Problem Set 2

Due: March 5, 2019

1) [2 pts] Prove the duality property of the discrete Fourier transform: If $M \times N$ image, $x[m, n]$ has DFT $X[k, l]$ then $X[m, n]$ has DFT $MNx[-k \mod M, -l \mod N]$.

2) [4 pts] a) Consider a 3×3 mask that averages the 4 non-diagonal neighbors of a pixel $[m, n]$, but excludes the pixel itself. Find the equivalent filter $H(u, v)$ in frequency domain. Is this a low-pass or high-pass filter?

b) For an $M \times N$ image, you are asked to repeatedly apply a Gaussian filter, with $H(u, v) = e^{-(u^2+v^2)/2D^2}$ for some D. If you apply the filter repeatedly for K times for a sufficiently large K, what happens to the resulting image? If the lowest positive real number representable by the computer is ϵ, what is the minimum K to guarantee this behavior (here assume/note that $H(u, v)$ will also be stored digitally in the computer)?

3) [3 pts] In class we saw both mean and median filters as examples of neighborhood operations. Let $\mathcal{N}(i)$ denote a given neighborhood of pixel i. For the rest of the exercise, you may assume that the number of pixels in the neighborhood, i.e. $|\mathcal{N}(i)|$ is odd. Let I_j denote the intensity of pixel j.

a) Show that the output of the mean filter applied to the neighborhood $\mathcal{N}(i)$ is the solution to

$$\min_{I_i^{new}} \sum_{j \in \mathcal{N}(i)} |I_i^{new} - I_j|^2$$

b) Show that the output of the median filter applied to the neighborhood $\mathcal{N}(i)$ is the solution to

$$\min_{I_i^{new}} \sum_{j \in \mathcal{N}(i)} |I_i^{new} - I_j|$$

c) Suppose we try to solve

$$\min_{I_i^{new}} \sum_{j \in \mathcal{N}(i)} w_j |I_i^{new} - I_j|$$

for some weights $\{w_j\}$. Then how can I_i^{new} be calculated from the knowledge of I_j for $j \in \mathcal{N}(i)$?

4) [6 pts] Matlab Exercise: In this exercise, you will re-generate three examples we saw in class.

a) [1 pt] Use the phase of the Fourier spectrum of Lena image, and the magnitude of the Fourier spectrum of the kneeling man image (provided to you), to generate a combined image with the corresponding phase and magnitude Fourier spectra. Display the input images and their relevant Fourier specta, and the final output image.

b) [2 pt] Take the DCT of each distinct (i.e. not sliding) 8×8. Keep the largest (in magnitude) 10 DCT coefficients, and set the rest to zero. Take the inverse DCT of each

block to generate a new image. Do the same with DFT and inverse DFT (i.e. FFT). Display the images.
(*Hint:* The built-in function blkproc may be helpful. Also defining a function that performs the given transform, then keeps the largest 10 coefficients and then does the inverse transform will also be useful.)

c) [3 pts] Perform the sharpening exercise with the coins image (imread('eight.tif') in MATLAB). Read-in the image, then perform lowpass filtering with an averaging filter (use imfilter). Generate the high-pass image by subtracting this low-pass image from the original. Generate the sharpened image as original image plus 2 times the high-pass image. Display the original, low-pass, high-pass and sharpened images.

Bonus MATLAB Exercise below

5) [3 pts] Bonus Matlab Exercise: In class we studied the use of median filtering for removing salt-and-pepper noise. However, if the noise level is too high, then median filter will fail. In this exercise, we will try to do a simple modification to median filtering for improved performance. The original and noisy image (corrupted at 40% level) are provided to you in a file called HW2_bonus.mat.

Let $\mathcal{P}$ be the set of all pixels in the image. Following the notation of Question 3, the median filter solves

$$\min_{\{U_i\}} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}(i)} |U_i - I_j|$$

to generate a new image, where we replaced I_i^{new} from Question 3 with U_i for brevity. This essentially makes sure our output is consistent with the noisy observations $\{I_j\}$ we had. Now, we can enforce an additional level of consistency in our image, by saying that the filtered values $\{U_i\}$ should be similar to their neighbors as well. Thus, we set this up as

$$\min_{\{U_i\}} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}(i)} |U_i - I_j| + \lambda \sum_{(i,j) \text{ are neighbors}} |U_i - U_j|$$

This is hard to solve. Hence, we will do a simple iterative solution to this. We will use 3×3 neighborhoods for both terms, and set $\lambda = 1$. We will start our algorithm by taking the median filter of the noisy dataset, call this iteration 0. Then at iteration t, we will use our estimate from iteration $t-1$ to solve the above problem at each neighborhood as:

$$\min_{U_i^{(t)}} \sum_{j \in \mathcal{N}(i)} |U_i^{(t)} - I_j| + \sum_{(i,j) \text{ are neighbors}} |U_i^{(t)} - U_j^{(t-1)}|$$

Below is what I get after 50 iterations (from the left: original, noisy, median filter, proposed filter):