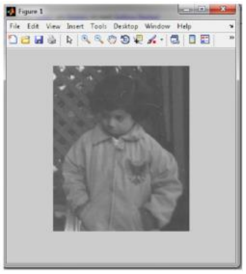
407 Comp Lab 1

The goals of this lab is:

- How to read, show, access pixels, save an image in MATLAB.
- Treat image as a matrix after reading it
- use the MATLAB help if you want more information on any function: >>help imread
  or use the help documentation on the main MATYLAB interface
- There are known and stored images for MATLAB image such as ‘pout.tif’



MATLAB Basics

Matrices in MATLAB

- Matrix is a main MATLAB’s data type.
- MATLAB workspace
- How to build a matrixexample: >> A=[1 2 3; 4 5 6; 7 8 9]; Creates matrix with size 3x3.
- Special matrices :zeros(n,m), ones(n,m),eye (n,m)
- who, whos - current variables in workspace
- save - save workspace variables to a *.mat file
- load - load variables from .mat file
- clear all - clear workspace variables

Basic Operations on Matrices

- All the operators in MATLAB defined on matrices : +, -, *, /, ^, sqrt, sin, cos, etc.
- Element wise operators defined with preceding dot : .*, ./, .^ .
- [m, n] = size(A) - size of a matrix A where m is no. of rows and n is no. of columns
- sum(A) - columns sums of A gives one row vector
- sum(sum(A)) - all the elements sum of A gives one value

relational and Logical operations

- == , < , > , (not equal) ~= ,
- and(&), or(||), (not) ~
- find('condition') - Returns indexes of A's elements that satisfies the condition.

Flow Control

- MATLAB has five flow control constructs:

– if statements

– switch statements

– for loops

–while loops

– break statements.

Scripts and Functions

There are two kinds of M-files:

– Scripts, which do not accept input arguments or return output arguments. They operate on data in the workspace.

–Functions, which can accept input arguments and return output arguments. Internal variables are local to the function.

Figures

- plot(x,y) - plot x data against y data
- figure , figure(k) - open a new figure
- hold on, hold off – refreshing to add many plots on one figure

Reading the image Pout.tif

```
I = imread('Pout.tif');
```

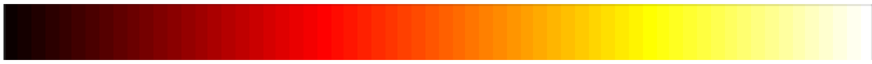
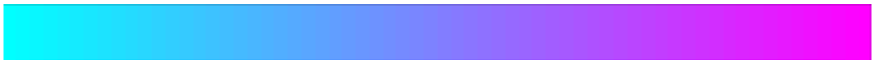
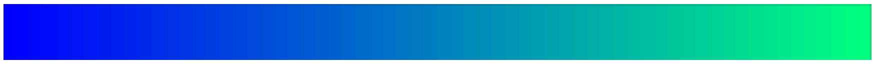
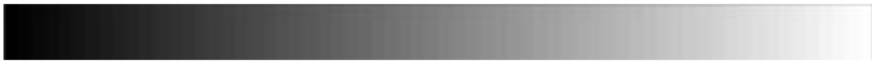
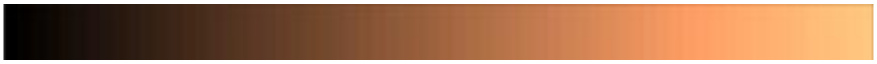
- The image is now stored in the 2D matrix I as intensity values if image is gray level and in 3D matrix if I is colored image.
- Since an image is stored as a matrix in MATLAB, all matrix operations available can be used on the image (e.g., addition, subtraction etc).
- *Investigate the image I in the workspace, its size, pixel values at specific (x,y)*

Image Show methods and Saving Images

1-imshow shows an image

2-imagesc works just as image but rescales the gray levels to use a specific colormap. To see the colors of the color map you can use the command colorbar after you have displayed the image, or you can choose Insert →Colorbar in the menu of the figure window.

Color maps are collection of selected colors having one name. Examples of some colormaps in MATLAB:

Color map	**Color Scale**
hot	
cool	
winter	
gray	
copper	

Examples:

```
I = imread('Pout.tif'); % read the image
```

```
imshow(I); % show image I using ’imshow’ in the (only one) default figure
figure % open a new figure window in order not to overwrite the above figure
imagesc(I) % show matrix I using ’imagesc’
colormap(hot)
colorbar % add a colorbar
```

3-imtool(I)

Opens the image I in Image Tool,

- To measure pixel values in the image you can move the pointer over the pixel and check Pixel info: (x, y) f, where f is the gray level at the position (x, y) in the picture.
- You can also choose Tools →Pixel Region from menu (or by pressing the Inspect pixel values button in the toolbar). This will open the Pixel Region window and show a cross hair marker in the image. Check the gray level value in the pixel (1, 1) located and in command window type I(1,1). Did you get the same value?

In general, *imshow* if you just want to view an image, *imtool* if you want to examine an image thoroughly, and *imagesc* if you want to display a matrix that is not necessarily an image.

4-imwrite command, To save an image in the current directory or specify your directory

```
imwrite(I,’Pout.tif’) % write image I to file ’Pout.tif’
```

use the MATLAB help on how to know how to save images to different file formats.

You can save the contents of a figure by using File →Save As..., and choose a suitable file format and filename, to save the contents of the figure.

<u>Contrast Enhancement</u>

- Read ill-contrast image(‘LabImage1Ori.png’)
- using imtool and display the gray level histograms by selecting Tools →Adjust Contrast in the menu or by pressing the Adjust contrast button in the toolbar.
- The image histogram will appear in the new Adjust Contrast window with a pink interval defining the gray level interval of the image file that is displayed on the screen then you can change contrast by moving the red handles and changing the length of the pink interval in the histogram.
- Another way to display an image histogram is to use the imhist and hist commands.

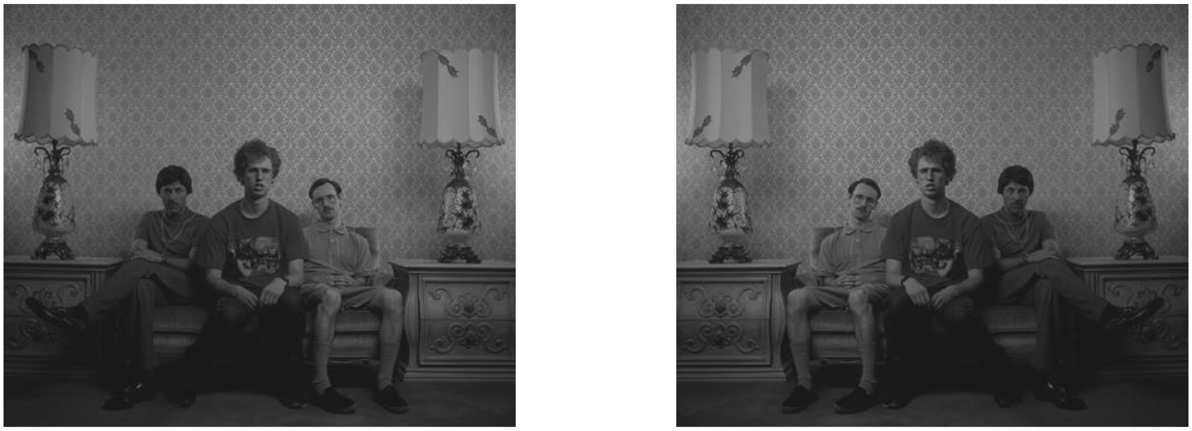
  imhist(I) % show the histogram of I
- Note: The image histogram is defined as : the x axis is the rang of gray levels in the image(ex from 0 to 255 for a gray image), the y-axis is the number of pixels that have that gray value

Problem 1

Write a MATLAB code that reads a gray scale image and generates the reversed

image of original

image. Your output should looks like:



```
a=imread('Labimage1Ori.png');
[r,c]=size(a);
for i=1:1:r
    k=1;
    for j=c:-1:1
        result(i,k)=a(i,j);
        k=k+1;
    end
end
subplot(1,2,1),imshow(a)
subplot(1,2,2),imshow(result)
```

Problem 2

Write a MATLAB code that will do the following

1. Read and display a gray scale image.

2. Display the image such that the pixels having intensity values below than 30 will display as black and pixels having intensity values above than 100 will display as white and the pixels in between displayed as it is.

Problem 3

Write a MATLAB code that

- reads a gray scale image
- generates a flipped image of original image. Your output should be like the one given below