Chapter 7

The Red-Black Wavelet Transform

In this chapter we present a new kind of second-generation wavelets on a rectangular grid. These wavelets are constructed using two-dimensional lifting 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 Red-Black wavelets is compared to tensor product wavelets in an image denoising application.

7.1 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 [30]) or on hexagonal grids [44]. All these are heavily based on Fourier transforms, 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 chapter, 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 [52].

7.2 The Red-Black Wavelet Transform

In Section 3.2, 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 non-trivial 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 CDF $(2,2)$ wavelet.

7.2.1 A Two-Step Method

Our approach is inspired by the well-known Red-Black Gauss-Seidel technique for the iterative solution of linear systems. In the literature [30], 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 black subset γ_j from its immediate neighbors in the red set λ_j. This does not leave much choice and so, a value is predicted from its horizontal and vertical neighbors, i.e., the elements at the N(orth), S(outh), E(ast), and W(est) positions. In accordance with the CDF $(2,2)$ lifting scheme, we take for this prediction an average of these four neighbors. The elements in γ_j are replaced by their prediction errors. Next, we have to make an updating step for the data in the red subset λ_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 7.1, 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

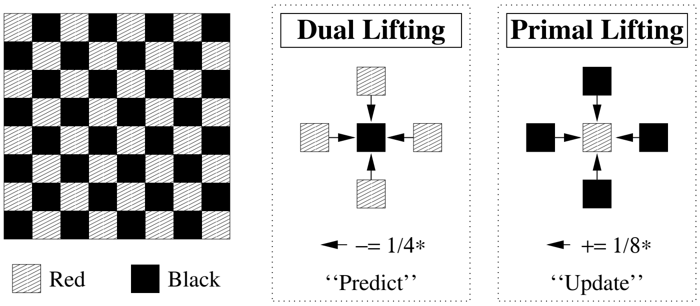


Figure 7.1: Horizontal/vertical lifting in the odd steps: *Red-Black*.

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 7.2). 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 alternating red-black and blue-yellow splittings 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 $X = x_{i,j}$.

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 7.1).

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

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

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

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

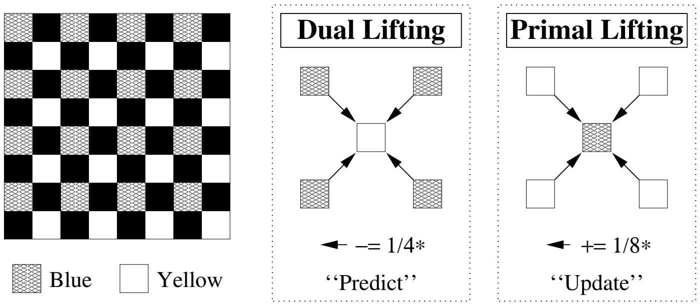


Figure 7.2: Diagonal lifting in the even steps: *Blue-Yellow*.

As a result, because the averaging operation smoothes 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.

Diagonal Lifting

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

Splitting The *red* squares in Figure 7.1 (the data λ_j) are partitioned in the *blue* (the data λ_{j-1}) and *yellow* (the data γ_{j-1}) squares in Figure 7.2.

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

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

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

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

Thus 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.

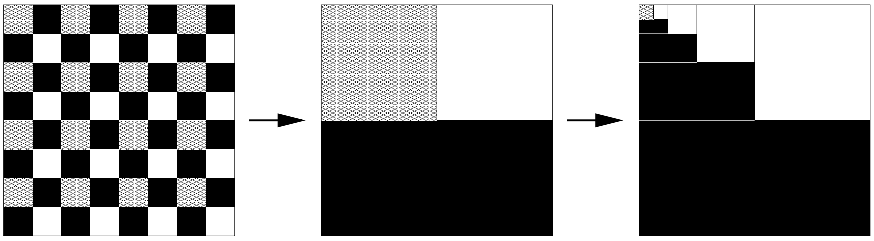


Figure 7.3: Reordering of the interleaved coefficients and multilevel decomposition.

7.2.2 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 [4]. 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.

7.2.3 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 an ordering more similar to the Mallat ordering, one has to reorder the wavelet coefficients after each diagonal lifting step. For multiple levels, we get the representation of Figure 7.3.

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

7.2.4 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 (curly braces denote rounding):

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

and similarly for all the other steps.

7.2.5 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 biorthogonal 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:

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

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

$\psi_{\text{high}}(x,y)$ 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 non-zero 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 7.4.

7.2.6 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.

In parallel with our work, Kovačević and Sweldens completed a similar construction, which considers our example as a special case in a more general multidimensional lifting framework [31].

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

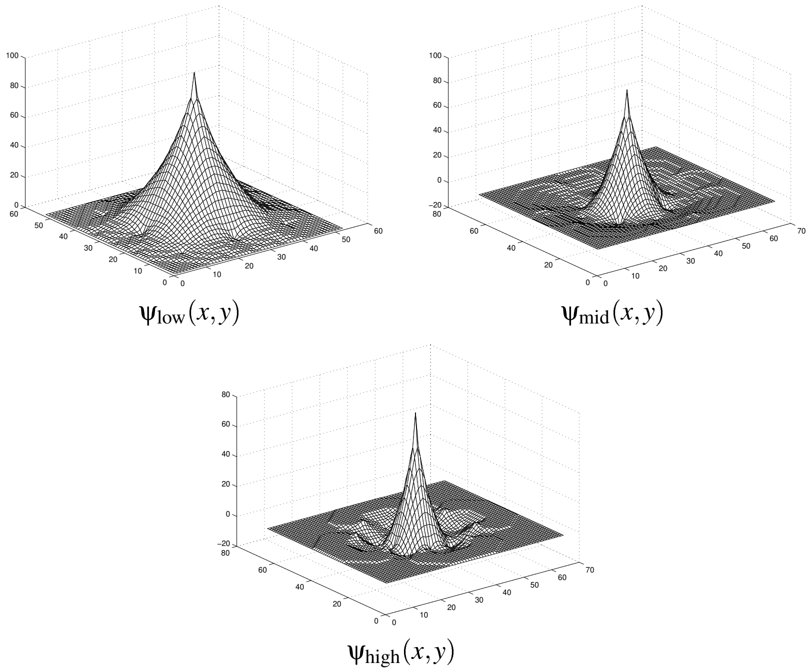


Figure 7.4: Graphs of the basis functions for the integer Red-Black wavelet transform.

7.3 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 thus it 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 7.1.

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 [49].

Smoothness: The wavelet functions (both primal and dual) have two vanishing moments: constant, linear and bilinear 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 ($O(n)$, with n the number of elements in the grid), due to the underlying lifting technology.

7.4 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 ($O(n)$), 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.

The solution to this problem is called the redundant wavelet transform (see Section 2.5). 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 still 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.

7.5 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 7.5.

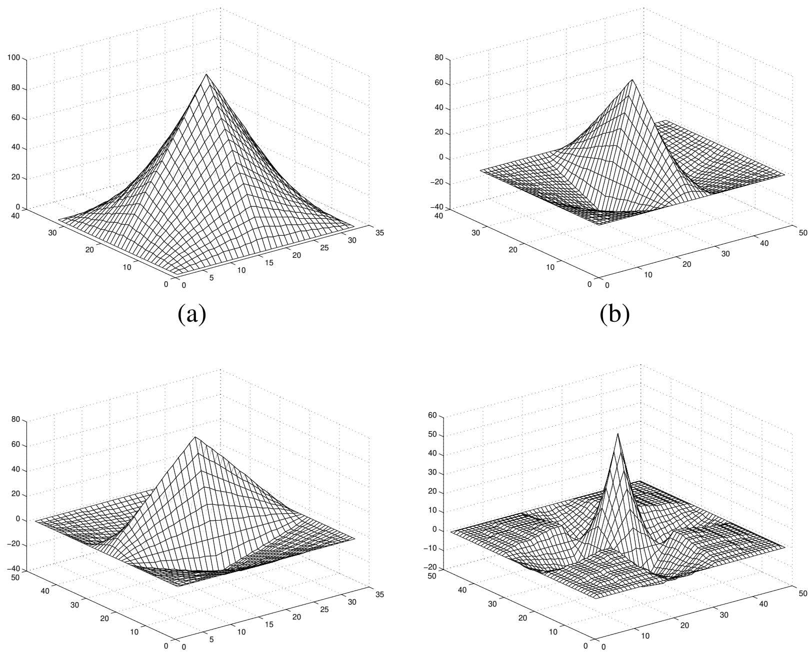


Figure 7.5: Graphs of the basis functions for the integer Cohen-Daubechies-Feauveau $(2,2)$ tensor product wavelets.

7.5.1 Test case 1

We considered an image processing operation on an grey scale aerial image: noise reduction using wavelet shrinkage based on Generalized Cross Validation [24].

For Red-Black and CDF $(2,2)$, we tested both the fast wavelet transform and the redundant wavelet transform. The results, obtained by visual inspection, 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. Because the redundant wavelet transform does not use subsampling and is translation invariant, it is better in preserving fine edges.

7.5.2 Test case 2

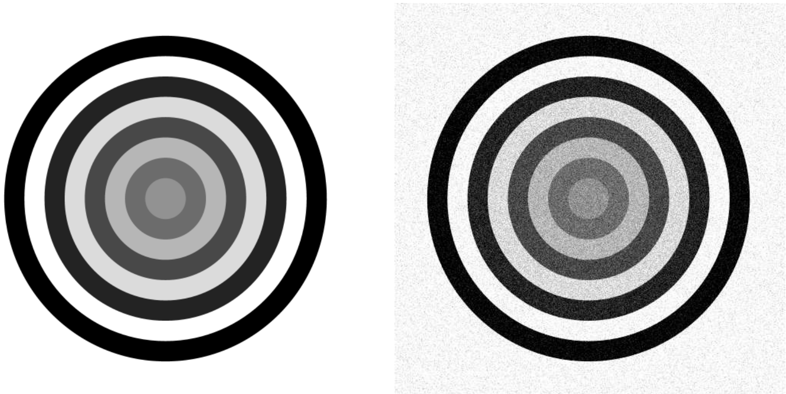
For the second case, we started from an image containing anti-aliased concentric circles, in different shades of grey (Figure 7.6a), and added white noise to it (Figure 7.6b).

The three-level integer wavelet transform of the noise-free image is shown in Figure 7.7, 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. 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 Figure 7.8. 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.

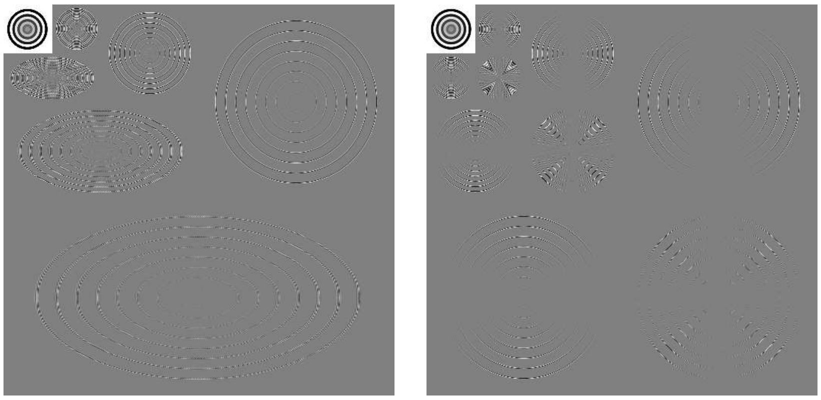
7.6 Conclusions

Red-Black wavelets are less anisotropic than tensor product wavelets because they favor the horizontal and vertical directions less than tensor product wavelets. Com-



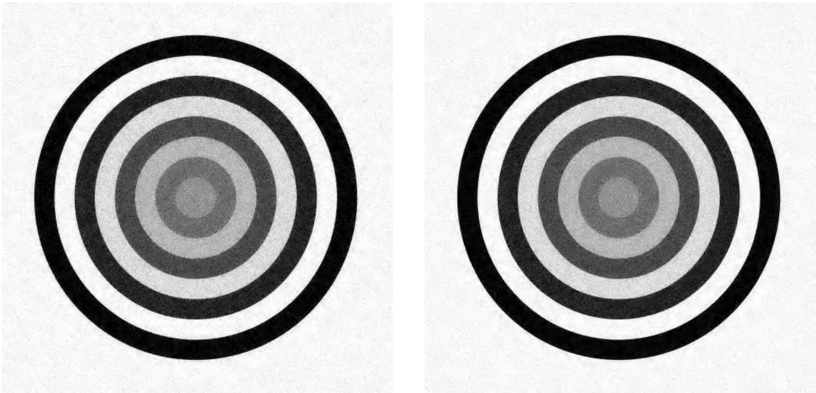
(a) (b)

Figure 7.6: Grey scale test images (512×512 pixels): (a) without and (b) with additive white noise ($\sigma = 20$).



(a) (b)

Figure 7.7: Three-level wavelet decomposition of the test image without noise, using (a) Red-Black and (b) tensor product CDF $(2,2)$ wavelets.



(a) (b)

Figure 7.8: 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). Although the PSNR is lower when using Red-Black wavelets, the Red-Black wavelets give sharper concentric circles in low-contrast areas, i.e. in the center of the image, according to the human eye.

pared 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.