Version: 26.04.2023

Assignment 2

Gradient of Gaussian Filtering, Förstner Interest Operator

Submission Deadline: 10.05.2023, 11 am

Use the grayscale version of the provided image `ampelmaennchen.png`. For gray conversion use built-in functions such as `rgb2gray(img)` in Octave or `cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)` in Python. Normalize the image to **range** $[0.0, 1.0]$ and make sure to use **double-precision floating-point** format.

Task 1 – Gradient of Gaussian (GoG) Filtering

a) Compute continuous 5×5 GoG-filter kernels for convolution in x- and y-direction.

$$G_x = \frac{\partial G(x,y,\sigma)}{\partial x} = -\frac{x}{2\pi\sigma^4}\, e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Example: for standard deviation $\sigma = 0.5$ the two 2D kernels are:

$$G_x = \frac{\partial G(x,y,0.5)}{\partial x} = \begin{bmatrix} 0.0000 & 0.0001 & 0.0000 & -0.0001 & -0.0000 \\ 0.0002 & 0.0466 & 0.0000 & -0.0466 & -0.0002 \\ 0.0017 & 0.3446 & 0.0000 & -0.3446 & -0.0017 \\ 0.0002 & 0.0466 & 0.0000 & -0.0466 & -0.0002 \\ 0.0000 & 0.0001 & 0.0000 & -0.0001 & -0.0000 \end{bmatrix}, \; G_y = G_x^T.$$

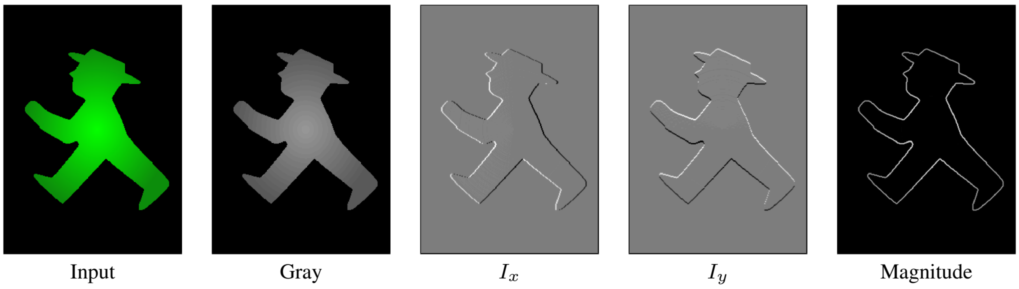
b) Apply these filters to your input image I to derive two **gradient images**: I_x and I_y (one in x- and one in y-direction). Write a function for the convolution of the image with the kernel and ignore the boundaries of the image for simplicity, i.e. no padding needed (you may use built-in convolution function).

c) Compute and visualize the **gradient magnitude** image G.

$$G = \sqrt{I_x^2 + I_y^2}$$

d) Show and interpret the results on the provided image and an image of your choice.

Sample Results for GoG Filtering:



Input Gray I_x I_y Magnitude

Task 2 – Förstner Interest Operator

Use the gradient images to identify Förstner interest points in your input image.

a) Compute the autocorrelation matrix M for each pixel using a moving window w of 5×5 pixels. Perform convolution based on this window in order to include the local neighborhood around each pixel (use I_x, I_y and ignore the boundaries of the images).

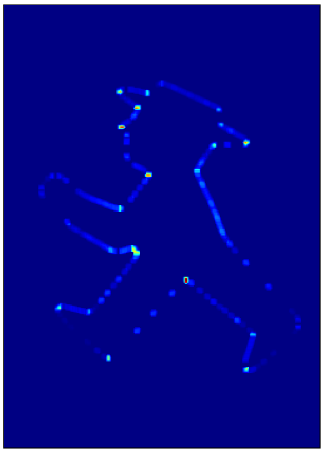
$$M = \mathrm{w} * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

b) Compute the **cornerness** w and **roundness** q from M for each pixel and store the values in two matrices W and Q. Plot W and Q with an appropriate colormap (`imagesc`, e.g. `jet`).

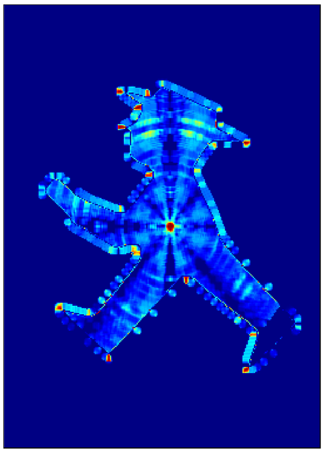
c) Derive a binary mask of potential interest points by simultaneously applying the thresholds $t_W = 0.004$ and $t_Q = 0.5$ on W and Q, respectively.

d) Plot an overlay of the initial input image with the detected points (`plot`).

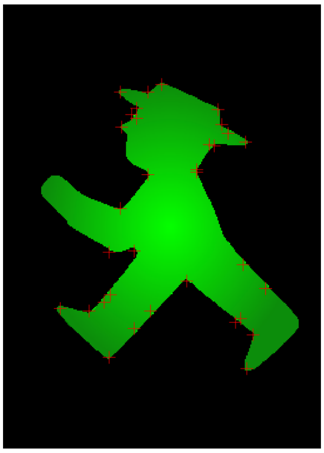
Sample Results for Förstner Interest Operator:



Cornerness



Roundness



Förstner

Submit your source code, the resulting images, and a `.pdf` (or `.ipynb`) file containing your answers.