LAB LESSON TWO

IMAGE FUNDAMENTALS

Dr. Nursuriati Jamil

File Conversion

You can convert from one image type to another in one step, using one of the image conversion functions.

Function	Purpose
`ind2gray`	Indexed image to intensity image
`ind2rgb`	Indexed image to RGB image (MATLAB)
`gray2ind`	Intensity image to indexed image
`rgb2gray`	RGB image or colormap to grayscale
`rgb2ind`	RGB image to indexed image
`mat2gray`	Matrix to intensity image
`im2bw`	Image to binary image by thresholding
`im2double`	Image array to double precision
`im2uint8`	Image array to 8-bit unsigned integers
`im2uint16`	Image array to 16-bit unsigned integers

Sampling

Sampling is the process of digitizing an analog image in a continuous space to a digital image in a discrete space. Digitizing the coordinate values is called *sampling.*

To reduce sampling of an image, (undersampling), you can do the following. Take note of the checkerboard effects.

```
% undersampling
i = imread (‘pout.tif’); imshow(i);
[row,col,dim]=size(i);
s = 4;            %sampling factor
undersampled=i([1:s:row],[1:s:col],[1:1:dim]);
figure, imshow(undersampled);

OR
sub_k = imresize(i, 0.5);
```

Likewise, to add more samples (oversampling), you can do the following.

```
% oversampling
i  imread(‘peppers.png’); imshow(i);
super_k = imresize(i, 6);
figure, imshow(super_k);
```

Quantization

The process of representing the amplitude of the continuous signal at a given coordinate as an integer value with different gray levels is usually referred to as *quantization*. Digitizing the amplitude is known as quantization.

To convert a color image with 24-bit depth to 8-bit depth, use `RGB2IND` function.

```
RGB = imread('peppers.png’);
[X,map] = rgb2ind(RGB, 256);
imwrite(X,map,'peppers.tif');
```

*** Change the bit-depth to 4 bits, describe what happens to the image**.

`IM2UINT8` and the `IM2UINT16` functions are also useful when attempting to convert between different bit depths. The following code sample illustrates the conversion from a 8 bit depth image to a 24 bit depth image.

```
[Xtrees,maptrees] = imread('trees.tif');
XtreesRGB = ind2rgb(Xtrees,maptrees);
XtreesRGB = im2uint8(XtreesRGB);
imwrite(XtreesRGB,'myindexedtrees.tiff')
imfinfo('myindexedtrees.tiff')
```

Pixel Value Information

The Image Processing Toolbox provides several functions that return information about the data values that make up an image. These functions return information about image data in various forms

```
X = imread(‘peppers.png’);
imshow(X);

%% PIXVAL: Display image pixel information
title('pixval')
pixval on
pixval off
hold on

title('impixel')
col = [260, 454];
row = [203, 290];
plot(col,row,'*k','markersize', 10)
pixels = impixel(X,col,row);
```

Cropping

The following commands allows the user to crop the image manually, which is an important process especially if the programmer would like to manipulate the data more in details.

```
img = imread('peppers.png’);
imshow(img);
imgCrop = imcrop(img);
imshow(imgCrop);

I= imread('peppers.png');
imshow(I);
rect = [290 160 220 165];
I2= imcrop(I,rect);
figure
imshow(I2);
```

Rotation

```
I = imread('peppers.png');
J = imrotate(I,35);
figure, imshow(J);
```

Assignment

1. Read and display ‘trees.tif’ into Matlab. Note: This file is an indexed file.
   (2 marks)

2. Mark all the red leaves in the river with a marker.
   (6 marks)

3. Double the size of the image and change the bit depth of the image to 16 bits. Save the image using a new name.
   (7 marks)