Example Solutions for Homework Assignment 4 (H4)

Problem 1: Transformations

(a) **General remark:** Consider a matrix $A \in \mathbb{R}^{N\times N}$ with entries $a_{i,j}$ ($i,j \in [0, N-1]$) and vectors $\boldsymbol{x} = (x_0, \ldots, x_{N-1})^\top$, $\boldsymbol{y} = (y_0, \ldots, y_{N-1})^\top \in \mathbb{R}^N$. A matrix-vector multiplication $\boldsymbol{y} = A\boldsymbol{x}$ can be written componentwise for the entries of $\boldsymbol{y}$ as

$$y_p = \sum_{m=0}^{N-1} a_{p,m} x_m \qquad \forall p = 0, \ldots, N-1 \,.$$

We will exploit this for the construction of the appropriate basis transformation matrices.

(i) Using the notation given on in the problem task, the DFT for a signal with length $N = 8$ is given by

$$g_p = \frac{1}{\sqrt{8}} \sum_{m=0}^{7} f_m \exp\left(-\frac{i2\pi pm}{8}\right) \qquad p = 0, \ldots, 7 \,,$$

and the corresponding inverse transform by

$$f_m = \frac{1}{\sqrt{8}} \sum_{p=0}^{7} g_p \exp\left(\frac{i2\pi pm}{8}\right) \qquad m = 0, \ldots, 7 \,.$$

Thus the entries of the transformation matrix are given as:

$$a_{p,m} = \frac{1}{\sqrt{8}} \exp\left(-\frac{i\pi pm}{4}\right) \,,$$

and for the back transformation matrix:

$$b_{m,p} = \frac{1}{\sqrt{8}} \exp\left(\frac{i\pi pm}{4}\right) \,.$$

for $p, m = 0, \ldots, 7$. Simplifying the entries we get:

$$A = \frac{1}{\sqrt{8}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \bar{z} & -i & -z & -1 & -\bar{z} & i & z \\ 1 & -i & -1 & i & 1 & -i & -1 & i \\ 1 & -z & i & \bar{z} & -1 & z & -i & -\bar{z} \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -\bar{z} & -i & z & -1 & \bar{z} & i & -z \\ 1 & i & -1 & -i & 1 & i & -1 & -i \\ 1 & z & i & -\bar{z} & -1 & -z & -i & \bar{z} \end{pmatrix}$$

where $\bar{z}$ is the complex conjugate of $z = \frac{1+i}{\sqrt{2}}$.

$$B = \frac{1}{\sqrt{8}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & z & i & -\bar{z} & -1 & -z & -i & \bar{z} \\ 1 & i & -1 & -i & 1 & i & -1 & -i \\ 1 & -\bar{z} & -i & z & -1 & \bar{z} & i & -z \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -z & i & \bar{z} & -1 & z & -i & -\bar{z} \\ 1 & -i & -1 & i & 1 & -i & -1 & i \\ 1 & \bar{z} & -i & -z & -1 & -\bar{z} & i & z \end{pmatrix}$$

(ii) The DCT is given by the formula

$$g_p \quad = \quad \sum_{m=0}^{7} f_m c_p \cos\left(\frac{(2m+1)p\pi}{16}\right) \qquad p = 0, \ldots, 7\ ,$$

and the corresponding back transformation

$$f_m \quad := \quad \sum_{p=0}^{7} g_p c_p \cos\left(\frac{(2m+1)p\pi}{16}\right) \qquad m = 0, \ldots, 7\ .$$

with $c_0 := \sqrt{\frac{1}{8}}$ and $c_p := \sqrt{\frac{1}{4}}$ for $p > 0$. So the entries of the transformation and back transformation matrix are in this case given by:

$$a_{p,m} = c_p \cos\left(\frac{(2m+1)p\pi}{16}\right) \quad \text{and} \quad a_{m,p} = c_p \cos\left(\frac{(2m+1)p\pi}{16}\right)\ .$$

Simplifying this, we get explicitly

$$
A = \sqrt{\frac{1}{8}} \begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
a_1 & b_1 & c_1 & d_1 & -d_1 & -c_1 & -b_1 & -a_1 \\
a_2 & b_2 & -b_2 & -a_2 & -a_2 & -b_2 & b_2 & a_2 \\
b_1 & -d_1 & -a_1 & -c_1 & c_1 & a_1 & d_1 & -b_1 \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
c_1 & -a_1 & d_1 & b_1 & -b_1 & -d_1 & a_1 & -c_1 \\
b_2 & -a_2 & a_2 & -b_2 & -b_2 & a_2 & -a_2 & b_2 \\
d_1 & -c_1 & b_1 & -a_1 & a_1 & -b_1 & c_1 & -d_1
\end{pmatrix}
$$

with

$$
\begin{aligned}
a_1 &= \sqrt{2}\cos\left(\frac{\pi}{16}\right) & a_2 &= \sqrt{2}\cos\left(\frac{2\pi}{16}\right) \\
b_1 &= \sqrt{2}\cos\left(\frac{3\pi}{16}\right) & b_2 &= \sqrt{2}\cos\left(\frac{6\pi}{16}\right) \\
c_1 &= \sqrt{2}\cos\left(\frac{5\pi}{16}\right) & d_1 &= \sqrt{2}\cos\left(\frac{7\pi}{16}\right)
\end{aligned}
$$

The corresponding back transformation matrix is

$$
B = \sqrt{\frac{1}{8}} \begin{pmatrix}
1 & a_1 & a_2 & b_1 & 1 & c_1 & b_2 & d_1 \\
1 & b_1 & b_2 & -d_1 & -1 & -a_1 & -a_2 & -c_1 \\
1 & c_1 & -b_2 & -a_1 & -1 & d_1 & a_2 & b_1 \\
1 & d_1 & -a_2 & -c_1 & 1 & b_1 & -b_2 & -a_1 \\
1 & -d_1 & -a_2 & c_1 & 1 & -b_1 & -b_2 & a_1 \\
1 & -c_1 & -b_2 & a_1 & -1 & -d_1 & a_2 & -b_1 \\
1 & -b_1 & b_2 & d_1 & -1 & a_1 & -a_2 & c_1 \\
1 & -a_1 & a_2 & -b_1 & 1 & -c_1 & b_2 & -d_1
\end{pmatrix}
$$

(iii) Regarding the DWT, we do not have a formula, which depends on the vector index p directly, but instead the basis vectors depend on some scaling and shift index. For a signal of length 8 we have to consider the 8 basis vectors

$$
\begin{aligned}
&\mathbf{\Phi}_{3,0}, \\
&\mathbf{\Psi}_{3,0}, \\
&\mathbf{\Psi}_{2,0}, \mathbf{\Psi}_{2,1}, \\
&\mathbf{\Psi}_{1,0}, \mathbf{\Psi}_{1,1}, \mathbf{\Psi}_{1,2}, \mathbf{\Psi}_{1,3}
\end{aligned}
$$

(see also Lecture 7, Slide 13). The Wavelet coefficients of the

DWT are given by a vector-vector multiplication:

$$c_{3,0} = \boldsymbol{f}^\top \boldsymbol{\Phi}_{3,0} = \sum_{m=0}^{7} f_m(\Phi_{3,0})_m \;, \quad d_{1,0} = \boldsymbol{f}^\top \boldsymbol{\Psi}_{1,0} = \sum_{m=0}^{7} f_m(\Psi_{1,0})_m$$

$$d_{3,0} = \boldsymbol{f}^\top \boldsymbol{\Psi}_{3,0} = \sum_{m=0}^{7} f_m(\Psi_{3,0})_m \;, \quad d_{1,1} = \boldsymbol{f}^\top \boldsymbol{\Psi}_{1,1} = \sum_{m=0}^{7} f_m(\Psi_{1,1})_m$$

$$d_{2,0} = \boldsymbol{f}^\top \boldsymbol{\Psi}_{2,0} = \sum_{m=0}^{7} f_m(\Psi_{2,0})_m \;, \quad d_{1,2} = \boldsymbol{f}^\top \boldsymbol{\Psi}_{1,2} = \sum_{m=0}^{7} f_m(\Psi_{1,2})_m$$

$$d_{2,1} = \boldsymbol{f}^\top \boldsymbol{\Psi}_{2,1} = \sum_{m=0}^{7} f_m(\Psi_{2,1})_m \;, \quad d_{1,3} = \boldsymbol{f}^\top \boldsymbol{\Psi}_{1,3} = \sum_{m=0}^{7} f_m(\Psi_{1,3})_m$$

with $\boldsymbol{g} = (c_{3,0}, d_{3,0}, d_{2,0}, d_{2,1}, d_{1,0}, d_{1,1}, d_{1,2}, d_{1,3})^\top$.

To get the basis vectors explicitly, the corresponding continous functions $\Psi_{j,k}(x)$ and $\Phi_{j,k}(x)$ have to be sampled at the equidistant grid points $\left\{\frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \frac{7}{2}, \frac{9}{2}, \frac{11}{2}, \frac{13}{2}, \frac{15}{2}\right\})$. That means for example: For $\boldsymbol{\Phi_{3,0}}$, the continous version is given by

$$\Phi_{3,0}(x) \quad = \quad \frac{1}{\sqrt{8}} \Phi\left(\frac{x}{8} - 0\right)$$

which results in the vector

$$\boldsymbol{\Phi_{3,0}} \quad = \quad \frac{1}{\sqrt{8}}(1,1,1,1,1,1,1,1)^\top.$$

For $\boldsymbol{\Psi_{3,0}}$ we have

$$\Psi_{3,0}(x) \quad = \quad \frac{1}{\sqrt{8}} \Psi\left(\frac{x}{8}\right)$$

which results in the vector

$$\boldsymbol{\Psi_{3,0}} \quad = \quad \frac{1}{\sqrt{8}}(1,1,1,1,-1,-1,-1,-1)^\top.$$

The rest of the vectors can be computed analogously. The resulting transformation matrix can now be stated explicitly as:

$$A \quad = \quad \begin{pmatrix} \boldsymbol{\Phi}_{3,0}^\top \\ \boldsymbol{\Psi}_{3,0}^\top \\ \boldsymbol{\Psi}_{2,0}^\top \\ \boldsymbol{\Psi}_{2,1}^\top \\ \boldsymbol{\Psi}_{1,0}^\top \\ \boldsymbol{\Psi}_{1,1}^\top \\ \boldsymbol{\Psi}_{1,2}^\top \\ \boldsymbol{\Psi}_{1,3} \end{pmatrix} = \sqrt{\frac{1}{8}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{pmatrix}$$

On the slides of the lecture, there is no explicit description how to backtransform the coefficients without using the Fast Wavelet Transform. However, we know that the backtransformation should result in the original signal $\boldsymbol{f}$ again. Thus $\boldsymbol{f} = B\boldsymbol{g} = B(A\boldsymbol{f})$ has to hold. In Assignement C4, Problem 1, we have proven that the DWT-vectors form an orthnormal basis. Thus the backtransformation matrix is obtained by using $B := A^\top$ (see also part (b)):

$$
\begin{aligned}
B &= \sqrt{\frac{1}{8}}\,(\boldsymbol{\Psi}_{3,0}, \boldsymbol{\Phi}_{3,0}, \boldsymbol{\Phi}_{2,0}, \boldsymbol{\Phi}_{2,1}, \boldsymbol{\Phi}_{1,0}, \boldsymbol{\Phi}_{1,1}, \boldsymbol{\Phi}_{1,2}, \boldsymbol{\Phi}_{1,3}) \\
&= \sqrt{\frac{1}{8}} \begin{pmatrix}
1 & 1 & \sqrt{2} & 0 & 2 & 0 & 0 & 0 \\
1 & 1 & \sqrt{2} & 0 & -2 & 0 & 0 & 0 \\
1 & 1 & -\sqrt{2} & 0 & 0 & 2 & 0 & 0 \\
1 & 1 & -\sqrt{2} & 0 & 0 & -2 & 0 & 0 \\
1 & -1 & 0 & \sqrt{2} & 0 & 0 & 2 & 0 \\
1 & -1 & 0 & \sqrt{2} & 0 & 0 & -2 & 0 \\
1 & -1 & 0 & -\sqrt{2} & 0 & 0 & 0 & 2 \\
1 & -1 & 0 & -\sqrt{2} & 0 & 0 & 0 & -2
\end{pmatrix}
\end{aligned}
$$

(b) As exploited in part (a)(iii), we can also observe in part (a)(ii) that $B := A^\top$. For part (a)(i) we see, that B is the conjugate transpose of A, i.e. $B := \bar{A}^\top$. Furthermore, by construction of A and B, we have

$$\boldsymbol{f} \;=\; B\boldsymbol{g} = B(A\boldsymbol{f}) = BA\boldsymbol{f}$$

which means BA has to be the identity matrix I which is only true for $B := A^{-1}$. These observations are resonable, since all matrices A contain row vectors, which form orthonormal basises with respect to the corresponding inner products (see also Assignments H3, Problem 2 and C4, Problem 1). Such matrices are called orthogonal matrices and $A^\top = A^{-1}$ holds by definition.

Problem 2: Transformations of Shifted Signals

(a) Since the given signals have length 4, the DFT of our signals is given by:

$$\widehat{f}_p \;=\; \frac{1}{2} \cdot \sum_{m=0}^{3} f_m \cdot \exp\left(\frac{-i\pi pm}{2}\right).$$

Plugging in Euler's formula, we get:

$$\widehat{f_p} \;=\; \frac{1}{2} \cdot \sum_{m=0}^{3} f_m \cdot \left(\cos\left(\frac{-\pi p m}{2} \right) + i \cdot \sin\left(\frac{-\pi p m}{2} \right) \right)$$

Since sin and cos are both 2π-periodic, we only need to consider four cases to evaluate the exponential term:

$p \cdot m \mod 4$	0	1	2	3
$\cos\left(\frac{-\pi pm}{2}\right) + i \cdot \sin\left(\frac{-\pi pm}{2}\right)$	1	$-i$	-1	i

Thereby, we can easily compute the DFT of our signals f_1 and f_2:

$$\begin{aligned}
\widehat{f_0} &= 1/2 \cdot (6+8+2+4) &&= 10 \\
\widehat{f_1} &= 1/2 \cdot (6 \cdot 1 + 8 \cdot (-i) + 2 \cdot (-1) + 4 \cdot i) &&= 2-2i \\
\widehat{f_2} &= 1/2 \cdot (6 \cdot 1 + 8 \cdot (-1) + 2 \cdot 1 + 4 \cdot (-1)) &&= -2 \\
\widehat{f_3} &= 1/2 \cdot (6 \cdot 1 + 8 \cdot i + 2 \cdot (-1) + 4 \cdot (-i)) &&= 2+2i
\end{aligned}$$

So $\widehat{\boldsymbol{f}}$ is given as $(10,\ 2-2i,\ -2,\ 2+2i)$. As expected, $\widehat{f_1} = \overline{\widehat{f}_3}$, i.e. the conjugate complex symmetry holds. This has to be the case, since we used a real-valued input signal.

$$\begin{aligned}
\widehat{g_0} &= 1/2 \cdot (4+6+8+2) &&= 10 \\
\widehat{g_1} &= 1/2 \cdot (4 \cdot 1 + 6 \cdot (-i) + 8 \cdot (-1) + 2 \cdot i) &&= -2-2i \\
\widehat{g_2} &= 1/2 \cdot (4 \cdot 1 + 6 \cdot (-1) + 8 \cdot 1 + 2 \cdot (-1)) &&= 2 \\
\widehat{g_3} &= 1/2 \cdot (4 \cdot 1 + 6 \cdot i + 8 \cdot (-1) + 2 \cdot (-i)) &&= -2+2i
\end{aligned}$$

So $\widehat{\boldsymbol{g}}$ is given as: $(10,\ -2-2i,\ 2,\ -2+2i)$. Again, $\widehat{f_1}$ is the complex conjugate of $\widehat{f_3}$. Evidently, the DFT of the original and the shifted signal are different.

Let now take a look at the spectra of the two signals. They are given as follows:

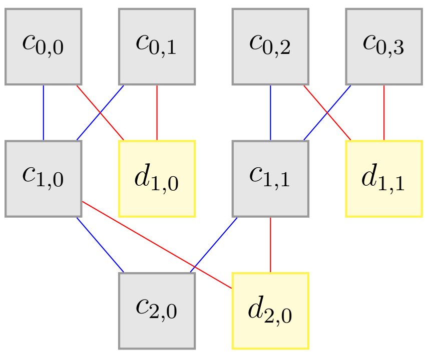
$$\begin{array}{rclcl}
|\widehat{f}_0| & = & \sqrt{\mathrm{Re}^2(\widehat{f}_0) + \mathrm{Im}^2(\widehat{f}_{1,0})} & = & \sqrt{10^2} \quad = \quad 10 \\
|\widehat{f}_1| & = & \sqrt{2^2 + (-2)^2} & = & \sqrt{8} \\
|\widehat{f}_2| & = & \sqrt{(-2)^2} & = & 2 \\
|\widehat{f}_3| & = & \sqrt{2^2 + 2^2} & = & \sqrt{8}
\end{array}$$

$$\begin{array}{rclcl}
|\widehat{g}_0| & = & \sqrt{10^2} & = & 10 \\
|\widehat{g}_1| & = & \sqrt{(-2)^2 + (-2)^2} & = & \sqrt{8} \\
|\widehat{g}_2| & = & \sqrt{2^2} & = & 2 \\
|\widehat{g}_3| & = & \sqrt{(-2)^2 + 2^2} & = & \sqrt{8}
\end{array}$$

(b) Before considering the task of the exercise itself, let us first discuss how a Haar wavelet decomposition and reconstruction is performed and how a visualisation of the algorithm can help to compute the wavelet transform quickly. Therefore, let us consider a signal of length $N = 4$. The wavelet coefficients $d_{2,0}$, $d_{1,0}$, and $d_{1,1}$ can be obtained by starting out with $c_{0,k} = f_k$ $(k = 0, \ldots, 7)$ and applying the two recursion formulas:

$$\begin{aligned}
c_{j,k} &= \frac{1}{\sqrt{2}} \left(c_{j-1,2k} + c_{j-1,2k+1}\right) \\
d_{j,k} &= \frac{1}{\sqrt{2}} \left(c_{j-1,2k} - c_{j-1,2k+1}\right)
\end{aligned}$$

The scaling coefficients $c_{j,k}$ can be computed as the sum of two coarser scaling coefficients (rescaled by $1/\sqrt{2}$) while the wavelet coefficients $d_{j,k}$ result from the corresponding difference. This observation leads to the following simple visualisation of the scheme:

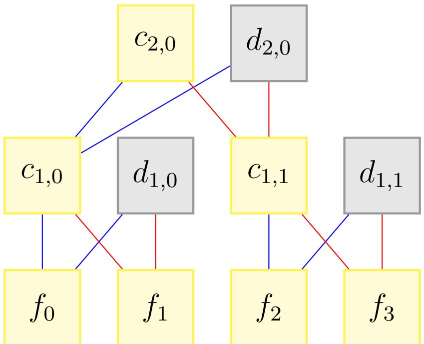


Blue lines indicate rescaled sums, red lines rescaled differences. Only the coefficients highlighted in yellow form the final result of the computation, all other coefficients contain redundant information and are usually not kept.

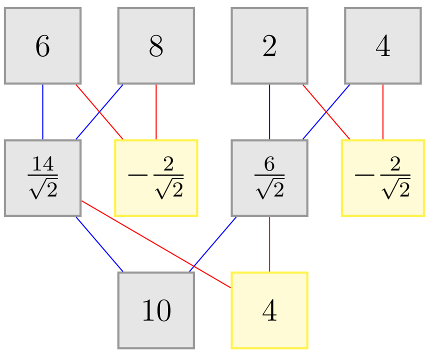
The reconstruction is performed by using the formulas

$$c_{j,2k} = \frac{1}{\sqrt{2}}\left(c_{j+1,k} + d_{j+1,k}\right)$$
$$c_{j,2k+1} = \frac{1}{\sqrt{2}}\left(c_{j+1,k} - d_{j+1,k}\right)$$

In this case, we have the following setup (blue lines indicate scaled sums and red ones indicate scaled differences). Also note that the detail coefficients $d_{i,j}$ as well as $c_{n,0}$ (marked in gray) are all known and need not to be computed. Only the coefficients highlighted in yellow need to be determined.



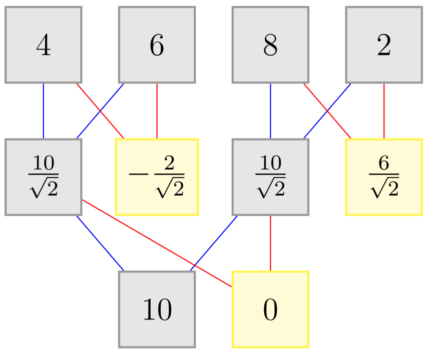
This is however just a general remark, for this exercise, we do not need the back transformation. With this simple scheme, we can easily compute the DWT for the signals $\boldsymbol{f}$ and $\boldsymbol{g}$ Plugging in the numbers from this exercise into the scheme yields for the unshifted signal $\boldsymbol{f}$:



The wavelet decomposition of the unshifted signal therefore yields the coefficients

$$\begin{aligned} c_{2,0} &= 10 \\ d_{2,0} &= 4 \\ (d_{1,0}, d_{1,1}) &= \left(-\frac{2}{\sqrt{2}}, -\frac{2}{\sqrt{2}}\right). \end{aligned}$$

For the shifted signal, we obtain the following computations:



Which yields the wavelet transform

$$\begin{aligned} c_{2,0} &= 10 \\ d_{2,0} &= 0 \\ (d_{1,0}, d_{1,1}) &= \left(-\frac{2}{\sqrt{2}}, \frac{6}{\sqrt{2}}\right). \end{aligned}$$

(c) Although the DFTs are different, the spectra of the two signals are identical. This is due to the shift theorem that tells us that a shift in the spatial domain results in a rotation in the fourier domain. This rotation, however, does not become visible in the Fourier spectrum, since $\left|\exp(-\frac{i\,2\pi\,pm_0}{M})\right| = 1$ for all shifts m_0. So the Fourier transform is shift invariant, with respect to its spectrum.

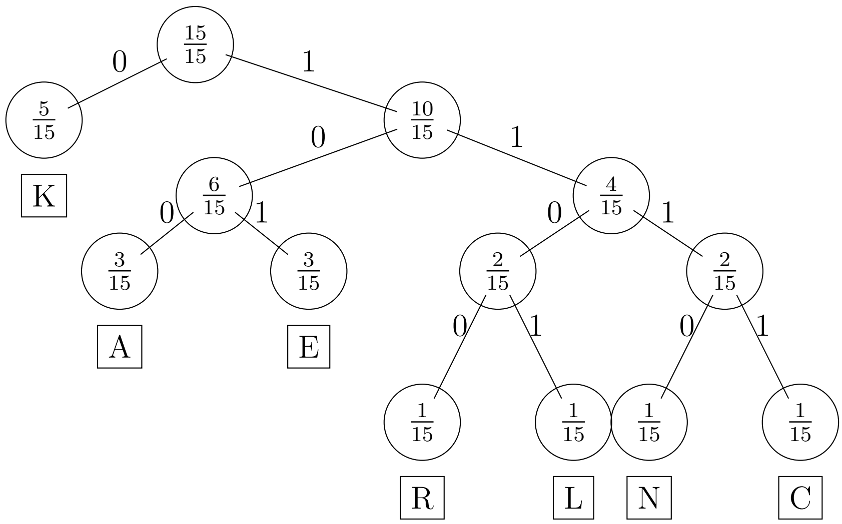
In contrast, the Wavelet coefficients of the original signal and its shifted version are not similar at all. This shows, that the Discrete Haar Wavelet Transform is not shift invariant but highly depends on the local structure of the given signal.

Problem 3: Huffman Coding

(a) The word "'KAKERLAKENKACKE" consists of 15 (7 unique) letters with the following frequency.

Letter	A	C	E	K	L	N	R
Frequency	3	1	3	5	1	1	1

A naive way would use 3 bit per letter (there are more than 4 and less than 8 unique letters in this word), thus the encoding of the whole word would need 45 bits. Let us now consider the Huffman tree. It is given by:

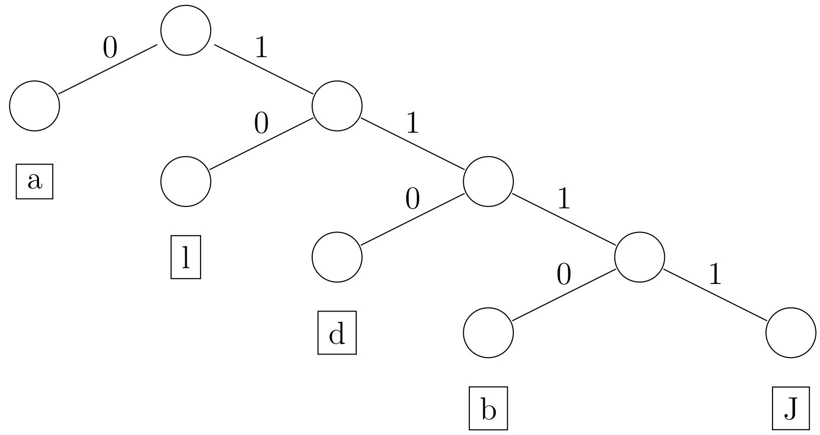


And thus we have the coding:
K = 0, A = 100, E = 101, R = 1100, L = 1101, N = 1110, C = 1111
Using these codes, we obtain the word

010001011100110110001011110010011110101.

This results in an average of 2.6 bits per character and 39 bits in total to encode this word. Note that this code is not unique. Letters with the same frequencies can be swapped without affecting the compression rate.

(b) The Huffman tree for this code is given by



The decoding is quite easy. One starts at the far left of the codeword and iterates through the tree until we end on a letter. This gives us

1111	J
0	a
10	l
0	a
10	l
0	a
1110	b
0	a
110	d

which results in the word “Jalalabad”.

Problem 4: Discrete Cosine Transform

(a) The supplemented code for the discrete cosine transform (DCT) is given by

```
/* ---- DCT in y-direction ---- */
for (i=0; i<nx; i++)
 for (p=0; p<ny; p++)
     {
     tmp[i][p]=0;

     for (m=0; m<ny; m++)
         tmp[i][p] += cy[p] * u[i][m] * cosf((2.0*m+1)*p*ny_1);
     }

/* ---- DCT in x-direction ---- */

for (p=0; p<nx; p++)
 for (j=0; j<ny; j++)
     {
     c[p][j]=0;

     for (m=0; m<nx; m++)
         c[p][j] += cx[p] * tmp[m][j] * cosf((2*m+1)*p*nx_1);
     }
```

The supplemented code for the inverse discrete cosine transform (IDCT) is given by

```
/* ---- DCT in y-direction ---- */
for (i=0; i<nx; i++)
 for (m=0; m<ny; m++)
     {
     tmp[i][m]=0;

     for (p=0; p<ny; p++)
         tmp[i][m] += cy[p] * c[i][p] * cosf((2*m+1) * p * ny_1);
     }

/* ---- DCT in x-direction ---- */
for (m=0; m<nx; m++)
 for (j=0; j<ny; j++)
```

```
{
u[m][j] = 0;

for (p=0; p<nx; p++)
    u[m][j] += cx[p] * tmp[p][j] * cosf ((2*m+1) * p * nx_1);
}
```

(b) Let us now compare the original DCT and the 8 × 8 DCT with respect to the DCT spectrum and the required runtime. To this end, we used the test image `boats.pgm`. This image and the corresponding spectra are depicted on the next page. While the spectrum of the normal DCT shows a concentration of low frequencies in the upper left corner, the 8 × 8 DCT shows essentially the same but for each block separately. Since the position of the blocks in the DCT corresponds to the position of these blocks in the image, you can still recognize some lines describing the original boats. This in turn means, that in contrast to the original DCT, at least the spatial information on the location of the different blocks is preserved. With respect to the runtimes, you notice that the 8 × 8 DCT is about 64 times faster. This is not surprising, since the DCT basis functions are only of size 8 instead of size 512. See Table 1.

(c) Let us now remove about 90% of all frequencies. This yields the spectra seen in Table 2.

As one can see from the spectra, the frequencies that have been removed were exclusively high frequencies. Both, in the spectrum of the normal DCT and the spectrum of the 8 × 8 DCT only the low frequencies in the upper left corner of the spectrum/block spectrum remain. The compressed (backtransformed) images are shown on the next page. While the DCT is sharper but shows ringing artifacts due to the removal of global high frequencies, the 8 × 8 DCT image shows slight block artifacts. See Table 3.

(d) Instead of removing simply the highest frequencies, one could try a more adaptive approach that removes the lowest coefficients of the 8 × 8 DCT spectrum. This can be done in terms of a quantisation step. Here, two different approaches are compared: (i) a simple approach that treats the coefficients of all frequencies equal and (ii) an approach that uses ideas from the previous task and gives more weight to low frequencies, since they are more important to the human eye. Please note that strategy (ii) is actually used by JPEG. The corresponding results that show similar compression rates as the ones in the previous task are presented in Table 4.

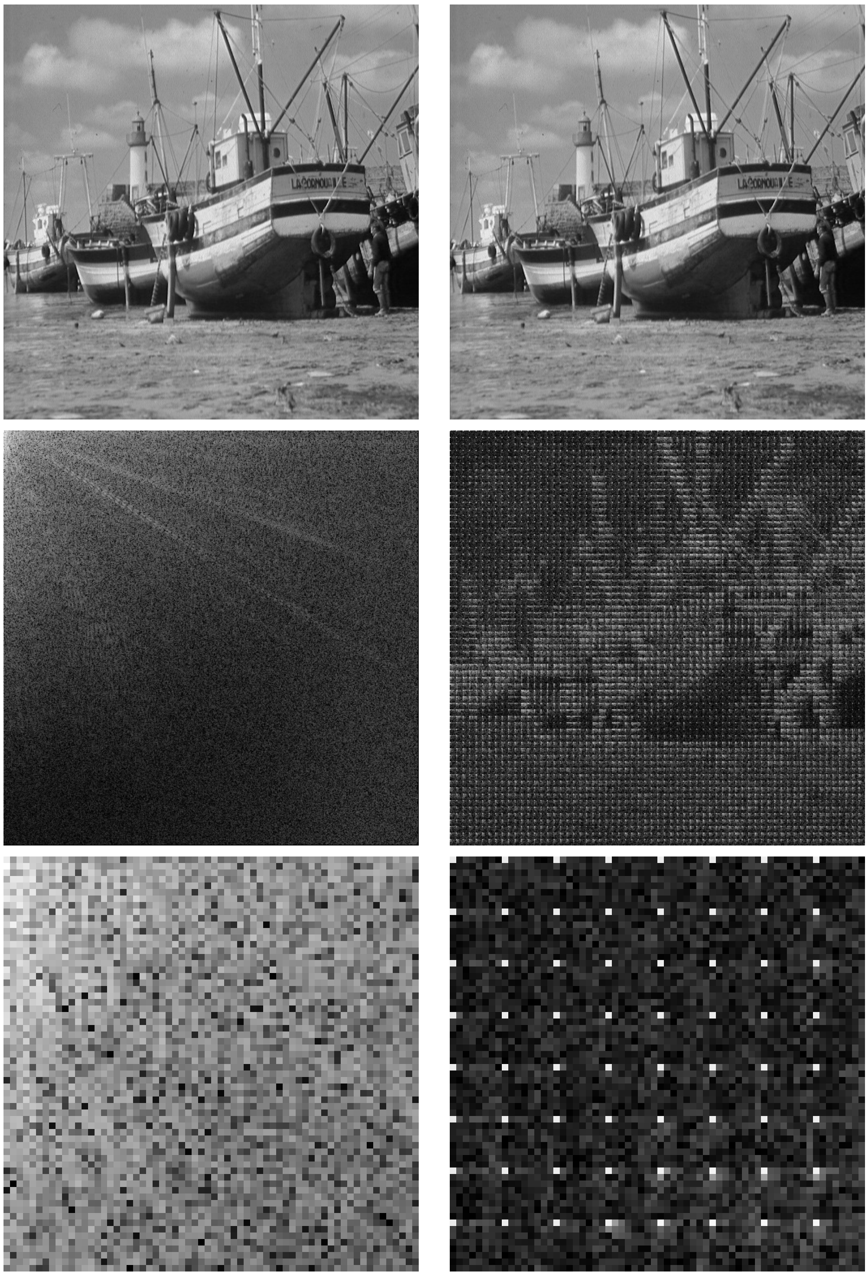


Table 1: Comparison of the DCT and the 8×8 DCT for the image `boats.pgm`. *(a) Top left:* Original image. *(b) Top right:* Backtransformed image (no change). *(c) Centre left:* Spectrum of the DCT. *(d) Centre right:* Ditto for the 8×8 DCT. *(e) Bottom left:* Spectrum of the DCT (Zoom in the upper left corner). *(f) Bottom right:* Ditto for the 8×8 DCT.

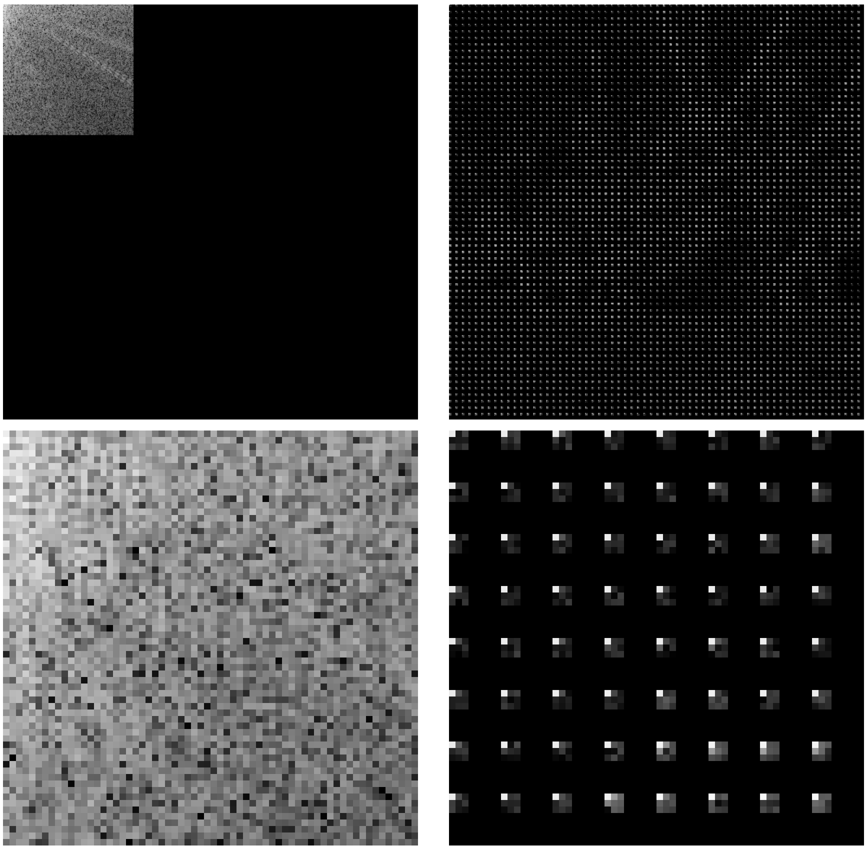


Table 2: Comparison of the DCT and 8 × 8 DCT under the removal of high frequencies. *(a) Top left:* Spectrum of the DCT. *(b) Top right:* Ditto for the 8 × 8 DCT. *(c) Bottom left:* Zoom of DCT spectrum. *(d) Bottom right:* Ditto for the 8 × 8 DCT.

Table 3: Comparison of the DCT and 8×8 DCT under the removal of high frequencies. *(a) Top left:* Compressed by DCT (backtransformed image). *(b) Top right:* Ditto for the 8×8 DCT. *(c) Bottom left:* Zoom of DCT image. *(d) Bottom right:* Ditto for the 8×8 DCT.

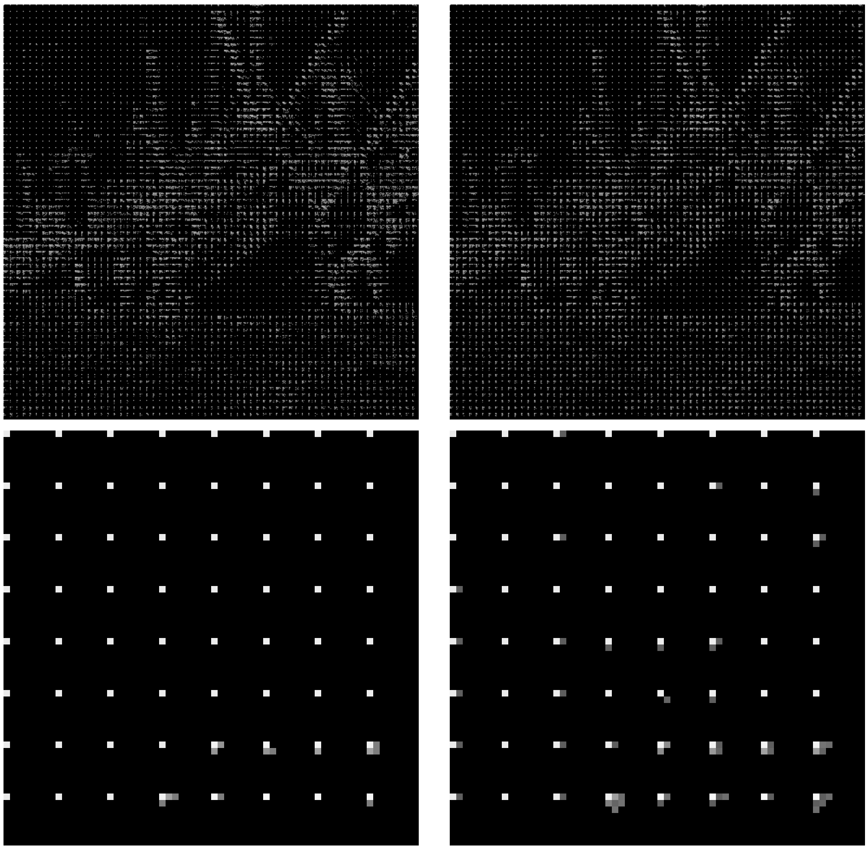


Table 4: Comparison of different quantisation strategies for the 8 $\times$ 8. *(a) Top left:* Equal treatment of all frequencies. *(b) Top right:* Giving more weight to low frequencies. *(c) Bottom left:* Zoom of (a). *(d) Bottom right:* Zoom of (b).

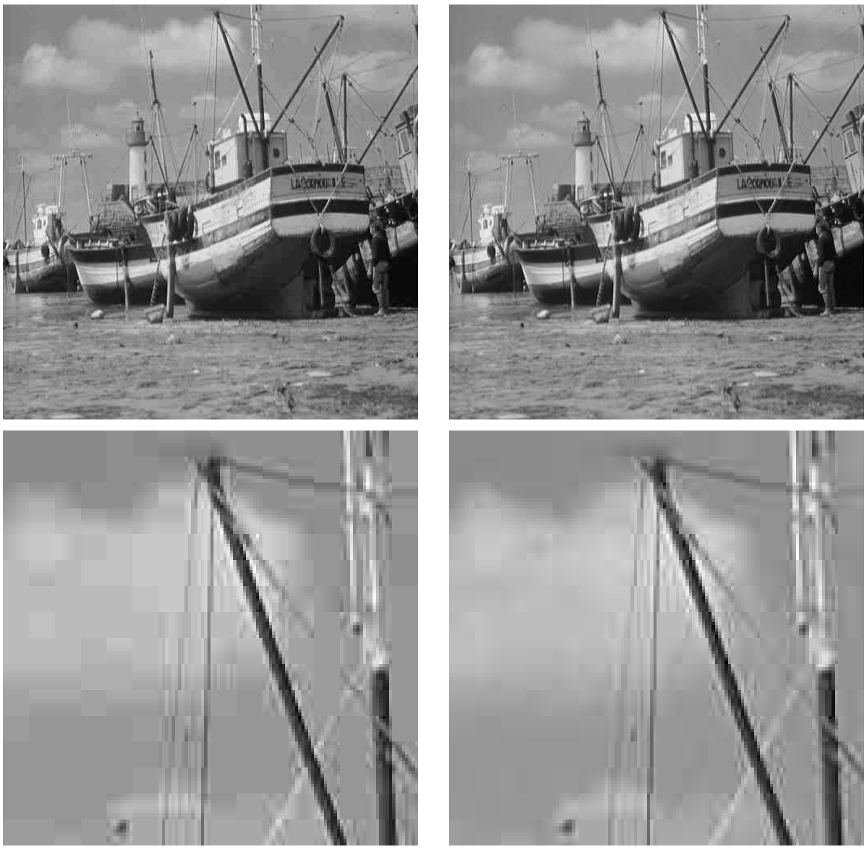


Table 5: Comparison of different quantisation strategies for the 8×8. *(a) Top left:* Compressed with equal treatment of all frequencies. *(b) Top right:* Ditto for the approach that gives more weight to low frequencies. *(c) Bottom left:* Zoom of (a). *(d) Bottom right:* Zoom of (b).

While the spectra look very similar, the corresponding compressed (backtransformed) images on the next page show quite different results. Although both images are relatively sharp, the one that gives more weight to low frequencies shows much less block artifacts (see e.g. the sky). This is due to the fact that storing a few more coefficients for the low frequencies significantly improves the overall result, while a few missing details (removed coefficients for the high frequencies) do not change the sharpness to much (see Table 5).