BİL 467/561 – Image Processing

Exercise #2

1. For this question, you will implement the box filter as detailed in our lecture slides. **You are not allowed to use OpenCV's or any other image/signal processing API's built-in box filter function.** Your implementation must be for an arbitrary filter size, that is, your box filter must take the filter size as a parameter and work for any filter size. To handle border pixels, use zero padding. You can assume that your inputs will be 8-bit grayscale images. You do not need handle any other input types.

   Once you implement your own box filter function, you will apply it to the provided test image ("lena_grayscale_hq.jpg" on Piazza/Resources) for three filter sizes, namely, 3x3, 11x11 and 21x21. You will display your outputs and make sure that they do not have any visual artifacts. Call these output images output_1_1, output_1_2 and output_1_3.

   You will then apply OpenCV's built-in box filter function **(use blur() or boxFilter() – be careful about the normalization flag of the second API function)** to the same test image with the same filter sizes. You will display your outputs and make sure that they do not have any visual artifacts. Call these output images output_2_1, output_2_2 and output_2_3.

   Finally, you will take the absolute difference between these two sets of output images

   (abs(output1_x – output_2_x))

   and display these difference images. You will compute and print the max absolute difference in all absolute difference images. **The maximum allowed absolute difference is 3**.

2. For the second part you will implement the box filter as a separable filter. **You are not allowed to use OpenCV's or any other image/signal processing API's built-in separable box filter function.** Your implementation must be for an arbitrary filter size, that is, your separable box filter must take the filter size as a parameter and work for any filter size. To handle border pixels, use zero padding. You can assume that your inputs will be 8-bit grayscale images. You do not need handle any other input types.

   Once you implement your own separable box filter function, you will apply it to the provided test image ("lena_grayscale_hq.jpg" on Piazza/Resources) for three filter sizes, namely, 3x3, 11x11 and 21x21. You will display your outputs and make sure that they do not have any visual artifacts. Call these output images output_3_1, output_3_2 and output_3_3.

   Finally, you will take the absolute difference between these two sets of output images

   (abs(output3_x – output_2_x))

   and display these difference images. You will compute and print the **max absolute difference** in all absolute difference images. **The maximum allowed absolute difference is 3**.

Hint: The 1D filters for the 3x3 box filter case are given as:

$$\frac{1}{3}\begin{bmatrix}1 & 1 & 1\end{bmatrix} \text{ and } \frac{1}{3}\begin{bmatrix}1\\1\\1\end{bmatrix}$$

3. We discussed that separable filters are more efficient in the sense that they can obtain the same output with less computation. Can you imagine a more efficient implementation for the box filter even compared to the separable implementation?