EE 301 Lab 10 – Images and Imaging Systems -- K. Nayak

Introduction

In this lab, you will load, create, and manipulate images in Matlab. You can think of images as two-dimensional (2D) signals!

Your lab report should be organized by Exercise #. There is also the opportunity to earn extra credit by completing some or all of the "Optional Exercises". If you have other ideas that you'd like to explore for extra credit, contact Krishna via email for approval.

Exercise #1: Warm-Up

Load and unpack the folder of 256x256 or 512x512 test images into your Matlab working directory. For this exercise we will just focus on grayscale images, but there are a variety of useful test images out there. See http://sipi.usc.edu/database/ for more information.

View some of these images using the following loop.

```
for imnum = 1:48
   fname = sprintf('testimages512/%d.gif',imnum);
   im = imread(fname,'gif');
   imshow(im);
   title(num2str(imnum));
   pause(0.2);
end
```

Optional: Load photos from your own personal library, or from the internet. Use the `imread()` function. http://www.mathworks.com/help/matlab/ref/imread.html Convert the color images into gray scale by averaging or performing a weighted combination of the red, green, and blue channels. Compare this with use of the `rgb2gray()` function. http://www.mathworks.com/help/images/ref/rgb2gray.html

Exercise #2: Fourier Transform

Use the following code to examine the Fourier transform of these images. The following lines plot the log-magnitude and the phase of the

```
ftx = fft2c(im);                    % Compute the Fourier
transform

subplot(131);
imshow(im,[0 256]);                 % Image
title('Image');

subplot(132);
imshow(log(abs(ftx)),[-4 10]);      % Fourier Log-Magnitude
```

```
title('Fourier log magnitude');

subplot(133);
imshow(angle(ftx),[-pi pi]);      % Fourier phase
title('Fourier phase');
```

Examine the Fourier transform for the 48 test images and provide your interpretation. Do you see any common trends among all (or most) images? Do you see any unique features in the Fourier domain, and do they relate to features in the image domain? Include images in your report and superimpose arrows to identify features that you discuss.

Optional: Acquire your own test images to test your inferences made above. Take pictures with your camera/phone/etc. Load them into Matlab (see Optional exercise above). Then apply the same processing. Do you see the expected features in the Fourier transform?

Exercise #3: What's more important Fourier magnitude, or Fourier phase?

Write a Matlab function that combines the Magnitude of the Fourier transform from one image with the Phase of the Fourier transform of another image. This is precisely what was done in Figure 6.2 of the Oppenheim & Willsky textbook, and was demonstrated in class. Use the provided function `ifft2c()` to take the inverse 2D Fourier Transform. You may get complex values and therefore will have to take the `abs()` of the output prior to display.

Submit your code, and a few examples. From this process, can you infer the type of information contained in the magnitude versus phase of the Fourier transform?

Exercise #4: Camera Shake

In conventional photography, it is possible to model motion blur caused by camera motion as a 2D convolution. This is because when the camera shutter is open, each detector is seeing the sum of several spatially shifted versions of the scene. The spatial shifts are the same for all pixels if the motion is caused by camera motion. The following code simulates camera motion, where the motion is described by m.

```
m = [2 1 0 0 0;
     1 2 1 0 0;
     0 1 2 1 0;
     0 0 1 2 1;
     0 0 0 1 2];
m = m/sum(m(:));
im2 = conv2(double(im),m,'same');
subplot(121);
imshow(im,[0 256]);
subplot(122);
imshow(im2,[0 256]);
```

This example considers camera motion that is predominantly down and to the right (or equivalently up and to the left) by 5 pixels. Use a similar framework to simulate other

motions (left-right, up-down) and shifts by larger than 5 pixels. What general trends do you observe? Show examples.

Optional: We know from Chapter 4 that convolution in one domain is equivalent to multiplication in the Fourier domain. Verify this by examining the Fourier transform of `m`, `im`, and `im2` in Matlab. (*Note: this will help you implement camera shake reduction*)

Open Ended Problems

For those wishing to earn more substantial extra credit, consider one or more of the following open-ended exercises:

1. Removing "Camera Shake" Blur. Adobe Photoshop has a feature for removing camera shake blur.
   http://helpx.adobe.com/photoshop/using/reduce-camera-shake-induced-blurring.html
   http://petapixel.com/2013/04/17/a-sneak-peek-of-the-magical-new-shake-reduction-tool-coming-to-photoshop/
   Try to implement this yourself, using what you know about Fourier transforms. Note that you will need to devise and inverse system that "un-does" the blurring.

2. (more to be added)