Filtering & Edge detection

1 Numerical Exercises

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

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

   (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.

$$B = A*F$$

   Solution:
   $F = [1, 2, 1]$

   (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.

$$B = A\otimes F$$

   Solution:
   $F = [1, 2, 1]$

   (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.
   Solution:
   To achieve an output with the same size as the input arrays, the padding can be applied either to A or to B depending on which array is the filter. Another important observation is that both arrays have an even number of entries, thus there exists an ambiguity on which side the output is cropped. This ambiquity should be considerd while using standard programming libraries. We provide the solution with the uncropped signal.

   If B is the filter, the uncropped solution is:
   $C = [50, 51, 39, 30, 35]$ with the padded signal $A = [1, 3, 3, 1, 2, 1, 1, 2]$

   If A is the filter, the uncropped solution array is:
   $C = [49, 46, 39, 35, 36]$ with the padded signal $B = [7, 7, 7, 7, 6, 4, 4, 6]$

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?

Solution:

(a) The 3D bilateral filter is ideal because it smooths noise while preserving important anatomical edges by adapting to the local intensity. It outperforms Gaussian filters in maintaining critical details, such as fine tissue structures or tumor margins.

(b) Here are the detailed steps for implementing a 3D edge detection pipeline:

- Noise Reduction: First apply a 3D bilateral filter to reduce noise while preserving edges.
- 3D Gradient Calculation: (i) Compute 3D gradients using 3D Sobel operators or 3D Prewitt operators. (ii) Calculate gradients in x, y, and z directions separately. (iii) Combine the directional gradients to obtain the gradient magnitude.
- Edge Detection: Apply a 3D extension of the Canny edge detection algorithm: (i). Use the computed gradient magnitudes. (ii). Threshold the gradient magnitudes to identify potential edges. (iii). Apply 3D non-maximum suppression to thin the detected edges.

(c) To optimize performance, use separable filters to break down 3D convolutions into multiple 1D convolutions, significantly reducing complexity. Additionally, leverage GPU acceleration for parallel processing, which allows faster computation, especially for large 3D datasets.

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

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

Solution:

$$F{*}A = \begin{bmatrix} 1 & -3 & -3 & -3 \\ 7 & 1 & 4 & 2 \\ 14 & 10 & 11 & 3 \\ 11 & 9 & 7 & 1 \end{bmatrix}$$

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

Solution:
$a = [0, 1, 0]$ $\qquad b = [2, 2, 2]$

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

Solution:
$a = [1, 0, 0]$ $\qquad b = [0, 0, 1]$

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

Solution:
$a = [2, 1, 2]$ $\qquad b = [1, 2, 1]$

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)

Solution:

To derive this filter, we can consider a 1D function f(x) as our desired filter computes a derivative only in one direction. The task is now simplified and we need to approximate $f''(x)$ using a central difference scheme with step size Δx, i.e. calculate an estimate of $f''(x)$ using the function values at the positions $f(x - \Delta x)$, $f(x)$, $f(x + \Delta x)$. A taylor expansion of $f(x)$ around the points x evaluated at $x + \Delta x$ and $x - \Delta x$ yields

$$\begin{aligned} f(x+\Delta x) &= f(x) + \Delta x f'(x) + \Delta x^2 \frac{f''(x)}{2} + \Delta x^3 \frac{f'''(x)}{6} + \Delta x^4 \frac{f''''(x)}{24} + \text{H.O.T} \\ f(x-\Delta x) &= f(x) - \Delta x f'(x) + \Delta x^2 \frac{f''(x)}{2} - \Delta x^3 \frac{f'''(x)}{6} + \Delta x^4 \frac{f''''(x)}{24} + \text{H.O.T} \end{aligned}$$

where H.O.T. denotes higher-order terms. Since we are dealing with pixels here, our discretization step size Δx is equal to one. Therefore, we obtain

$$\begin{aligned} f(x+1) &= f(x) + f'(x) + \frac{f''(x)}{2} + \frac{f'''(x)}{6} + \frac{f''''(x)}{24} + \text{H.O.T} \\ f(x-1) &= f(x) - f'(x) + \frac{f''(x)}{2} - \frac{f'''(x)}{6} + \frac{f''''(x)}{24} + \text{H.O.T} \end{aligned}$$

To approximate the second derivative we can sum up the two terms above and solve for $f''(x)$

$$f(x+1) + f(x-1) - 2f(x) = f''(x) + \frac{f''''(x)}{12} + \text{H.O.T}$$

Since only an approximation is desired, we can discard the higher-order terms of order ≥ 3 and get the central second order finite difference

$$f''(x) \approx f(x-1) - 2f(x) + f(x+1)$$

In matrix notation, we can write

$$G_{xx,c} = \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$$

(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.

Solution:

Similarly to the central difference case, we can write the taylor approximations for the neighboring pixels. Consider the points x, $x + \Delta x$, $x + 2\Delta x$, $x + 3\Delta x$. Following the same procedure as above, we can find that the second order forward difference can be written as

$$f''(x) = 2f(x) - 5f(x+1) + 4f(x+2) - f(x+3)$$

In matrix notation we get

$$G_{xx,f} = \begin{bmatrix} 2 & -5 & 4 & -1 \end{bmatrix}$$