Assignment 1

Fundamental Concepts

Objective

The aim of this introductory home work is to introduce you to the basic functions in the Matlab Image Processing Toolbox. By the end of these tasks, you should be able to read images from the disk display them, write them back to the disk and perform conversions between different image classes.

Submission Requirements

Some of the assigned tasks are for practice purposes only while others (marked as 'Exercise' or 'Question') have to be answered in the form of a (printed) report that you need to prepare. Following guidelines will be helpful to you in carrying out the tasks and preparing the report.

Guidelines

- In the exercises, when you are asked to display an image, you have to put the image displayed in your report. You may either save the image as 'jpeg' (File->Save As) and add it to the report or use the 'Print Screen' command on your keyboard to get a snapshot of the displayed image. This point will become clear to you once you actually carry out the assigned tasks.

Tasks

1. Reading an Image

Images can be read into Matlab environment using the function imread.

```
>> Img = imread('path\filename.ext')
```

You may use the following image[1] to test the function:

```
>> Img = imread('rice.png')
```

The following table shows the commonly used file formats supported by Matlab

Image Format	Description	Recognized Extensions
TIFF	Tagged Image File Format	tif, tiff
JPEG	Joint Photographic Experts Group	jpeg, jpg
BMP	Windows Bitmap	bmp
GIF	Graphics Interchange Format	gif
PNG	Portable Network Graphics	png

2. Image Size

Function size can be used to determine the dimensions of an image.

```
>> [M N] = size(Img)
```

OR

```
>> [M N B] = size(Img)
```

Where B gives the number of channels in the image. (3 for colored image and 1 for gray image)

The function whos gives additional information about the array. For example, whos Img gives:

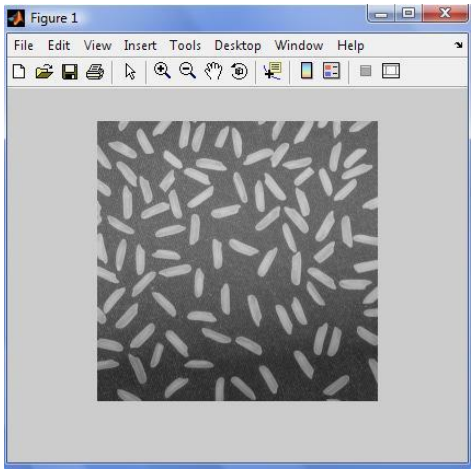
```
Name      Size          Bytes      Class
Img       256x256       65536      uint8
```

3. Displaying an Image

Images can be displayed using the function imshow.

```
>> imshow(Img)
```

[1] A set of images that comes with Matlab can be accessed directly using the 'name.ext' without the need to specify the complete path. That is why we are able to write: imread('rice.png')



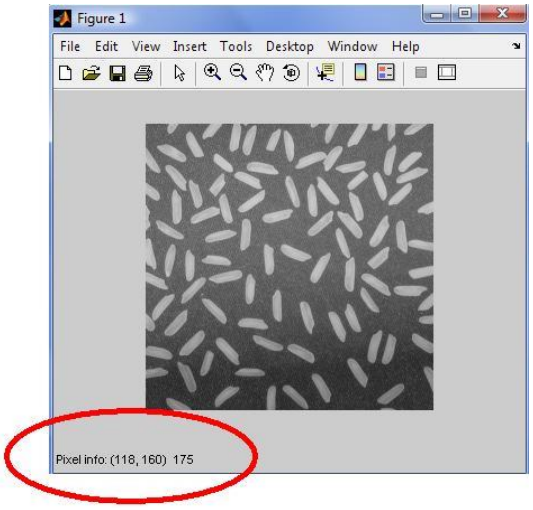
To display another image using imshow while keeping the first image, use:

```
>> imshow(Img), figure, imshow(Img2)
```

To see the pixel values in an image, use:

```
>>Figure, imshow(Img)
>>impixelinfo
```

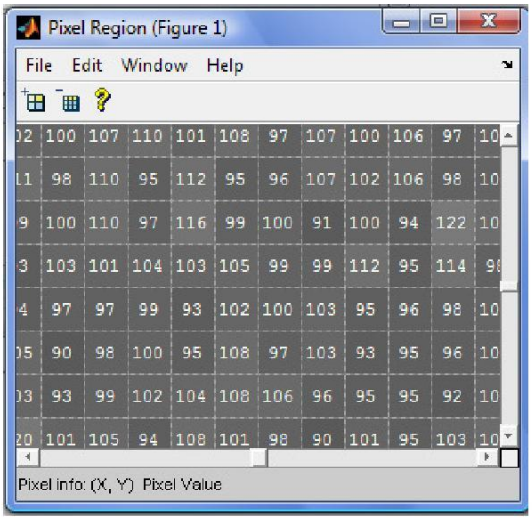
Move the mouse over the image and you will get the pixel coordianates and the respective value at the bottom of the window as indicated in the following figure.



You may also use the command impixelregion to see the pixel values in an image interactively:

```
>> figure, imshow(Img)
```

>> impixelregion



Exercise 1

Load the image ‘cameraman.tif’, find its dimensions and number of channels and display it.

4. Writing Images

Images are written to disk using the function imwrite which has the following basic syntax:

>>imwrite (Img, ‘filename’)

>>imwrite (Img, ‘testImage.tif’)

The imwrite function can have additional parameters depending upon the file format selected. E.g.

>> imwrite (Img, ‘filename.jpg’, ‘quality’, q)

Where q is an integer between 0 and 100.

5. Getting Information about Images

The function imfinfo can be used to get information about a file stored on the disk.

>> imfinfo ‘rice.png’

Filename: 'C:\Program Files\MATLAB\R2007b\toolbox\images\imdemos\rice.png'

FileModDate: '26-janv.-2003 05:03:06'

FileSize: 44607

Format: 'png'

FormatVersion: []

Width: 256

Height: 256

BitDepth: 8

ColorType: 'grayscale'

You may use this function to get the size of the file as:

```
>> K = imfinfo('rice.png')
>> ImageSize = K.FileSize.
```

Exercise 2

Load the image 'cameraman.tif', and save it in 'jpeg' format with a compression factor of 25.

Display the two images (original and compressed) and compare them visually. What do you observe?

6. Using the function 'imfinfo', find the size of files before and after compression and hence compute the compression ratio.

Some

Name	Description
double	Double precision floating point numbers 8 bytes per element
uint8	Unsigned 8 bit integers [0 255]
uint16	Unsigned 16 bit integers [0 65535]
uint32	Unsigned 32 bit integers
int8	Signed 8 bit integers [-128 127]
int16	Signed 16 bit integers [-32768 32767]
int32	Signed 32 bit integers
single	Single precision floating point numbers 4 bytes per element
char	Characters (2 bytes)

logical	Values 0 or 1

Dealing with images, you will mostly encounter uint8 and logical data types.

7. Image Types

The Image Processing Toolbox supports four types of images:

- Intensity images
- Binary images
- RGB images
- Indexed images

Intensity Images

Pixel values represent image intensities.

Class uint8 [0 255]

Class double [0 1]

Binary Images

Binary images are logical arrays of 0s and 1s. An array of type uint8 having values 0 and 1 is NOT considered a binary image. An array can be converted to binary using:

```
>> A =
   5  0  1
   2  0  3

>>B = logical (A)
>>B =
    1   0  1
    1   0  1
```

The function islogical can be used to check if in array is logical or not. The other two types will be discussed at a later stage.

8. Conversion between Data Classes

The general syntax of conversion between data classes is:

>> B = data_class_name (A)

For example:

B= double(A) % Converts the type of A to double

Exercise 3

What happens if you convert a double having values out side the range [0 255] to an uint8?

9. Conversion between Image Types and Classes

The toolbox provides a number of functions to convert between image classes and types. Some of these are:

im2uint8, im2double, im2bw, … etc.

For example consider the following 2x2 image of type double which could result from some intermediate calculations:

f=[-0.5 0.5
0.75 1.5]

Performing the conversion:
>> g = im2uint8(f) yields:

g = [0 128
255] 191

So this function sets:

All values less than 0 to 0.

All values greater than 1 to 255.

And multiplies all other values by 255.

The conversion of an arbitrary array of class double to an array of class double scaled to the range [0 1] can be done via function mat2gray.

```
>> g= mat2gray(A)
```

The output values are in the range 0 (black) to 1 (white).

Finally, for conversion between an intensity image and a binary image, the function im2bw can be used:

```
>> g=im2bw(f,T)
```

Produces a binary image g from the intensity image f by thresholding (will be covered in detail in the lectures). The output image g has a value 0 for all pixels in the input image with a value less than T, and 1 for all other pixels. The value of T has to be in the range [0 1] regardless of the type of input image f. (If the input image is of type uint8, im2bw will first divide all pixels by 255 and then apply the threshold).

Exercise 4

Load the image 'rice.png', binarize it using the function 'im2bw' with a threshold of 0.5 and display both original and binarized images.

Other than these, you have to perform all following tasks as well.

a) Read a colored image using "imread command" as:

   f=imread('flower.jpg');

   Display the image using "imshow" command.

b) Convert the image to gray scale using "rgb2gray" command. Display the image.

c) Convert this image matrix in part (b) to data type uint8 like

   f=uint8(f)

d) Create a new image by multiplying the above image "f" with a constant of 0.33.

Display the new image.

e) Now write the created image in part (d) using “imwrite” command as”

imwrite(g, ‘newflower.jpg’)

f) Now add the images in part (c) and part (d) using “imadd” command:

Z=imadd(f,g)

Display the image.

g) Now subtract the images in part (c) and part (d) using “imsubtract” command. Display the image.

h) Flip the image in part (b) vertically. Display the vertical image.

i) Flip the image in part (b) horizontally. Display the image.

j) Take the image negative of the image in part (b) using “imcomplement” command. Display the image.

Use “figure” to show each of the figures separately. Use “title” to show the title on each figure. Your solution to this assignment should include figures and Matlab code for each part along with any necessary explanations to questions.

Image Fundamentals

a) Read a colored image. Convert the image to gray scale. Display the image.

b) Displays only lower half portion of an input image.

c) Study “imrotate” command and rotate the original image by 45 degrees using bilinear interpolation.

d) Study the command of “imadjust”. Now adjust the gray levels of the image. E.g.

g=imadjust(f,[0.5 1],[0.9 0]); %% gray value 0.5 is mapped to 0.9 and gray value 1 is mapped to 0 in the above statement.

e) Study false contouring in the image read in part a). For this use “imshow” command.

imshow(f, 256); %% shows the 8 bit image…2^8

Display 7 bit, 6 bit, 5 bit, 4 bit, 3 bit, 2 bit and 1 bit images. In which image you observe a greater false contouring.

f) Decimate the image in part a) by 2, 4, 8, 16 and 32.

im_de2=f(1:2:row,1:2:col); %% for Decimation by 2

What would be the effect on image if you decimate it by 32?

g) Use histogram equalization to enhance the image given in part a). Use "histeq" command for creating contrast image. Display the contrast image.

h) Display the histogram of the contrast image using "imhist" command.

imhist(j,255); %% for 8 bit image.

i) Study power law transformation. Read a grayscale darker image. Now take a value of gamma less than 1 to convert the image to lighter tone. Display the image. Now read a grayscale washout image. Take a value of gamma greater than to convert the image to darker tone. Display the image.

Use "figure" to show each of the figures separately. Use "title" to show the title on each figure. Your solution to this assignment should include figures and Matlab code for each part along with any necessary explanations to questions.

Spatial Image Enhancement

a) Read a colored image. Display the image.

b) Study the command of "fspecial" and "imfilter". Use fspecial to generate a given filter mask than use imfilter to filter the image with a given filter mask.

c) Enhance the image with 3x3 and 5x5 averaging filters. Which one gives better results and why?

d) Enhance the image with 3x3 Gaussian filter with sigma 0.8.

e) Show edge detection of image by using 3x3 laplacian filter of alpha 0.5.

f) Enhance the image with 3x3 prewitt filter.

g) Enhance the image with 3x3 vertical and horizontal sobel masks.

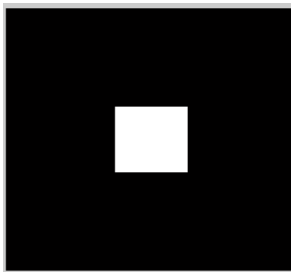
h) Enhance the image with 3x3 unsharp mask using alpha 0.5.

Use “figure” to show each of the figures separately. Use “title” to show the title on each figure. Your solution to this assignment should include figures and Matlab code for each part along with any necessary explanations to questions.

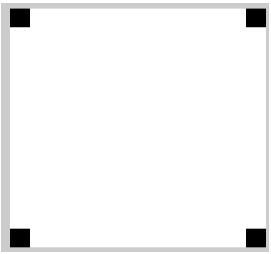
Basic Image Processing Operations

Black and White Images

a) Create an image matrix of zeros of 200x200. Now create a white square at the center of this image from rows 76 to 125 and columns 76 to 125. You can use the function zeros and ones. The output image should be like this.



b) Now create an image like this by creating an image matrix of ones of 256x256. Square size is 20x20 pixels.



Colored Images

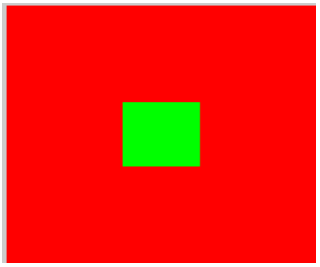
c) Generate a colored image using Matlab code similar to the one shown in figure below. Use following specifications.

a. Image size = 200 x 200 pixels

b. Image type = rgb

c. Square size = 76 row to 125 columns

Sample image:



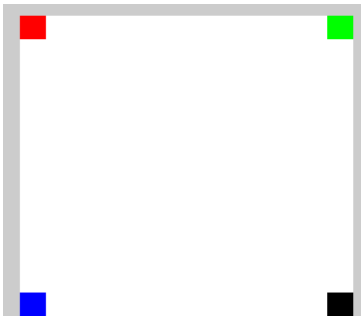
d) Generate a colored image using Matlab code similar to the one shown in figure below. Use following specifications.

a. Image size = 256 x 256 pixels

b. Image type = rgb

c. Square size = 20 x 20 pixels

Sample image:



Converting the image to Binary

a) Read a grayscale image and display it.

b) Convert this image to binary by using inbuilt functions of graythresh and im2bw. Graythresh determines the level of an image and im2bw converts the image to binary by using that level. Display the image.

c) Now convert the image in part a) to binary using for loops. Define the level of the image. Run the loops up to rows and columns of that image. And if the pixel is at any point greater than the defined level than that pixel should be one otherwise zero. Display the image also.

Use “figure” to show each of the figures separately. Use “title” to show the title on each figure. Your solution to this assignment should include figures and Matlab code for each part along with any necessary explanations to questions.