ELEN 444 DSP Course Project

Prof. J. Ji

Due Monday, December 6 before class

Project Description: You are working in a manufacturing facility where there are lots of interference and noises. One day there are some VIPs visiting your place and you took a picture using your camera phone. After you downloaded the picture you found that the image are very noisy and contains many stripes due to a sinusoidal interference. You truly like the picture and you'd like to show it to your friends. As a signal processing expert, you decided to process the corrupted image to make it ``acceptable''. Specifically, you will analyze the corruptions (interference and noise) and then design digital filters to suppress them. Following the step-by-step instructions below and using your DSP knowledge learned in this class, design and implement a Matlab program to process the image.

Evaluation: Your projects will be evaluated based on all materials you turned in. In addition, the final results will be displayed and the whole class vote for three Best Image Effect Award projects.

Procedures:

1. Go to the webct, download the image file proj444.jpg. In Matlab, read in the image using "*data = imread('project444.jpg'); [M,N]=size(data);*". Display the image using "*imagesc(data); colormap('gray'); axis equal; axis off;*" and observe the stripe artifacts and noise. *data* is an M by N image (two dimensional signal) that you will process.

2. Pickup the central line of the image by "*x=data(129,:)*". Compute the spectrum of *x* using DFT and plot the magnitude spectrum. Identify the peaks of the spectrum and estimate the index of DFT terms corresponding to the sinusoidal interferences.

3. Reduce the stripes by removing the interference components. You may removing the DFT terms corresponding to the interference frequency and then linearly interpolate them from the neighbors. For example, assume *X(p)* is the interference term, you may set "*X(p)=0.5*X(p-1)+0.5*X(p+1);"*. Perform an IDFT to obtain *y*, the line without interference.

4. Repeat step 2-3 for each line of the image to obtain an image with reduced stripes. Since all lines have the same interference frequency, you don't need to re-estimate *p* each time. You may process the whole image by using a loop in matlab "*for thisline=1:M; x=data(thisline,*:); *y*={processing result of x}; ...; *image1(thisline,:) =real(*y); *end*". Display *image1* as in Step 1.

5. Design a 9-point FIR low-pass filter of the first type (symmetric, odd number) using windowing method to suppress the noise (Since most signal energy is in low frequency band and the noise is uniform across all frequencies.). Specify the characteristics of your filter. Sketch your desirable *Hd(ω)* and stem *h(n)*, your filter coefficients. Compute and plot the frequency response of the designed filter for $\omega = -\pi : stepsize : \pi$ with a step size no large than $\pi/128$.

6. Process your image using convolution, i.e., run "*image2=filter2(image1,h,'same');*" on horizontal direction and then run "*image3=filter2(image2,transpose(h),'same')*;". Display image3 as in step 1. Output your data to a file by "*save LastnameFirstname image1 image2 image3 h*;".

What to turn in: Your Matlab scripts and plots, and email me (jimji@tamu.edu) the yourlastnameFirstname.mat file. Have fun!