prediction

$$\gamma_j \leftarrow \gamma_j + \mathcal{P}(\lambda_j),$$

Merge

$$\lambda_{j+1} \leftarrow \lambda_j + \gamma_j.$$

2.4 Integer Transforms

In practice, discrete signals are represented by integers. Doing the filtering operations on these numbers however will transform them in rational or real numbers because the filter coefficients need not be integers. To obtain an efficient implementation of the discrete wavelet transform, it is of great practical importance that the wavelet transform is represented by a set of integers as well (possibly up to some scaling factor). Of course, this can always be obtained by scaling and rounding the real numbers, but this rounding process will loose information, and after rounding, the original signal can not be reconstructed from its transform without an error.

Here is another advantage of the lifting scheme that can be used, because a discrete wavelet transform computation using the lifting scheme can easily be converted to a transform that maps integers to integers [2, 15].

Since also the most elementary lifting steps involve for example divisions by 2, we obtain in general prediction values $\mathcal{P}(\lambda_j)$ and update values $\mathcal{U}(\gamma_j)$ which are not integer. We shall round these numbers to integers (for example the nearest integer) and indicate this operation by square braces. Thus, we actually compute rounded values:

$$\begin{aligned}\gamma_j &\leftarrow \gamma_j - [\mathcal{P}(\lambda_j)]\,,\\ \lambda_j &\leftarrow \lambda_j + [\mathcal{U}(\gamma_j)]\end{aligned}$$

To see that perfect reconstruction is always possible, we note the following. The last computation in a forward transform step is to compute $\lambda_j \leftarrow \lambda_j + [\mathcal{U}(\gamma_j)]$. Thus the first computation in the inverse transform step is to recompute $[\mathcal{U}(\gamma_j)]$, and this will give exactly the same result as before if γ_j has been perfectly reconstructed so far. Therefore $\lambda_j \leftarrow \lambda_j - [\mathcal{U}(\gamma_j)]$ reconstructs exactly the original λ_j. Thus also $[\mathcal{P}(\lambda_j)]$ can be reproduced in the reconstruction step, exactly as in the forward step, and thus also $\gamma_j \leftarrow \gamma_j + [\mathcal{P}(\lambda_j)]$ will give the original γ_j back. This shows that each step of the lifting scheme with rounding, is perfectly invertible and thus the whole signal is perfectly reconstructable, whatever the rounding rule we use, on condition of course that the rounding is deterministic.

2.5 Example: Cohen-Daubechies-Feauveau

One popular family of classical biorthogonal wavelets that fits in the above scheme are the wavelets constructed by Cohen, Daubechies and Feauveau [3]. Especially its member with two vanishing moments for both the primal and dual wavelet (hence named CDF $(2,2)$ wavelet) is widely used.

Thanks to the lifting scheme, the accompanying wavelet transform can be implemented in an efficient way (see e.g. [15]). From the second generation viewpoint, one transform step of a discrete signal $x = \{x_k\}$ looks like:

Splitting Split the signal x (i.e. λ_{j+1}) into even samples (i.e. λ_j) and odd samples (i.e. γ_j):

$$\begin{aligned}s_i &\leftarrow x_{2i},\\ d_i &\leftarrow x_{2i+1}\end{aligned}$$

Prediction Predict the odd samples using linear interpolation:

$$d_i \leftarrow d_i - \frac{1}{2}(s_i + s_{i+1}).$$

Update Update the even samples to preserve the mean value of the samples:

$$s_i \leftarrow s_i + \frac{1}{4}(d_{i-1} + d_i).$$

As a result, the signal $s = \{s_k\}$ is a coarse representation of the original signal x, while the signal $d = \{d_k\}$ contains the high frequency information that is lost when going from resolution level $j+1$ to resolution level j.

The same kind of interpolation scheme can be used for interpolating polynomials of higher degree. These lifting schemes need some special care at the boundaries if one wants the wavelets to live explicitly on the discrete set where the data are defined. A whole family of lifting schemes can be constructed in this way [14], of which the above example is just the simplest possible case.

Note that this transform works on one-dimensional data. For two-dimensional data, it can be applied row and columnwise, resulting in a tensor product wavelet transform. This is a separable transform.

3 The Red-Black Wavelet Transform

In the previous section, we presented the principle of constructing wavelet transforms by the lifting scheme in a very general setting. It was not assumed that the data sets were one-dimensional for example. The same principle and the same description can be applied for two or more dimensional data as well. As an illustration, we apply these ideas to the two-dimensional case in its simplest possible appearance: a two-dimensional analog of the one-dimensional CDF (2, 2) wavelet. We assume that the data are given on a regular rectangular grid. It should be clear form our previous exposition, that this is about the simplest nontrivial case that can be conceived, and that much more complex schemes are possible. However, because in the one-dimensional case, the CDF (2, 2) wavelets are widely used, we are convinced that the two-dimensional analog that we present below has the same potentials as the the CDF (2, 2) wavelet.

3.1 A Two-Step Method

Our approach is inspired by the well-known Red-Black Gauss-Seidel technique for the iterative solving of linear systems. In the literature [7], the kind of lattice we use is known as a quincunx lattice. However, we prefer the name Red-Black wavelet transform because it is simpler and it is more appropriate to describe the splitting step in the lifting scheme.

The idea, just like in the CDF (2, 2) wavelet, is that we first split the data in two subsets. The even/odd splitting from the one-dimensional case is replaced by a checkerboard splitting, with red and black squares. Thus we have a red subset λ_j and a black subset γ_j. Then we predict the values of the red subset λ_j from its immediate neighbors in the black set. This does not leaves much choice and so, a value is predicted from its horizontal and vertical neighbors, i.e., the elements at the N, S, E, and W positions. In accordance with the CDF (2, 2) lifting scheme, we take for this prediction an average of these four neighbors. The elements in λ_j are replaced by their prediction errors. Next, we have to make an updating step for the data in the red subset γ_j, which is based on the (new) black values, to preserve the mean value of the data.

For the subsequent resolution level $j-1$, we are left with the (new) red set λ_j from the previous level. However, it is clear by looking at Figure 1, that these data are arranged along diagonals on the grid. Thus we cannot repeat the same operation on a horizontal/vertical basis as on the previous resolution level, but we can do it on the diagonals instead. Again we start by splitting λ_j into two subsets: the blue subset λ_{j-1} and the yellow subset γ_{j-1} (see Figure 2). The same kind of averaging and updating steps are then computed for the blue-yellow squares. For the next resolution level, we have to decompose the blue subset further, but since this subset is again arranged in a horizontal/vertical manner, we can do again a red-black update. It is seen that we will have alternatingly a red-black and a blue-yellow splitting to pass through the subsequent resolution levels. Therefore we say that the Red-Black wavelet transform is a 'two level' transform.

We give now the formulas which are based on these ideas in a more schematic presentation. Since we are working on a rectangular grid, the input data set λ_{j+1} is an image, i.e. a rectangular matrix $X = x_{i,j}$.

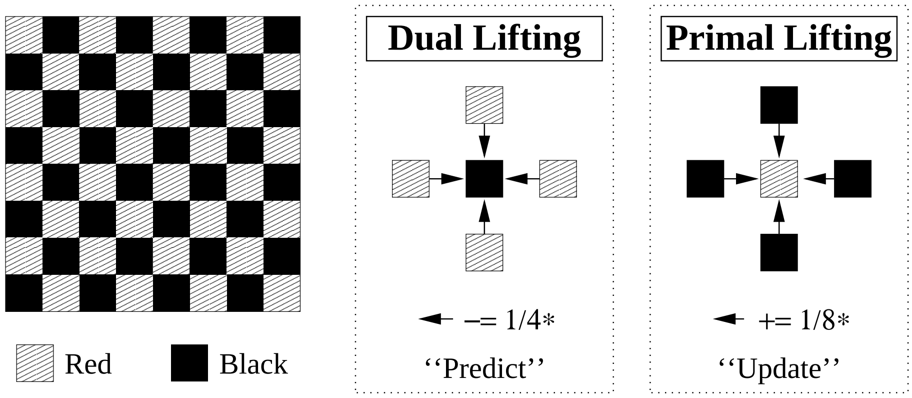


Figure 1: Horizontal/vertical lifting in the odd steps: *Red-Black*.

3.1.1 Horizontal/Vertical Lifting

Splitting: The image λ_{j+1} is split in the *red* squares λ_j, and the *black* squares γ_j (cfr. Figure 1).

Prediction: The black squares are predicted using linear interpolation based on the 4 neighboring red squares:

$$x_{i,j} \leftarrow x_{i,j} - \frac{1}{4}\left(x_{i-1,j} + x_{i,j-1} + x_{i,j+1} + x_{i+1,j}\right) \qquad \text{for } i \bmod 2 \neq j \bmod 2.$$

Update: The red squares are updated using the black squares to preserve the mean value:

$$x_{i,j} \leftarrow x_{i,j} + \frac{1}{8}\left(x_{i-1,j} + x_{i,j-1} + x_{i,j+1} + x_{i+1,j}\right) \qquad \text{for } i \bmod 2 = j \bmod 2.$$

As a result, because the averaging operation smooths the data, the values corresponding to the *red* squares are a low resolution representation of the original image, while the *black* squares contain the detail information. The inverse transform is straightforward.

3.1.2 Diagonal Lifting

For this step we consider the checkerboard to be rotated over $45°$.

Splitting The *red* squares in Figure 1 (the data λ_j) are partitioned in the *blue* (the data λ_{j-1}) and *yellow* (the data γ_{j-1}) squares in Figure 2.

Prediction The yellow squares are predicted using linear interpolation based on the 4 neighboring blue squares:

$$x_{i,j} \leftarrow x_{i,j} - \frac{1}{4}\left(x_{i-1,j-1} + x_{i-1,j+1} + x_{i+1,j-1} + x_{i+1,j+1}\right) \qquad \text{for } \begin{cases} i \bmod 2 = 1, \\ j \bmod 2 = 1. \end{cases}$$

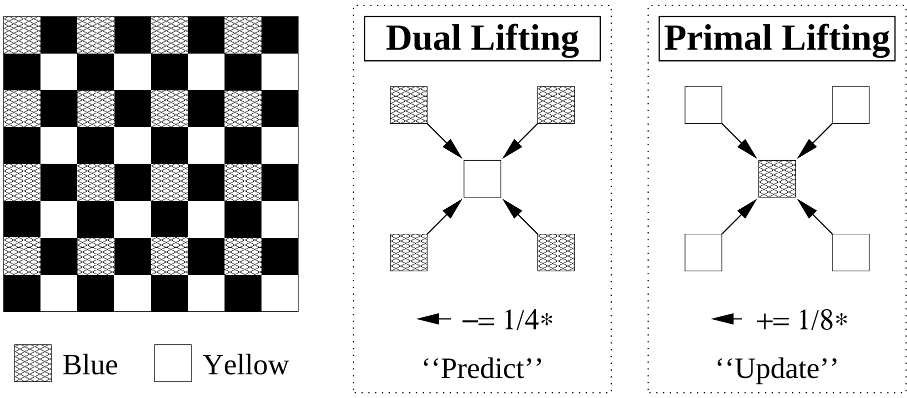


Figure 2: Diagonal lifting in the even steps: *Blue*-*Yellow*.

Update The blue squares are updated using the yellow squares to preserve the mean value:

$$x_{i,j} \leftarrow x_{i,j} + \frac{1}{8}\left(x_{i-1,j-1} + x_{i-1,j+1} + x_{i+1,j-1} + x_{i+1,j+1}\right) \qquad \text{for} \begin{cases} i \bmod 2 = 0, \\ j \bmod 2 = 0. \end{cases}$$

Result: the *blue* squares (λ_{j-1}) are a low resolution representation of the original image, while the *yellow* squares contain the detail information. Again, the inverse transform is straightforward.

The next step (horizontal/vertical lifting, again) will be performed on the blue squares only, yielding a multi-resolution decomposition.

3.2 Borders

As in the one dimensional case, one could arrange for the wavelets to live exactly on the grid where the original data are given. However, it is common practice to extend the data symmetrically for the computations near the border [1]. This gives good results in practice for the classical one-dimensional or two-dimensional tensor product wavelets. To avoid further complications, we assume that also in our case the data are extended symmetrically across the borders.

3.3 Reordering

The above transform is 100% in place and all coefficients will be interleaved. This is one of the powerful features of the lifting scheme: it does not require additional memory to calculate the forward or inverse transform.

If one wants a representation more similar to the Mallat representation, one has to reorder the wavelet coefficients after each diagonal lifting step. For multiple levels, we get the representation of Figure 3.

Note that this reordering scheme is not possible if either the number of rows or columns of the image is odd.

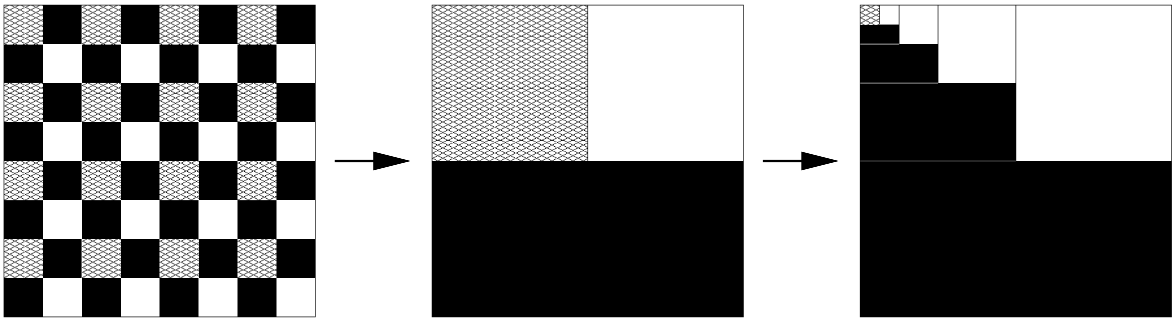


Figure 3: Reordering of the interleaved coefficients and multilevel decomposition.

3.4 Integer Wavelet Transform

Because the Red-Black wavelet transform is based on the lifting scheme, a version that maps integers to integers is immediately deduced: all computations giving a floating point number are rounded to an integer, e.g.

$$\mu \leftarrow \mu - \frac{1}{4}(\rho + \sigma + \tau + \upsilon)$$

becomes

$$\mu \leftarrow \mu - \left[\frac{1}{4}(\rho + \sigma + \tau + \upsilon)\right]$$

and similarly for all the other steps.

3.5 Basis Functions

In a classical one-dimensional wavelet transform, all basis functions are derived from two functions — the mother and the father wavelet — by dilation and translation. In the case of one-dimensional bi-orthogonal wavelets, there is also a dual basis. However, in our construction we use different wavelet transforms for the even and odd steps. As a consequence all basis functions are derived from three[1] functions:

$\psi_{\mathrm{low}}(x, y)$ Basis function corresponding to the low pass part of the diagonal lifting step.

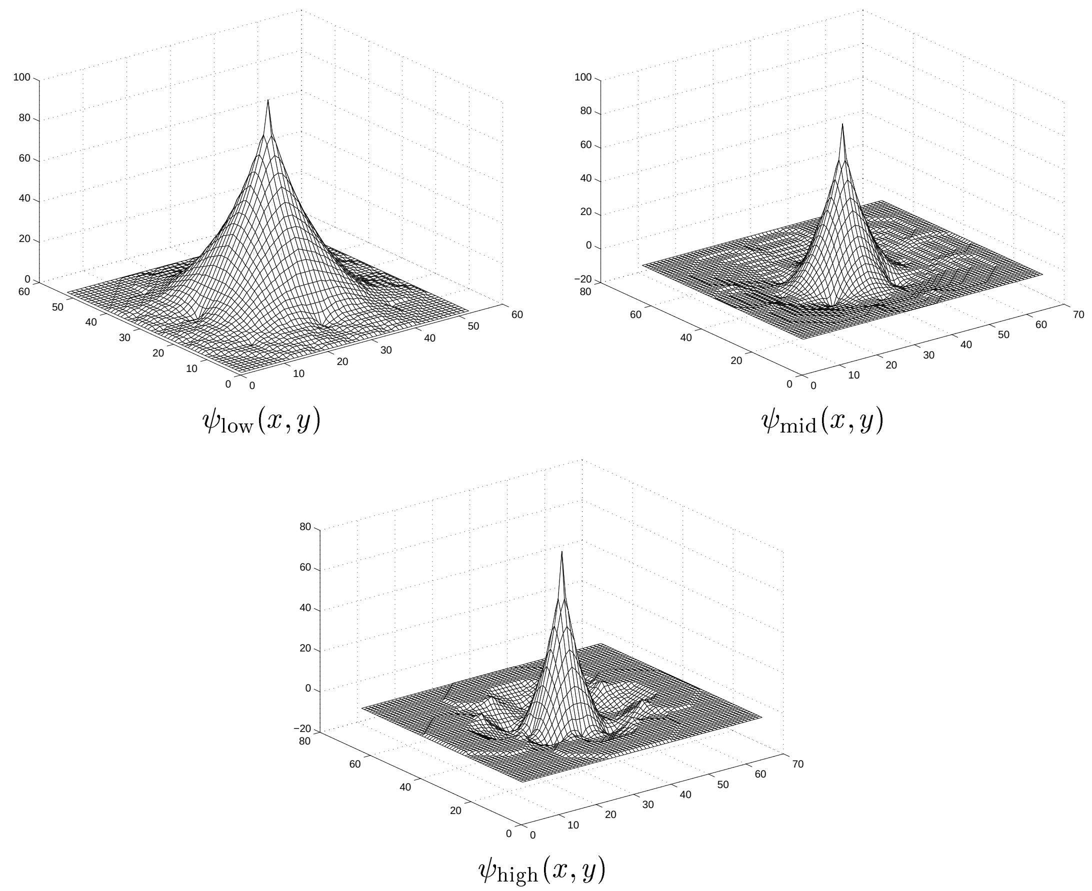
$\psi_{\mathrm{mid}}(x, y)$ Basis function corresponding to the high pass part of the diagonal lifting step.

$\psi_{\mathrm{high}}(x, y)$ Basis function corresponding to the high pass part of the horizontal/vertical lifting step.

These three functions can be obtained by choosing a blue, a yellow or a red square to take a nonzero value and setting all other coefficients equal to zero. Then the inverse transform is applied to these data, in principle for an infinite number of levels, which results in the basis functions .

Graphs of these basis functions (for the integer variant of the lifting scheme) are shown in Figure 4.

[1]We do not consider the intermediate basis functions corresponding to the low pass parts after the horizontal/vertical lifting steps.



$\psi_{\text{low}}(x, y)$ $\psi_{\text{mid}}(x, y)$ $\psi_{\text{high}}(x, y)$

Figure 4: Graphs of the basis functions for the integer Red-Black wavelet transform.

3.6 Other Members of the Family

As we already mentioned at the beginning of this section, we used in the description above very simple prediction and update ‘molecules’. The molecules have the form of a cross: a normal cross for the odd steps and a diagonal cross for the even steps. These give us wavelet functions with 2 vanishing moments, cfr. the Cohen-Daubechies-Feauveau biorthogonal wavelets. By using larger molecules and more advanced prediction and update formulas, one can create smoother basis functions with larger supports.

4 Properties of the Red-Black Wavelet Transform

The Red-Black wavelet transform has the following properties:

2D: The Red-Black wavelet transform is a true two-dimensional transform. It is not constructed using a tensor product of a one-dimensional transform and is not separable.

Isotropy: It is less anisotropic than tensor product wavelets on the same rectangular grid.

The edges which are manifest in the figures of the tensor product wavelets show the pronounced vertical/horizontal orientation, while this is not visible in the figures of the Red-Black wavelets.

This property was our main motivation for developing the Red-Black wavelet transform. Hexagonal data cells would be more anisotropic, but not many input data formats have hexagonal data cells. Most have square or rectangular data cells. And it is not trivial to convert a rectangular grid to a hexagonal grid in a satisfactory way.

Symmetry: All basis functions have four symmetry axes: horizontal, vertical, and diagonal.

This follows immediately by construction of these wavelets and is visually checked in the pictures of Figure 1.

Biorthogonality: The splitting in red and black or blue and yellow pixels creates trivial biorthogonal basis functions. Since lifting steps preserve the biorthogonality properties the transform as a whole has biorthogonal basis functions, too [11].

Smoothness: The wavelet functions (both primal and dual) have two vanishing moments: constant, linear and bi-linear functions can be represented exactly by the scaling functions. This is similar to the Cohen-Daubechies-Feauveau (2, 2) biorthogonal wavelets, from which we borrowed the construction scheme and extended it to two dimensions.

Smoother wavelets with larger support can be constructed by enlarging the prediction and update molecules, and by using multiple lifting steps.

Implementation: Both the forward and inverse transform are very simple to implement and have fast execution times ($\mathcal{O}(n)$, with n the number of elements in the grid), due to the underlying lifting technology.

5 The Redundant Wavelet Transform

In the classical 'first generation' framework, there is a decimation step after each filter step to reduce the amount of data in every subband. In the 'second generation' case, this is reflected in the splitting step: the original data set is split in two distinct data sets. Thus at each transform step, the amount of work is reduced, compared to the previous step. This makes the total amount of work proportional to the size of the input data ($\mathcal{O}(n)$), hence the name *Fast Wavelet transform*.

But for some applications, this is a disadvantage: it makes the transform *translation variant*: if the input data is translated, at each resolution level data samples may migrate to a different data set due to the partitioning in distinct data sets at the subsequent resolution levels. As a consequence, in most cases (depending on the translation direction and distance) the wavelet transform of the translated data is not equal to the translation of the wavelet transform of the original data. E.g. feature detection and noise reduction suffer from translation variance.

The solution is called the *Redundant Wavelet Transform* (or stationary, or non-decimated wavelet transform) [8]. Here one gets rid of the decimation step, causing all subbands to have the same size as the size of the input data set. At each resolution level, the filters have to be upsampled[2] to keep a consistent multi-resolution analysis.

The computing complexity is no longer $\mathcal{O}(n)$, but $\mathcal{O}(n \log n)$. The redundant transform also requires more memory. For the inverse transform, we have to calculate some kind of mean value due to the redundance in the forward transform.

Using the lifting scheme, it is still possible to implement the redundant wavelet transform. The following changes are required:

Copying Instead of partitioning the data set λ_{j+1} into two distinct data sets λ_j and γ_j, we copy λ_{j+1} to λ_j and γ_j. Thus both λ_j and γ_j contain the whole input data set.

Prediction We use a different prediction at every transform level to mimic the upsampling of the filters:

$$\gamma_j \leftarrow \gamma_j - \mathcal{P}_j(\lambda_j).$$

Update We use a different update at every transform level to mimic the upsampling of the filters:

$$\lambda_j \leftarrow \lambda_j + \mathcal{U}_j(\gamma_j).$$

The operators $\mathcal{P}_j$ and $\mathcal{U}_j$ were given an index, because they compute values that are indeed based on four samples, but they need not be the "nearest" samples anymore because the position of the elements must be computed by a formula that depends on the resolution level.

6 Comparison with the CDF (2, 2) Tensor Product Wavelets

We compared the performance of the integer Red-Black wavelet transform with the integer Cohen-Daubechies-Feauveau (2, 2) tensor product wavelet transform. The basis functions for the latter transform are shown in figure 5.

[2] A filter is upsampled by putting zeros between the successive filter coefficients.

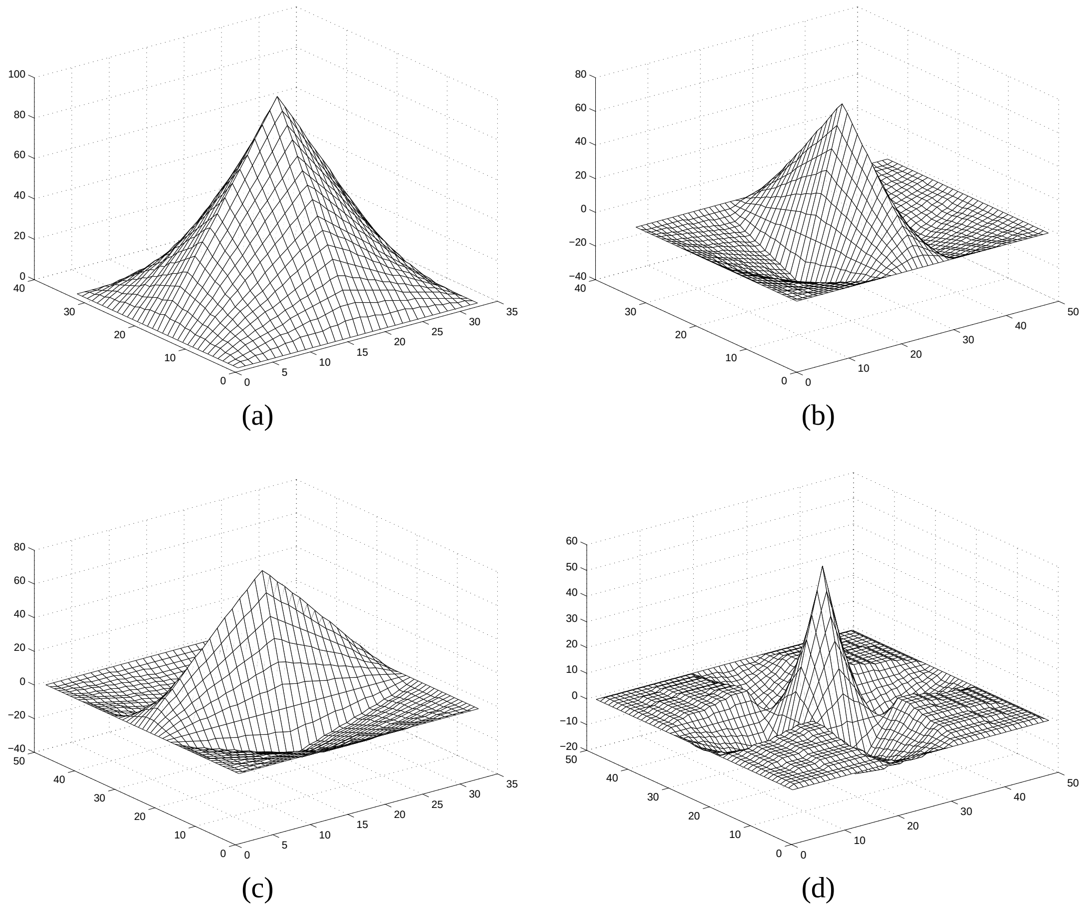
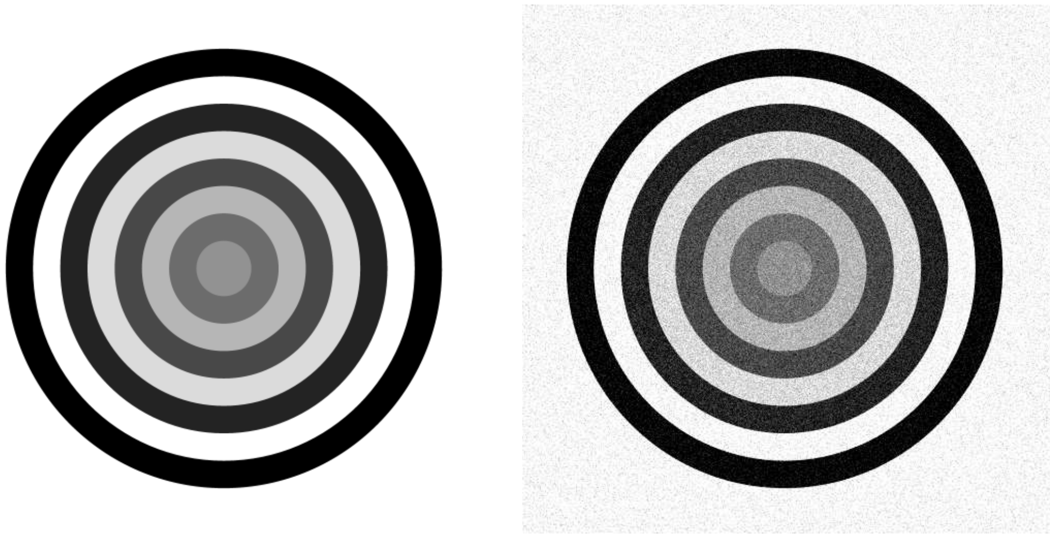


Figure 5: Graphs of the basis functions for the integer Cohen-Daubechies-Feauveau (2, 2) tensor product wavelets.



(a) (b)

Figure 6: Greyscale test images (512×512 pixels): (a) without and (b) with additive white noise ($\sigma = 20$).

6.1 Test case 1

We considered an image processing operation on an greyscale aerial image: noise reduction using wavelet shrinkage based on generalized cross validation [5].

For Red-Black and CDF (2, 2), we tested both the fast wavelet transform and the redundant wavelet transform. The results are summarized below:

- The tensor product CDF (2, 2) wavelets perform marginally better than the Red-Black wavelets and result in more pronounced edges. This is caused by the larger support for the Red-Black wavelets.

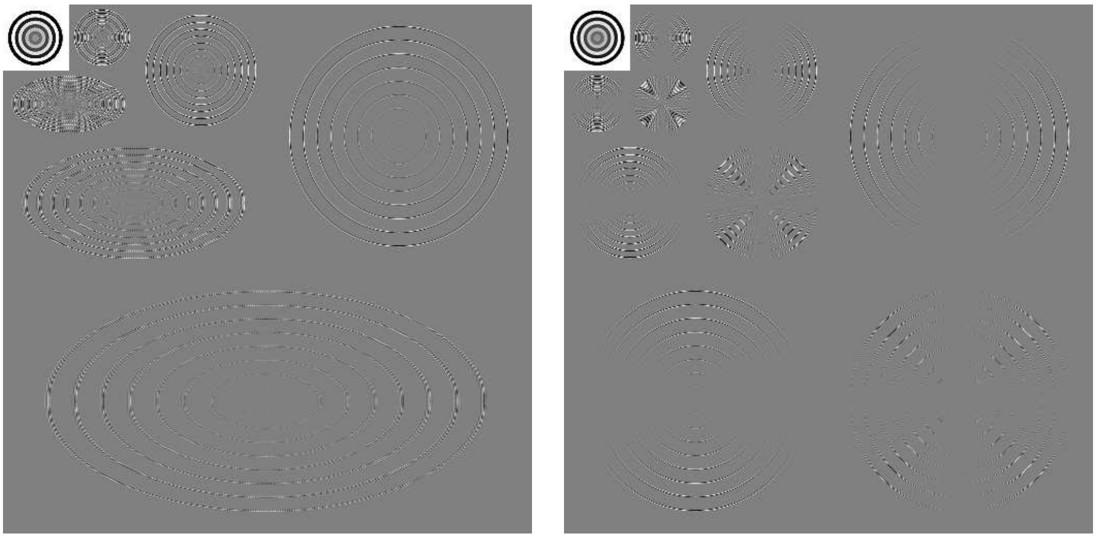
  There is one exception: faint lines that are not horizontal, vertical or diagonal are better preserved by the Red-Black transform, due to the less anisotropic properties.

- As expected, the redundant wavelet transform always performs better than the corresponding fast wavelet transform.

6.2 Test case 2

For the second case, we started from an image containing anti-aliased concentric circles, in different shades of grey (fig. 6a), and added white noise to it (fig. 6b).

The three-level integer wavelet transform of the noise-free image is shown in fig. 7, for both the Red-Black and the tensor product CDF (2, 2) wavelets. The concentric circles are more pronounced in the decomposition when using the Red-Black wavelets, due to their better anisotropic properties.



(a) (b)

Figure 7: Three-level wavelet decomposition of the test image without noise, using (a) Red-Black and (b) tensor product CDF (2, 2) wavelets.

The tensor product CDF (2, 2) wavelets clearly show to detect only horizontal, vertical and diagonal features.

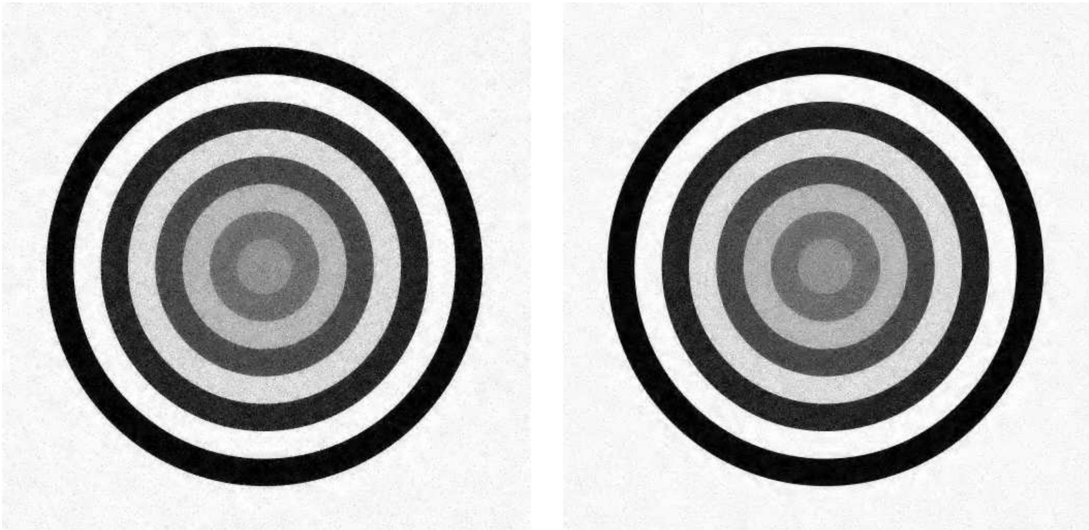
Then we reduced the noise in the noisy image using the same wavelet shrinkage technique as in the first test case, again for both the Red-Black and the tensor product CDF (2, 2) wavelets. The resulting images are shown in fig. 8. The Peak Signal to Noise Ratio (PSNR) measure indicates that the tensor product CDF (2, 2) wavelets perform a bit better (approx. 0.4 dB better). However, according to the human eye, the Red-Black wavelets give sharper concentric circles in low contrast areas, i.e. in the center of the image.

7 Conclusion

Red-Black wavelets are less anisotropic than tensor product wavelets. Compared to a similar tensor product wavelet, image denoising using Red-Black wavelets performs better for lines that are not horizontal, vertical or diagonal.

Besides image processing, other possible applications are situated in partial differential equations.

The 'two level' principle of the Red-Black wavelet transform is not restricted to a quincunx lattice. It can also be extended to triangular or hexagonal regular grids: an update and predict pair on a triangular grid will yield a hexagonal low pass grid, while an update and predict pair on a hexagonal grid will yield a triangular low pass grid. Note that one is not limited to planar grids: (semi-)regular grids of triangles or hexagons on a sphere are also possible.



(a) (b)

Figure 8: Reconstruction of the original image after denoising using (a) Red-Black (PSNR = 26.8 dB) and (b) CDF (2, 2) wavelets (PSNR = 27.2 dB).

8 Acknowledgements

This paper presents research results of the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture. The scientific responsibility rests with its authors. This research is also supported by the Flemish Information Technology Action Program ('Vlaams Actieprogramma Informatietechnologie'), project number ITA/950244.

Note

After finishing this work, we learned that Kovačević and Sweldens have completed a similar construction[6].

References

[1] C. M. Brislawn. Classification of nonexpansive symmetric extension transforms for multirate filter banks. *Appl. Comput. Harmon. Anal.*, 3:337–357, 1996.

[2] R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo. Wavelet transforms that map integers to integers. Technical report, Department of Mathematics, Princeton University, 1996.

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

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

[5] M. Jansen, M. Malfait, and A. Bultheel. Generalized cross validation for wavelet thresholding. *Signal Processing*, 56(1):33–44, January 1997.

[6] J. Kovačević and W. Sweldens. Wavelet families of increasing order in arbitrary dimensions. Technical report, Bell Laboratories, Lucent Technologies, 1997.

[7] J. Kovacevic and M. Vetterli. Nonseparable multidimensional perfect reconstruction filter banks and wavelet bases for $\mathbf{R}^n$. *IEEE Trans. Inform. Theory*, 38(2):533–555, March 1992.

[8] G. P. Nason and B. W. Silverman. The stationary wavelet transform and some statistical applications. In A. Antoniadis and G. Oppenheim, editors, *Wavelets and Statistics*, Lecture Notes in Statistics, pages 281–299, 1995.

[9] P. Schröder and W. Sweldens. Spherical wavelets: Efficiently representing functions on the sphere. *Computer Graphics, (SIGGRAPH '95 Proceedings)*, 1995.

[10] E.P. Simoncelli and E.H. Adelson. Non-separable extensions of quadrature mirror filters to multiple dimensions. *Proceedings of the IEEE*, 78:652–664, April 1990.

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

[12] W. Sweldens. The lifting scheme: A new philosophy in biorthogonal wavelet constructions. In A. F. Laine and M. Unser, editors, *Wavelet Applications in Signal and Image Processing III*, pages 68–79. Proc. SPIE 2569, 1995.

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

[14] W. Sweldens and P. Schröder. Building your own wavelets at home. Technical Report 1995:5, Industrial Mathematics Initiative, Department of Mathematics, University of South Carolina, 1995.

[15] G. Uytterhoeven, F. Van Wulpen, M. Jansen, D. Roose, and A. Bultheel. WAILI: Wavelets with Integer Lifting. TW Report 262, Department of Computer Science, Katholieke Universiteit Leuven, Belgium, July 1997.