

Computer Vision I - Sheet 2

Dominik Autaler, Jonas Otto (Group 2)

May 30, 2019

1 Filter algebra

1.1

Given the filter $H = \begin{pmatrix} -1 & -2 & 0 \\ -2 & 0 & 2 \\ 0 & 2 & 1 \end{pmatrix}$ and an Image $I \in [0, 255]$ the exercise arises the question what minimum and maximum output values this filter could create. Because all values in the image are positive, the maximum of 1275 is reached for a pixel region like $\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 255 \\ 0 & 255 & 255 \end{pmatrix}$. Analogous the minimum value of -1275 is for instance reached in a pixel region like $\begin{pmatrix} 255 & 255 & 0 \\ 255 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$.

1.2

The source code used for this exercise is located in sh02ex01.m. The resulting images for task 2 are shown in the figures 2 and 3. For comparison purposes the original image is displayed in figure 1. The filter H emphasises large contrasts along the diagonal from bottom left to the top right corner. Because the filter only emphasises high contrasts, it belongs to the class of high-pass filters.



Figure 1: Original image

1.3

The figures 4 to 6 show the results to task 3. Although the image shown in figure 6 looks quite dark, the sum over all pixels in the image is approximately 31. Compared to the size of the image (256px x 256px) this number is so small, the one could argue that the error is coming



Figure 2: Gaussian kernel applied to lena.tif

from the small inaccuracies related to calculating with floating point numbers. Therefore the example shows the commutative property the convolution owns quite well.

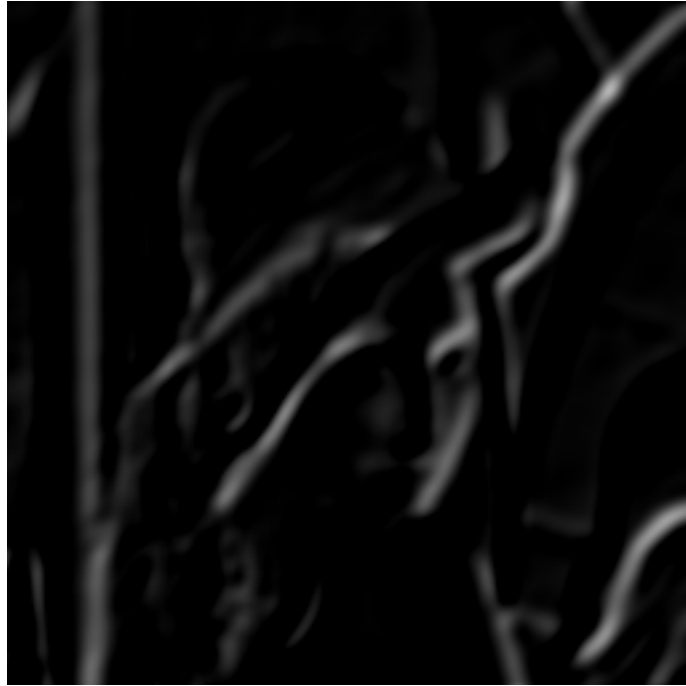


Figure 3: Resulting image for $R1 = I * G_\sigma * H$



Figure 4: Filter H applied to lena.tiff



Figure 5: Resulting image for $R2 = I * H * G_\sigma$

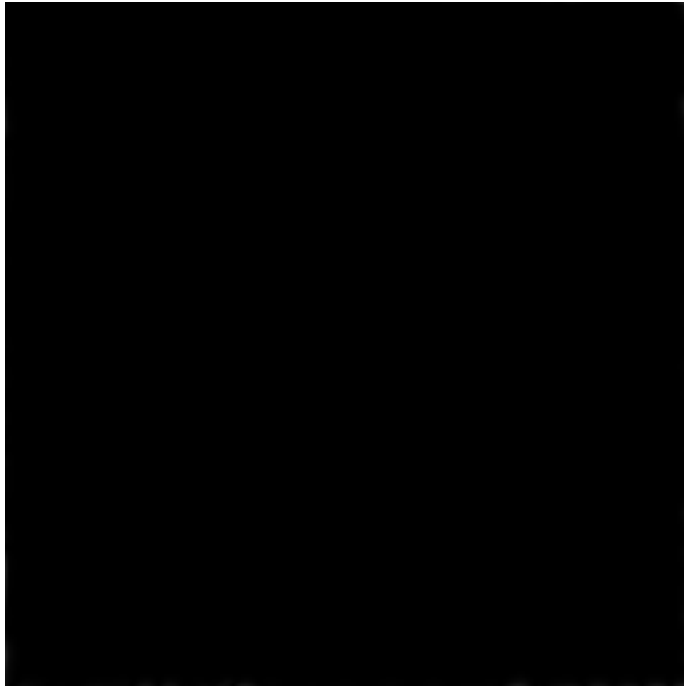


Figure 6: Resulting image for $|R1 - R2|$

2 Discrete Fourier Transform

2.1

Our implementation of the Discrete Fourier Transform can be found in the file `fourier.m`. Figure 7 shows the results of applying our DFT to the given signal.

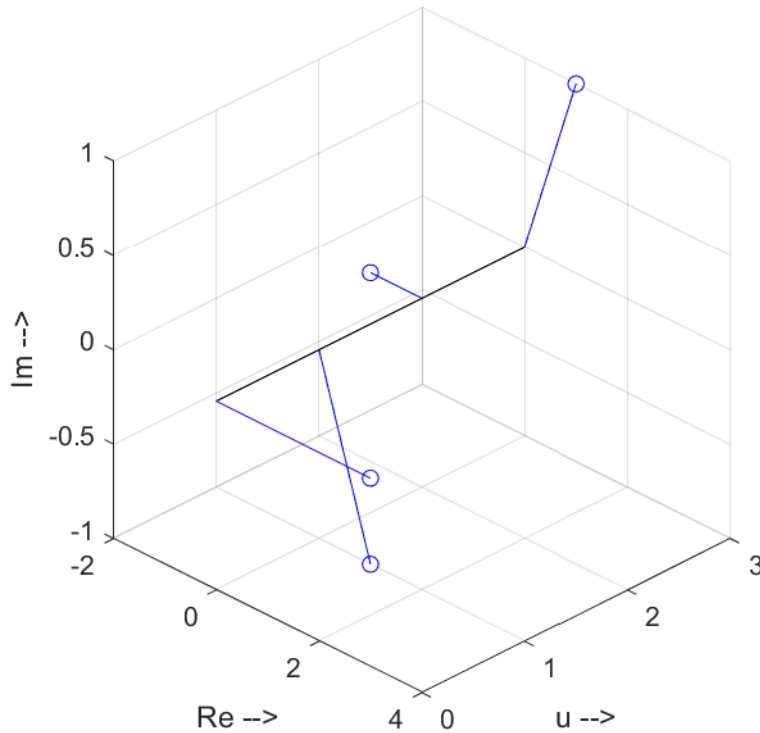


Figure 7: Complex visualization of the results of the DFT

2.2

Figure 8 shows the sine wave which corresponds to the vector at $u = 1$ in Figure 7. Because this vector has both a real and an imaginary component not equal to zero it would also create a cosine wave. Both waves together would then represent this part in the original signal.

2.3

The source code used for this task can be found in the file `ifourier.m`.

2.4

Figure 9 shows the results gained from task a) to c). Because the third value of the reconstructed signal is not exact zero but of the magnitude 10^{-16} Matlab expands the vertical axis in the third plot a bit into the negative direction. Besides this small difference, the reconstructed signal looks the same compared to the original signal. Because our signal was sampled with a sampling frequency of 4Hz and the sampled signal has 4 points, the frequency resolution of the frequency spectrum is $\frac{4\text{Hz}}{4} = 1\text{Hz}$.

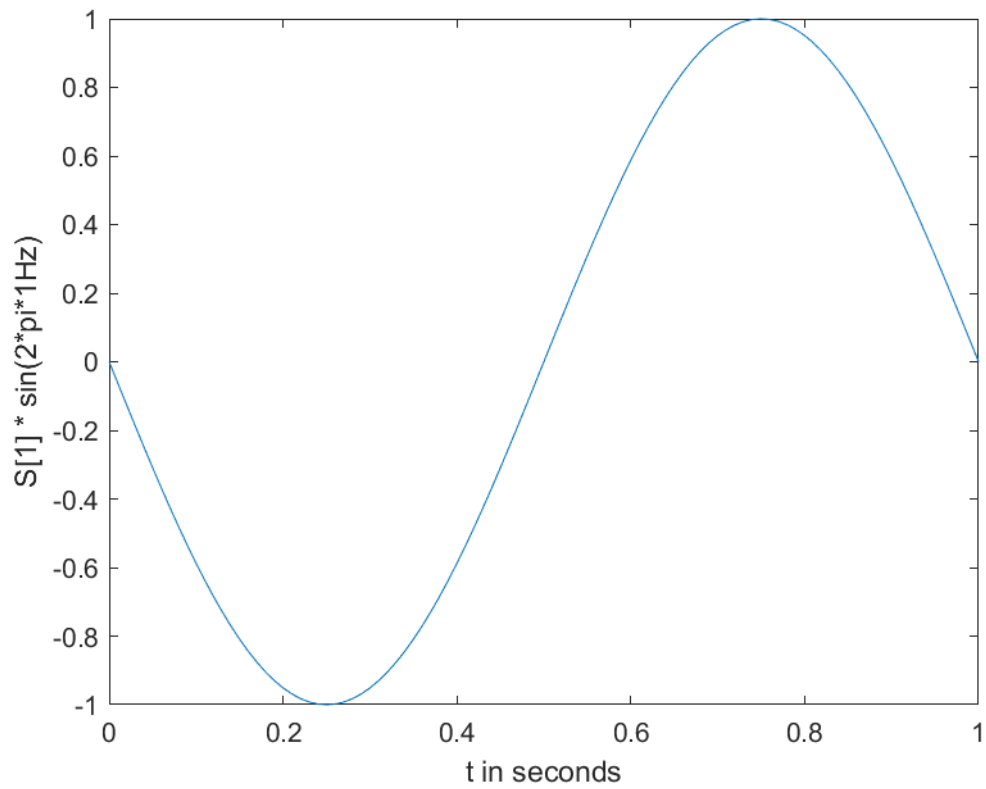


Figure 8: Sinus component of the complex value representing 1Hz

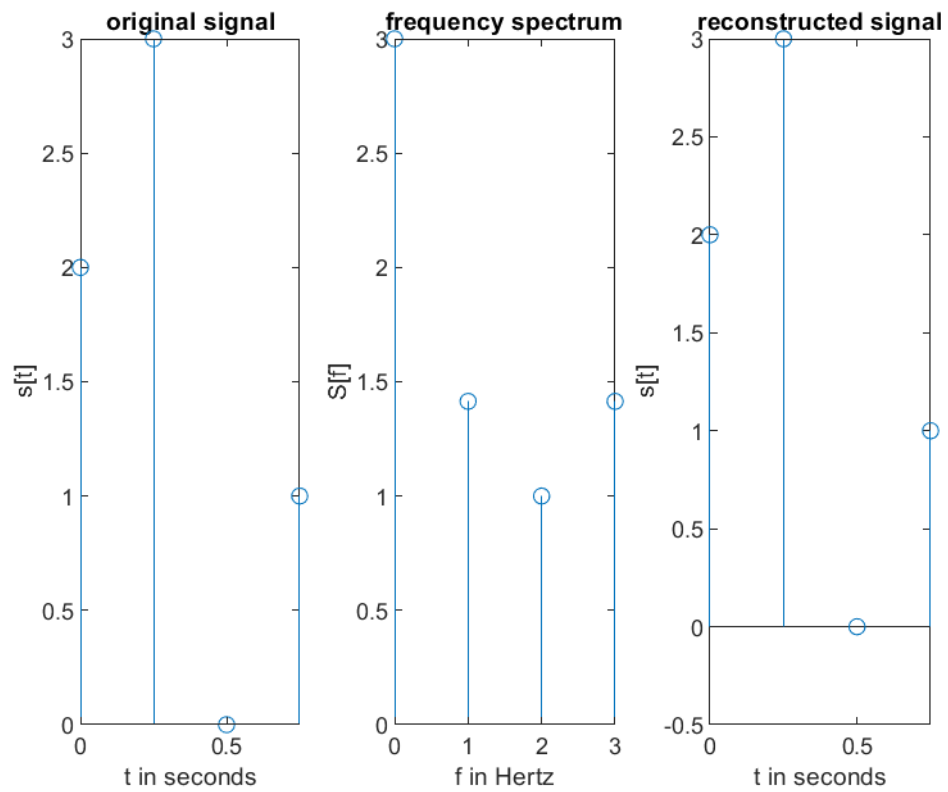


Figure 9: Plots showing results from task a) to c)

3 Fourier Transform for Image Quality Assessment

The source used to create the figures shown in this section can be found in the Matlab script sh03ex03.m. Figure 10 shows the original images together with their frequency spectra calculated with `fft2`. The spectra have been shifted by using `fftshift`. Moreover they are showing the absolute values with an logarithmic scale. Figure 11 shows the highpass filter which was used in the next tasks.

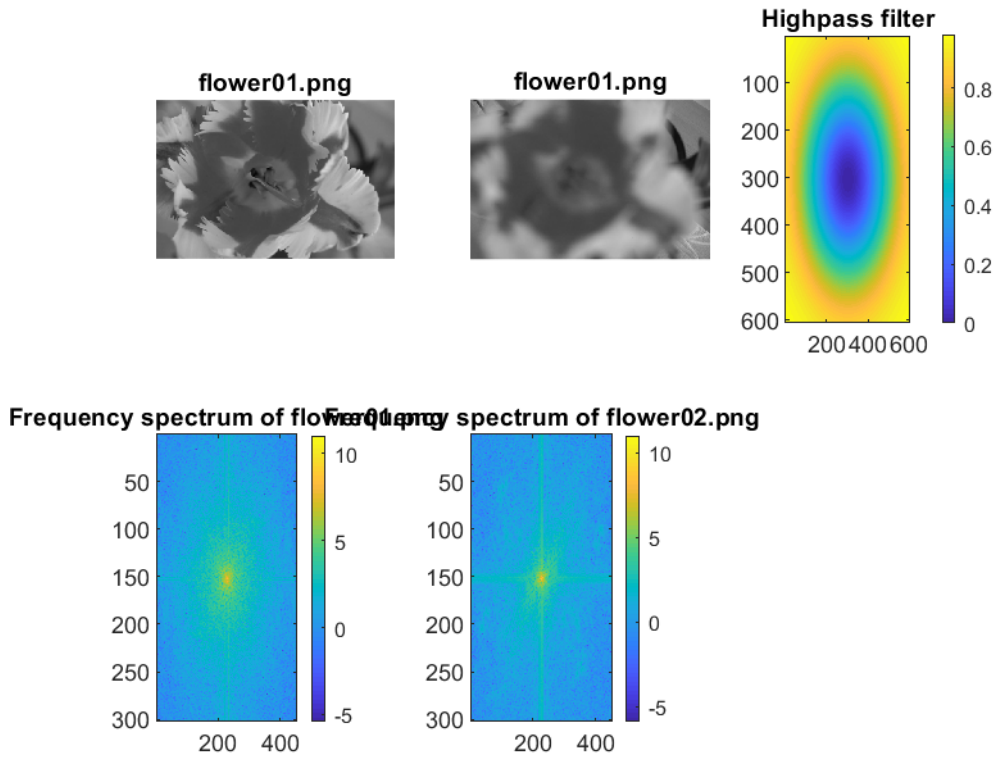


Figure 10: Original images and their frequency spectra

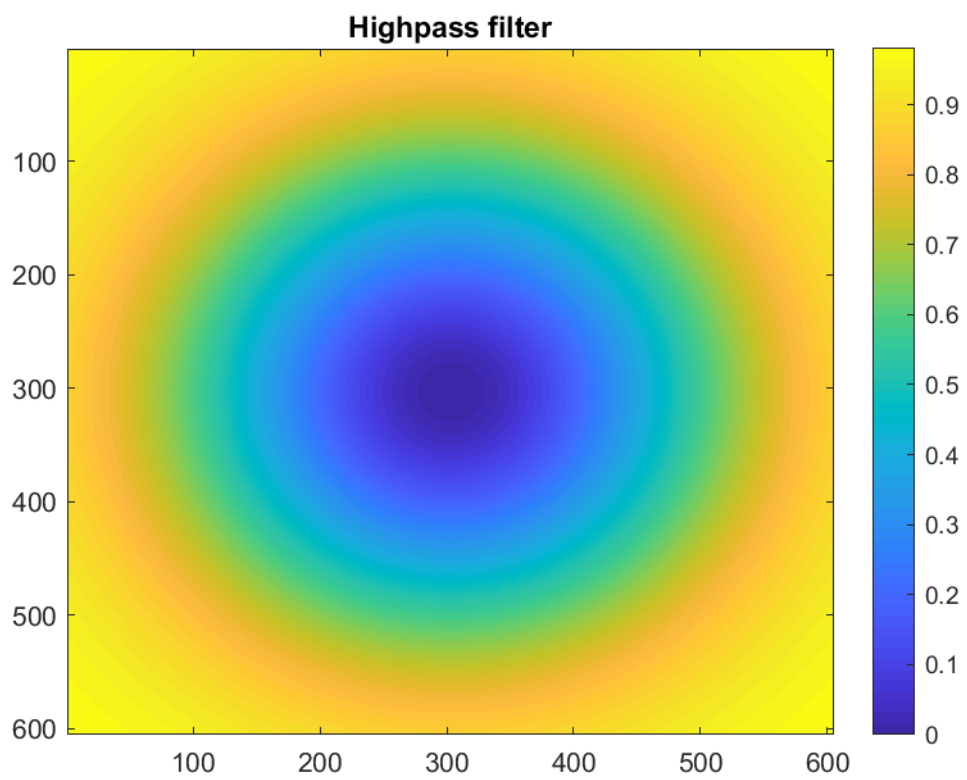


Figure 11: Highpass filter