

## Assignment 3

- Line detection based on Hough-voting

You may use own photograph or the provided image (input\_ex3.jpg). As always, please compute a **grayscale image** (value range  $[0, \dots, 1]$ , data type double) and use it here!

**A) Implement a program** which detects lines in an image based on **Hough-voting**. Do **not** use the MATLAB function *hough* here! You may use it for a comparison with your own results.

- Read the input image and convert it to a grayscale image with a value range  $[0, \dots, 1]$ . Plot the result image.
- Apply a GoG filtering (use your code from assignment 2) to the grayscale image in order to derive gradient images in x- and y-direction.
- Compute the gradient magnitude image.
- Find and apply an appropriate threshold on the gradient magnitudes in order to derive representative edge pixels (trial and error). Plot the binary edge image.
- Implement a function for Hough line detection.
  - Inputs: Binary edge image (from d.), GoG-filter outputs (from b.)
  - Outputs: Hough voting array  $H$ , index arrays for the parameters  $\theta$  and  $\rho$
- Hints (see slides):
  - Use the polar line representation
  - Incorporate available information about the gradient direction to speed up processing.
- Plot the resulting Hough voting array  $H$ .
- Use the MATLAB function *houghpeaks* to find local maxima of  $H$ .
- Plot the found extrema on your figure of step g.
- Use the MATLAB function *houghlines* to derive the corresponding lines.
- Plot the lines on the figure of step a.