Chapter 8

Tensor products in a wavelet setting

In Chapter 7 we defined tensor products in terms of vectors, and we saw that the tensor product of two vectors is in fact a matrix. The same construction can be applied to other vector spaces, in particular to vector spaces that are function spaces. As we will see, the tensor product of two univariate function spaces will be a space of functions in two variables. Recall that wavelets are defined in terms of function spaces, so this construction will allow us to define tensor products of wavelets. Through this we will be able to define wavelet transforms that can be applied to images.

Definition 8.1 (Tensor product of function spaces)**.** Let V and W be two vector spaces of functions defined on the intervals $[0, M)$ and $[0, N)$, respectively, and suppose that $f_1 \in V$ and $f_2 \in W$. The tensor product of f_1 and f_2, denoted $f_1 \otimes f_2$, denotes the function in two variables defined on $[0, M) \times [0, N)$ given by $f_1(t_1)f_2(t_2)$. The function $f_1 \otimes f_2$ is also referred to as the separable extension of f_1 and f_2 to two variables. The tensor product of the two spaces $V \otimes W$ denotes the set of all functions in two variables defined on $[0, M) \times [0, N)$ and on the form $f_1(t_1)f_2(t_2)$, where $f_1 \in V$ and $f_2 \in W$.

We will always assume that the spaces V and W consist of functions which are at least integrable. In this case $V \otimes W$ is also an inner product space, with the inner product given by a double integral,

$$\langle f, g\rangle = \int_0^N \int_0^M f(t_1, t_2)g(t_1, t_2)dt_1 dt_2. \tag{8.1}$$

In particular, this says that

$$\begin{aligned}\langle f_1 \otimes f_2, g_1 \otimes g_2 \rangle &= \int_0^N \int_0^M f_1(t_1) f_2(t_2) g_1(t_1) g_2(t_2) dt_1 dt_2 \\ &= \int_0^M f_1(t_1) g_1(t_1) dt_1 \int_0^N f_2(t_2) g_2(t_2) dt_2 = \langle f_1, g_1 \rangle \langle f_2, g_2 \rangle.\end{aligned} \tag{8.2}$$

This means that for tensor products, a double integral can be computed as the product of two one-dimensional integrals.

The tensor product space defined in Definition 8.1 is useful for approximation of functions of two variables if each of the two spaces of univariate functions have good approximation properties.

Idea 8.2. If the spaces V and W can be used to approximate functions in one variable, then $V \otimes W$ can be used to approximate functions in two variables.

We will not state this precisely, but just consider some important examples.

Example 8.3. Let $V = W$ be the space of all polynomials of finite degree. We know that V can be used for approximating many kinds of functions, such as continuous functions, for example by Taylor series. The tensor product $V \otimes V$ consists of all functions on the form $\sum_{i,j} \alpha_{i,j} t_1^i t_2^j$. It turns out that polynomials in several variables have approximation properties analogous to univariate polynomials.

Example 8.4. Let $V = W = V_{N,T}$ be the Nth order Fourier space which is spanned by the functions

$$e^{-2\pi i N t/T}, \ldots, e^{-2\pi i t/T}, 1, e^{2\pi i t/T}, \ldots, e^{2\pi i N t/T}$$

The tensor product space $V \otimes V$ now consists of all functions on the form $\sum_{k,l=0}^{N} \alpha_{k,l} e^{2\pi i k t_1/T} e^{2\pi i l t_2/T}$. One can show that this space has approximation properties similar to $V_{N,T}$. This is the basis for the theory of Fourier series in two variables.

In the following we think of $V \otimes W$ as a space which can be used for approximating a general class of functions. By associating a function with the vector of coordinates relative to some basis, and a matrix with a function in two variables, we have the following parallel to Theorem 7.10:

Theorem 8.5. If $\{f_i\}_{i=0}^{M-1}$ is a basis for V and $\{g_j\}_{j=0}^{N-1}$ is a basis for W, then $\{f_i \otimes g_j\}_{(i,j)=(0,0)}^{(M-1,N-1)}$ is a basis for $V \otimes W$. Moreover, if the bases for V and W are orthogonal/orthonormal, then the basis for $V \otimes W$ is orthogonal/orthonormal.

Proof. The proof is similar to that of Theorem 7.10: if

$$\sum_{(i,j)=(0,0)}^{(M-1,N-1)} \alpha_{i,j}(f_i \otimes g_j) = 0,$$

we define $h_i(t_2) = \sum_{j=0}^{N-1} \alpha_{i,j} g_j(t_2)$. It follows as before that $\sum_{i=0}^{M-1} h_i(t_2) f_i = 0$ for any t_2, so that $h_i(t_2) = 0$ for any t_2 due to linear independence of the f_i. But then $\alpha_{i,j} = 0$ also, due to linear independene of the g_j. The statement about orthogonality follows from Equation 8.2. □

We can now define the tensor product of two bases of functions as before: if $\mathcal{B} = \{f_i\}_{i=0}^{M-1}$ and $\mathcal{C} = \{g_j\}_{j=0}^{N-1}$, we set $\mathcal{B} \otimes \mathcal{C} = \{f_i \otimes g_j\}_{(i,j)=(0,0)}^{(M-1,N-1)}$. Coordinate matrices can also be defined as before: if $f(t_1,t_2) = \sum X_{i,j}(f_i \otimes g_j)(t_1,t_2)$, the coordinate matrix of f is the matrix X with elements $X_{i,j}$. Theorem 7.12 can also be proved in the same way in the context of function spaces. We state this as follows:

Theorem 8.6 (Change of coordinates in tensor products of function spaces)**.** Assume that $\mathcal{B}_1, \mathcal{C}_1$ are bases for V, and $\mathcal{B}_2, \mathcal{C}_2$ are bases for W, and that S is the change of coordinates matrix from $\mathcal{C}_1$ to $\mathcal{B}_1$, and that T is the change of coordinates matrix from $\mathcal{C}_2$ to $\mathcal{B}_2$. Both $\mathcal{B}_1 \otimes \mathcal{B}_2$ and $\mathcal{C}_1 \otimes \mathcal{C}_2$ are bases for $V \otimes W$, and if X is the coordinate matrix in $\mathcal{C}_1 \otimes \mathcal{C}_2$, and Y the coordinate matrix in $\mathcal{B}_1 \otimes \mathcal{B}_2$, then

$$Y = SXT^T. \tag{8.3}$$

8.1 Adopting the tensor product terminology to wavelets

In the remaining part of this chapter we will apply the tensor product construction to wavelets. In particular the spaces V, W from Definition 8.1 are defined from function spaces V_m, W_m, constructed from a given wavelet. We can in particular form the tensor products $\phi_{0,n_1} \otimes \boldsymbol{\phi}_{0,n_2}$. We will assume that

1. the first component ϕ_{0,n_1} has period M (so that $\{\phi_{0,n_1}\}_{n_1=0}^{M-1}$ is a basis for the first component space),
2. the second component ϕ_{0,n_2} has period N (so that $\{\phi_{0,n_2}\}_{n_2=0}^{N-1}$ is a basis for the second component space).

When we speak of $V_0 \otimes V_0$ we thus mean an MN-dimensional space with basis $\{\phi_{0,n_1} \otimes \boldsymbol{\phi}_{0,n_2}\}_{(n_1,n_2)=(0,0)}^{(M-1,N-1)}$, where the coordinate matrices are $M \times N$. This difference in the dimension of the two components is done to allow for images where the number of rows and columns may be different. In the following we will implicitly assume that the component spaces have dimension M and N, to ease notation. If we use that $\boldsymbol{\phi}_{m-1} \oplus \boldsymbol{\psi}_{m-1}$ also is a basis for V_m, we get the following corollary to Theorem 8.5:

Corollary 8.7. Let ϕ, ψ be a scaling function and a mother wavelet. Then the two sets of tensor products given by

$$\boldsymbol{\phi}_m \otimes \boldsymbol{\phi}_m = \{\phi_{m,n_1} \otimes \phi_{m,n_2}\}_{n_1,n_2}$$

and

$$\begin{aligned}
&(\boldsymbol{\phi}_{m-1} \oplus \boldsymbol{\psi}_{m-1}) \otimes (\boldsymbol{\phi}_{m-1} \oplus \boldsymbol{\psi}_{m-1}) \\
&= \{\phi_{m-1,n_1} \otimes \phi_{m-1,n_2}, \\
&\qquad \phi_{m-1,n_1} \otimes \psi_{m-1,n_2}, \\
&\qquad \psi_{m-1,n_1} \otimes \phi_{m-1,n_2}, \\
&\qquad \psi_{m-1,n_1} \otimes \psi_{m-1,n_2}\}_{n_1,n_2}
\end{aligned}$$

are both bases for $V_m \otimes V_m$. This second basis is orthogonal/orthonormal whenever the first basis is.

From this we observe that while the one-dimensional wavelet decomposition splits V_m into a direct sum of the two vector spaces V_{m-1} and W_{m-1}, the corresponding two-dimensional decomposition splits $V_m \otimes V_m$ into a direct sum of four tensor product vector spaces which deserve individual names.

Definition 8.8. We define the following tensor product spaces:

1. The space $W_m^{(0,1)}$ spanned by $\{\phi_{m,n_1} \otimes \psi_{m,n_2}\}_{n_1,n_2}$. This is also called the 01-subband, or the LH-subband,
2. The space $W_m^{(1,0)}$ spanned by $\{\psi_{m,n_1} \otimes \phi_{m,n_2}\}_{n_1,n_2}$. This is also called the 10-subband, or the HL-subband,
3. The space $W_m^{(1,1)}$ spanned by $\{\psi_{m,n_1} \otimes \psi_{m,n_2}\}_{n_1,n_2}$. This is also called the 11-subband, or the HH-subband.

The names L and H stand for *Low-pass filters* and *High-pass filters*, reflecting the interpretation of the corresponding filters G_0, G_1, H_0, H_1 as lowpass/highpass filters. The use of the term subbands comes from the interpretation of these filters as being selective on a certain frequency band. The splitting of $V_m \otimes V_m$ into a direct sum of vector spaces can now be summed up as

$$V_m \otimes V_m = (V_{m-1} \otimes V_{m-1}) \oplus W_{m-1}^{(0,1)} \oplus W_{m-1}^{(1,0)} \oplus W_{m-1}^{(1,1)}. \tag{8.4}$$

Also in the setting of tensor products we refer to $V_{m-1} \otimes V_{m-1}$ as the space of low-resolution approximations. The remaining parts, $W_{m-1}^{(0,1)} \oplus W_{m-1}^{(1,0)} \oplus W_{m-1}^{(1,1)}$, is refered to as the detail space. Note that the coordinate matrix of

$$\sum_{n_1,n_2=0}^{2^{m-1}N}(c_{m-1,n_1,n_2}(\phi_{m-1,n_1}\otimes\phi_{m-1,n_2})+w^{(0,1)}_{m-1,n_1,n_2}(\phi_{m-1,n_1}\otimes\psi_{m-1,n_2})+$$
$$w^{(1,0)}_{m-1,n_1,n_2}(\psi_{m-1,n_1}\otimes\phi_{m-1,n_2})+w^{(1,1)}_{m-1,n_1,n_2}(\psi_{m-1,n_1}\otimes\psi_{m-1,n_2})) \tag{8.5}$$

in the basis $(\boldsymbol{\phi}_{m-1}\oplus\boldsymbol{\psi}_{m-1})\otimes(\boldsymbol{\phi}_{m-1}\oplus\boldsymbol{\psi}_{m-1})$ is

$$\left(\begin{array}{cc|cc} c_{m-1,0,0} & \cdots & w^{(0,1)}_{m-1,0,0} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ \hline w^{(1,0)}_{m-1,0,0} & \cdots & w^{(1,1)}_{m-1,0,0} & \cdots \\ \vdots & \vdots & \vdots & \vdots \end{array}\right). \tag{8.6}$$

We see that the coordinate matrix is split into four submatrices:

- The c-values, i.e. the coordinates for $V_{m-1}\oplus V_{m-1}$. This is the upper left corner in (8.6), and is also called the 00-subband, or the LL-subband.
- The $w^{(0,1)}$-values, i.e. the coordinates for $W^{(0,1)}_{m-1}$. This is the upper right corner in (8.6), and corresponds to the LH-subband.
- The $w^{(1,0)}$-values, i.e. the coordinates for $W^{(1,0)}_{m-1}$. This is the lower left corner in (8.6), and corresponds to the HL-subband.
- The $w^{(1,1)}$-values, i.e. the coordinates for $W^{(1,1)}_{m-1}$. This is the lower right corner in (8.6), and corresponds to the HH-subband.

The $w^{(i,j)}$-values are as in the one-dimensional situation often refered to as wavelet coefficients. Let us consider the Haar wavelet as an example.

Example 8.9. If V_m is the vector space of piecewise constant functions on any interval of the form $[k2^{-m},(k+1)2^{-m})$ (as in the piecewise constant wavelet), $V_m\otimes V_m$ is the vector space of functions in two variables which are constant on any square of the form $[k_12^{-m},(k_1+1)2^{-m})\times[k_22^{-m},(k_2+1)2^{-m})$. Clearly $\phi_{m,k_1}\otimes\phi_{m,k_2}$ is constant on such a square and 0 elsewhere, and these functions are a basis for $V_m\otimes V_m$.

Let us compute the orthogonal projection of $\phi_{1,k_1}\otimes\phi_{1,k_2}$ onto $V_0\otimes V_0$. Since the Haar wavelet is orthonormal, the basis functions in (8.4) are orthonormal, and we can thus use the orthogonal decomposition formula to find this projection. Clearly $\phi_{1,k_1}\otimes\phi_{1,k_2}$ has different support from all except one of $\phi_{0,n_1}\otimes\phi_{0,n_2}$. Since

$$\langle\phi_{1,k_1}\otimes\phi_{1,k_2},\phi_{0,n_1}\otimes\phi_{0,n_2}\rangle=1/2$$

when the supports intersect, we obtain

$$\mathrm{proj}_{V_0\otimes V_0}\phi_{1,k_1}\otimes\phi_{1,k_2} = \begin{cases} \frac{1}{2}(\phi_{0,k_1/2}\otimes\phi_{0,k_2/2}) & \text{when } k_1, k_2 \text{ are even} \\ \frac{1}{2}(\phi_{0,k_1/2}\otimes\phi_{0,(k_2-1)/2}) & \text{when } k_1 \text{ is even, } k_2 \text{ is odd} \\ \frac{1}{2}(\phi_{0,(k_1-1)/2}\otimes\phi_{0,k_2/2}) & \text{when } k_1 \text{ is odd, } k_2 \text{ is even} \\ \frac{1}{2}(\phi_{0,(k_1-1)/2}\otimes\phi_{0,(k_2-1)/2}) & \text{when } k_1, k_2 \text{ are odd} \end{cases}$$

So, in this case there were 4 different formulas, since there were 4 different combinations of even/odd. Let us also compute the projection onto the orthogonal complement of $V_0\otimes V_0$ in $V_1\otimes V_1$, and let us express this in terms of the $\phi_{0,n}, \psi_{0,n}$, like we did in the one-variable case. Also here there are 4 different formulas. When k_1, k_2 are both even we obtain

$$\begin{aligned}
&\phi_{1,k_1}\otimes\phi_{1,k_2} - \mathrm{proj}_{V_0\otimes V_0}(\phi_{1,k_1}\otimes\phi_{1,k_2}) \\
&= \phi_{1,k_1}\otimes\phi_{1,k_2} - \frac{1}{2}(\phi_{0,k_1/2}\otimes\phi_{0,k_2/2}) \\
&= \left(\frac{1}{\sqrt{2}}(\phi_{0,k_1/2}+\psi_{0,k_1/2})\right)\otimes\left(\frac{1}{\sqrt{2}}(\phi_{0,k_2/2}+\psi_{0,k_2/2})\right) - \frac{1}{2}(\phi_{0,k_1/2}\otimes\phi_{0,k_2/2}) \\
&= \frac{1}{2}(\phi_{0,k_1/2}\otimes\phi_{0,k_2/2}) + \frac{1}{2}(\phi_{0,k_1/2}\otimes\psi_{0,k_2/2}) \\
&+ \frac{1}{2}(\psi_{0,k_1/2}\otimes\phi_{0,k_2/2}) + \frac{1}{2}(\psi_{0,k_1/2}\otimes\psi_{0,k_2/2}) - \frac{1}{2}(\phi_{0,k_1/2}\otimes\phi_{0,k_2/2}) \\
&= \frac{1}{2}(\phi_{0,k_1/2}\otimes\psi_{0,k_2/2}) + \frac{1}{2}(\psi_{0,k_1/2}\otimes\phi_{0,k_2/2}) + \frac{1}{2}(\psi_{0,k_1/2}\otimes\psi_{0,k_2/2}).
\end{aligned}$$

Here we have used the relation $\phi_{1,k_i} = \frac{1}{\sqrt{2}}(\phi_{0,k_i/2}+\psi_{0,k_i/2})$, which we have from our first analysis of the Haar wavelet. Checking the other possibilities we find similar formulas for the projection onto the orthogonal complement of $V_0\otimes V_0$ in $V_1\otimes V_1$ when either k_1 or k_2 is odd. In all cases, the formulas use the basis functions for $W_0^{(0,1)}$, $W_0^{(1,0)}$, $W_0^{(1,1)}$. These functions are shown in Figure 8.1, together with the function $\phi\otimes\phi\in V_0\otimes V_0$.

Example 8.10. If we instead use any of the wavelets for piecewise linear functions, the wavelet basis functions are not orthogonal anymore, just as in the one-dimensional case. The new basis functions are shown in Figure 8.2 for the alternative piecewise linear wavelet.

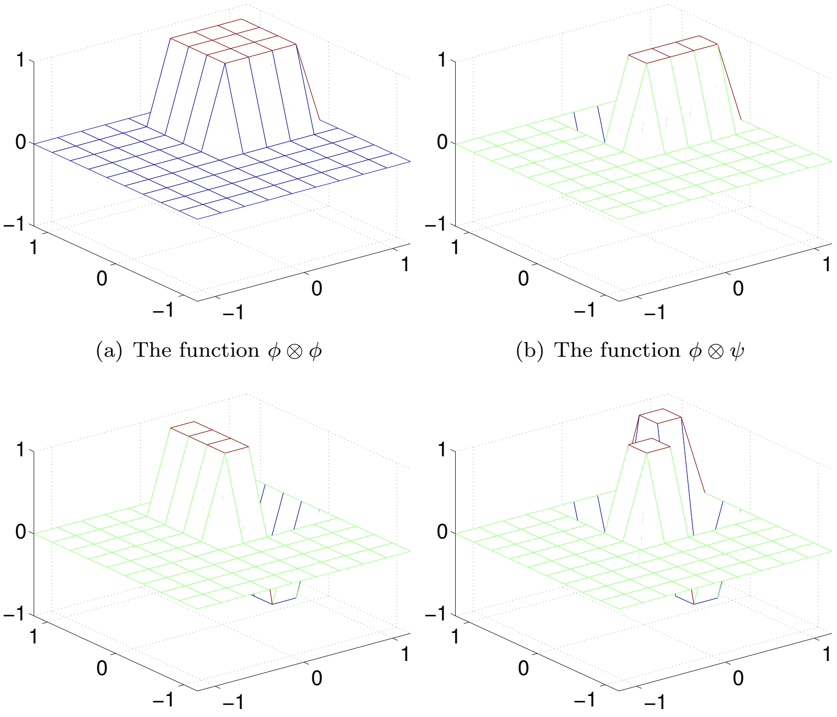
An immediate corollary of Theorem 8.6 is the following:

Corollary 8.11. Let

$$\begin{aligned} A_m &= P_{(\boldsymbol{\phi}_{m-1}\oplus\boldsymbol{\psi}_{m-1})\leftarrow\boldsymbol{\phi}_m} \\ B_m &= P_{\boldsymbol{\phi}_m\leftarrow(\boldsymbol{\phi}_{m-1}\oplus\boldsymbol{\psi}_{m-1})} \end{aligned}$$

be the stages in the DWT and the IDWT, and let

$$X = (c_{m,i,j})_{i,j} \qquad Y = \begin{pmatrix} (c_{m-1,i,j})_{i,j} & (w^{(0,1)}_{m-1,i,j})_{i,j} \\ (w^{(1,0)}_{m-1,i,j})_{i,j} & (w^{(1,1)}_{m-1,i,j})_{i,j} \end{pmatrix} \tag{8.7}$$



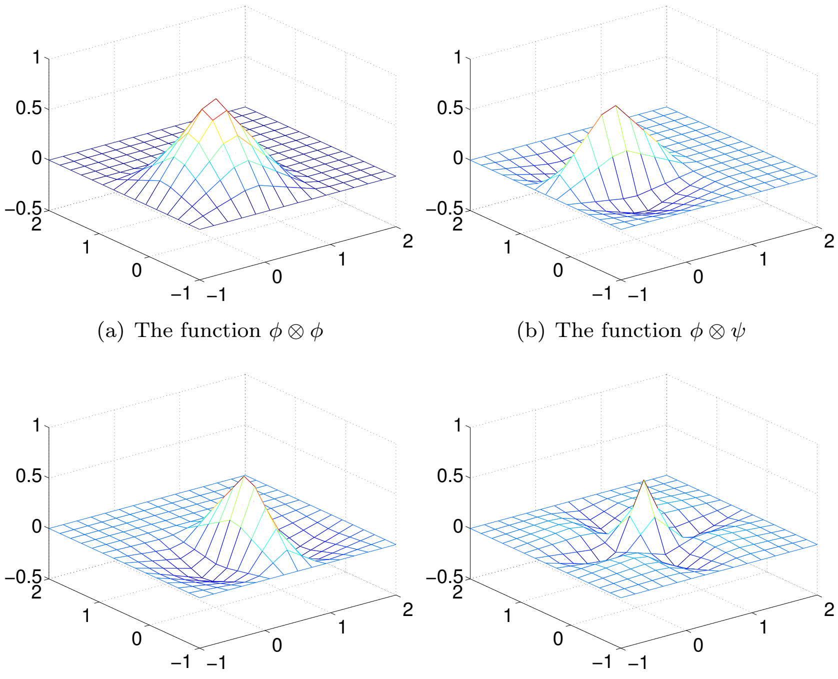
(a) The function $\phi \otimes \phi$

(b) The function $\phi \otimes \psi$

(c) The function $\psi \otimes \phi$

(d) The function $\psi \otimes \psi$

Figure 8.1: The basis functions for $(V_0 \otimes V_0) \oplus W_0^{(0,1)} \oplus W_0^{(1,0)} \oplus W_0^{(1,1)}$ for the Haar wavelet.



(a) The function $\phi \otimes \phi$

(b) The function $\phi \otimes \psi$

(c) The function $\psi \otimes \phi$

(d) The function $\psi \otimes \psi$

Figure 8.2: The basis functions for $(V_0 \otimes V_0) \oplus W_0^{(0,1)} \oplus W_0^{(1,0)} \oplus W_0^{(1,1)}$ for the alternative piecewise linear wavelet.

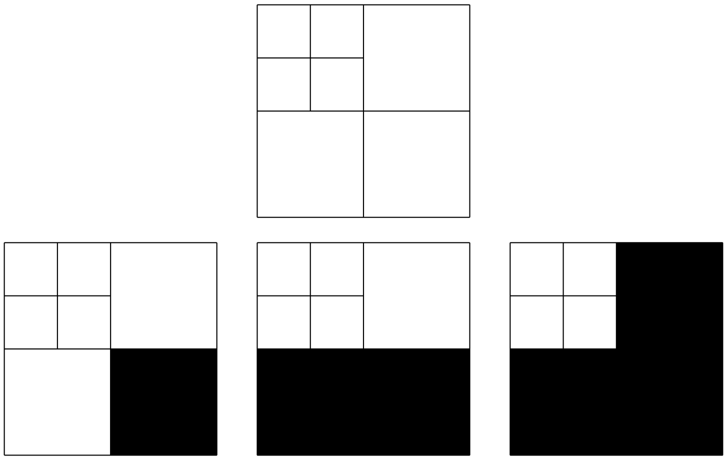


Figure 8.3: Graphical representation of neglecting the wavelet coefficients at the first level. In the first figure, all coefficients are present. Then we remove the ones from $W_1^{(1,1)}$, $W_1^{(1,0)}$, and $W_1^{(0,1)}$, respectively.

be the coordinate matrices in $\boldsymbol{\phi}_m \otimes \boldsymbol{\phi}_m$, and $(\boldsymbol{\phi}_{m-1} \oplus \boldsymbol{\psi}_{m-1}) \otimes (\boldsymbol{\phi}_{m-1} \oplus \boldsymbol{\psi}_{m-1})$, respectively. Then

$$Y = A_m X A_m^T \tag{8.8}$$

$$X = B_m Y B_m^T \tag{8.9}$$

By the m-level two-dimensional DWT/IDWT (or DWT2/IDWT2) we mean the change of coordinates where this is repeated m times as in a DWT/IDWT.

Each stage in DWT2 and IDWT2 can now be implemented by substituting the matrices A_m, B_m above into Theorem 7.13. This implementation can reuse an efficient implementation of the one-dimensional DWT/IDWT. When using many levels of the DWT2, the next stage is applied only to the upper left corner of the matrix, just as the DWT at the next stage only is applied to the first part of the coordinates. At each stage, the upper left corner of the coordinate matrix (which gets smaller at each iteration), is split into four equally big parts. To illustrate this, assume that we have a coordinate matrix, and that we perform the change of basis at two levels, i.e. we start with a coordinate matrix in the basis $\boldsymbol{\phi}_2 \otimes \boldsymbol{\phi}_2$. Figure 8.3 illustrates first the collection of all coordinates, and then the resulting collection of coordinates after removing subbands at the first level successively. The subbands which have been removed are indicated with a black colour. Figure 8.4 illustrates in the same way incremental removal of the subbands at the second level.

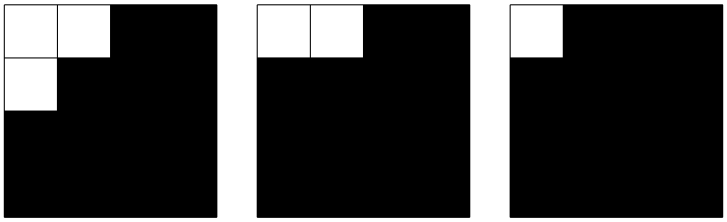


Figure 8.4: Graphical representation of neglecting the wavelet coefficients at the second level. We remove the ones from $W_2^{(1,1)}$, $W_2^{(1,0)}$, and $W_2^{(0,1)}$, respectively.

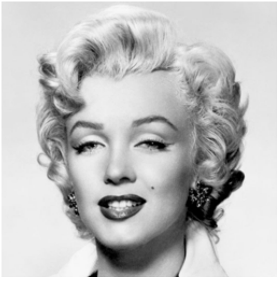


Figure 8.5: Image of Marilyn Monroe, used in our experiments.

Let us round off this section with some experiments with images using the wavelets we have considered[1]. Our theory is applied to images in the following way: We visualize the pixels in the image as coordinates in the basis $\boldsymbol{\phi}_m \otimes \boldsymbol{\phi}_m$ (so that the image has size $(2^m M) \times (2^m N)$), and perform change of coordinates with the DWT2. We can then, just as we did for sound, and for the DCT/DFT-values in images, either set the the part from the $W_k^{(i,j)}$-spaces (the detail) to zero, or the part from $V_0 \otimes V_0$ (the $M \times N$-low-resolution approximation) to zero, depending on whether we want to inspect the detail or the low-resolution approximation in the image. Finally we apply the IDWT2 to end up with coordinates in $\boldsymbol{\phi}_m \otimes \boldsymbol{\phi}_m$ again, and display the image with pixel values being these coordinates.

Example 8.12 (Creating thumbnail images)**.** Let us take the sample image of Marilyn Monroe, shown in Figure 8.5, and first use the Haar wavelet. In Exercise 1 you will be asked to implement a function which compute DWT2 for the Haar wavelet. After the DWT2, the upper left submatrices represent the low-resolution approximations from $V_{m-1} \otimes V_{m-1}$, $V_{m-2} \otimes V_{m-2}$, and so on. We can now use the following code to store the low-resolution approximation for $m = 1$:

[1] Note also that Matlab has a wavelet toolbox which could be used for these purposes, but we will not go into the usage of this.

(a) (b) (c) (d)

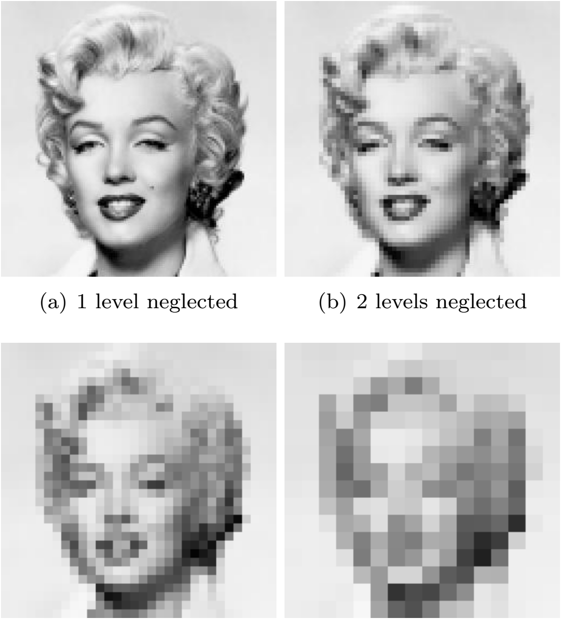
Figure 8.6: The corresponding thumbnail images for the Image of Marilyn Monroe, obtained with a DWT of 1, 2, 3, and 4 levels.

```
img = double(imread('mm.gif','gif'));
[l1,l2]=size(img);
x=DWT2HaarImpl(img,1);
x=x(1:(l1/2),1:(l2/2));
imwrite(uint8(x),'mm1thumbnail.jpg','jpg');
```

In Figure 8.6 the results are shown up to 4 resolutions.

Example 8.13 (Detail and low-resolution approximations with the Haar wavelet)**.** In Exercise 2 you will be asked to implement a function `showDWTlower` which displays the low-resolution approximations to our image test file `mm.gif`, for the Haar wavelet, using functions we implement in the exercises. Let us take a closer look at the images generated. Above we viewed the low-resolution approximation as a smaller image. Let us compare with the image resulting from setting the wavelet detail coefficients to zero, and viewing the result as an image of the same size. In particular, let us neglect the wavelet coefficients as pictured in Figure 8.3 and Figure 8.4. Since the Haar wavelet has few vanishing moments, we should expect that the lower order resolution approximations from V_0 are worse when m increase. Figure 8.7 confirms this for the lower order resolution approximations. Alternatively, we should see that the higher order detail spaces contain more information. In Exercise 3 you will be asked to implement a function `showDWTlowerdifference` which displays the detail components in the image for a given resolution m for the Haar wavelet. The new images when this function is used are shown in Figure 8.8. The black colour indicates values which are close to 0. In other words, most of the coefficients are close to 0, which reflects one of the properties of the wavelet.

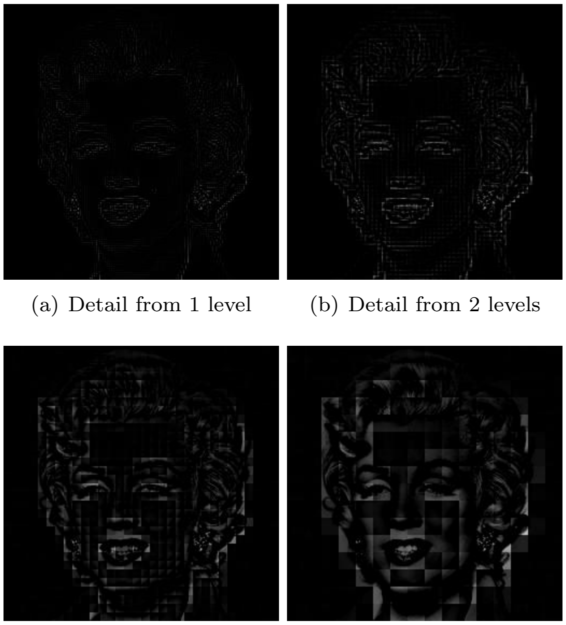
Example 8.14 (The alternative piecewise linear wavelet, and neglecting bands in the detail spaces)**.** In Exercise 5 you will be asked to implement a function `showDWTfilterslower` which displays the low-resolution approximations to our image test file `mm.gif`, for any type of wavelet, using functions we implement in the exercises. With this function we can display the result for all the wavelets we have considered up to now, in succession, and at a given resolution, with the following code:



(a) 1 level neglected (b) 2 levels neglected

(c) 3 levels neglected (d) 4 levels neglected

Figure 8.7: Image of Marilyn Monroe, with higher levels of detail neglected for the Haar wavelet.



(a) Detail from 1 level

(b) Detail from 2 levels

(c) Detail from 3 levels

(d) Detail from 4 levels

Figure 8.8: The corresponding detail for the images in Figure 8.7, with the Haar wavelet.

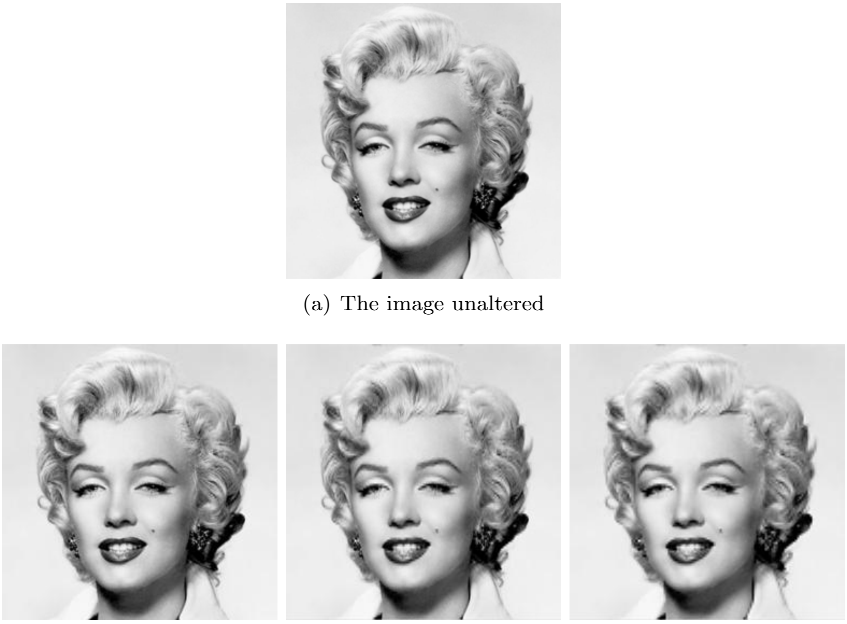
```
function showDWTall(m)
disp('Haar wavelet');
showDWTlower(m);
disp('Wavelet for piecewise linear functions');
showDWTfilterslower(m,[sqrt(2)],...
    [sqrt(2) -1/sqrt(2)],...
    [1/sqrt(2) 1/(2*sqrt(2))],...
    [1/sqrt(2)]);
disp('Wavelet for piecewise linear functions, alternative  version');
showDWTfilterslower(m,[3/(2*sqrt(2)) 1/(2*sqrt(2)) -1/(4*sqrt(2))],...
   [sqrt(2) -1/sqrt(2)],...
   [1/sqrt(2) 1/(2*sqrt(2))],...
   [3/(4*sqrt(2)) -1/(4*sqrt(2)) -1/(8*sqrt(2))]);
```

The call to `showDWTlower` first displays the result, using the Haar wavelet. The code then moves on to the piecewise linear wavelet and the alternative piecewise linear wavelet. In Example 5.50 we found the parameters `h0,h1,g0,g1` to use for these wavelets. These are then sent as parameters to the function `showDWTfilterslower`, which displays the corresponding image when a wavelet with a given set of filter coefficients are used.

Let us use some other filter than the Haar wavelet. Once the filters are known, and the image has been read from file, the functions `DWT2Impl` and `IDWT2Impl` from Exercise 4, a new image with detail at m levels neglected can be produced with the following code:

```
x=DWT2Impl(h0,h1,img,m);
newx=zeros(size(x));
newx(1:(l1/2^m),1:(l2/2^m))=x(1:(l1/2^m),1:(l2/2^m));
x=IDWT2Impl(g0,g1,newx,m);
```

We can repeat this for various number of levels, and compare the different images. We can also neglect only parts of the detail, since it at each level is grouped into three bands ($W_m^{(1,1)}$, $W_m^{(1,0)}$, $W_m^{(0,1)}$), contrary to the one-dimensional case. Let us use the alternative piecewise linear wavelet. This is used in the JPEG2000 standard for lossless compression, since the filter coefficients here turned out to be dyadic fractions, which are suitable for lossless operations. The resulting images when the bands on the first level indicated in Figure 8.3 are removed are shown in Figure 8.9. The resulting images when the bands on the second level indicated in Figure 8.4 are removed are shown in Figure 8.10. The image is seen still to resemble the original one, even after two levels of wavelets coefficients have been neglected. This in itself is good for compression purposes, since we may achieve compression simply by dropping the given coefficients. However, if we continue to neglect more levels of coefficients, the result will look poorer. In Figure 8.11 we have also shown the resulting image after the third and fourth level of detail have been neglected. Although we still can see details in the image, the quality in the image is definitely poorer. Although the quality is poorer when we neglect levels of wavelet coefficients, all information is kept if



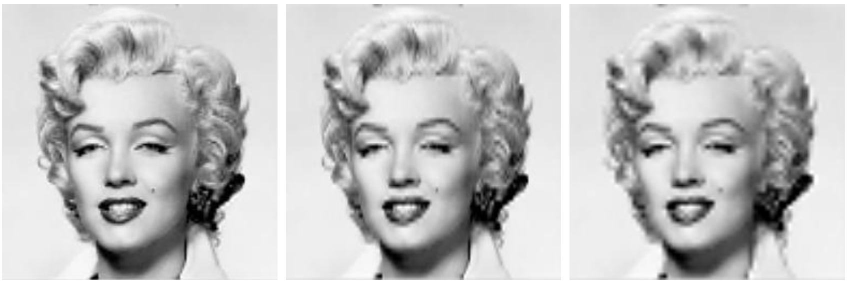
(a) The image unaltered

(b) Resulting image after neglecting detail in $W_1^{(1,1)}$, as illustrated in Figure 8.3(b)

(c) Resulting image after neglecting also detail in $W_1^{(1,0)}$, as illustrated in Figure 8.3(c).

(d) Resulting image after neglecting also detail in $W_1^{(0,1)}$, as illustrated in Figure 8.3(d).

Figure 8.9: Image of Marilyn Monroe, with various bands of detail at the first level neglected. The alternative piecewise linear wavelet was used.

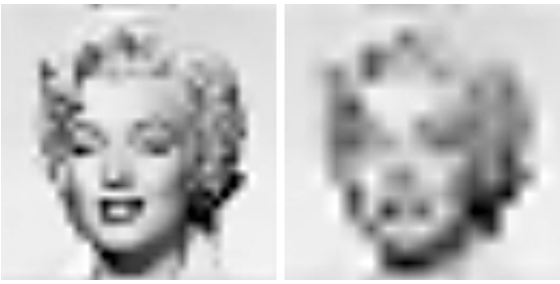


(a) Resulting image after also neglecting detail in $W_2^{(1,1)}$, as illustrated in Figure 8.10(a).

(b) Resulting image after also neglecting detail in $W_2^{(1,0)}$, as illustrated in Figure 8.10(b).

(c) Resulting image after also neglecting detail in $W_2^{(0,1)}$, as illustrated in Figure 8.10(c).

Figure 8.10: Image of Marilyn Monroe, with various bands of detail at the second level neglected. The alternative piecewise linear wavelet was used.



(a) 3 levels neglected (b) 4 levels neglected

Figure 8.11: Image of Marilyn Monroe, with higher levels of detail neglected. The alternative piecewise linear wavelet was used.

we additionally include the detail/bands. In Figure 8.12, we have shown the corresponding detail for Figure 8.9(d), Figure 8.10(c), and Figure 8.11. Clearly, more detail can be seen in the image when more of the detail is included.

Example 8.15. The JPEG2000 standard uses a more advanced wavelet for lossy compression than the piecewise linear wavelet used for lossless compression. This uses a more advanced scaling function ϕ than the ones we have used, and adds more vanishing momemnts to it, similarly, to how we did for the alternative piecewise linear wavelet. We will not deduce the expression for this wavelet[2], only state that it is determined by the filters

$$H_0 = \{0.1562, -0.0985, -0.4569, 1.5589, \underline{3.5221}, 1.5589, -0.4569, -0.0985, 0.1562\}$$
$$H_1 = \{0.0156, -0.0099, \underline{-0.1012}, 0.1909, -0.1012, -0.0099, 0.0156\}.$$

for the DWT, and the filters

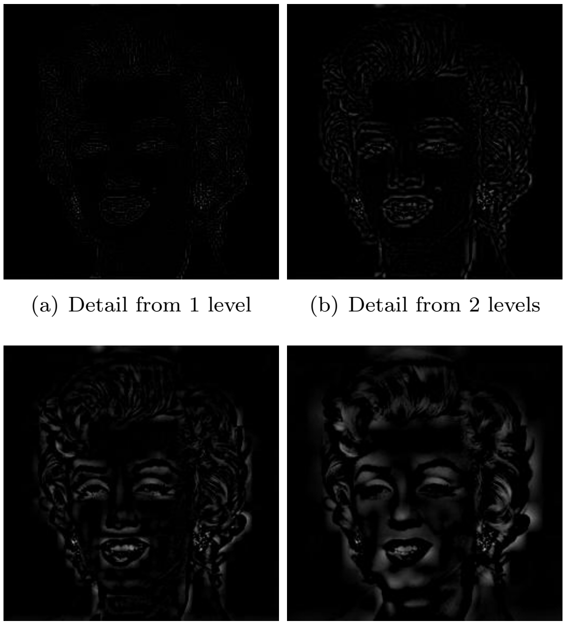
$$G_0 = \{-0.0156, -0.0099, 0.1012, \underline{0.1909}, 0.1012, -0.0099, -0.0156\}$$
$$G_1 = \{0.1562, 0.0985, -0.4569, \underline{-1.5589}, 3.5221, -1.5589, -0.4569, 0.0985, 0.1562\}$$

for the IDWT. The length of the filters are 9 and 7 in this case, so that this wavelet is called the CDF 9/7 wavelet (CDF represents the first letters in the names of the inventors of the wavelet). The corresponding frequency responses are

$$\lambda_{G_0}(\omega) = -0.0312\cos(3\omega), -0.0198\cos(2\omega), 0.2024\cos\omega + 0.1909$$
$$\lambda_{H_0}(\omega) = 0.3124\cos(4\omega) - 0.1970\cos(3\omega) - 0.9138\cos(2\omega), 3.1178\cos\omega + 3.5221.$$

In Figure 8.13 we have plotted these. It is seen that both filters are lowpass

[2] it can be obtained by hand, but is more easily automated on a computer



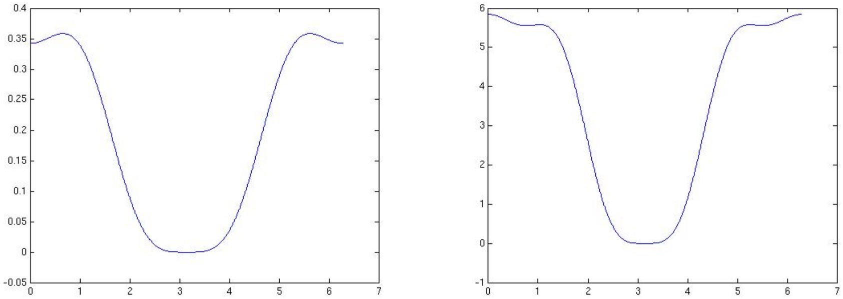
(a) Detail from 1 level

(b) Detail from 2 levels

(c) Detail from 3 levels

(d) Detail from 4 levels

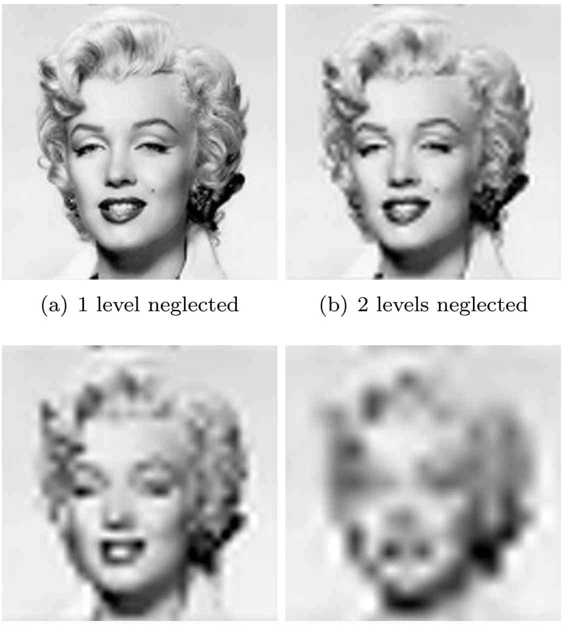
Figure 8.12: The corresponding detail for the image of Marilyn Monroe. The alternative piecewise linear wavelet was used.



(a) $\lambda_{G_0}(\omega)$

(b) $\lambda_{H_0}(\omega)$

Figure 8.13: The frequency responses for the filters used in lossy compression with JPEG2000.



(a) 1 level neglected

(b) 2 levels neglected

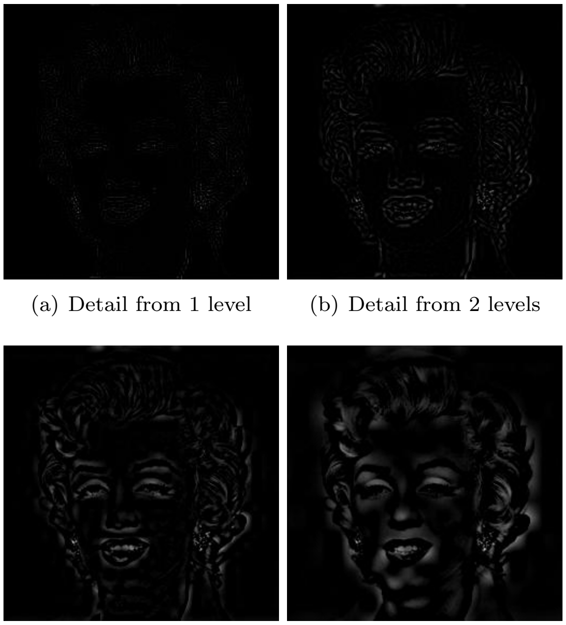
(c) 3 levels neglected

(d) 4 levels neglected

Figure 8.14: Image of Marilyn Monroe, with higher levels of detail neglected. The CDF 9/7 wavelet was used.

filters also here, and that they are closer to an ideal bandpass filter. Although there exist wavelets with better properties when it comes to compression, it can be shown that this wavelet has a good tradeoff between complexity and compression, and is therefore much used. With the CDF 9/7 wavelet, we should see improved images when we discared the detail in the images. Figure 8.14 confirms this for the lower resolution spaces, while Figure 8.15 confirms this for the higher order detail spaces.

As mentioned, the procedure developed in this section for applying a wavelet transform to an image with the help of the tensor product construction, is adopted in the JPEG2000 standard. This lossy (can also be used as lossless) image format was developed by the Joint Photographic Experts Group and published in 2000. After significant processing of the wavelet coefficients, the final coding with JPEG2000 uses an advanced version of arithmetic coding. At the cost of increased encoding and decoding times, JPEG2000 leads to as much as 20 % improvement in compression ratios for medium compression rates, possibly more for high or low compression rates. The artefacts are less visible than in JPEG and appear at higher compression rates. Although a number of components in JPEG2000 are patented, the patent holders have agreed that the core software should be available free of charge, and JPEG2000 is part of most



(a) Detail from 1 level (b) Detail from 2 levels

(c) Detail from 3 levels (d) Detail from 4 levels

Figure 8.15: The corresponding detail for the image of Marilyn Monroe. The CDF 9/7 wavelet was used.

Linux distributions. However, there appear to be some further, rather obscure, patents that have not been licensed, and this may be the reason why JPEG2000 is not used more. The extension of JPEG2000 files is `.jp2`.

Exercises for Section 8.1

Ex. 1 — Implement functions

```
function xnew=DWT2HaarImpl(x,m)
function x=IDWT2HaarImpl(xnew,m)
```

which implements DWT2 and IDWT2 for the Haar-wavelet.

Ex. 2 — In this exercise we will experiment with applying an m-level DWT to a sound file.

a. Write a function

   ```
   function showDWTlower(m)
   ```

   which
   1. reads the image file `mm.gif`,
   2. performs an m-level DWT2 to the image samples using the function `DWT2HaarImpl`,
   3. sets all wavelet coefficients representing detail to zero (i.e. keep only wavelet coefficients from $V_0 \otimes V_0$),
   4. performs an IDWT2 on the resulting coefficients using the function `IDWT2HaarImpl`,
   5. displays the resuting image.

b. Run the function `showDWTlower` for different values of m. Describe what you see for different m. degraded? Compare with what you saw with the function `showDCThigher` in Exercise 2, where you performed a DCT on the image samples instead, and set DCT coefficients below a given threshold to zero.

c. Do the image samples returned by `showDWTlower` lie in $[0, 255]$?

Ex. 3 — Repeat Exercise 2, but this time instead keep only wavelet coefficients from the detail spaces. Call the new function `showDWTlowerdifference`. What kind of image do you see? Can you recognize the original image in what you see?

Ex. 4 — Implement functions

```
function xnew=DWT2Impl(h0,h1,x,m)
function x=IDWT2Impl(g0,g1,xnew,m)
```

where `DWT2Impl` performs the m-level DWT2 on the image given by `x`, and `IDWT2Impl` performs the m-level IDWT2. The functions should at each stage call `DWTImpl` and `IDWTImpl` with $m = 1$, which you implemented in exercises 4 and 5, and each call to these functions should alter the appropriate upper left submatrix in the coordinate matrix. You can assume that the number of rows and columns both are powers of 2 in this matrix (but they may not be equal).

Ex. 5 — Write a function

```
function showDWTfilterslower(m,h0,h1,g0,g1)
```

which reimplements the function `showDWTlower` from Exercise 2 so that it takes as input the positive parts of the four different filters as in Section 5.6. Look at the result using the different wavelets we have encountered and for different m, using the code from Example 8.13. Can you see any difference from the Haar wavelet? If so, which wavelet gives the best image quality?

Ex. 6 — In this exercise we will change the code in Example 8.13 so that it instead only shows the contribution from the detail spaces.

a. Reimplement the function you made in Exercise 5 so that it instead shows the contribution from the detail spaces. Call the new function `showDWTfilterslowerdifference`.

b. In Exercise 3 we implemented a function `showDWTlowerdifference` for looking at the detail/error when the Haar wavelet is used. In the function `showDWTall` from Example 8.13, replace `showDWTlower` and `showDWTfilterslower` with `showDWTlowerdifference` and `showDWTfilterslowerdifference`. Describe the images you see for different m. Try to explain why the images seem to get clearer when you increase m.

8.2 Summary

We extended the tensor product construction to functions by defining the tensor product of functions as a function in two variables. We explained with some examples that this made the tensor product formalism useful for approximation of functions in several variables. We extended the wavelet transform to the tensor product setting, so that it too could be applied to images. We also performed several experiments on our test image, such as creating low-resolution images and neglecting wavelet coefficients. We also used different wavelets, such as the Haar wavelet, the alternative piecewise linear wavelet, and a new wavelet which is much used in lossy ompression of images. The experiments confirmed what we previously have proved, that wavelets with many vanishing moments are better suited for compression purposes.