Introduction to MATLAB Image Processing

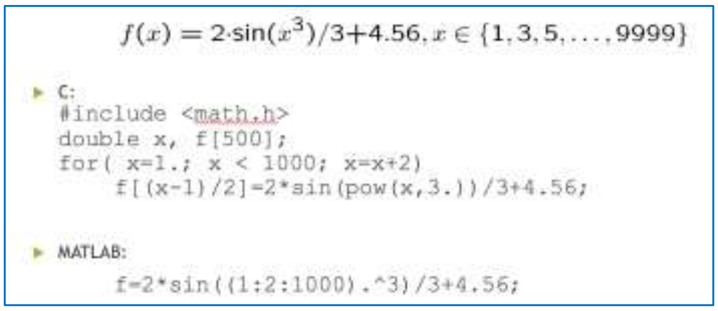
The purpose of this tutorial is to gain familiarity with MATLAB's Image Processing Toolbox. This tutorial does not contain all of the functions available in MATLAB. It is very useful to go to Help\MATLAB Help in the MATLAB window if you have any questions not answered by this tutorial. The help tool is especially useful in image processing applications, since there are numerous filter examples.

1. Prior Knowledge

What is MATLAB? It is called Matrix Laboratory—a high-level language for matrix calculations, numerical analysis, and scientific computing. As for programming, MATLAB

- ➢ Can type on command line, or use a program file ("m"-file)
- ➢ Semicolon at end of line is optional (suppresses printing)
- ➢ Control flow (if, for, while, switch, etc) similar to C and java
- ➢ Differences from C: no variable declarations, no pointers

So why MATLAB? Because we focus on our ideas only and we just write a short piece of code but can get abundant results:



$$f(x) = 2\cdot\sin(x^3)/3+4.56, x \in \{1,3,5,\ldots,9999\}$$

- C:

```
#include <math.h>
double x, f[500];
for( x=1.; x < 1000; x=x+2)
    f[(x-1)/2]=2*sin(pow(x,3.))/3+4.56;
```

- MATLAB:

```
    f=2*sin((1:2:1000).^3)/3+4.56;
```

What is the Image Processing Toolbox? The Image Processing Toolbox is a collection of functions that extend the capabilities of MATLAB's numeric computing environment. This toolbox supports a wide range of image processing operations, including:

- ✓ Geometric operations
- ✓ Neighbourhood and block operations
- ✓ Linear filtering and filter design
- ✓ Transformations
- ✓ Image analysis and enhancement
- ✓ Morphological image operations
- ✓ And many more….

2. Images in MATLAB

MATLAB can import/export several image formats:

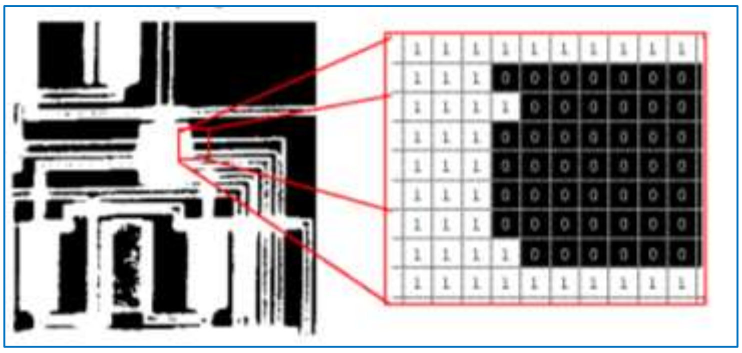
- BMP (Microsoft Windows Bitmap)
- GIF (Graphics Interchange Files)
- HDF (Hierarchical Data Format)
- JPEG (Joint Photographic Experts Group)
- PCX (Paintbrush)
- PNG (Portable Network Graphics)
- TIFF (Tagged Image File Format)
- XWD (X Window Dump)
- raw-data and other types of image data

Data types in MATLAB:

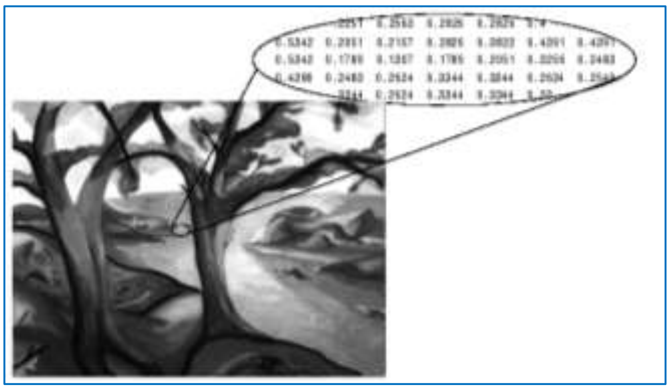
- Double (64-bit double-precision floating point)
- Single (32-bit single-precision floating point)

- Int32 (32-bit signed integer)
- Int16 (16-bit signed integer)
- Int8 (8-bit signed integer)
- Uint32 (32-bit unsigned integer)
- Uint16 (16-bit unsigned integer)
- Uint8 (8-bit unsigned integer)

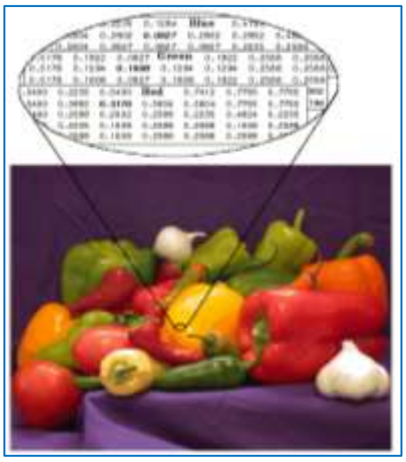
Binary images: {0,1}



Intensity images:[0,1] for double and [0,255] for uint8 etc.

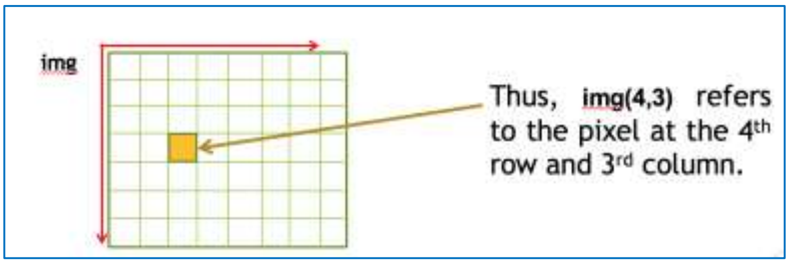


RGB images : m × n × 3



MATLAB Image Coordinates

- MATLAB stores images as matrices.
- In MATLAB, image pixels are referenced using (row, col) values.
- Origin of the coordinate system (1,1) is the top left corner of the image.



Thus, img(4,3) refers to the pixel at the 4th row and 3rd column.

Image Matrix

By Default, MATLAB reads an image in uint8 (unsigned 8-bit integer) format. And each pixel has values in the range [0,255].

For some function you may need to convert image to double format. Double format has pixel values in the range [0,1]. To convert any image format to double, use MATLAB function “im2double()”

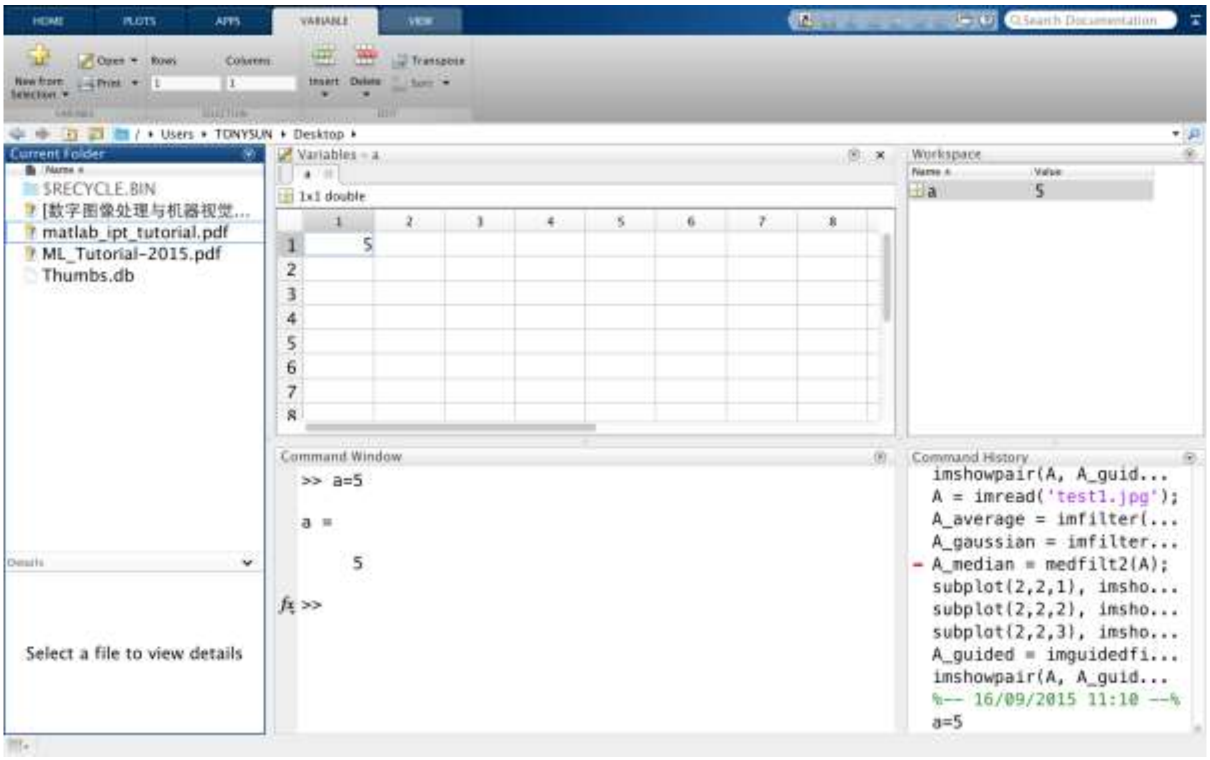
e.g.: A = imread(‘filename.extension’);

A_double = im2double(A);

3. Get Started

The first thing you need to do it so open MATLAB installed on your lab pcs. Access the Start Menu, proceed to Programs, Select MATLAB x.x folder and select Matlab.exe. You can find more helpful tips by checking official MATLAB document to warm up: https://www.mathworks.in/help/pdf_doc/matlab/getstart.pdf

When MATLAB opens, the screen should look something like what is pictured below (ignore the Matlab version):



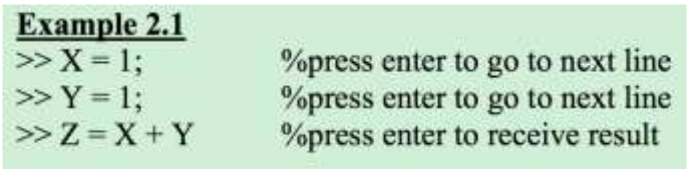
```
>> a=5

a =

     5

fx >>
```

```
imshowpair(A, A_guid...
A = imread('test1.jpg');
A_average = imfilter(...
A_gaussian = imfilter...
A_median = medfilt2(A);
subplot(2,2,1), imsho...
subplot(2,2,2), imsho...
subplot(2,2,3), imsho...
A_guided = imguidedfi...
imshowpair(A, A_guid...
%-- 16/09/2015 11:10 --%
a=5
```

- The **Command Window** is the window on the lower centre of the screen. This window is used to both enter commands for MATLAB to execute, and to view the results of these commands.
- The **Command History** window, in the lower right side of the screen, displays the commands that have been recently entered into the Command Window.

- In the upper left hand side of the screen there is **Current Folder**, which shows files in current folder. You can change the current fold by simply select your destination in the address bar or directly type your address in it. The windows below Current Folder is the **Details** window, you can view file details if you have a file selected.
- In the upper fight hand side of the screen is the **Workspace** window which contains the variables you have defined. The contents of these variable could be seen in the **Variables** window, which is located in the top centre of the screen.

In order to gain some familiarity with the Command Window, try Example 2.1, below. You must type code after the >> prompt and press return to receive a new prompt. If you write code that you do not want to reappear in the MATLAB Command Window, you must place a semi colon after the line of code. If there is no semi colon, then the code will print in the command window just under where you typed it.



Example 2.1

```
>> X = 1;           %press enter to go to next line
>> Y = 1;           %press enter to go to next line
>> Z = X + Y        %press enter to receive result
```

As you probably noticed, MATLAB gave an answer of Z = 2 under the last line of typed code. If there had been a semi colon after the last statement, the answer would not have been printed. Also, notice how the variables you used are listed in the Workspace Window and the commands you entered are listed in the Command History window. If you want to retype a command, an easy way to do this is to press the ↑ or ↓ arrows until you reach the command you want to reenter. Please check another PDF document named "MATLAB Basics" to learn more about MATLAB foundation.

4. Basic Image Manipulation

Once you get familiar with MATLAB, we can start to do some basic image manipulation using the built-in MATALB image processing toolbox.

4.1 Read an Image

MATLAB can read images as color or grayscale. Color images are generally 3-channel images with separate channels for each color, for e.g., RGB image has individual color channels for R, G & B.

```
A = imread('filename.extension');
```

The return value A is an array containing the image data. If the file contains a grayscale image, A is an M-by-N array. If the file contains a truecolor image, A is an M-by-N-by-3 array. M and N are height (no. of rows) and width (no. of columns) respectively of the read image.

File extension is the extension associated with a file type. For e.g., for JPEG files it is .jpg, for BMP files it is .bmp, etc. Now if you want to display the read image, the command would be:

```
figure, imshow(A);
```

Getting to know image height, width and number of channels.

```
height = size(A, 1);
```

```
width = size(A, 2);
```

```
number_of_channels = size(A, 3) ;
```

3 for RGB images and 1 for grayscale images.

4.2 Write an Image

MATLAB can write single channel (Grayscale) as well as multi-channel (Color) images. Let us first read an image:

```
A = imread('filename.extension');
```

As we can recall from previous lecture, the return value A is an array containing the image data. If the file contains a grayscale image, A is an M-by-N array. If the file contains a true color image, A is an M-by-N-by-3 array. M and N are height (no. of rows) and width (no. of columns) respectively of the read image.

File extension is the extension associated with a file type. For e.g., for JPEG files it is .jpg, for BMP files it is .bmp, etc. MATLAB supports writing images in a number of formats such as JPEG, BMP, TIFF, PNG, etc.

Now let us say you read a color image and want to save it as grayscale image, this process can be performed as:

```
A_gray = rgb2gray(A);

imwrite(A_gray, 'filename.extension');
```

Now you can open the file from your disk to see the original file and written file.