ECE 253a	Digital Image Processing	Handout #21	Pamela Cosman	12/2/08

Solutions to Homework 5

1. **2-D Sampling and Aliasing:**

a) The 2-D continuous Fourier transform of $\cos(2\pi(u_0x + v_0y))$ is:

$$\begin{aligned}\mathcal{F}(\cos(2\pi(u_0x + v_0y))) &= \frac{1}{2}\mathcal{F}(e^{j2\pi u_0x}e^{j2\pi v_0y} + e^{-j2\pi u_0x}e^{-j2\pi v_0y}) \\ &= \frac{1}{2}(\delta(u - u_0)\delta(v - v_0) + \delta(u + u_0)\delta(v + v_0)) \\ &= \frac{1}{2}(\delta(u - u_0, v - v_0) + \delta(u + u_0, v + v_0))\end{aligned}$$

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

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

For z, 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.

```
z1 = cos(2 * pi * 1/4 .* x -2 * pi * 7/8 .* y);
```

For z1, the sampling rate is not fast enough for y. The sampling rate (1) is much less than twice the y frequency (7/8). Aliasing will replicate the two deltas to (-1/4,1/8) and (1/4,-1/8).

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

This corresponds to Nyquist rate, since the sampling frequency is exactly twice the highest frequency. When you view it with the imshow command, it will look like a constant gray, because the human visual system can't resolve that high spatial frequency. Looking at the actual values in the array shows that they alternate between 1 and -1.

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

```
z3 = cos(2 * pi * a .* x - 2 * pi * b .* y);
```

the sampled function would be aliased and would have an appearance identical to that of the sampled and displayed function `z`. Other values of a and b (e.g., 33/32 & 129/128, etc.) will have the same effect. In fact, all values of the form $a = k \pm \frac{1}{32}$ and $b = k \pm \frac{1}{128}$ will work, where k is any integer. Several people proposed using $b = 65/128$. Note that this *is* large enough to produce aliasing, but it is not the right amount of aliasing to produce exactly the appearance of the first image z.

2. **2-D DFTs:**

   a) The DC component is located in the (0,0) position which is Matlab matrix element (1,1). You can check that by summing the intensity at all points of the image. That should give a large number corresponding to the DC component. Another way to check it is be creating a matrix of all 1's and showing that there is then only one non-zero component in the DFT, and it is located at matrix element (1,1). Several people said that the DC component is in all four corners of the fft2 array. This is not correct; just as in the one-dimensional case, the DC component is only a single coefficient, and it occupies only a single position in the array.

   `fftshift` - shifts the FFT so that the DC component is located at the center of the data, i.e. for a 128x128 image, the DC component goes from (0,0) to (64,64) which is matrix position (65,65).

   Since the dynamic range of the frequency components is large, the small components will be wiped out by the intensity of the larger components if the FFT is displayed directly. By using a `log()` you basically enhance the small components.

   b) There shouldn't be any difference between images since a shift in one domain introduces a linear phase factor in the other domain, but since we are looking at the magnitude, the phase factor term does nothing.

3. **Reducing Aliasing:**

   a) The first image, `lena64_1`, looks somewhat 'jagged', especially at edges such as at the hat or the hair. In contrast, the second image, `lena64_2`, is much more smooth. The filter essentially averages the intensities of the original image, thus reducing the aliasing. The comparison between the filtered and unfiltered versions is more striking with the baboon image. Because this image has a lot more high spatial frequencies, the unfiltered version looks particularly noisy and blocky, and the low-pass filter has more of a beneficial impact.

   b) 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. Some people looked at the power spectra only for the downsampled 64×64 versions. But after subsampling, that is after the aliasing has occurred (frequencies are folded over), it is no longer possible to separate out what was part of the original signal power from what was folded over. Note that to get *power* you have to take the square of the coefficient magnitudes:

```
ln_fft_1 = fftshift(abs(fft2(lena)).^2);
ln_fft_2 = fftshift(abs(fft2(lnlow)).^2);
```

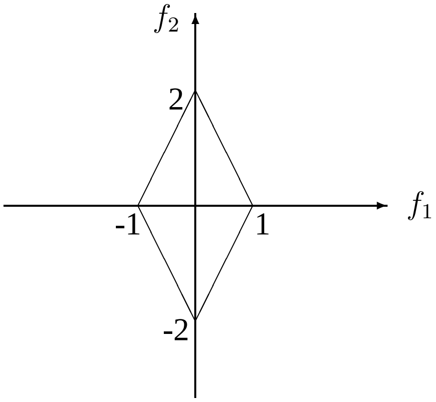
   Because we made the assumption that the underlying continuous image field was sampled at the Nyquist rate to form the original 512×512 discrete image, the sampling distance is 1, and the sampling frequency is therefore 1, which is the critical sampling frequency. Downsampling by 8 corresponds to a sampling distance of 8, hence a sampling frequency of 1/8. The part of the spectrum corresponding to the signal (unaliased component) is therefore the 64x64 square centered around the DC-component. The rest of the spectrum corresponds to the aliasing power. Thus:

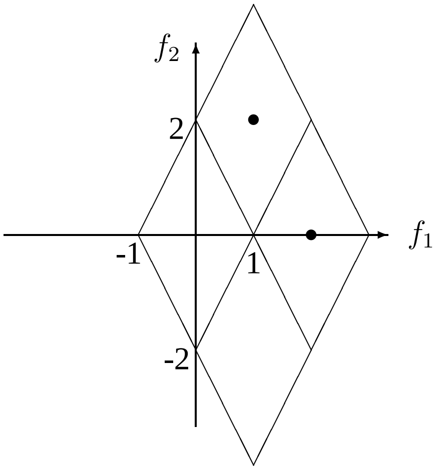
```
sig_1 = ipextract(ln_fft_1,226,226,64,64);
sig_2 = ipextract(ln_fft_2,226,226,64,64);
ps_1 = sum(sig_1(:));
ps_2 = sum(sig_2(:));
(ps_1-ps_2)/ps_1
alias_1 = ipplace(zeros(64),ln_fft_1,226,226);
alias_2 = ipplace(zeros(64),ln_fft_2,226,226);
pa_1 = sum(alias_1(:));
pa_2 = sum(alias_2(:));
(pa_1-pa_2)/pa_1
```

which reveals that the signal power was reduced by about 2% and the aliasing power by 81%.

4. **Non-rectangular Sampling**

   The baseband is confined to the region where $2 \mid f_1 \mid + \mid f_2 \mid < 2$, as shown below on the left, so the replications can be placed as shown below on the right.





The replications are located at lattice points given by

$$B = \begin{bmatrix} 1 & 2 \\ 2 & 0 \end{bmatrix}$$

The sampling points are therefore

$$A = (B^{-1})^T = \begin{bmatrix} 0 & 1/2 \\ 1/2 & -1/4 \end{bmatrix}$$

5. **Scalar Quantization – optimality conditions**

The random variable X with the two-sided exponential pdf

$$f_X(x) = \frac{\lambda}{2} e^{-\lambda|x|}$$

is to be quantized with a three level quantizer q:

$$q(x) = \begin{cases} +b & x > a \\ 0 & -a \leq x \leq +a \\ -b & x < -a \end{cases}$$

(a) The centroid condition stipulates that b should be the centroid of the probability that is mapping into that region. Looking only at positive x:

$$b = \frac{\int_a^\infty x \frac{\lambda}{2} e^{-\lambda x} dx}{\int_a^\infty \frac{\lambda}{2} e^{-\lambda x} dx}$$

After eliminating the $\lambda/2$ on top and bottom, we integrate the numerator by parts to obtain

$$numerator = \frac{ae^{-\lambda a}}{\lambda} + \frac{e^{-\lambda a}}{\lambda^2}$$

and

$$denominator = \frac{e^{-\lambda a}}{\lambda}$$

$$b = a + \frac{1}{\lambda}$$

(b) The other Lloyd condition says that the decision level must be halfway between the reconstruction levels. So

$$a = \frac{b}{2}$$

Therefore, the 3-level quantizer is characterized by

$$a = \frac{1}{\lambda} \qquad \text{and} \qquad b = \frac{2}{\lambda}$$