DATA.ML.300 Computer Vision
Exercise Round 5

For these exercises you will need Python or Matlab and a webcam. The second exercise can only be done with Python. Return your answers as a pdf along with your modified code 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 are using Python, make sure you have *OpenCV* library for Python installed.

pip install –user –upgrade opencv-python

If you are using Matlab, make sure you have *Support Package for USB Webcams* installed. This can be done through the add-on explorer.

Task 1. Similarity transformation from two point correspondences. (pen & paper) (1 point)
A similarity transformation consists of rotation, scaling and translation and is defined in two dimensions as follows:

$$\mathbf{x}' = s\mathbf{R}\mathbf{x} + \mathbf{t} \quad \Leftrightarrow \quad \begin{pmatrix} x' \\ y' \end{pmatrix} = s \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \tag{1}$$

Describe a method for solving the parameters s, θ, t_x, t_y of a similarity transformation from two point correspondences $\{\mathbf{x}_1 \rightarrow \mathbf{x}'_1\}$, $\{\mathbf{x}_2 \rightarrow \mathbf{x}'_2\}$ using the following stages:

a) Compute the correspondence between vectors $\mathbf{v}' = \mathbf{x}'_2 - \mathbf{x}'_1$ and $\mathbf{v} = \mathbf{x}_2 - \mathbf{x}_1$ using the similarity transform above. Use corresponding unit vectors to solve the scale factor s from this correspondence. *Hint: There should be no scaling in a transformation between two unit vectors*

b) Solve also the rotation angle θ from this correspondence.

c) After solving s and θ compute $\mathbf{t}$ using equation (1) and either one of the two point correspondences.

d) Use the procedure to compute the transformation from the following point correspondences: $\{(\frac{1}{2}, 0) \rightarrow (0, 0)\}$, $\{(0, \frac{1}{2}) \rightarrow (-1, -1)\}$.
(Hint: Drawing the point correspondences on a grid paper may help you to check your answer.)

Task 2. Homography using SIFT (Programming exercise) (1 point)
This exercise can only be done in Python. Look up the code in **homography.py** and complete the missing parts. Include the code and its outputs in your submission. Feel free to try your own images albeit not required.

Task 3. Real-time face point tracking (Programming exercise) (2 points)
We'll be using KLT-tracker to track points detected from a face. Open **face_tracking** and follow the instructions written in the comments. Answer the following questions in your pdf. **You do not have to include an output image.**

a) How does this program work, i.e. what are its main parts?

b) Do you notice any problems with the tracking? How do you think these could be avoided?