Project 2
Computing the fundamental matrix between 2 images
(due date: 24/12/2024)

The goal of this assignment consists of automatically computing the fundamental matrix for pairs of images. Students are encouraged to take digital photographs themselves.

A corner detector (which is provided in matlab image processing toolbox, e.g. `canny` (for canny corners)or `detectHarrisPoints` (for harris corners)) should be used to extract interest features in both images. Features should be compared using some similarity measure, e.g. zero-mean normalized cross-correlation, sum-of-square-differences or sum-of-absolute differences of a small window centered on the feature. The search region for putative correspondences is typically limited to part of the image. Features that are each other mutual best matches can be used as putative correspondences.

Students are expected to implement a complete algorithm for automatic computation of the fundamental matrix. This includes the 7-point algorithm, RANSAC, the normalized 8-point algorithm and a non-linear iterative algorithm to compute the fundamental matrix. Additional matches should be identified along the epipolar lines.

Students are encouraged to experiment with the algorithm: use different image pairs, vary the ratio of relative scene depth vs. baseline, and use different parameter settings for the algorithm: vary similarity measures, number of features, search range, etc. A small viewer that allows clicking a point in one of the images and having the corresponding epipolar lines appear in both images should also be implemented.

Students are encouraged to implement the solution for this assignment using Matlab. Matlab provides all the necessary function to read, display and manipulate images (try `help images`), as well as algorithms for linear least-squares (use `svd`) and non-linear least squares (`lsqnonlin`). Matlab also allows obtaining coordinates of clicked points (`ginput`), random numbers (`rand`), etc.

The different image pairs overlaid with the matched points and with some epipolar lines should be placed on the report, together with the obtained residual errors obtained at the different stages of the algorithm (use symmetrical epipolar distance for all the matches obtained in the last stage of the algorithm) and a discussion of the results and findings. Source code should also be attached to the end of report.

The following is a suggested structure for your report, as well as the rubric that we will follow when evaluating reports. You don't necessarily have to organize your report using these sections in this order, but that would likely be a good starting point for most projects.

- **Title, Author(s)**
- **Introduction (10%)**: Describe the problem you are working on, and an overview of your results
- **Methods (30%)**: Discuss your approach for solving the problems. Why is your approach the right thing to do? How to implement your approach?
- **Data (10%)**: Describe the data you are working with for your project. What type of data is it? Where did it come from? How much data are you working with? Did you have to do any preprocessing to use this data in your project?
- **Experimental results (30%)**: Discuss the experiments that you performed to demonstrate that your approach solves the problem. You should use known angles, length ratios etc to test your results and include graphs, tables, or other figures to illustrate your experimental results and the residual errors.
- **Conclusion (5%)** Summarize your key results - what have you learned?
- **Attachment(5%)** Source code.
- **Writing / Formatting (10%)** Is your paper clearly written and nicely formatted?