Take-Home Exam #1

1. 2D systems (10 pts). For the following 2D continuous domain systems, determine if the system is linear and/or shift invariant. If the system is linear, give its impulse response.
   
   a. \[ g(x, y) = S[f(x, y)] = \int_{x=-L}^{x+L} \int_{y=-L}^{y+L} f(x', y')dx'dy' \]
   
   b. \[ g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x', y')h(ax - x', ay - y')dx'dy' \]
   
   c. \[ g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x', y')\exp(-i2\pi(x'x + y'y))dx'dy' \]
   
   d. \[ g(x, y) = |f(x, y) \ast h(x, y)| \]
   
   e. \[ g(x, y) = F_{2D}^{-1}\{F_{2D}\{f(x, y)\}\} \cdot H^2(u, v) \]

2. Sampling (10 pts). Consider an image, \( f \), with a band-limited spectrum that lies completely in the first quadrant of the continuous Fourier domain \( (u, v) \), that is \( F \) is non-zero only for \( 0 \leq u < u_m \) and \( 0 \leq v < v_m \). Now, suppose \( f \) is sampled at two rates: \( f_{s1}(x, y) = f(x, y)\text{comb}(2u_m x, 2v_m y) \) and \( f_{s2}(x, y) = f(x, y)\text{comb}(u_m x, v_m y) \) and the discrete domain images \( f_{d1}(n, m) \) and \( f_{d2}(n, m) \), respectively, are extracted.
   
   a. Sketch the Fourier domain for \( f_{s1} \) and \( f_{s2} \), identifying where spectrum is non-zero.
   
   b. Can the original image be reconstructed from \( f_{d2} \)? If so, describe the relationship between the discrete samples, \( f_{d2}(n, m) \), and \( f(x, y) \).
   
   c. Completely specify \( F_{d2}(\omega_x, \omega_y) \) in terms of \( F_{d1}(\omega_x, \omega_y) \).
3. Filters (40 pts). In this problem you will design 4 different 7x7 zero-phase, high-pass filters using the methods described below and implement them in Matlab. The desired spatial frequency response is 0 for zero spatial frequency and the 1 for high frequencies. For each filter (a.-d.), calculate the 2D DSFT, $H(\omega_x, \omega_y)$, for at least a 64x64 array and display the real part of $H$ using `imagesc`. Include a colorbar and please make sure your axes are labeled and correct.

a. Starting with a 1-D high-pass filter

$$h_1(n) = [-0.141 -1.111 -0.2319 0.7143 -0.2319 -1.111 -0.141]$$

construct a 7x7 separable high-pass filter.

b. Convert $h_1(n)$ into its corresponding low-pass filter equivalent and construct a 7x7 separable low-pass filter, $h_{lp}(n,m)$. Now convert $h_{lp}(n,m)$ into its corresponding high-pass equivalent.

c. Create a 7x7 high pass filter using the 2-D frequency sampling method. Use this definition for the 2D DSFT domain:

$$H(\omega) = \begin{cases} 0 & \omega < \pi / 7 \\ (\omega - \pi / 7)^2 / 2\pi & \pi / 7 \leq \omega \leq 3\pi / 7 \\ \omega = \sqrt{\omega_x^2 + \omega_y^2} & \omega > 3\pi / 7 \end{cases}$$

(Hint: carefully read description of functions `fftshift` and `ifftshift`).

d. Design an “ideal” (jinc-like) 7x7 filter low pass filter for a cut-off frequency $\omega_c = \frac{2\pi}{7}$. You should use some form of circularly symmetric windowing function. Use this filter as the basis of a high-pass filter.

e. Comment on differences that you see in these different designs. Which filters are closed to the ideal spectrum (inverse of the ideal low pass in d.)? Are some methods particularly poor and why? Which methods have the narrowest transition band and which have the largest passband ripple? What symmetries exist? Are some computationally better than others?

4. Convolution (40 pts). We wish to filter with the following 5x5 convolution kernel, $h(n,m) =$

```
0  -1  -1  -1  0
-1  -1  -1  -1  -1
3   5   5   5   3
-1  -1  -1  -1  -1
0  -1  -1  -1  0
```

where the underlined “5” at the center of the matrix is the (0,0) pixel. You will apply this filter to the 58x58 object in the variable $r$srq produced by e1_p1_template.m. Assume that $r$sqr(30,30) (in Matlab) is the center pixel of the input image. For parts a., b., d., and e., do the following: i) Display the filtered image using `imagesc` and include a colorbar, ii) specify the output image size, and iii) specify the Matlab coordinate of the center pixel of the output image.

a. Implement 2D convolution using `conv2`.

b. Implement 2D linear convolution using 2D FFT’s.

c. Convert $r$sqr into a 1D vector: $rld(p) = rld(n + mN) = rsqr(n,m)$ (column-wise stacking in Matlab). Show that a 2D convolution can be implemented by using a 1D convolution of $rld$ with $hld$, a similar 1D stacking of zero-padded version of $h$. What modifications to the above procedure are necessary to insure a linear convolution result? What is the minimum length of $hld$ that yields the linear convolution of part a?

d. Implement 2D linear convolution using the 1D convolution function `conv`, specially please implement the ideas of part c.

e. Implement 2D linear convolution using 1D FFT’s. Here, you will implement the 1D convolution of part d. using FFT’s.