Prof. Dr. Thomas Schultz
Rasha Sheikh (rasha@cs.uni-bonn.de)

University of Bonn
B-IT / Institute of CS II
November 05, 2020

Winter term 2020/21

Image Acquisition and Analysis in Neuroscience

Assignment Sheet 1

Solution has to be uploaded by November 16, 2020, 8:00 a.m., via eCampus

If you have questions concerning the exercises, please use the forum on eCampus.

- Please work on this exercise in **small groups** of 3 students. Submit each solution only once, but clearly indicate who contributed to it by forming a team in eCampus. Remember that all team members have to be able to explain all answers.
- Please submit your answers in PDF format, and your scripts as *.py/*.ipynb files. If you are using Jupyter notebook, please also export your scripts and results as PDF.

Exercise 1 (Complex numbers and their geometric intuition, *4 Points*)

Perform the following two operations in the polar representation $z = |z|e^{i\phi}$:

a) Multiplication of two complex numbers: $(1 - i) \cdot (0 + 2i)$

b) Division of two complex numbers: $(0 + i)/(1 - \sqrt{3}i)$

Illustrate the geometric intuition behind these operations by plotting the complex operands and the result of each operation in the complex plane.

Exercise 2 (Discrete Fourier transform, *5 Points*)

Similar to the continuous Fourier transform treated in the lecture, a Fourier transform for discrete signals is commonly defined as

$$X_k = \sum_{n=0}^{N-1} e^{-2\pi i k (n/N)} x_n \tag{1}$$

and it can be written as a linear equation $X = Ax$.

a) Please write down an equation for element $a_{\alpha\beta}$ of A. (1P)

b) The inverse of the discrete Fourier transform as defined above is given by

$$x_n = \frac{1}{N} \sum_{n=0}^{N-1} e^{2\pi i k (n/N)} X_k. \tag{2}$$

Briefly argue why Eq. (2) is the same as stating that $A^{-1} = \frac{1}{N} A^*$, where A^* is the conjugate transpose (adjoint) of A. (2P)

c) What normalization factor would be required in Eq. (1) to make it unitary, i.e., to achieve a matrix A that satisfies $A^{-1} = A^*$? (2P)

Exercise 3 (Fourier transform of two-sided decaying exponentials, *6 Points*)

Please determine the Fourier transform $F(k)$ of the two-sided decaying exponentials function:

$$f(t) = \begin{cases} e^{-t} & \text{for } t \geq 0 \\ e^{t} & \text{for } t < 0 \end{cases} \tag{3}$$

You may use all theorems and results provided in the lecture.

Exercise 4 (Install and try Python, *10 Points*)

Throughout the semester, we will use the programming language Python to perform image processing operations. You will need the following open source software, which is available for Linux, Windows, and Macs:

- The Python interpreter itself, which is the basis for running any Python program.
- The Python packages NumPy, SciPy, matplotlib, and scikit-image. Please note that these packages may not be supported by all version of Python. We test our own model solutions using Python 3.
- It's very helpful to use Jupyter notebook, or an integrated development environment that supports Python.

Once you have the required software, please perform the following two simple tasks:

a) Read the grayscale image `brain.png`, which is provided along with this sheet on eCampus, and turn it into a binarized version `brain-bin.png`, so that all pixels with values below 100 are completely black, all others completely white. (5P)

b) Create an image that is 255 pixels wide and 100 pixels high. Fill its pixels with a horizontal intensity gradient so that the final image is black on the left and becomes gradually brighter to the right. (5P)

Please submit the resulting image files and your Python code.

Hint: If you are used to C++ or Java, you should be aware of the fact that Python does not use brackets to group statements. Instead, code with exactly the same indentation (be careful not to confuse spaces and tabs!) forms a block, e.g., the body of a loop. For a more detailed introduction to Python, please refer to `https://docs.python.org/3/tutorial/`. A useful reference for SciPy is at `https://docs.scipy.org/doc/scipy/reference/`. In particular, routines for reading and writing files can be found in module `scipy.misc`.

Good Luck!