Exercise 3: Digital Images, Neighborhoods, Point Operations and Filters

Exercises

3.1 An analogue function f_a has as maximum spatial frequency value of f_{sx} = 10^2 mm^{-1} . This function is to be converted into a discrete function f_d by sampling with equidistant sampling points spaced at a distance $\Delta x = 0.05$ *mm.* What can be said in general about the sampled function f_d ?

3.2 For a square grid, mark the points of equal distance according to the discrete maximum distance D_8 for the neighboring points of a pixel p

3.3 An image b_{SB} of dimension 8x8 shows a regular black and white chequerboard pattern with 64 squares (size 1x1); the pixel $b_{SB}(0, 0)$ has the value black.

a) What type of image is the image b_{SB} ?

b) Sketch the brightness profile (intensity line profile) of the third line (j = 2) of b_{SB} (black=0, white=1).

c) Sketch the histogram $h(b_{SB})$ of the image b_{SB} .

3.4 What is the essential difference between
a) a point operation and a local operation
b) between a homogeneous and an inhomogeneous image operation?

3.5 Formally describe the transformation function T_{pI} for an inversion of
a) binary images $b(i, j)$
b) gray scale images $g(i, j)$.

3.6 Image Enhancement Filters

1. Load a grayscale image of your choice into MATLAB. Add a salt and pepper noise to the image (`imnoise`).
2. Try to reduce the noise using
   a. an *average filter* specified as a convolution kernel (`conv2` or `filter2`)
      i. Try different filter kernel sizes, e.g. 3x3 and 6x6.
   b. A g*aussian filter* approximated by a convolution kernel (`conv2, filter2`)
   c. The MATLAB function fspecial provides special convolution kernels for filters "with a name" (for example gaussian). Filter the image with a gaussian filter using `fspecial` and `filter2.`
3. Apply a *median filter* to the image with the added salt and pepper noise (`medfilt2`)
   a. Try different filter kernel sizes, e.g. 3x3 and 6x6.
4. Formulate a local image operation T_L (Laplace operator) that subtracts the pixel values of the N_4 neighborhood from four times the value of the central pixel and apply this kernel using `filter2` to the original noise-free image.