Image Processing and Computer Vision (IPCV)

Prof. J. Weickert
Mathematical Image Analysis Group

Summer term 2019
Saarland University

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 $\boldsymbol{x} = (x_0, \dots, x_{N-1})^\top$, $\boldsymbol{y} = (y_0, \dots, y_{N-1})^\top \in \mathbb{R}^N$. A matrix-vector multiplication $\boldsymbol{y} = A\boldsymbol{x}$ can be written componentwise for the entries of $\boldsymbol{y}$ as

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

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

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

and the corresponding inverse transform by

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

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

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

(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 $\left\{\frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \frac{7}{2}, \frac{9}{2}, \frac{11}{2}, \frac{13}{2}, \frac{15}{2}\right\}\big)$. 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 Assignment C4, Problem 1, we have proven that the DWT-vectors form an orthonormal 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

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

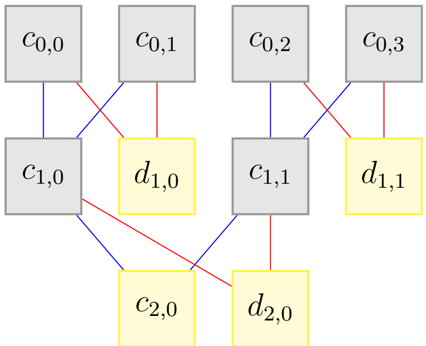
which means BA has to be the identity matrix I which is only true for $B := A^{-1}$. These observations are reasonable, since all matrices A contain row vectors, which form orthonormal bases with respect to the corresponding inner products (see also Assignments H3, Problem 1 and C4, Problem 1). Such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

Problem 2: Discrete Wavelet Transform

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 $M = 4$. The wavelet coefficients $d_{2,0}$, $d_{1,0}$, and $d_{1,1}$ can be obtained by starting out with $c_{0,k} = f_k$ $(k = 0, \dots, 7)$ and applying the two recursion formulas:

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

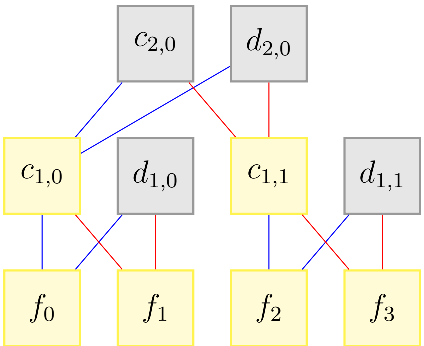
The scaling coefficients $c_{j,k}$ can be computed as the sum of two coarser scaling coefficients (rescaled by $1/\sqrt{2}$) while the wavelet coefficients $d_{j,k}$ result from the corresponding difference. This observation leads to the following simple visualisation of the scheme:



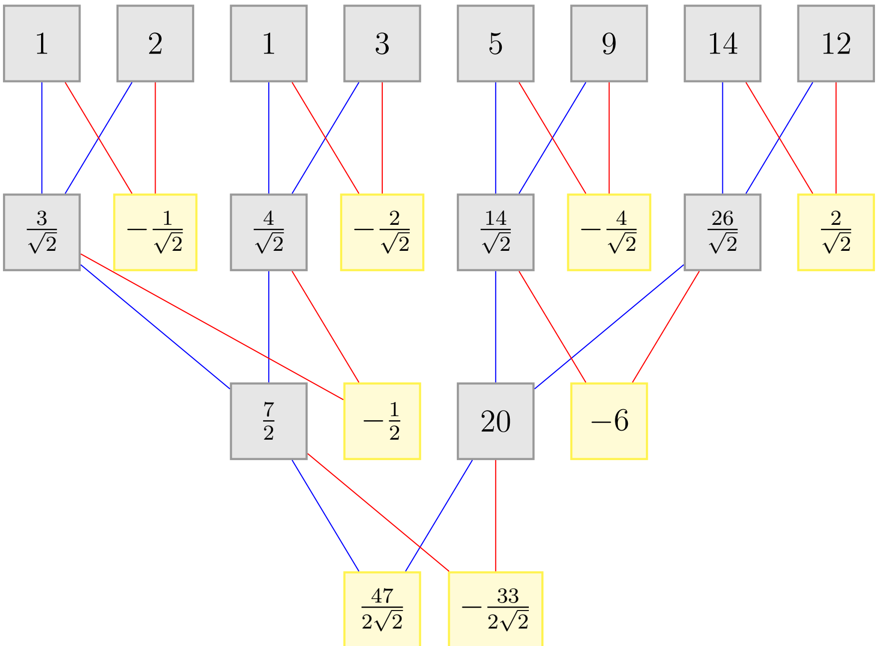
Blue lines indicate rescaled sums, red lines rescaled differences. Only the coefficients highlighted in yellow form the final result of the computation, all other coefficients contain redundant information and are usually not kept.
The reconstruction is performed by using the formulas

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

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.



(a) With this simple scheme, we can easily compute the DWT for signal $\boldsymbol{f}$. Plugging in the numbers from this exercise into an equivalent scheme for $M = 8$ yields



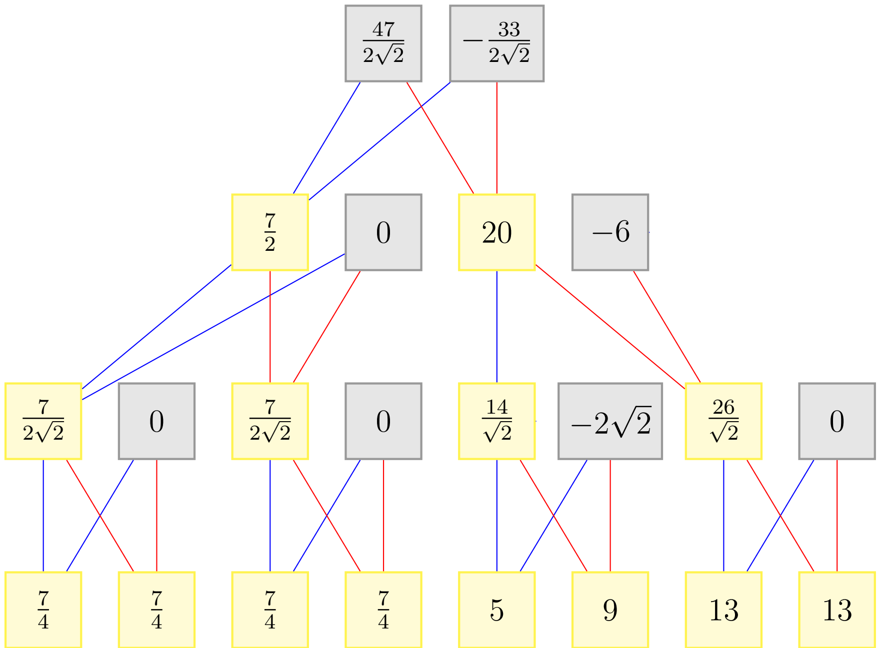
The wavelet decomposition of $\boldsymbol{f}$ is therefore

$$\left(\frac{47}{2\sqrt{2}}, -\frac{33}{2\sqrt{2}}, -\frac{1}{2}, -6, -\frac{1}{\sqrt{2}}, -\sqrt{2}, -2\sqrt{2}, -\sqrt{2}\right)^\top .$$

(b) The four coefficients with smallest values are $d_{2,0}, d_{1,0}, d_{1,1}$ and $d_{1,3}$. If we set those to zero, we obtain

$$\left(\frac{47}{2\sqrt{2}}, -\frac{33}{2\sqrt{2}}, 0, -6, 0, 0, -2\sqrt{2}, 0\right)^\top .$$

(c) Let us now compute the backtransformation of the modified transform. We use the scheme presented above and obtain



Thus, the filtered signal is

$$\left(\frac{7}{4}, \frac{7}{4}, \frac{7}{4}, \frac{7}{4}, 5, 9, 13, 13\right)^\top .$$

(d) Setting some of the wavelet coefficients to zero means that we can store the signal in the wavelet domain in a more compact way. But here, it also had another effect. The signal $\boldsymbol{f}$ can be regarded as a noisy version of the step signal

$$(1.75, 1.75, 1.75, 1.75, 7, 7, 13, 13)^\top .$$

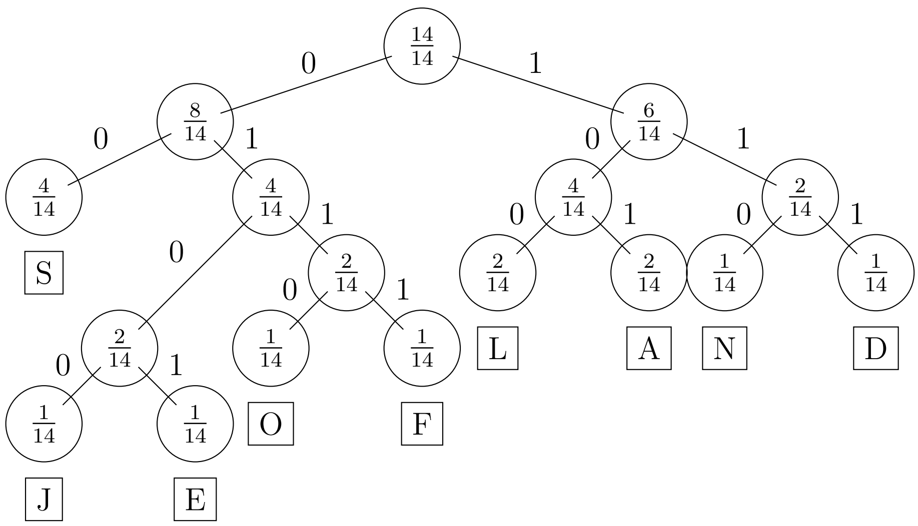
Two of the four wavelet coefficients corresponding to small scales have been set to zero. Since noise consists usually of small scale fluctuations, this results in a visible reduction of the noise at the left part of the signal. We mention that only the values of the first half on the right side were not changed, as this coincides with the wavelet coefficient that was not set to zero. We see therefore the spatial localisation of the Discrete Wavelet Transform. On the other hand, denoising with the Fourier Transform doesn't have any kind of localisation, every frequency influences the signal as a whole. The Fourier Transform also suffers from the implicit assumption of periodicity. Therefore the left boundary influences the right one and vice versa.

Problem 3: Huffman Coding

(a) The word "SELJALANDSFOSS" consists of 14 (9 unique) letters with the following frequency.

Letter	S	E	L	J	A	N	D	F	O
Frequency	4	1	2	1	2	1	1	1	1

A conventional way would use 4 bits per letter (there are more than 8 and less than 16 unique letters in this word), thus the encoding of the whole word would need $14 \cdot 4 = 56$ bits. Let us now consider the Huffman tree. It is given by:



And thus we have the coding:
S = 00, J = 0100, E = 0101, O = 0110, F = 0111, L = 100, A = 101, N = 110, D = 111.
Using these codes, we obtain the word

00 0101 100 0100 101 100 101 110 111 00 0111 0110 00 00.

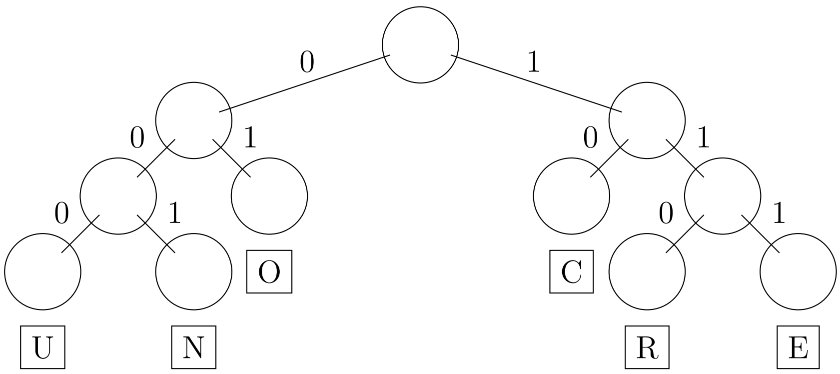
This results in an average of $\frac{42}{14} = 3$ bits per character and 42 bits in total to encode this word. Note that this code is not unique. Letters or nodes with the same frequencies can be swapped without affecting the compression rate. You can also change the label on each branch.

The entropy H of the word SELJALANDSFOSS is given by

$$\begin{aligned}
H =& - p_S \operatorname{ld} p_S - p_E \operatorname{ld} p_E - p_L \operatorname{ld} p_L - p_J \operatorname{ld} p_J - p_A \operatorname{ld} p_A \\
& - p_N \operatorname{ld} p_N - p_D \operatorname{ld} p_D - p_F \operatorname{ld} p_F - p_O \operatorname{ld} p_O \\
=& - \frac{4}{14} \operatorname{ld} \frac{4}{14} - \frac{1}{14} \operatorname{ld} \frac{1}{14} - \frac{2}{14} \operatorname{ld} \frac{2}{14} - \frac{1}{14} \operatorname{ld} \frac{1}{14} - \frac{2}{14} \operatorname{ld} \frac{2}{14} \\
& - \frac{1}{14} \operatorname{ld} \frac{1}{14} - \frac{1}{14} \operatorname{ld} \frac{1}{14} - \frac{1}{14} \operatorname{ld} \frac{1}{14} - \frac{1}{14} \operatorname{ld} \frac{1}{14} \\
\approx& \, 2.95
\end{aligned}$$

We can observe, that our Huffman code is very close to the optimum.

(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

10	C
01	O
01	O
10	C
10	C
000	U
110	R
110	R
111	E
001	N
10	C
111	E

which results in the word “COOCCURRENCE”.

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. Since no DCT coefficients are modified, the original images are obtained by the backtransform in both images. 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.

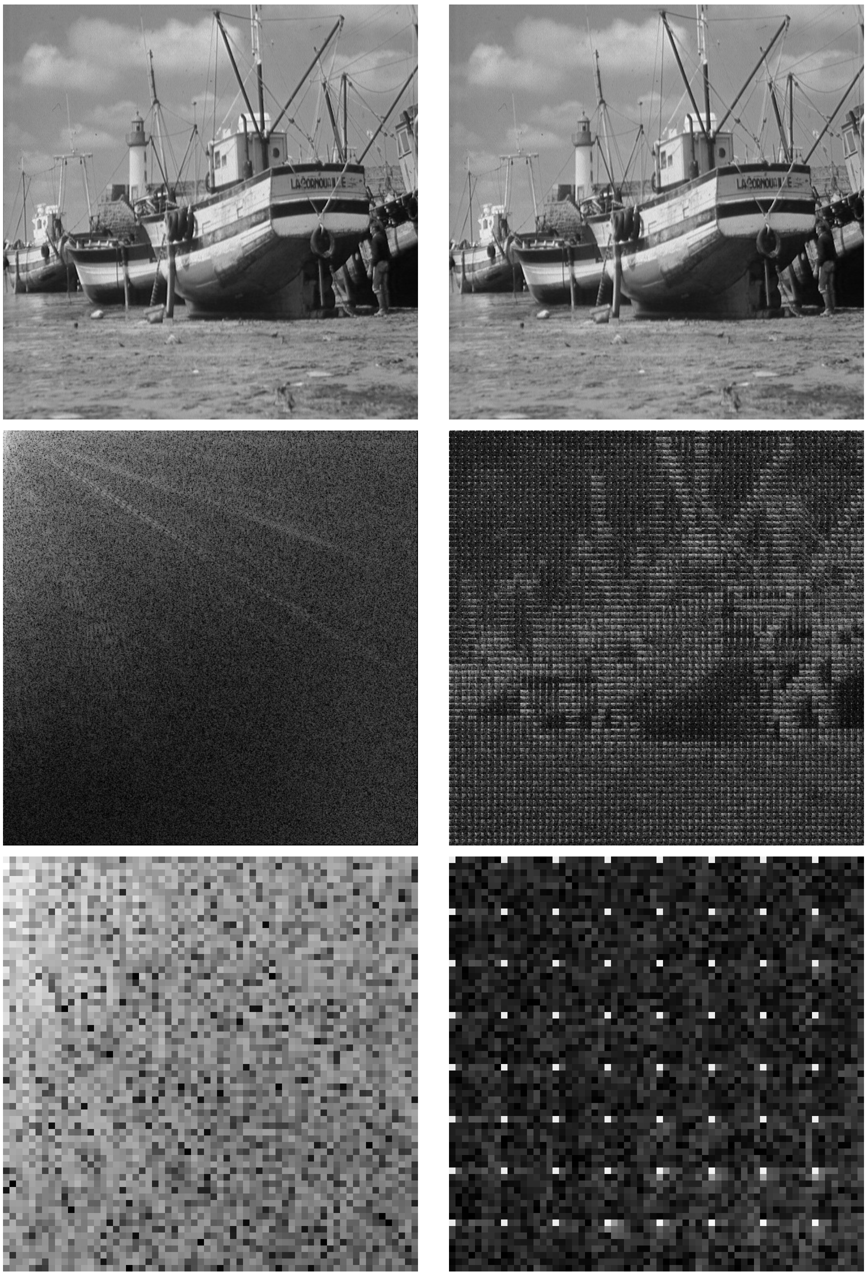


Table 1: Comparison of the DCT and the 8 × 8 DCT for the image `boats.pgm`. *(a) Top left:* Original image. *(b) Top right:* Backtransformed image (no change). *(c) Centre left:* Spectrum of the DCT. *(d) Centre right:* Ditto for the 8 × 8 DCT. *(e) Bottom left:* Spectrum of the DCT (Zoom in the upper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

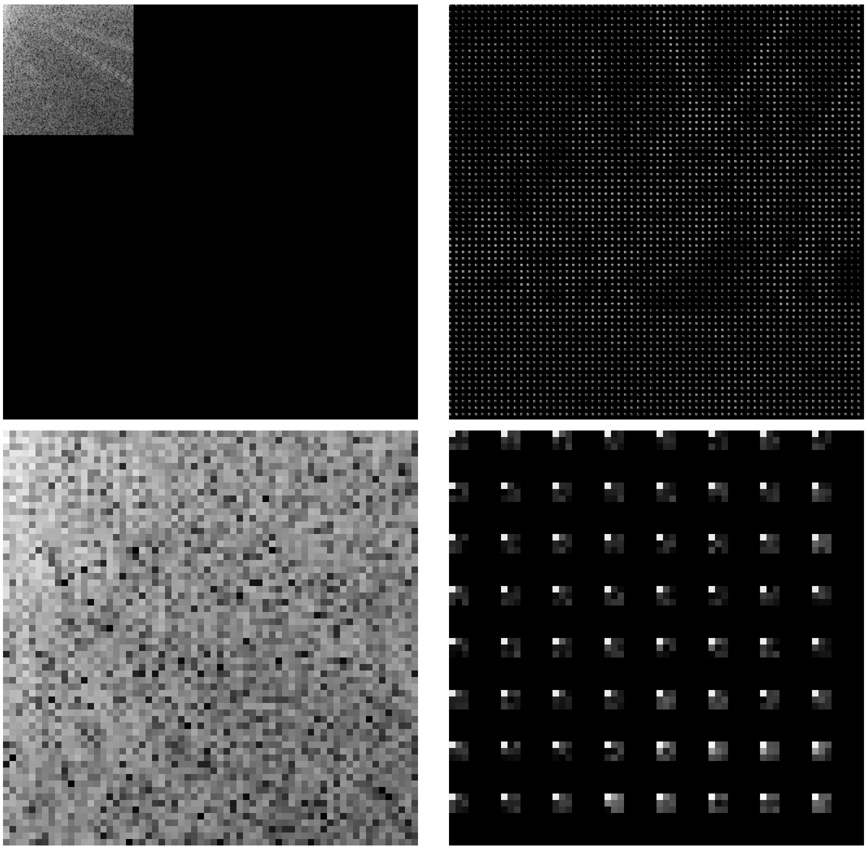


Table 2: Comparison of the DCT and 8 × 8 DCT under the removal of high frequencies. *(a) Top left:* Spectrum of the DCT. *(b) Top right:* Ditto for the 8 × 8 DCT. *(c) Bottom left:* Zoom of DCT spectrum. *(d) Bottom right:* Ditto for the 8 × 8 DCT.

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.

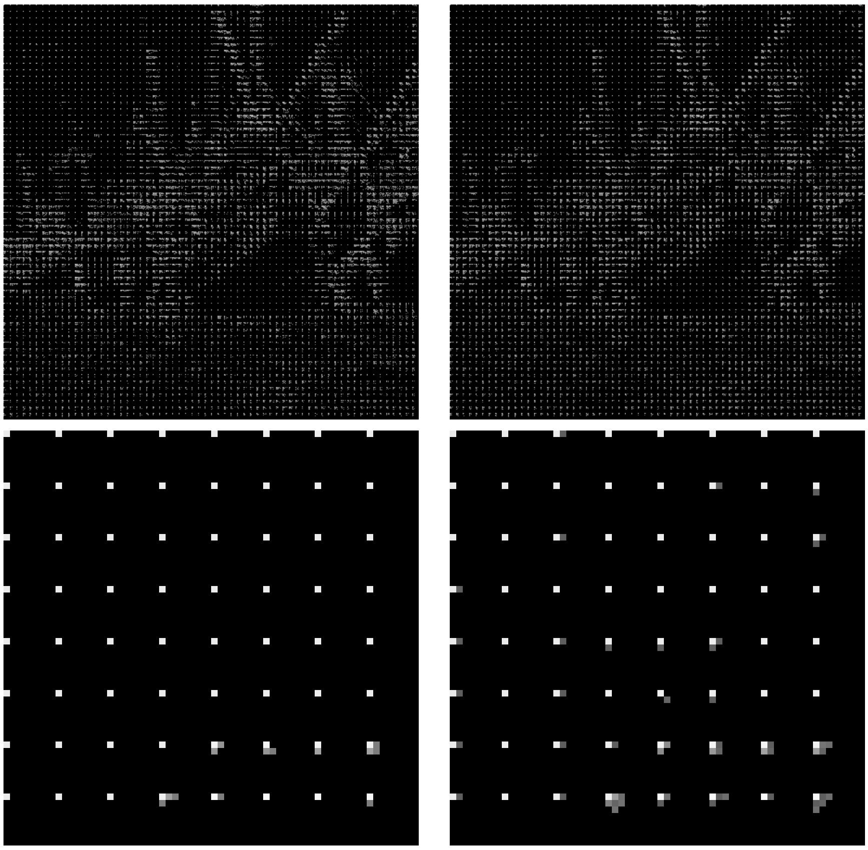


Table 4: Comparison of different quantisation strategies for the 8×8. *(a) Top left:* Equal treatment of all frequencies. *(b) Top right:* Giving more weight to low frequencies. *(c) Bottom left:* Zoom of (a). *(d) Bottom right:* Zoom of (b).

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.

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 too much (see Table 5).

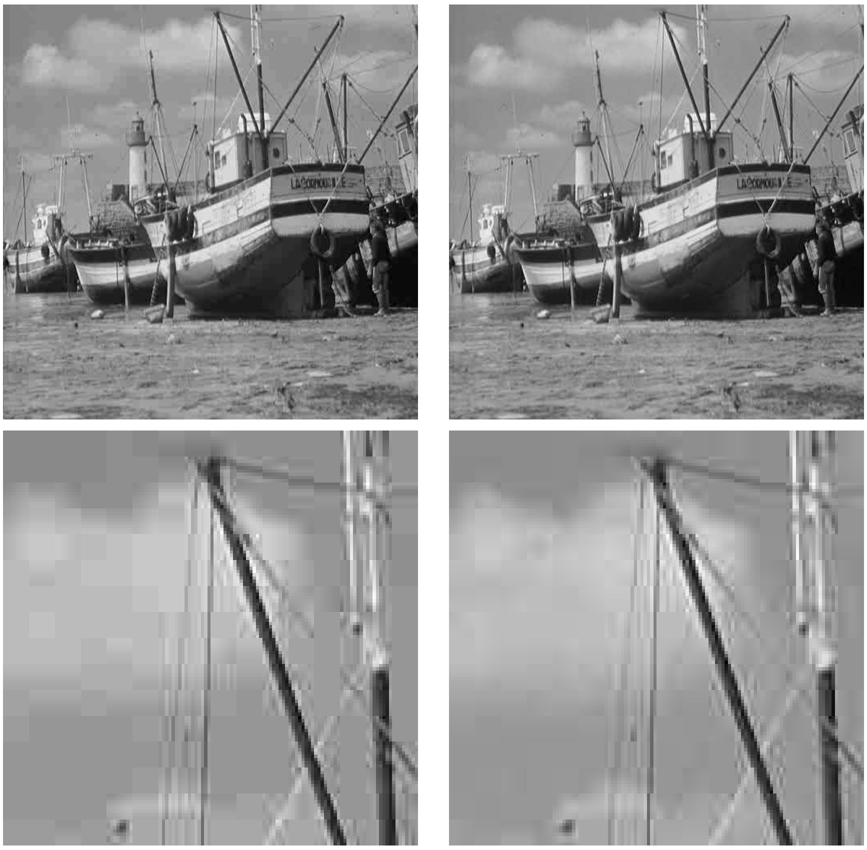


Table 5: Comparison of different quantisation strategies for the 8 × 8. *(a) Top left:* Compressed with equal treatment of all frequencies. *(b) Top right:* Ditto for the approach that gives more weight to low frequencies. *(c) Bottom left:* Zoom of (a). *(d) Bottom right:* Zoom of (b).