Concordia University

Department of Computer Science & Software Engineering

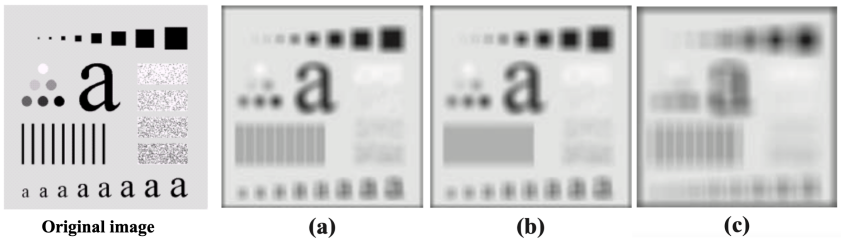
COMP 478/6771 Image Processing

Assignment 2 - Due Date: *October 17, 2023*

Part I: Theoretical questions

1. Image Filtering Questions:

   a) (**5 points**) We have the following images (a), (b), and (c) that are obtained by blurring the original image with square box filters of sizes 23x23, 25x25, and 45x45 elements, respectively. In the original image, there are a series of vertical bars in the bottom left, which are 5 pixels wide and 100 pixels long, and their separation is 20 pixels. We can see that the blurred vertical bars still show clear separation between them in image (a) and (c), but they have merged into one block in image (b), despite that the filter kernel size used falls between those for image (a) and (c). Please explain the reason for this.



Original image (a) (b) (c)

   b) (**7 points**) Write down a 5-by-5 Laplacian-like filter with the center element equal to -16. What general rules should you follow to build such a filter? If we apply this filter to an image *I*, do we get sharper edges compared to those images obtained using a real 3-by-3 Laplacian filter? Explain your answer.

2. Fourier Transform Questions:

   a) (**7 points**) Show the validity of the following translation (shifting) properties of 2D discrete Fourier transform pairs:

   i) $f(x,y)e^{j2\pi\left(\frac{u_0 x}{M}+\frac{v_0 y}{N}\right)} \Leftrightarrow F(u-u_0, v-v_0)$

   ii) $f(x-x_0, y-y_0) \Leftrightarrow F(u,v)e^{-j2\pi\left(\frac{ux_0}{M}+\frac{vy_0}{N}\right)}$

   b) (**7 points**) Please find the Fourier Transform of the following box function

$$f(t) = \begin{cases} 2, & 0 \le t \le W \\ 0, & otherwise \end{cases}$$

Part II: Programming questions – adaptive thresholding (22 points)

1. Download the two images from the assignment folder then carry out the **adaptive thresholding** algorithm to binarize the text in the image without the shadow/stains. For RGB images, please use rgb2gray() to convert the image for further processing. Please show your results, discuss your choice of parameters and filters, and compare your results with the ***adaptthresh()*** function in MATLAB (or the equivalent function in the software library of your choice).

> In ***adaptive threshold*** unlike fixed threshold, the threshold value at each pixel location depends on the neighboring pixel intensities. To calculate the threshold $T(x,y)$, the threshold value at pixel location (x, y) in the image, we perform the following steps:
>
> 1) A *m-by-m* region around the pixel location is selected. *m* is selected by the user.
> 2) The next step is to calculate the weighted average of the *m-by-m* region (averaging filter). You can choose any averaging filter of your choice and provide your reasons. We will denote the weighted average at location (x,y) by $WA(x, y)$.
> 3) The next step is to find the Threshold value $T(x, y)$ by subtracting a constant parameter, $\boldsymbol{c}$ from the weighted average value $WA(x, y)$ calculated for each pixel in the previous step. The threshold value $T(x, y)$ at pixel location (x, y) is then calculated using the formula:
>
> $$T(x, y) = WA(x, y) - c$$
>
> Now you will just need to apply the threshold transfer function to obtain the result.