Filtering & Edge detection

1 Numerical Exercises

1. Consider the following 1D image pixel arrays A and B

$$A = [3, 1, 2, 1] \qquad\qquad B = [7, 7, 6, 4]$$

   (a) Compute the filter F, which was applied as **convolution** to the pixel array A resulting in the output B. Assume that zero padding was applied.

$$B = A{*}F$$

   (b) Compute the filter F, which was applied as **cross-correlation** to the pixel array A resulting in the output B. Assume that zero padding was applied.

$$B = A \otimes F$$

   (c) Compute the convolution signal C between the pixel array A and B. Use "reflect across edge" padding. The output should have the same size as the input signal.

2. In a medical imaging context (e.g., CT or MRI scans), you are tasked with detecting the edges of anatomical structures in 3D.

   (a) What type of noise reduction filter would you apply to a 3D medical image to reduce noise while preserving fine details like organ boundaries?

   (b) Outline the rough steps involved in implementing a 3D edge detection pipeline. Be specific about each step, including preprocessing, noise reduction, gradient calculation, and edge detection.

   (c) What strategies would you use to optimize the performance of 3D edge detection for large volumes?

3. What is the convolution output if filter F is applied to the following A matrix assuming zero padding?

$$F = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \qquad\qquad A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 8 & 7 & 6 & 5 \\ 4 & 3 & 2 & 1 \end{bmatrix}$$

4. Find the two 1D separable filters $a, b \in R^{3\times 1}$ resulting in the following 2D filters such that:

$$A = ab^T$$

(a) $A = \begin{bmatrix} 0 & 0 & 0 \\ 2 & 2 & 2 \\ 0 & 0 & 0 \end{bmatrix}$

(b) $A = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

(c) $A = \begin{bmatrix} 2 & 4 & 2 \\ 1 & 2 & 1 \\ 2 & 4 & 2 \end{bmatrix}$

5. In the lecture you've seen the Prewitt filter which calculates a partial derivative. For example, the filter

$$G_x = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

calculates the first partial derivative in the x direction. It approximates $\frac{\partial I}{\partial x}$. In a similar fashion, derive a filter G_{xx} which approximates the second order partial derivative in x $\frac{\partial^2 I}{\partial x^2}$. The filter should be of size 1x3.

(a) Derive the filter $G_{xx,c}$ using central differences (i.e. one pixel on either side of the current pixel)

(b) Derive the filter $G_{xx,f}$ using forward differences (i.e. using only information to the right side of the current pixel). The filter size may be larger than 1x3.