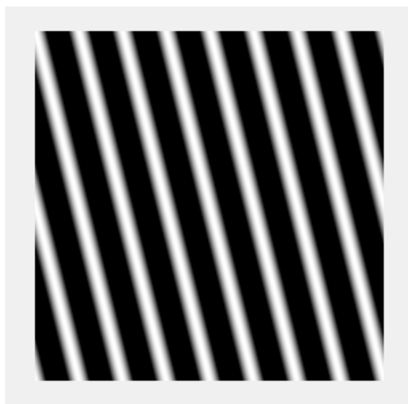
ECE 253 (Fall 2022), Homework 4 Solution

1) **2D Sampling and Aliasing**

a) The sampling period T is 1 in both the x and y directions. Therefore the sampling rate, which 1/T, is also equal to 1.

```
z1 = cos( 2 * pi * 1/32 .* x - 2 * pi * 1/128 .* y);
```

For z1, the frequency of the signal is 1/32 in the x direction and 1/128 in the y direction. So the sampling rate is much higher than twice the highest frequency. The signal is highly oversampled. It looks like this:



```
z2 = cos( 2 * pi * 1/4 .* x - 2 * pi * 7/8 .* y);
```

For z2, the frequency of the signal is 1/4 in the x direction, so the sampling rate is fast enough. But it is not fast enough for y. The sampling rate (1) is much less than twice the y frequency (7/8). Aliasing will fold over the frequency and make it appear like a frequency of 1/8, so lower frequency than the x direction. Looking at the array with imshow doesn't help us much because the frequencies are too high for our visual system to see what is going on.

But we can look at the numbers in the array:

1.0000	0.0000	-1.0000	-0.0000	1.0000	0.0000	-1.0000	-0.0000	1.0000	0.0000
0.7071	-0.7071	-0.7071	0.7071	0.7071	-0.7071	-0.7071	0.7071	0.7071	-0.7071
-0.0000	-1.0000	0.0000	1.0000	-0.0000	-1.0000	0.0000	1.0000	0.0000	-1.0000
-0.7071	-0.7071	0.7071	0.7071	-0.7071	-0.7071	0.7071	0.7071	-0.7071	-0.7071
-1.0000	-0.0000	1.0000	-0.0000	-1.0000	0.0000	1.0000	-0.0000	-1.0000	0.0000
-0.7071	0.7071	0.7071	-0.7071	-0.7071	0.7071	0.7071	-0.7071	-0.7071	0.7071
-0.0000	1.0000	0.0000	-1.0000	-0.0000	1.0000	0.0000	-1.0000	-0.0000	1.0000
0.7071	0.7071	-0.7071	-0.7071	0.7071	0.7071	-0.7071	-0.7071	0.7071	0.7071
1.0000	-0.0000	-1.0000	-0.0000	1.0000	-0.0000	-1.0000	-0.0000	1.0000	-0.0000
0.7071	-0.7071	-0.7071	0.7071	0.7071	-0.7071	-0.7071	0.7071	0.7071	-0.7071

Here we can see the period of the signal is 4 in the x direction, as it should be, but the period of the signal in the y direction is 8, which corresponds to the aliasing. So this is undersampled.

```
z3 = cos( 2 * pi * 1/2 .* x - 2 * pi * 1/2 .* y);
```

For z3, we have critical sampling, since the frequency is 1/2, which is half of the sampling frequency. And if we look at the values in the array, they alternate between -1 and +1.

b) If we choose a = 31/32 and b = 127/128 in the expression:

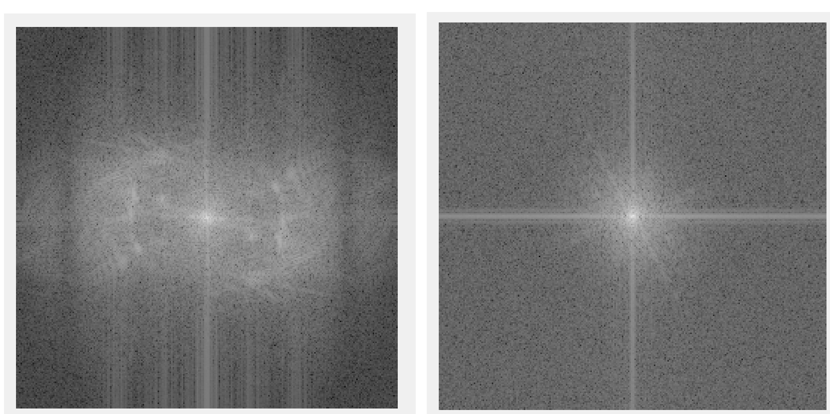
```
z4 = cos( 2 * pi * a .* x - 2 * pi * b .* y);
```

the sampled function would be alised and would have an appearance identical to that of the sampled and displayed function z1. Other values of a and b will have the same effect. All values of the form $a = k \pm 1/32$ and $b = k \pm 1/128$ would work, where k is any integer.

2) Pre-filtering to Reduce Aliasing before Sampling

a) The fft2() places the origin at top left corner. After fftshit(), the origin is moved to (257, 257).

b) The barbara image looks like it has more energy in the high spatial frequencies, so we would expect it to have more problems with aliasing from downsampling. The power spectra look like this, with the barbara spectrum on the left:



Also if you sum all the element in the array, the barbara image has more energy overall. (But you weren't asked to comment on the overall energy.)

c) The pre-filtering definitely helps more for barbara, which looks like quite strange with high frequency artifacts.

d) In order to separate the aliasing power from the signal power, we must take a look at the power

spectrum of the images before subsampling.
Because we made the assumption that the underlying continuous image field was sampled at the Nyquist rate to form the original 512×512 discrete image, the original sampling distance is 1, and the original sampling frequency is 1, which is the critical sampling frequency. So the foldover point in the original sampling scheme is equal to 1/2, because it is half of the original sampling frequency. But, since we were said to be sampling at Nyquist, no signal power is folded over.

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 the new foldover frequency is 1/16.
Translating all of these values to the Matlab pixel grid: The original spectrum goes from -256 to +256, which corresponds to going from -1/2 to +1/2. Under the new sampling scheme, the portion of the spectrum that is aliased is everything outside the foldover frequency, which is 1/16, so this corresponds to everything beyond the box that is from -32 to +32. So basically, we want to add up all the coefficients inside the 64 x 64 box centered in the array, and separately add up all the coefficient outside the box.
We can use:

```
>> [i,j] = find(barbfft_1 == max(max(barbfft_1)))
```

to verify that the DC component is located in position 257, 257. So we will take the box starting at 257-32=225 and of size 64x64.

Using `barbfft_2` to denote the power spectrum of the lowpass filtered version, the computation for reduction in aliasing power and in signal power is:

```
sig_1 = barbfft_1(225:225+63,225:225+63);
sig_2 = barbfft_2(225:225+63,225:225+63);
ps_1 = sum(sig_1(:));
ps_2 = sum(sig_2(:));
(ps_1-ps_2)/ps_1
alias_1 = barbfft_1;
alias_1(225:225+63,225:225+63) = zeros(64);
alias_2 = barbfft_2;
alias_2(225:225+63,225:225+63) = zeros(64);
pa_1 = sum(alias_1(:));
pa_2 = sum(alias_2(:));
(pa_1-pa_2)/pa_1
```

which reveals that the signal power was reduced by about 2% and the aliasing power by more than 89%.

Common error 1: Using the spectrum after down-sampling for calculation

Aliasing is high frequency components masquerading as lower frequency after sampling. Therefore, there is no way to distinguish between non-aliased legitimate signal power and aliasing power. That's why we can only estimate signal power and aliasing power from the spectrum before down-sampling.

Common error 2: Using the spectrum after logarithm for power calculation

The power spectrum is given by the square of the magnitude of the Fourier Transform. Applying logarithm to the power spectrum is just a visualization technique to enhance contrast.

3) **Non-rectangular sampling**

(a) A generator matrix for the rectangular spectral replication lattice is

$$B_{rect} = \begin{pmatrix} 6 & 0 \\ 0 & 2 \end{pmatrix}$$

and so the generator matrix A_{rect} for the sampling lattice in the spatial domain is the inverse transpose of B_{rect}:

$$A_{rect} = \begin{pmatrix} \frac{1}{6} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}$$

and the determinant is 1/12.

(b) A generator matrix for the non-rectangular spectral replication lattice is

$$B_{non-rect} = \begin{pmatrix} 3 & 0 \\ 1 & 2 \end{pmatrix}$$

and so the generator matrix for the sampling lattice in the spatial domain is the inverse transpose:

$$A_{non-rect} = \begin{pmatrix} \frac{1}{3} & \frac{-1}{6} \\ 0 & \frac{1}{2} \end{pmatrix}$$

and the determinant is 1/6.

(c) The lattice is less dense, and how much less dense can be determined by the ratio of the (magnitudes of the) determinants, which is 50%.