Mahidol University *Wisdom of the Land*

Chapter 1

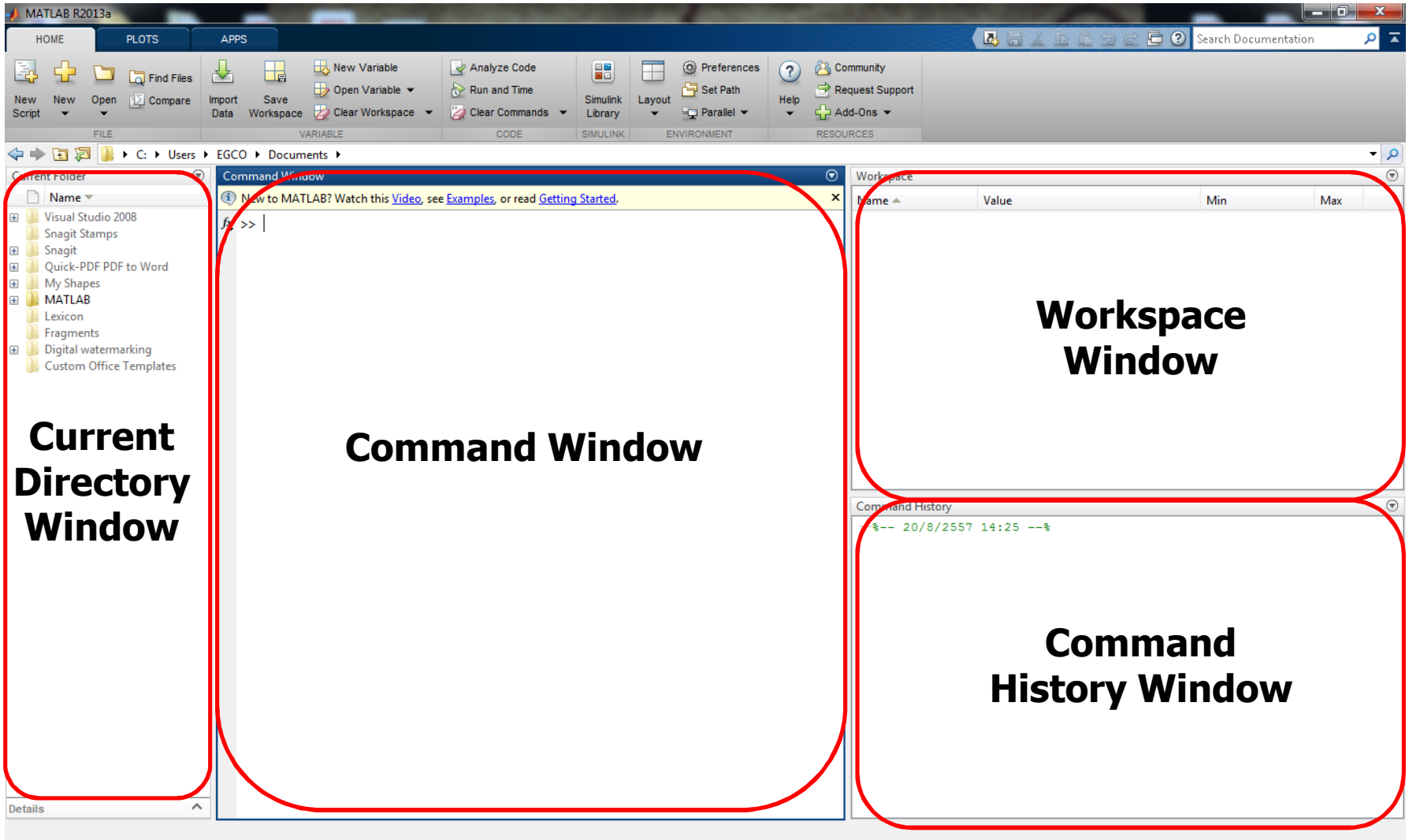
Image processing using MATLAB

Topics

This lecture will cover:

- MATLAB Basics
- Category of digital image
- Image types
- Importing and exporting images in MATLAB
- Converting between image formats

MATLAB Windows



Variable Name

- Variable naming rules:
  - must be unique in the first 63 characters
  - must begin with a letter
  - may not contain blank spaces or other types of punctuation
  - may contain any combination of letters, digits, and underscores
  - should not use MATLAB keyword

Basic Operators

Operation	Symbol
Addition	+
Subtraction	-
Multiplication	*
Division	/
Exponentiation	^

Matrix and Array Operations

- Matrix operations (linear algebra).

- Array arithmetic operations (element by element) via dot (".") operator.

- Matrix multiplication

```
>> A=[1 2;3 4], B=[2 5;-1 3]
     A = 1    2
         3    4
     B = 2    5
        -1    3
>> A*B
   ans = 0    11
         2    27
```

- Array Multiplication

```
>> A,B
     A = 1    2
         3    4
     B = 2    5
        -1    3
>> A.*B
   ans =  2   10
         -3   12
```

- Note that the array multiplication is performed by multiplying the corresponding entries of each matrix.

Boolean Operators

Operator	Meaning	Example
==	equal	x==3
~=	not equal	x~=y
>	grater	x>2
<	less	x<5
>=	grater than or equal	x>=y
<=	less than or equal	x<=y

Logical Operators

Operand		Not	AND	OR
A	B	~A	A&B	A|B
1	1	0	1	1
1	0	0	0	1
0	1	1	0	1
0	0	1	0	0

Logical Operators

- find('condition')
  - Returns indexes of the array elements that satisfy the condition

- Example:

```
» A=[7 3 5; 6 2 1], IndexA=find(A<4)
   A=
        7 3 5
        6 2 1
  IndexA=
              3
              4
              6
```

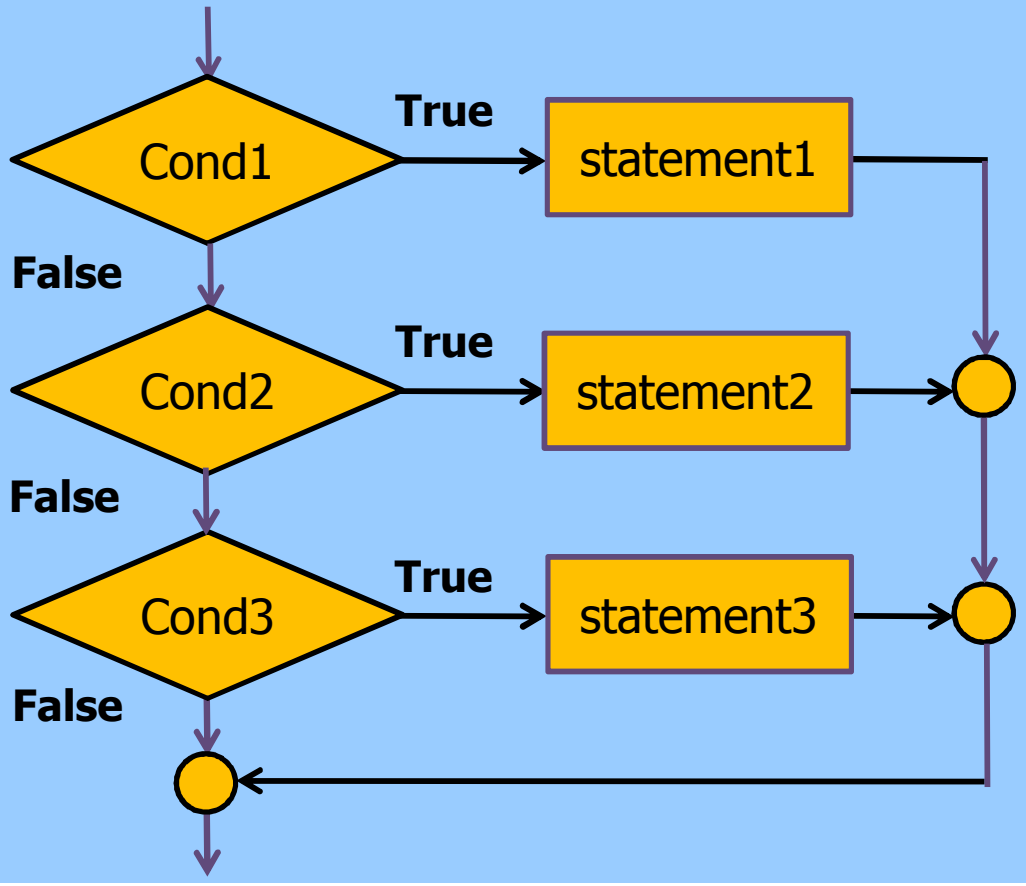
Flow Control

- MATLAB has five flow control constructs:
  - if-else statements
  - switch cases
  - while loops
  - for loops
  - break statements

If-else Statements

- The if-else statement is written as follows:

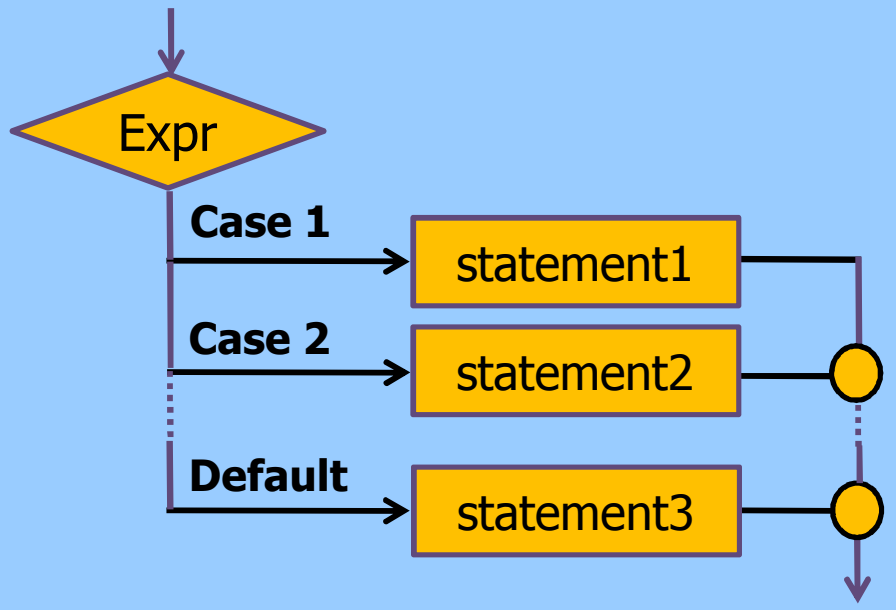
```
if (condition1)
   statement1
elseif (condition2)
        statement2
else
   statement3
end
```



Switch Cases

- The general form of the switch case is:

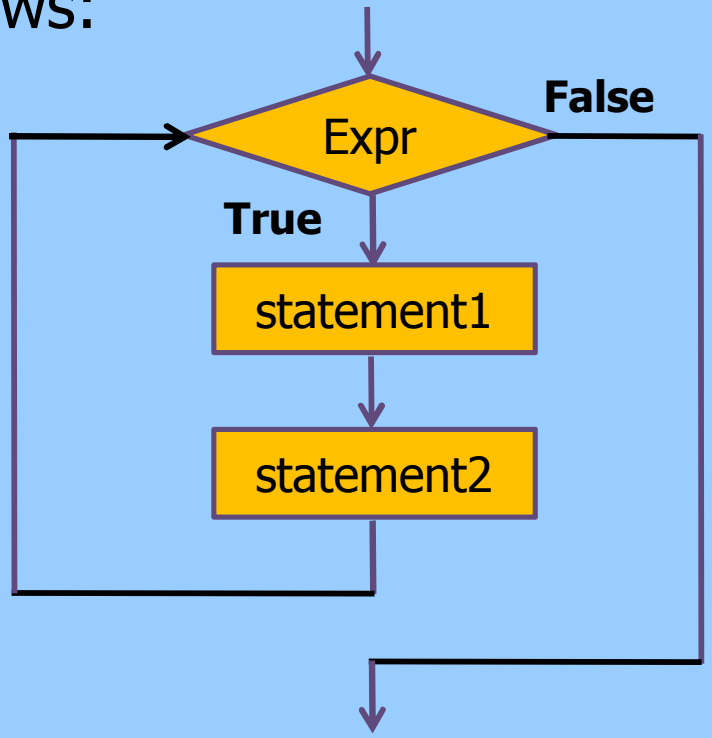
```
switch (expression)
        case value1
                statement1;
        case value2
                statement2;
        otherwise
        default statement3;
 end
```



While Loops

- The while loop is written as follows:

```
while (expression)
        statement1
        statement2
end
```

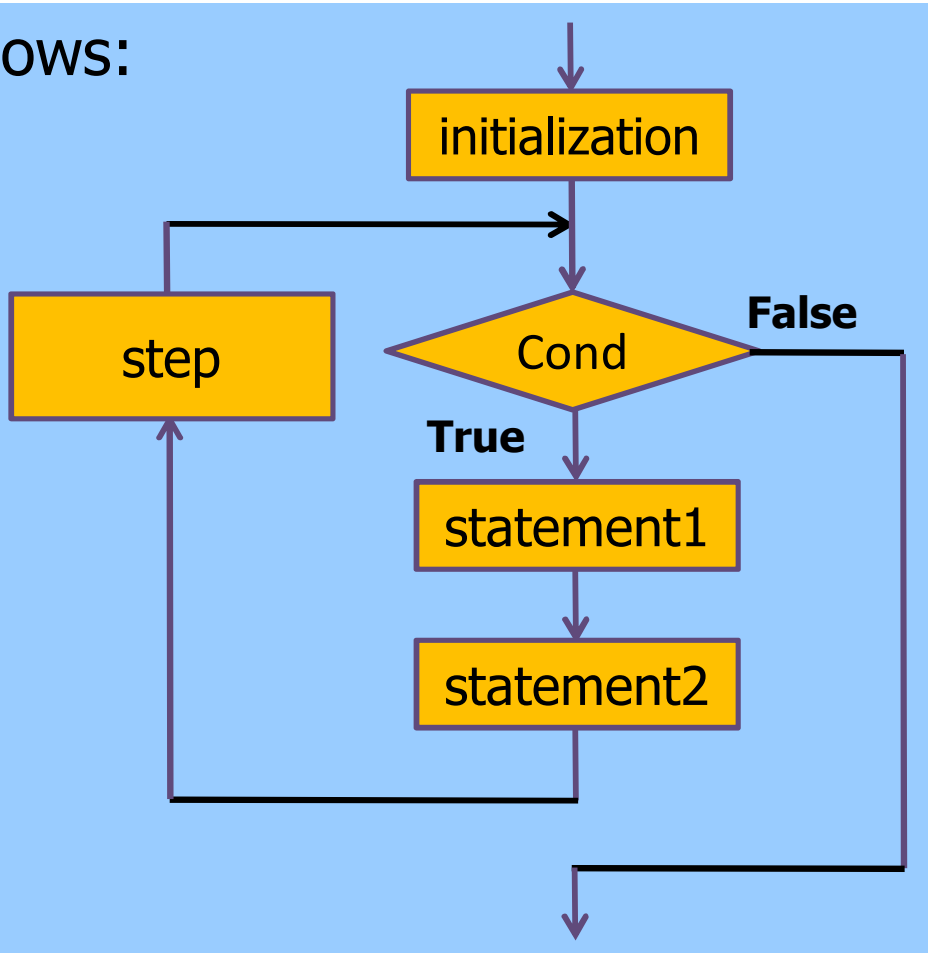


- While-loops are executed as long as the expression is true.
- The expression is tested before entering the loop.

For Loops

- The for loop is written as follows:

```
for index = start : step* : end
    statement1
    statement2
end
(*No step, assumed 1)
```



- Increments index by the value step on each loop, or decrements when step is negative.

Break Statements

- The break statement terminates execution of for or while loop.

Save and Load your Workspace Variables to File

- The save command is used for saving the variables in the workspace, as a file with .mat extension, in the current directory.

```
» save myfile
   % creates myfile.mat with all variables
» save myfile a b
   % save only variables a and b
» clear all
   % clear all variables
» clear a b
   % clear variables a and b
```

- You can reload the file using the load command.

```
» load Myfile
```

M-file

- The M Files, MATLAB allows writing two kinds of program files:
  - **Scripts** - script files are program files with .m extension. In these files, you write series of commands, which you want to execute together. Scripts do not accept inputs and do not return any outputs. They operate on data in the workspace.
  - **Functions** - functions files are also program files with .m extension. Functions can accept inputs and return outputs. Internal variables are local to the function.

Comment in M-file

- When writing code, you always should add comments to your MATLAB code by using the percent (%) symbol. Comments help clarify code by adding notes of what is being done in the code. For example,

```
% example.m
% program display message

    disp('Hello Image processing');
```

Comment lines

Useful Built-in Functions

- length(I)
  - Returns the length of the matrix I.
- [M,N] = size(I)
  - Returns the length of the rows and columns of I in separate variables.
- sum(I)
  - Sums the elements in I. (Can specify the dimension)
- I(:)
  - Selects elements of an array (column by column basis) and stacks them one atop the other.
  - sum(I(:)) == sum(sum(I))

Useful Built-in Functions

- ones(MxN)
  - Returns an MxN array of ones.
- zeros(MxN)
  - Returns an MxN array of zeros.
- rand(MxN)
  - Returns an MxN array of uniformly distributed random numbers from [0,1].

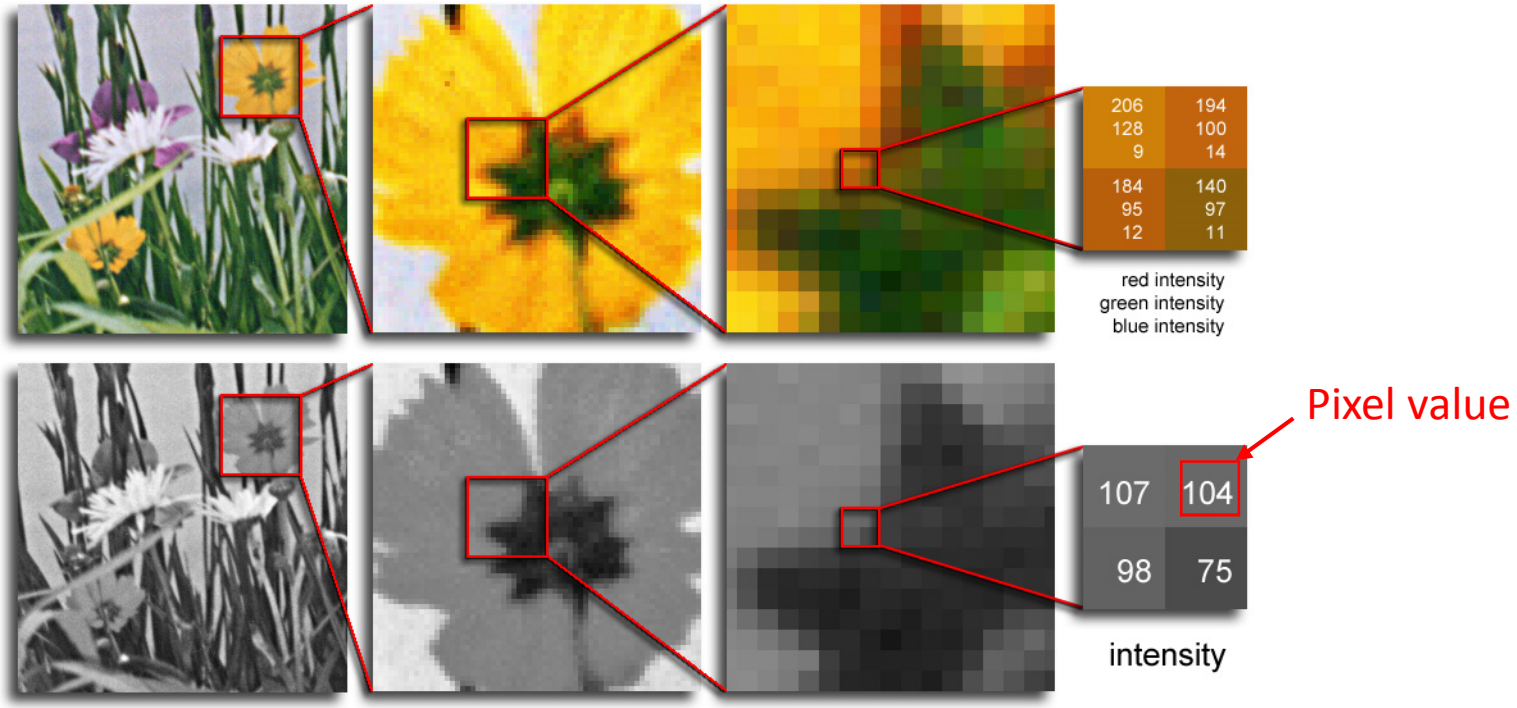
Topics

This lecture will cover:

- MATLAB Basics
- Category of digital image
- Image types
- Importing and exporting images in MATLAB
- Converting between image formats

What Is Digital Image?

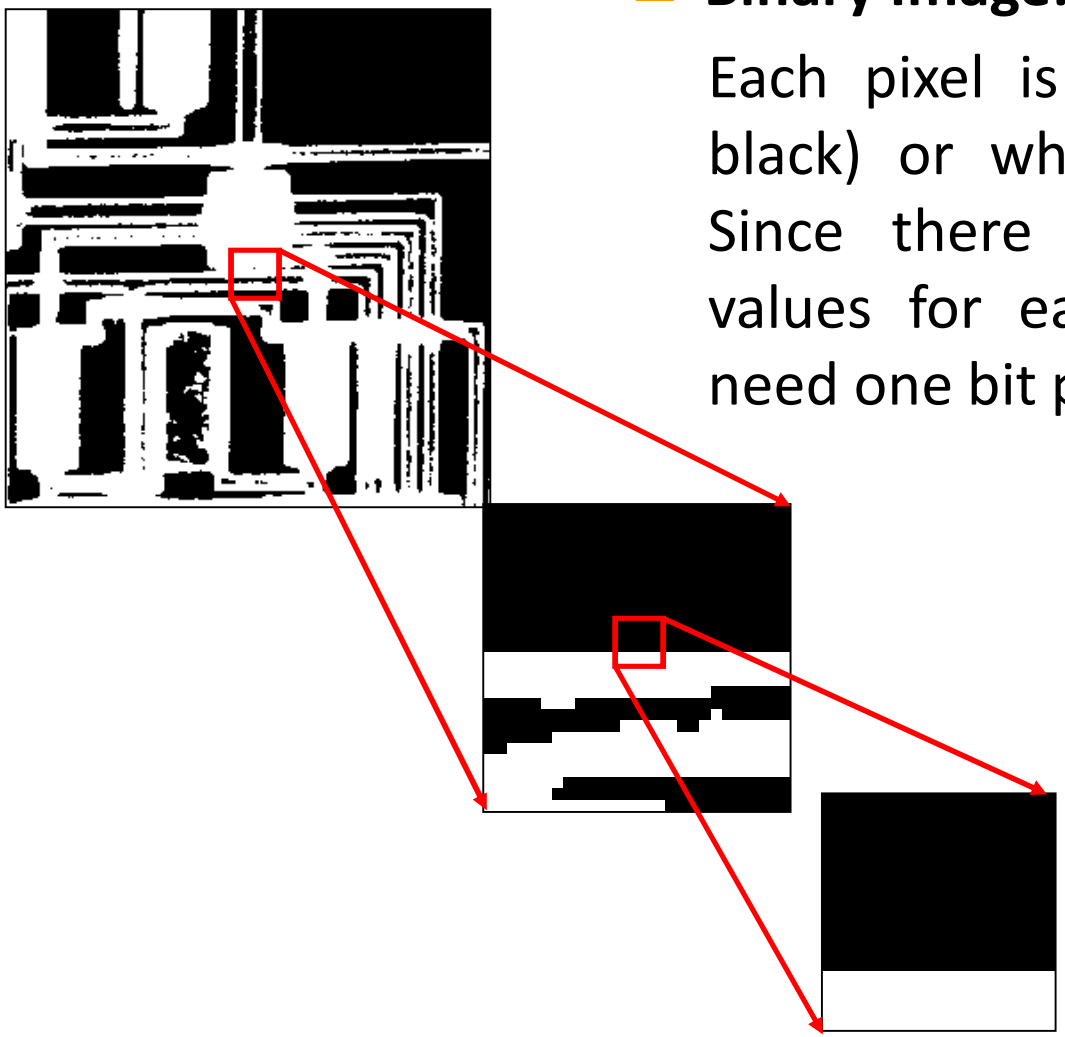
- If *(x, y)* and *f* are finite and discrete quantities, we call the image a digital image.
- A digital image is composed of a finite number of elements called pixels.



Color image: f (*x*, *y*)→ [{0,…,*N*}, {0,…,*N*}, {0,…,*N*}]

Gray scale image: f (*x*, *y*) →{0,1,…, *N*} *N* is finite, discrete quantity

Category of Digital Image



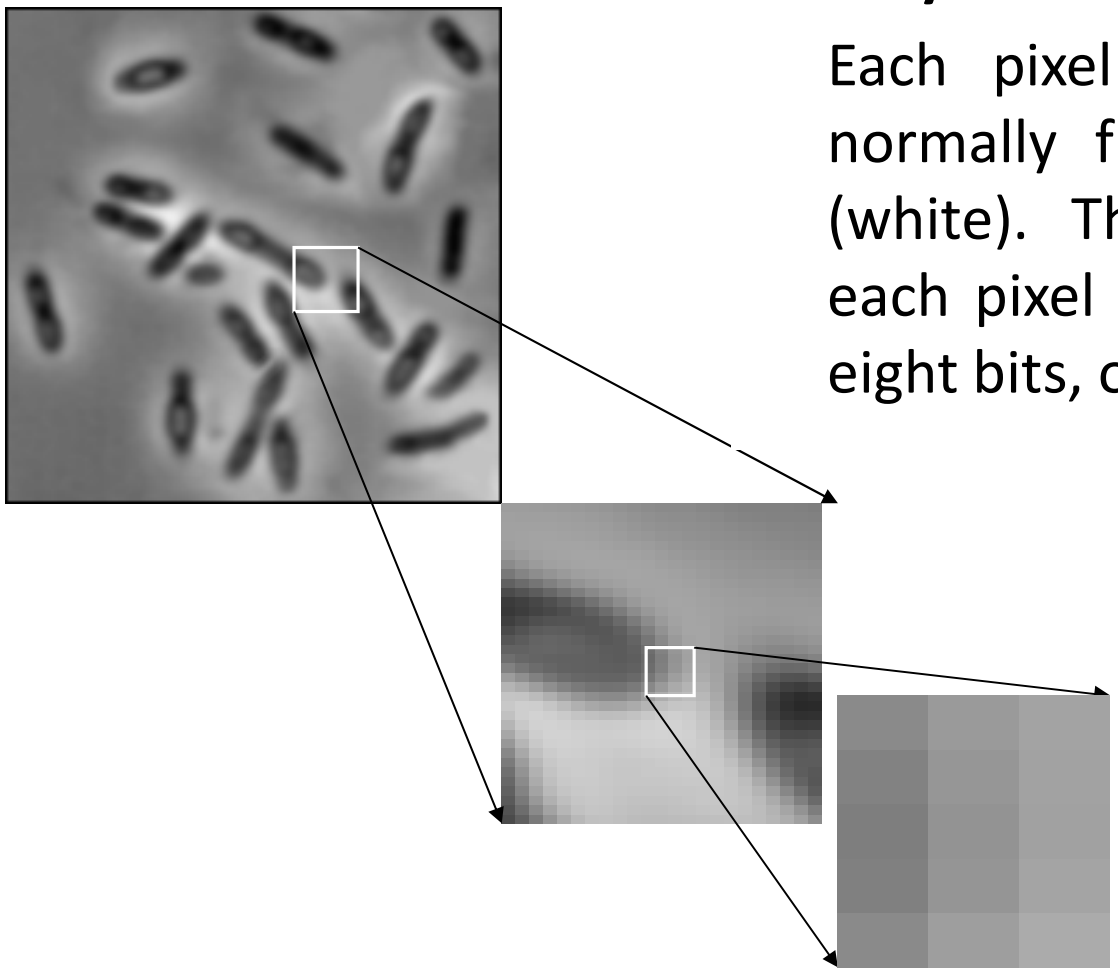
- **Binary Image:**

  Each pixel is just black (0 represent black) or white (1 represent white). Since there are only two possible values for each pixel (0,1), we only need one bit per pixel.

Binary data

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Category of Digital Image



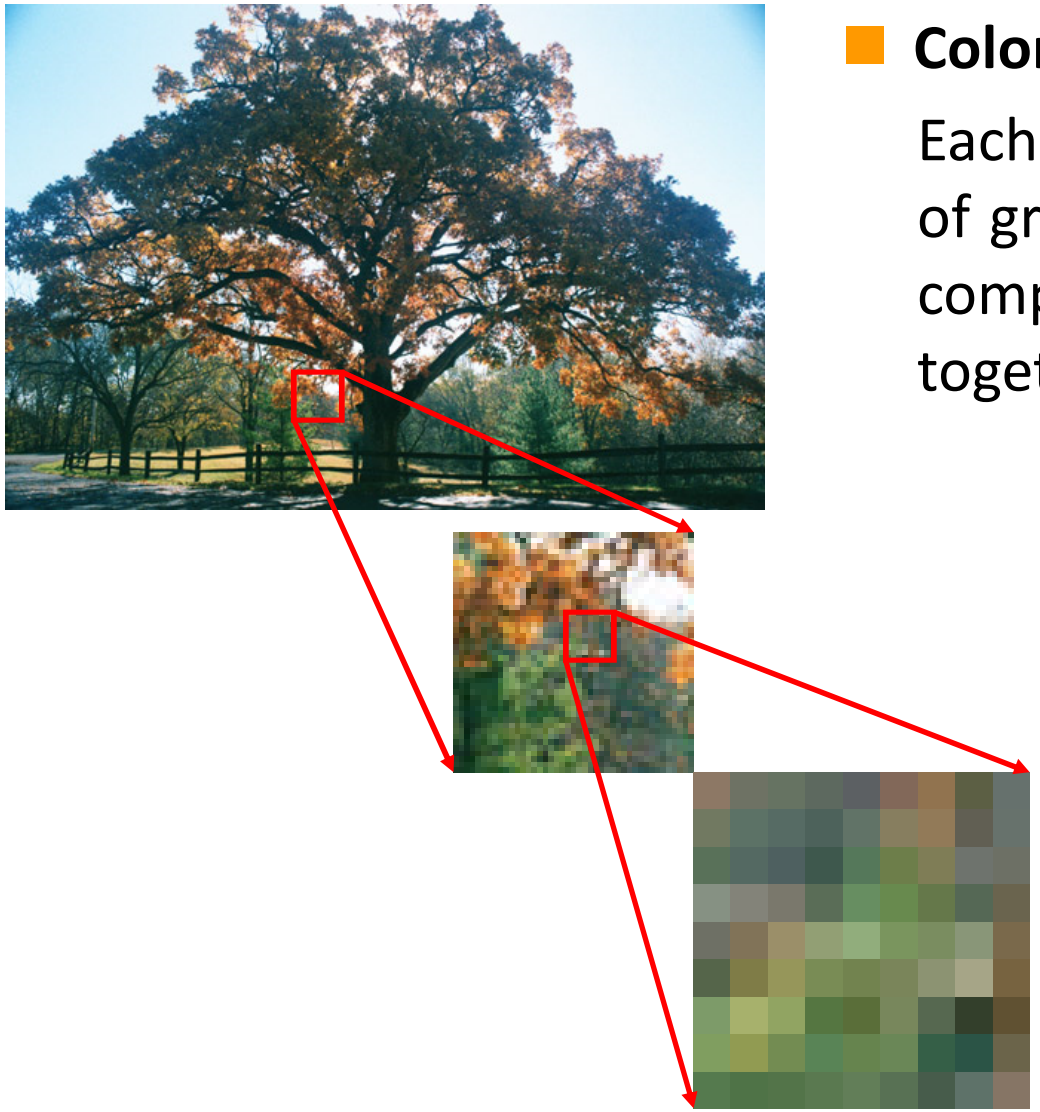
- **Gray Scale Image:**

  Each pixel is a shade of gray, normally from 0 (black) to 255 (white). This range means that each pixel can be represented by eight bits, or exactly one byte.

Gray scale values

$$\begin{bmatrix} 10 & 10 & 16 & 28 \\ 9 & 6 & 26 & 37 \\ 15 & 25 & 13 & 22 \\ 32 & 15 & 87 & 39 \end{bmatrix}$$

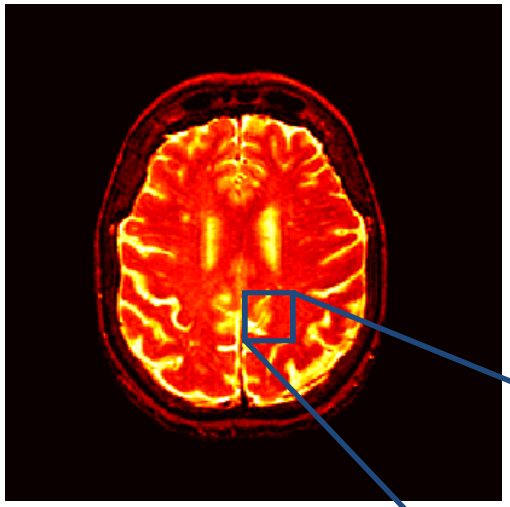
Category of Digital Image



- **Color Image:**

  Each component (R,G,B) is a shade of gray scale (0 - 255). R, G, and B components are combined together to form a color image.

Category of Digital Image



$$\begin{bmatrix} 1 & 4 & 9 \\ 6 & 4 & 7 \\ 6 & 5 & 2 \end{bmatrix}$$

Index value

- **Indexed Image:**

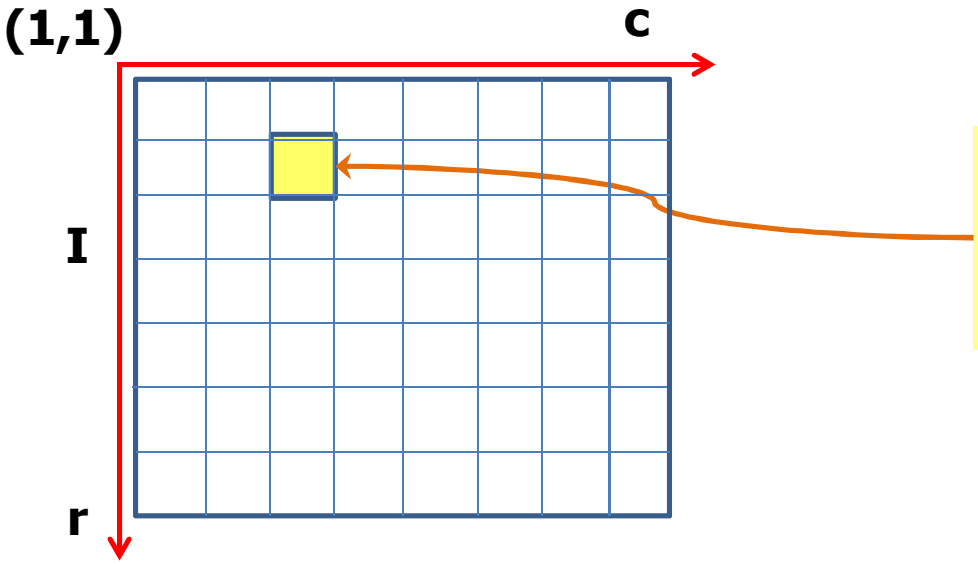
  The pixels in the image are represented by integers, which are pointers (indices) to color values stored in the color table.

Color Table

Index No.	Red component	Green component	Blue component
1	0.1	0.5	0.3
2	1.0	0.0	0.0
3	0.0	1.0	0.0
4	0.5	0.5	0.5
5	0.2	0.8	0.9
…	…	…	…

MATLAB Image Coordinates

- MATLAB stores images as matrices.
- In MATLAB, image pixels are referenced using (row, column) values.
- Origin of the coordinate system (1,1) is the top left corner of the image and can only be positive real integers.
- In Spatial, origin of the coordinate system (0,0).



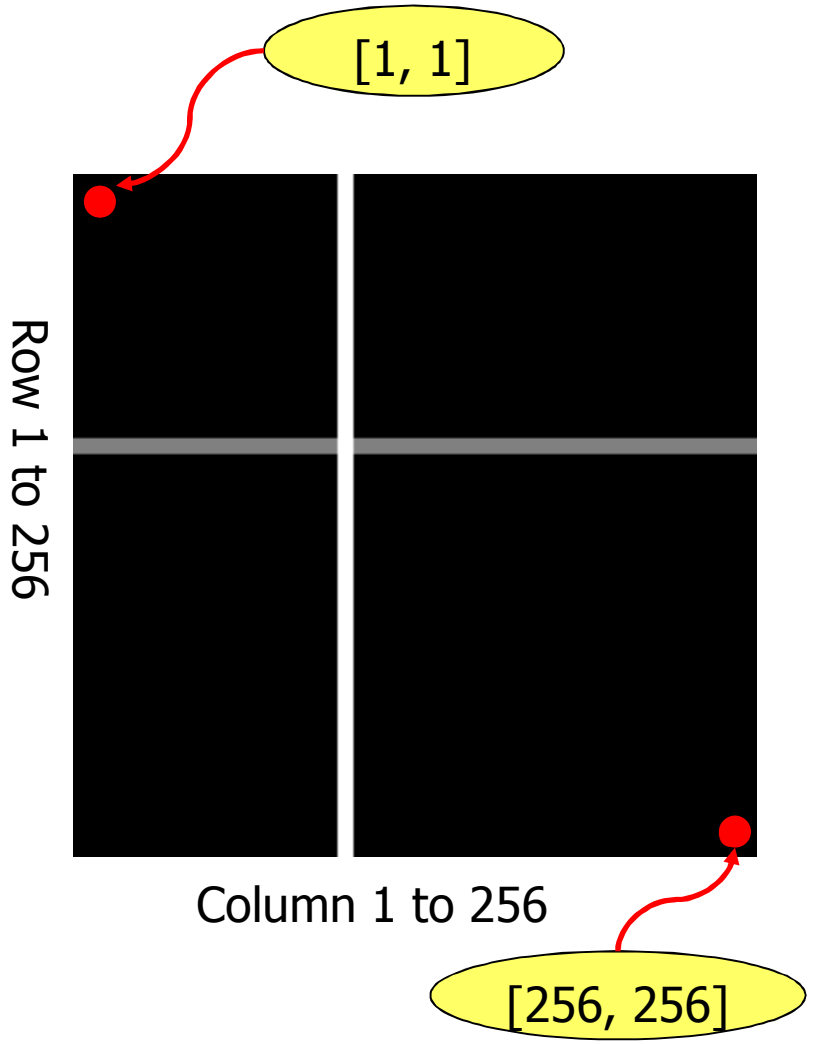
Thus, I(2,3) refers to the pixel at the 2th row and 3rd column.

MATLAB Image

- How to build an image?

- Grayscale image:

```
rows = 256;
cols = 256;
img = zeros(rows, cols);
img(100:105, :)= 0.5;
img(:, 100:105) = 1;
figure;
imshow(img);
```

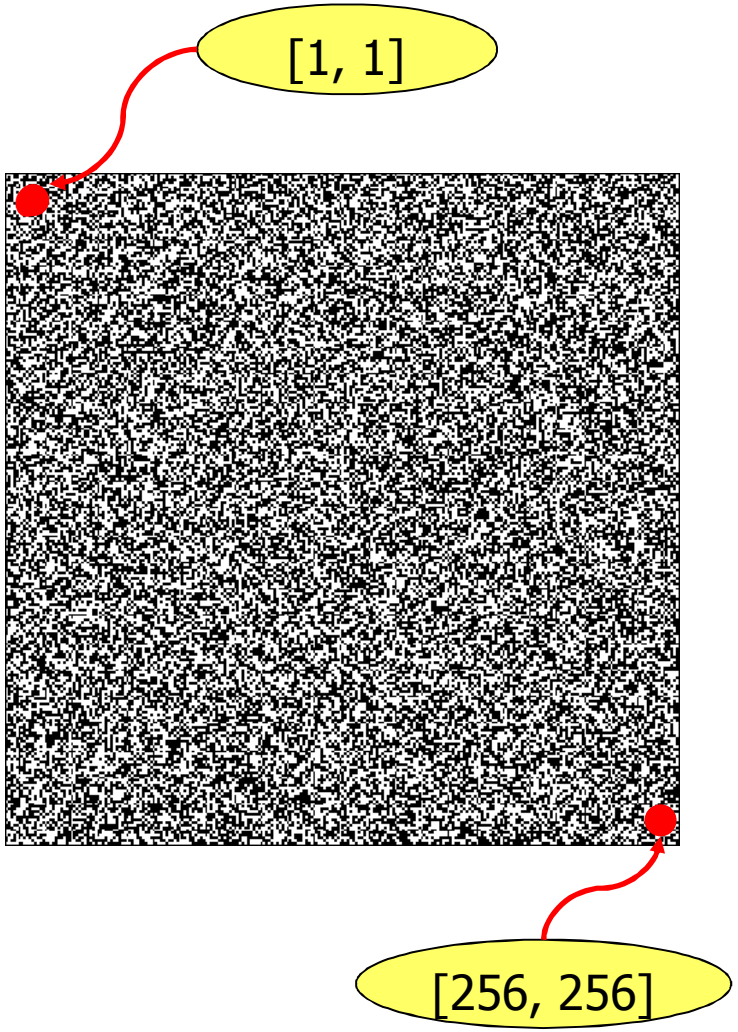


MATLAB Image

- How to build an image?

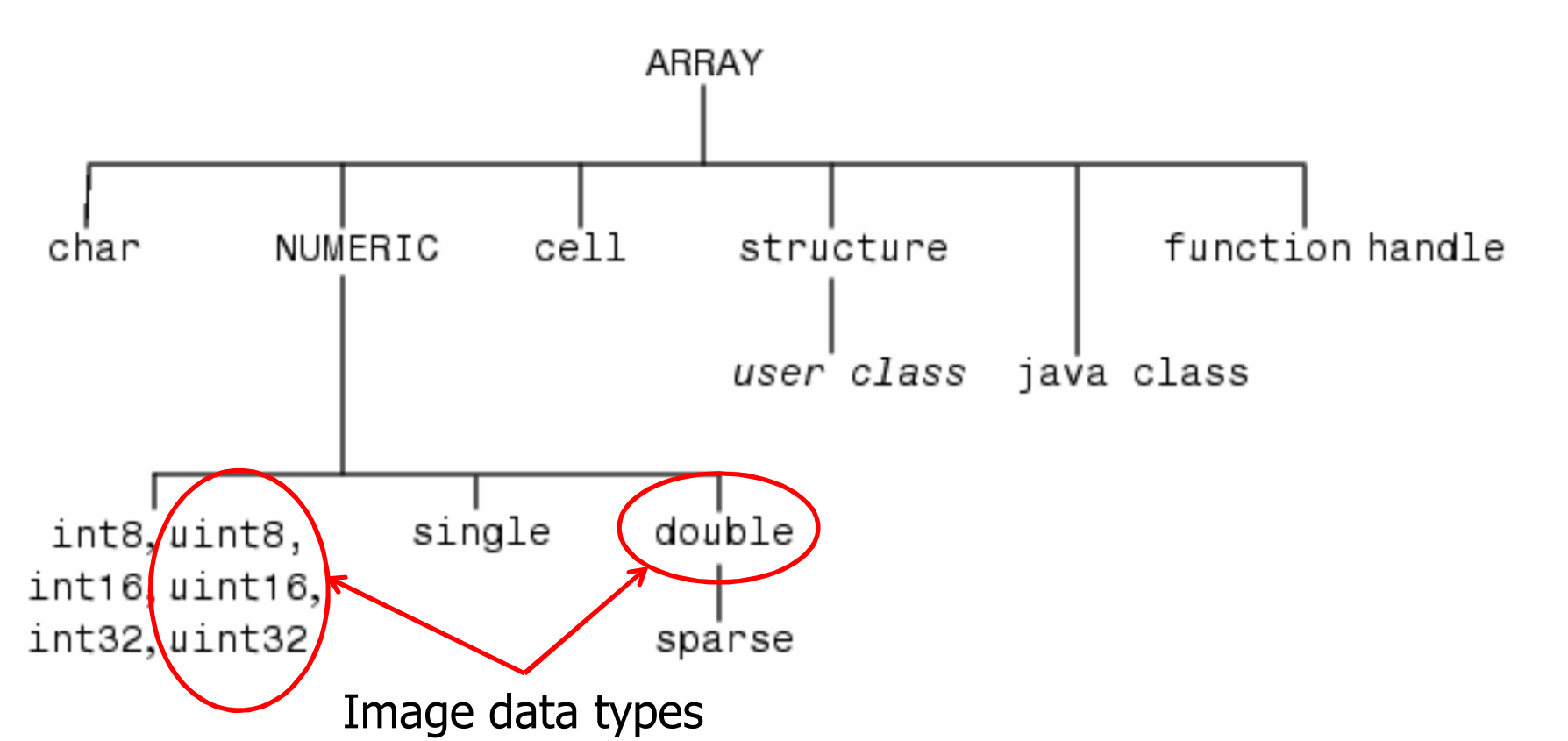
- Binary image:

```
rows = 256;
cols = 256;
img = rand(rows, cols);
img = round(img);
figure;
imshow(img);
```



MATLAB Data Types Used

- A wide array of different data types exist in MATLAB, but only a subset of the data types are used to represent images in MATLAB.



Topics

This lecture will cover:

- MATLAB Basics
- Category of digital image
- Image types
- Importing and exporting images in MATLAB
- Converting between image formats

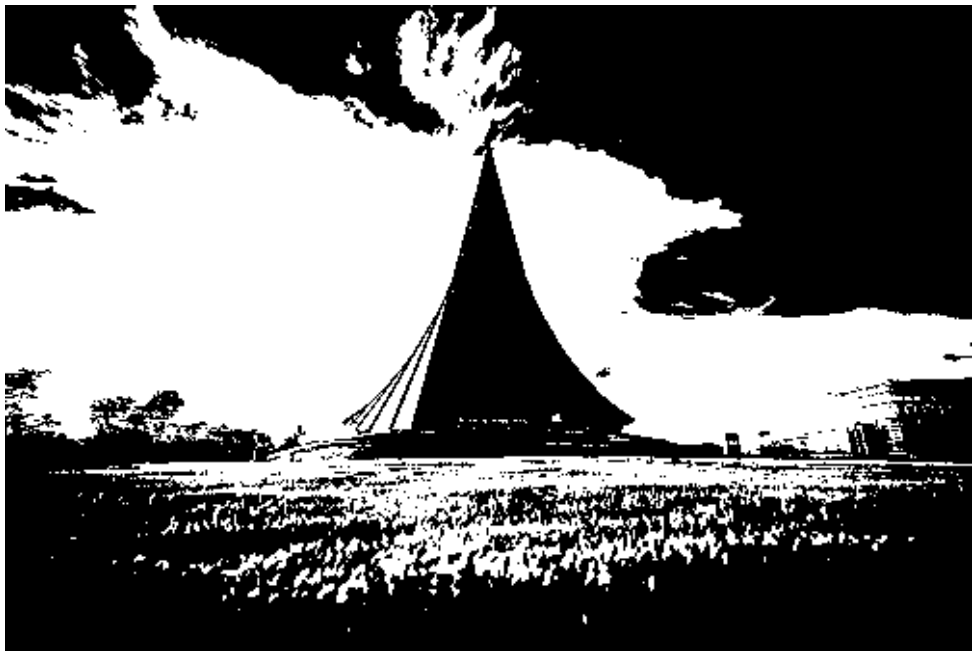
Image Types

- Four basic types of images are supported in MATLAB
  - Binary images : {0, 1}
  - Grayscale images : [0 1] or [0 255]
  - RGB images : m-by-n-by-3 matrix [0 255]
  - Index images : m-by-3 colormap matrix

Binary Images

- In a binary image, each pixel assumes one of only two discrete values: 0 (0 represent black) or 1 (1 represent white).

```
» Ibinary = imread('PrinceMahidol_Binary.bmp');
» imshow(Ibinary);
```



Grayscale Images

- An grayscale image only consists of one matrix, I, whose values represent intensities within some range, for example [0 1] or [0 255] (uint8).

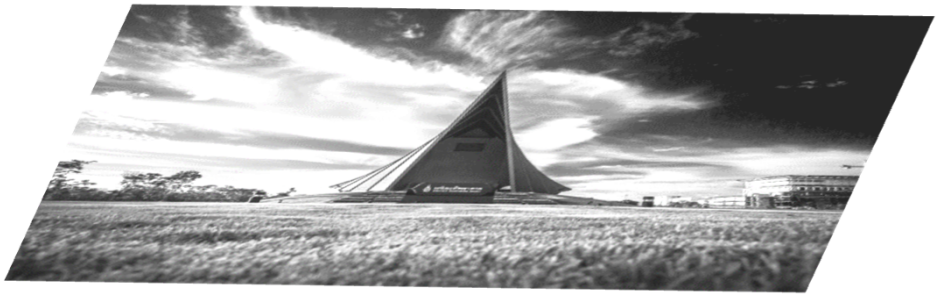
```
» Igray = imread('PrinceMahidol_Gray.bmp');
» imshow(Igray);
```

RGB Images

- A RGB image is stored in MATLAB as an m-by-n-by-3 data where each m-by-n page defines red (R), green (G) and blue (B) color components for each pixel.

```
» Irgb = imread('PrinceMahidol_RGB.bmp');
» imshow(Irgb);
```

RGB Images



Red Component

RGB image (M-by-N-by-3)

Green Component

Blue Component

RGB Images

R

G

B

Indexed Images

- An indexed image consists of a data matrix, Iindexed, and a colormap matrix, map.

```
» [Iindexed,map] = imread('PrinceMahidol_Indexed.bmp');
» imshow(Iindexed,map);
```

Topics

This lecture will cover:

- MATLAB Basics
- Category of digital image
- Image types
- Importing and exporting images in MATLAB
- Converting between image formats

Importing and Exporting Images in MATLAB

- Image Processing Toolbox provides a set of basic tools for image processing that typically includes the following:
  - imfinfo - Returns information about image file.
  - imread - Read image file.
  - imshow - Display image.
  - imwrite - Write image file.

```
» imfinfo('PrinceMahidolHall.bmp')
» I = imread('PrinceMahidolHall.bmp');
» imshow(I);
» imwrite(I,'PrinceMahidolHall.jpg');
```

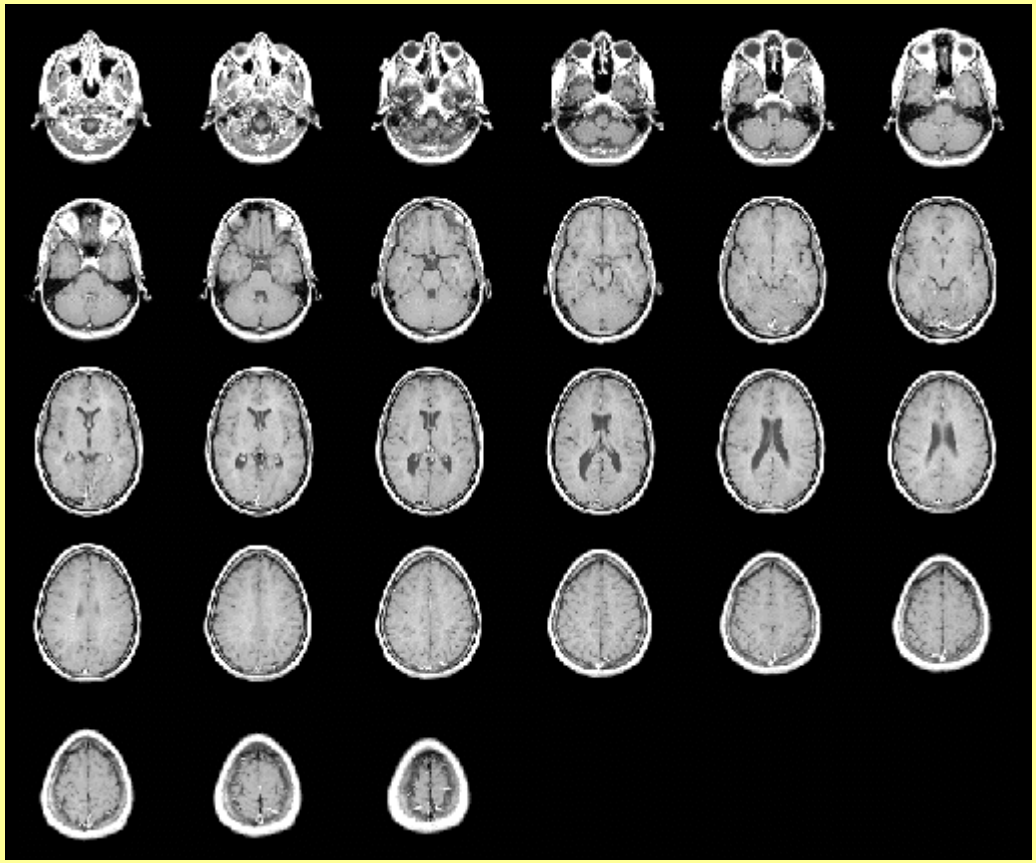
Display Images

- imshow - Display image.
- image - Create and display image object.
- imagesc - Scale data and display as image.
- colormap - Display colormap.
- montage - Display multiple image frames.
- warp - Display image as texture-mapped surface.

Montage

- An example of displaying multiple image frames as a rectangular montage.

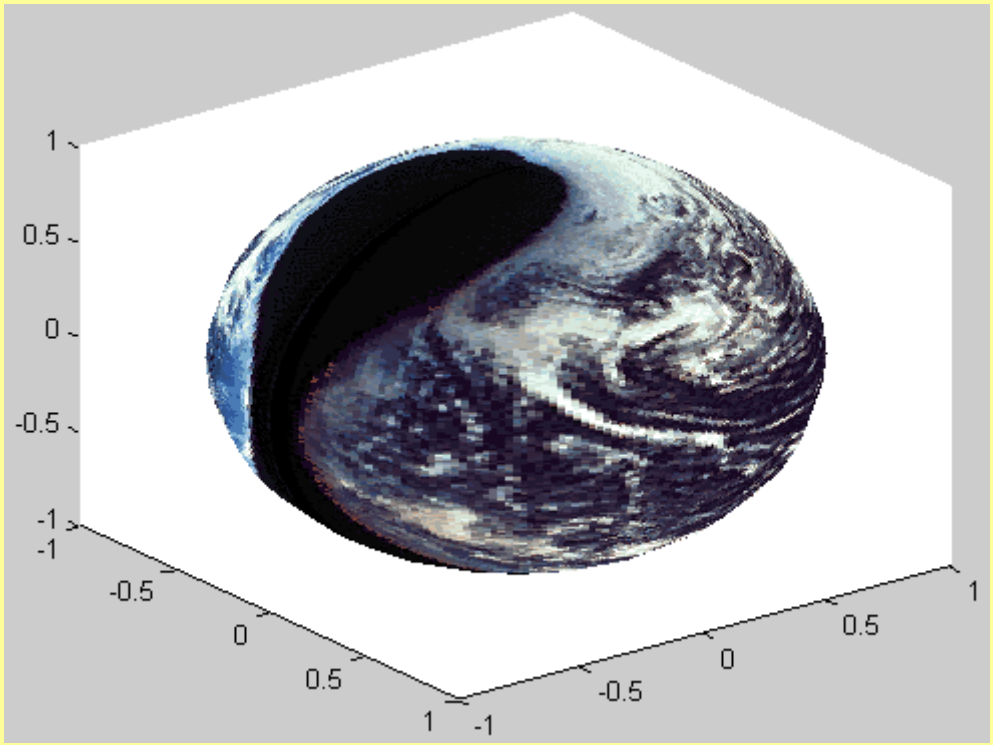
```
» load mri
» montage(D, map)
```



Warp

- The warp function allows you to display an image as a texture-mapped surface.

```
» [x, y, z] = sphere;
» load earth
» warp(x, y, z, X, map)
```



Topics

This lecture will cover:

- MATLAB Basics
- Category of digital image
- Image types
- Importing and exporting images in MATLAB
- Converting between image formats

Converting Image Formats

- gray2ind - Convert grayscale image to indexed image.
- ind2rgb - Convert indexed image to RGB image.
- rgb2ind - Convert RGB image to indexed image.
- rgb2gray - Convert RGB image or colormap to grayscale.

- 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.
- im2bw - Convert image to binary image by thresholding.

im2bw

- The adjusted cameraman image can be thresholded to make a black and white image of the man and the camera.

```
» Igray = imread('cameraman.tif');
» output = im2bw(Igray,0.06);
```

Exercise

- Write program to show only the middle part of cameraman image.
- You can use the images provided here, **http://goo.gl/q8UduL**

Input

output

Exercise

- Write a program to blend two images, using dance1 and dance2 images as input images.

dance1

dance2

output