CS-E4850 Computer Vision
Exercise Round 5

The following instructions are for the Matlab version. The instructions for the Python version are in github. The pen and paper problem is the same in both. Matlab is available on Aalto computers and also for students' own computers via `https://download.aalto.fi`.

The problems should be solved before the exercise session and solutions returned via the MyCourses page. Upload TWO files: (1) a PDF file illustrating your results with images and a few lines of text, and (2) a zip file which contains your Matlab source code files.

Get the example `m`-files by downloading `Exercise05.zip` from the MyCourses page.

Exercise 1. Total least squares line fitting. (Pen and paper problem)
An overview of least squares line fitting is presented on the slide 13 of Lecture 4. Study it in detail and present the derivation with the following stages:

1) Given a line $ax+by-d=0$, where the coefficients are normalized so that $a^2+b^2=1$, show that the distance between a point (x_i, y_i) and the line is $|ax_i + by_i - d|$.

2) Thus, given n points (x_i, y_i), $i = 1, \ldots, n$, the sum of squared distances between the points and the line is $E = \Sigma_{i=1}^{n}(ax_i + by_i - d)^2$. In order to find the minimum of E, compute the partial derivative $\partial E/\partial d$, set it to zero, and solve d in terms of a and b.

3) Substitute the expression obtained for d to the formula of E, and show that then $E = (a\ b)U^\top U(a\ b)^\top$, where matrix U depends on the point coordinates (x_i, y_i).

4) Thus, the task is to minimize $||U(a\ b)^\top||$ under the constraint $a^2 + b^2 = 1$. The solution for $(a\ b)^\top$ is the eigenvector of $U^\top U$ corresponding to the smallest eigenvalue, and d can be solved thereafter using the expression obtained above in the stage two.

Tasks continue on the next page...

Exercise 2. Robust line fitting using RANSAC. (Matlab exercise)
Run the example script `robustLineFitting.m`, which plots a set of points $(x_i, y_i), i = 1, \ldots, n$, and estimate a line that best fits to these points by implementing a RANSAC approach as explained in the slides of Lecture 4:

1) Repeat the following steps N times (set N large enough according to the guidelines given in the lecture):

   - Draw 2 points uniformly at random from set $\{(x_i, y_i)\}_i$.
   - Fit a line to these 2 points.
   - Determine the inliers to this line among the remaining points (i.e. points whose distance to the line is less than a suitably set threshold t).

2) Take the line with most inliers from previous stage and refit it using total least squares fitting to all inliers.

3) Plot the estimated line and all the points (x_i, y_i) to the same figure and report the estimated values of the line's coefficients.

Demo 1. Line detection by Hough transform. (Just a demo, no points given)
Run the example script `lineDetectionHough.m`, which illustrates line detection by Hough transform using Matlab built-in functions.