# ECE 253, Homework 4

## Due: Friday November 11, 2022 by 11:59pm

The first two problems should be done using Matlab, and the last one should be done by hand. Submit your homework in Canvas on Gradescope. Everything can be uploaded as one PDF file– include your answers to each question and your Matlab code (cut and paste it in). Include your full name and PID.

1) **2D Sampling and Aliasing**

For this problem, it might be useful to remember the following Fourier transform pair:

$$cos(2\pi(u_0 x + v_0 y)) \leftrightarrow \frac{1}{2}(\delta(u - u_0, v - v_0) + \delta(u + u_0, v + v_0))$$

a) Try the following in Matlab:

```
[x y] = meshgrid(0:256,0:256);
z1 = cos( 2 * pi * 1/32 .* x - 2 * pi * 1/128 .* y);
z2 = cos( 2 * pi * 1/4 .* x - 2 * pi * 7/8 .* y);
z3 = cos( 2 * pi * 1/2 .* x - 2 * pi * 1/2 .* y);
```

Look at the images (matrices) x, y, z1, z2, z3 (and you might want to inspect their numeric values too). The meshgrid function has the effect of sampling the three underlying continuous functions with sampling period T=1 in both the x and y directions. Discuss the appearances of the three sampled images, z1, z2, and z3. Are the underlying continuous functions getting undersampled? critically sampled (sampled at Nyquist)? oversampled? Explain what you see in terms of the spatial frequencies and the sampling frequency.

b) What values would we have to choose for "a" and "b" in the expression:

```
z4 = cos( 2 * pi * a .* x - 2 * pi * b .* y);
```

so that the sampled function would be aliased and would have an appearance identical to that of the sampled and displayed function z1?

2) Pre-filtering to Reduce Aliasing before Sampling

Aliasing can be analyzed by finding the foldover frequency, and looking at the area under the folded-over curve. Prefiltering reduces the aliasing power, but it also reduces the signal power in the part of the spectrum we would like to keep intact. This problem will examine this reduction in signal power and noise power that comes from pre-filtering.

Read in the images barbara.tif and baby.tif. We will use a 512x512 section of each, as follows:

```
>> barbBig = barbara(1:512,37:548);
>> babyBig = baby(201:712,201:712);
```

The power spectrum is given by the square of the magnitude of the Fourier Transform. So, the power spectrum for the cropped baby image, prior to downsampling, is given by:

```
babyfft_1 = fftshift(abs(fft2(babyBig)).^2);
```

and the total image energy can be found by summing all the values in that array. We can visualize the power spectrum by using

```
>> babyspectrum = log(babyfft_1 + 1);
>> imshow(babyspectrum / max(max(babyspectrum)))
```

a) In the power spectrum babyfft_1, where is the DC coefficient located? How can you check that? You will need to understand where the function fft2 puts the DC coefficient, and then where fftshift relocates it to.

b) Compare visually the images babyBig and barbBig, and their power spectra. Based on the visual comparison, which of the two images has more high frequency content? Which one do you expect to have more of an aliasing problem from downsampling?

c) We will use the following commands to generate two downsampled versions of babyBig and similarly generate two downsampled versions of barbBig. Be sure you understand what imresize with the 0.125 and 'nearest' parameters is doing:

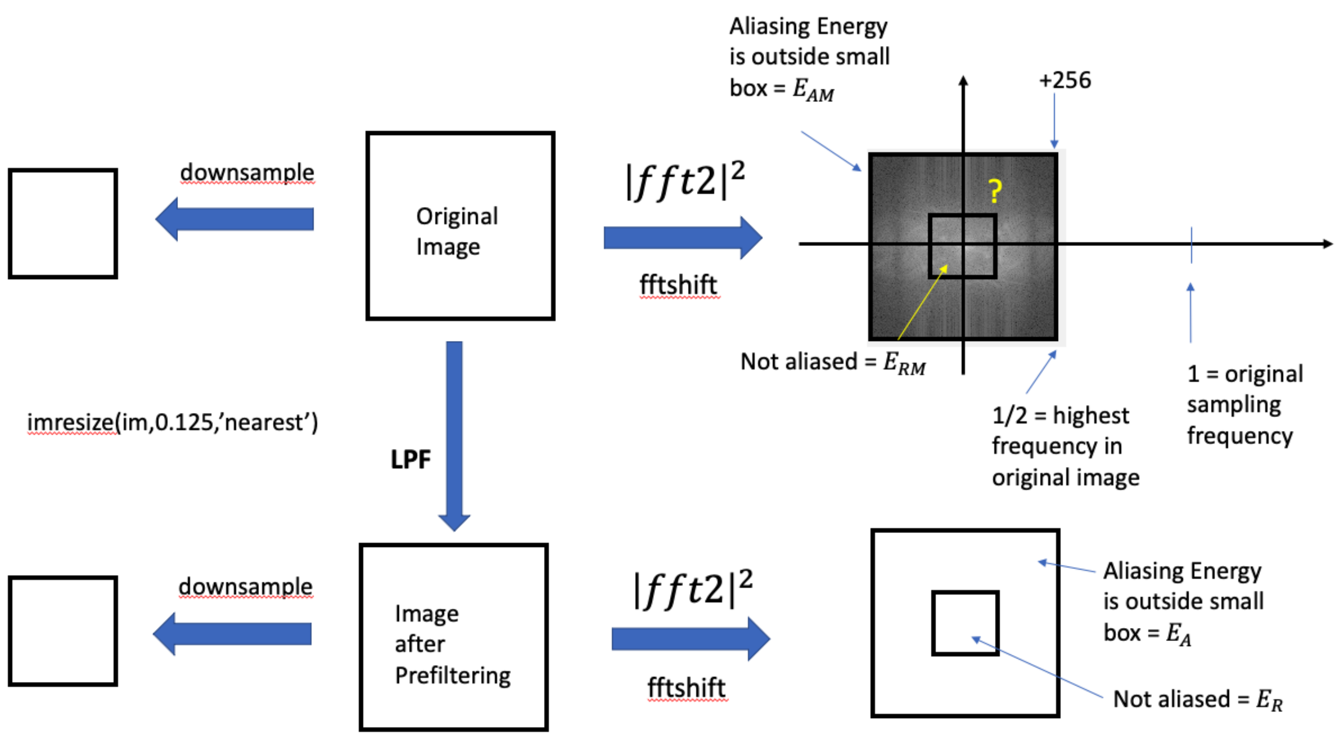
```
baby64_1 = imresize(babyBig,0.125,'nearest');
babylow = filter2(ones(8)/64,babyBig);
baby64_2 = imresize(babylow,0.125,'nearest');
```

Compare visually the two $64\times 64$ versions of baby, and of barbara. Comment on the differences and explain them. In particular, address whether the pre-filtering helps more for baby or for barbara.

d) Assume that (for each image) the underlying continuous image field was sampled at the Nyquist rate to form the original (cropped) $512\times 512$ discrete image. So the original sampling distance is 1, and the original sampling frequency is 1, which is the critical sampling frequency.

Since we are downsampling by a factor of 8, that corresponds to a new sampling distance of 8, and a sampling frequency of 1/8. So what portion of the spectrum is aliased and what portion is not? The tricky thing here is figuring out things in Matlab's pixel units and Matlab's coordinate system. Matlab actually indexes the array from 1 to 512 on both axes. But after doing the fftshift, it is useful to think of this as going from -256 to +256, where the highest frequency in u (or v) occurs at the first/last row/column.

For the barbBig image, for the subsampling from 512 x 512 down to size 64 x 64, estimate the reduction in aliasing power and the reduction in signal power that come from the prefiltering.



3) **Non-rectangular sampling**

An image has its spectrum confined to the region shown below.

(a) Find a generator matrix for the *rectangular* sampling lattice that has the lowest sampling density which can still ensure no aliasing (no overlap of spectral replications).

(b) Find a generator matrix for the *non-rectangular* sampling lattice that has lowest sampling density that could prevent the spectral replications from overlapping.

(c) What percentage reduction in samples can we get by sampling on the non-rectangular grid as opposed to the rectangular grid?