Exercise set 6: Features detection

Bonus submission deadline: 28.04.2021 23:59

The exercises marked with **BONUS** should be returned in Moodle by the submission deadline. Successfully solving it will give you bonus points to boost your final grade. During the exercise session, the solution of the exercises will be discussed.

Exercise 1: Image gradients (BONUS)

Computing the gradients of images is the core of feature detection. Given an image I its gradient magnitude at pixel (u, v) is defined as

$$\nabla I(u,v) = \sqrt{\nabla_x I(u,v)^2 + \nabla_y I(u,v)^2}, \tag{1}$$

where $\nabla_x I$ and $\nabla_y I$ are the partial derivatives of the image. These can be computed as a convolution of the image with an appropriate kernel, some popular choices are:

Roberts kernel

$$R_x = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad R_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{2}$$

Prewitt kernel

$$P_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad P_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{3}$$

Sobel kernel

$$S_x = \frac{1}{8}\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad P_y = \frac{1}{8}\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{4}$$

Your task in this exercise is to implement the function dI = gradient_image(I, method, th), which computes the magnitudes of the gradients in the image I using the given method. The magnitudes should be normalized to be between 0 and 1. The parameter method is a string that can be sobel, roberts or prewitt, which determines what method is used to compute the partial derivatives. If a third optional parameter th is given, the gradient image is binarized using the threshold th. Particularly, let dI_{max} denote the maximum value in dI, then each pixel in dI is set to 1 if its intensity is $\geq th \cdot dI_{max}$ and set to 0 otherwise. A few notes

- you can assume I is grayscale.
- You may want to check the matlab function conv2
- Note that ∇I,$\nabla_x I$ and $\nabla_y I$ should all have the same size of I.

Exercise 2: Canny edge detectors

Complete the (simplified) canny edge detector canny(I, thLow, thHigh). Follow the following steps:

1. Compute the gradients $\nabla_x I$, $\nabla_y I$ using the sobel operator as you did before. Set the magnitudes to 0 on the borders of the image.

2. Compute the magnitude of the gradients using (1) and the directions of the gradients with $\theta = \arctan \frac{\nabla_y I}{\nabla_x I}$. Cluster the directions to the main 8 directions of the compass (north, south, west, east, NE, NW, SE, SW).

3. Perform **non-maximum suppression**: for each pixel, check the neighbor pixels in the perpendicular direction of the gradient. If either of the neighbor pixels has a stronger gradient, set the gradient at that pixel to 0.

4. Perform **hysteresis threshold**: set all gradients below the lower threshold to 0 and all gradients above the higher threshold to 1. For the pixels with gradient between the thresholds, set those to 1 if there is a strong gradient in their 3×3 neighbourhood and to 0 otherwise.

Exercise 3: Harris corner detector

Nothing to do here, the TA will go through the function harris.m, which has a (slightly simplified) implementation of the harris corner detector and explain step-by-step the algorithm.