EE257/GP258 **Lab/Homework #7**
Applied Optimization Laboratory Handout #20
Due: Thurs., Feb. 28

Problem 1 – Spectra of interpolated functions.

a) Define a narrow-bandwidth function as follows:

$$f(i,j) = \cos (2 \pi \bullet 8 \bullet i / 1024 + 2 \pi \bullet 16 \bullet j / 1024)$$

for i,j ranging from 1 to 1024. Compute the 2D spectrum of this function, and display the power of each spectral coefficient as an image.

Now, form a subsampled version of this function, denoted fsamp(i,j), by selecting every eighth point in each dimension. Thus fsamp(i,j) has size 128 by 128.

b) Interpolate fsamp(i,j) to the original grid (1024 by 1024 points) using the nearest neighbor method. Compute the 2-D spectrum and display using the same scale you used for part (a). Note the difference between the two spectra.

c) Repeat part (b) but using bilinear interpolation.

d) Repeat again with a cubic spline interpolation. A cubic spline subroutine is available on the homework web page.

e) Finally, use Fourier methods to interpolate with sines and cosines. How does this method compare to the resuts above?

Problem 2 – Image sampling.

Download the file lab7prob2.dat from the web page. This is an image of the Bay Area from an orbital radar system, and is a byte file with size 1200x1180.

Sample this image at 1000 points that lie evenly spaced on a circle of radius 200, centered at location x=500, y=600 in the matrix. Take the first point directly to the right of the circle center, and proceed counter clockwise around the circle.

Compute the sample values at each point using nearest neighbor, linear, and cubic spline interpolators. Plot all three sets of samples as a function of sample number from 1 to 1000.

Problem 3 – Undistorting an image.

Download file lab7prob3.dat from the website. This image consists of 8-bit samples, and is 1536 pixels wide by 1280 lines long. Display the image so you can see what it depicts.

The image has been distorted because the coordinate system is not uniform and linearly spaced. The actual coordinates for each axis are related to the sample point numbers by:

X_actual = sqrt (X) / sqrt(1536) • 1024
Y_actual = sqrt (Y) / sqrt(1280) • 768

Resample the image and produce one that is sampled evenly in x and y. Bilinear interpolation is a good sampling function for this exercise.

Hint: In order to avoid "holes" in the output image, create your program loops over the output pixels rather than the input pixels and interpolate the input data.