DATA.ML.300 Computer Vision
Exercise Round 1

For these exercises you will need Matlab or Python. Return your answers and output images as a pdf along with your modified code (Python or Matlab) to Moodle. Exercise points will be granted after a teaching assistant has checked your answers. Returns done before the solution session will result in maximum of 4 points, whereas returns after the session will result in maximum of 1 point.

If you're using python, make sure you have scikit-image python library installed.

pip install –user scikit-image

Task 1. Homogeneous coordinates. (Pen & paper exercise) (1 point)
The equation of a line is

$$\mathbf{x}^\top \mathbf{l} = 0$$

This means that if a point $\mathbf{x}$ lies on the line $\mathbf{l}$ the equation is satisfied.

You are free to use Matlab for calculations.

a) Convert the four points below (cartesian x,y-coordinates) into their corresponding homogeneous coordinate form.

$$\begin{aligned} x_1 &= (2, -1) \\ x_2 &= (1, -2) \\ x_3 &= (1, 1) \\ x_4 &= (-1, 0) \end{aligned}$$

b) The line $\mathbf{l}$ through two points $\mathbf{x}$ and $\mathbf{x}'$ is $\mathbf{l} = \mathbf{x} \times \mathbf{x}'$. Use this to form two lines, line $\mathbf{l}_1$ through homogeneous points $\mathbf{x}_1$ and $\mathbf{x}_2$, and $\mathbf{l}_2$ through $\mathbf{x}_3$ and $\mathbf{x}_4$.

c) The intersection of two lines $\mathbf{l}$ and $\mathbf{l}'$ is the point $\mathbf{x} = \mathbf{l} \times \mathbf{l}'$. Use lines $\mathbf{l}_1$ and $\mathbf{l}_2$ to calculate their point of intersection and convert this back into cartesian coordinates.

Task 2. Image denoising (Programming exercise) (1 point)

In this exercise you will need to denoise an example image using three different filters. Open the image denoising file (Matlab or Python) and follow the instructions in the comments.

a) Gaussian filtering

From Szeliski's Book section 3.2.1: *"A more direct method is to treat the 2D kernel as a 2D matrix K and to take its singular value decomposition (SVD)"*

$$\mathbf{K} = \sum_i \sigma_i \mathbf{u}_i \mathbf{v}_i$$

"... $\sqrt{\sigma_0}\mathbf{u}_0$ and $\sqrt{\sigma_0}\mathbf{v}_0^\top$ provide the vertical and horizontal kernels"

Before proceeding to implement the separated Gaussian filter, separate the simpler version below to make sure you understand the process.

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Start by calculating the singular value decomposition (SVD), e.g. in Matlab, and calculate the 1D horizontal and vertical filters as described above. The resulting 1D filters **h** and **v** should be able to reconstruct the original 2D filter by simply matrix multiplying them together.

b) Median filtering

Section 3.3.1 of Szeliski's book.

c) Bilateral filtering

Section 3.3.1 of Szeliski's book

Task 3. Hybrid images (Programming exercise) (2 points)

In this task you will need to construct a hybrid image that combines facial images of a wolf and a man. In addition, visualize the log magnitudes of the Fourier transforms of the original images and their low-pass and high-pass filtered versions (i.e.constituents of the hybrid image). Open the hybrid image file (Matlab or Python) and follow the instructions in the comments.