**Computer Engineering Department**

Name-Surname Student No. .

**Faculty of Engineering Mahidol University**

# LAB01: Introduction to Image Processing Using MATLAB

## Objectives

Upon completion of this lab, you will be able to:

* Use MATLAB to read and display images in a variety of image file types.
* Describe the different types of images.
* Understand the basics of MATLAB programming.
* Write a user-defined function in MATLAB that converts the grayscale image to the binary image based on threshold.

**Introduction**

In MATLAB, an image is stored as a two-dimensional matrix, where the matrix contains the values of intensity of light of the pixels.

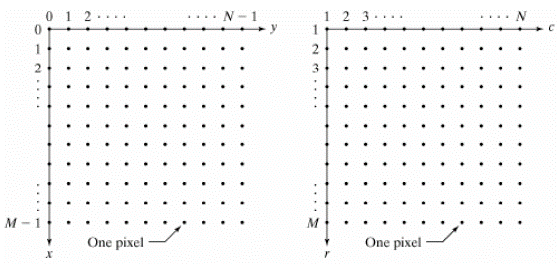


Figure 1: [Left] the usual image coordinates. [Right] the matrix coordinates of MATLAB.

Notice that the matrix coordinates in MATLAB originate at (r,c) = (1,1) instead of (x,y) = (0,0).

Four basic types of images are supported in MATLAB, including:

1. Binary image
2. Grayscale image
3. Indexed image
4. RGB image

The Image Processing Toolbox provides a set of basic tools for image processing that typically includes the following:

* **Converting Image Classes and Types**

im2bw - Convert image to binary image by thresholding.

im2double - Convert image array to double precision.

im2uint8 - Convert image array to 8-bit unsigned integers.

im2uint16 - Convert image array to 16-bit unsigned integers.

ind2gray - Convert indexed image to grayscale image.

gray2ind - Convert grayscale image to indexed image.

ind2rgb - Convert indexed image to RGB image.

rgb2gray - Convert RGB image or colormap to grayscale.

rgb2ind - Convert RGB image to indexed image.

* **Displaying Images**

image - Create and display image object.

imagesc - Scale data and display as image.

imshow - Display image.

colormap - Display color map.

* **Importing and exporting Images**

load - loads the variables from a MAT-file.

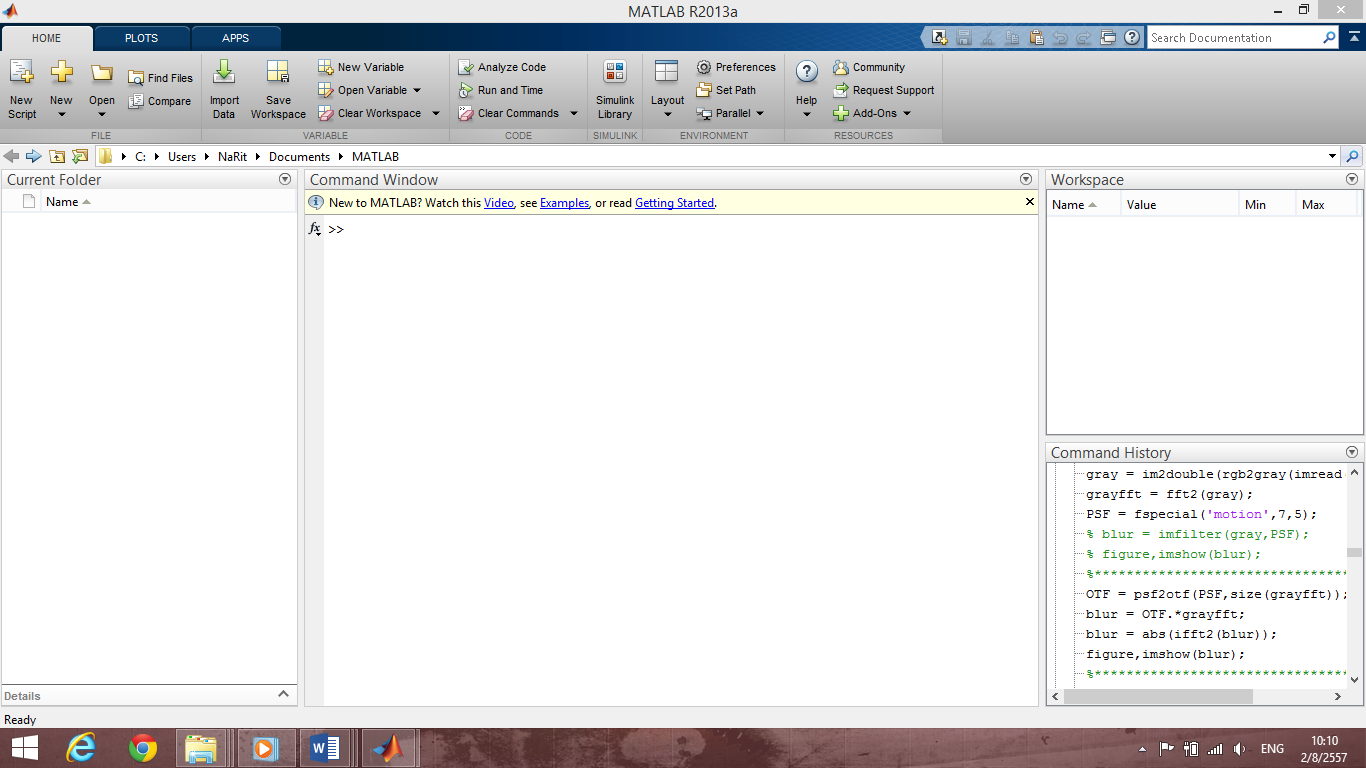
imread - Read image file.

imwrite - Write image file.

imfinfo - Return information about image file.

figure - Creates a figure on the screen.

* MATLAB Program

Figure 2: The MATLAB with the default layout.

**Current working Folder**

**Command window**

**Workspace**

**Listed variable**

## Exercise

This exercise can be used as a reference for basic operations in the Image Processing Toolbox. To find out what commands are available in the Image Processing Toolbox, type "help images" or type “help command-name” will give you detailed information on what a function does and how to use it.

You can use the images provided in the folder **LAB1** for your exercises. If you do not complete this exercise in one session, the variables you created will be erased when you exit MATLAB.

1. Use the **load** command, MATLAB built-in function, to import data into the MATLAB workspace from a file **ip01\_1.mat**. Record the command you used in the box below.

|  |
| --- |
| Load(‘ip01\_1.mat’) |

1. To list the currently active variables, use the **whos** command. Record your results in the box below.

|  |
| --- |
| X 258x350 722400 double  caption 1x66 132 char  map 128x3 |

What is the meaning of each variable?

|  |
| --- |
| X the loaded image value in double type ranging from 1-128  Caption – text description of the loaded image . variable type-> char. Containing 66 characters  Map is the index table containing corresponding values … |

1. Now consider the X variable in Exercise 2. Convert this X variable into binary image, grayscale image, and RGB image. Three images are stored in the X2, X3 and X4 variables, respectively. What command would you use to convert all images?

|  |
| --- |
| X2 = im2bw(X, map);  X3 = ind2grey(X, map);  X4 = im2rgb(X, map); |

What is the image type for X variable?

|  |
| --- |
| Double |

1. Display the X, X2, X3, and X4 images using the **imshow** command and save all your results.

|  |
| --- |
|  |
|  |

1. Record the value of the pixel at coordinates (2,4) of the X, X2, X3, and X4 images.

|  |
| --- |
| 106  1  0.7025  0.5490  respectively |

1. Use the **imread** command to read the following images “ip01\_2.tif”, “ip01\_3.tif”, and “ip01\_4.tif”. Keep these images in the I1, I2, and I3 variables, respectively. Display and save all your results.

|  |
| --- |
| Imshow(I1)    Imshow(I2)    Imshow(I3) |

1. Convert the I1 RGB image into indexed image with 256 colors, and indexed image with 16 colors. Keep these new images in the I4, and I5 variables, respectively. Display and save all your results.

|  |
| --- |
| Imshow(I4)    Imshow(I5) |

1. Make the diplay window of I2 image active and then use the **colormap**(jet) command. Observe output and record your result.

|  |
| --- |
|  |

1. Using the **colormap** command, convert the color map of the I2 image from jet to gray. After that save the resulting image.

|  |
| --- |
|  |

## When you have completed to convert the color map of the I2 image back to gray, you can use flipud command to flip the color map up to down. Record your result.

|  |
| --- |
|  |

## The im2bw command converts the grayscale image to the binary image based on threshold. Thresholding means that a output image is generated in which each pixel has intensity 1 (white) if the corresponding pixel in the input image has a value above the threshold and 0 (black) otherwise.

In order to create your own function in MATLAB, write a program for converting the grayscale image to the binary image, with the following function name: **Myim2bw.m**. Using this program on the image “cameraman.tif” should give you result as shown in Figure 3.

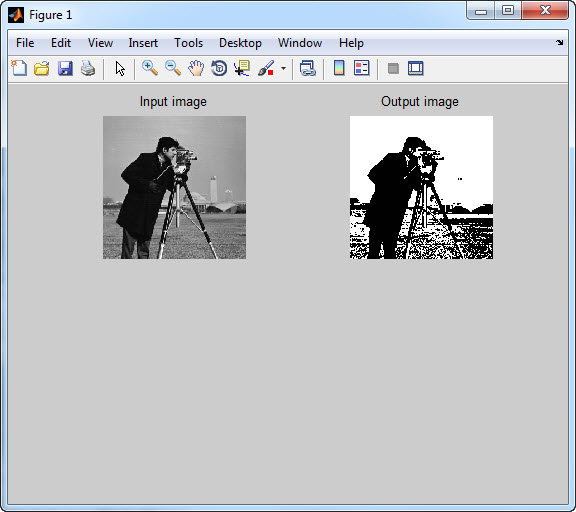


Figure 3: The result of converting the grayscale image

into the binary image using the threshold of 0.5.

## (Extra credit) Write a MATLAB program that makes the reverse image readable, with the following name: Myim2rv.m. For example, the reverse image of the word "ambulance" is used to caution the driver of the vehicle ahead of ambulance. Using this program on the image “ambulance.bmp” should give you result as shown in Figure 4.

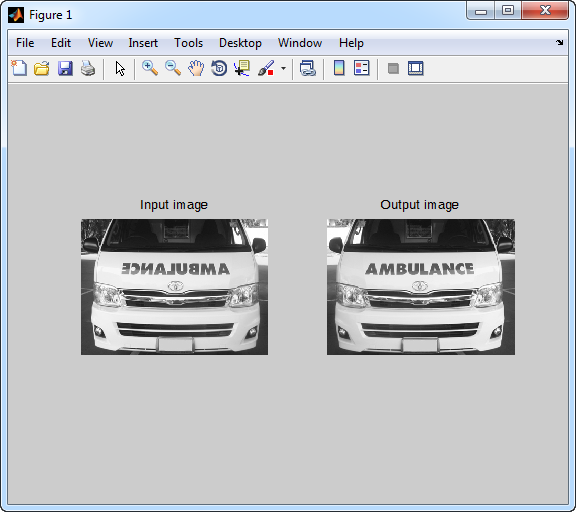


Figure 4: The result of making the reverse image readable.

## Question

## Describe the characteristics of the image types, including the binary image, the grayscale image, the indexed image, and the RGB image.

## What is the range of pixel values of each image?

## How many colors are there in each image?

## What is the size of each image?

* 1. Binary image

|  |
| --- |
| 0-1  Black & White  m\*n\*1 |

* 1. Grayscale image

|  |
| --- |
| 0-255  128 Black & 128 White  m\*n\*2 |

* 1. Indexed image

|  |
| --- |
| 0-255  256 colors. Color received from color table.  m\*n\*2 |

* 1. RGB image

|  |
| --- |
| 0-255  256 colors (combination of RED Green Blue)  m\*n\*3 |

1. What are the advantages and limitations of indexed image?

|  |
| --- |
| Advantage:  Indexed color saves a lot of memory, storage space, and transmission time: using truecolor. Limiting the image colors to 256, every pixel needs only 8 bits, or 1 byte each, so the example image now needs only 640×480×1 = 307,200 bytes (300 KiB), plus 256×3 = 768 additional bytes to store the palette map in itself (assuming RGB), approx. one third of the original size. Smaller palettes (4-bit 16 colors, 2-bit 4 colors) can pack the pixels even more (to 1/6 or 1/12), obviously at cost of color accuracy.  Disadvantage:  The main disadvantage of using indexed color is the limited set of simultaneous colors per image. Small 4- or 16-color palettes are still acceptable for little images (icons) or very simple graphics, but to reproduce real life images they become nearly useless. Some techniques, such as color quantization, anti-aliasing and dithering combined together can create indexed 256-color images comparable to the original up to an acceptable level. Also, if the original color palette for a given indexed image is lost, it can be nearly impossible to restore it. Here is an example of what happens when an indexed color image (the previous parrot) has been associated with an incorrect color palette: |

1. What is the differences between **image** and **imagesc** commands?

|  |
| --- |
| imagesc 🡪 the data is scaled to use the full colormap.  imagesc 🡪 can specify the scaling. Ex: CLIM = [CLOW CHIGH] |

**What you need to submit:**

Prepare a zip file that contains the following:

1. A report containing the text and image in the form of a word document.

2. The MATLAB program for implementing part in the form of a matlab file (m-file extension).

Double check that you have included all of the above in the zip file. Email the zip file to the account **send2narit@hotmail.com** with the following subject line: **EGCI486\_LABxx\_yyy**, which xx is a number of LAB and yyy is the last 3 digits of the student identification number. Your email should reach us before 11:59 PM on the due date.

\*\***Please preserve a copy of your email and all your work until the end of the course.**