Lecture Four
Multimedia Systems

Types of Images

By: Eng. Haydara Mahmoud.

Damascus University – Faculty of Informatics - 2019

Data Classes :

There are various data classes supported by Matlab for representing pixel values

Name	Description
`double`	Double-precision, floating-point numbers in the approximate range -10^{308} to 10^{308} (8 bytes per element).
`uint8`	Unsigned 8-bit integers in the range [0, 255] (1 byte per element).
`uint16`	Unsigned 16-bit integers in the range [0, 65535] (2 bytes per element).
`uint32`	Unsigned 32-bit integers in the range [0, 4294967295] (4 bytes per element).
`int8`	Signed 8-bit integers in the range [−128, 127] (1 byte per element).
`int16`	Signed 16-bit integers in the range [−32768, 32767] (2 bytes per element).
`int32`	Signed 32-bit integers in the range [−2147483648, 2147483647] (4 bytes per element).
`single`	Single-precision floating-point numbers with values in the approximate range -10^{38} to 10^{38} (4 bytes per element).
`char`	Characters (2 bytes per element).
`logical`	Values are 0 or 1 (1 byte per element).

All numeric computations in Matlab are done using **double** quantities.

Image Types:

The toolbox supports four types of images. They are:

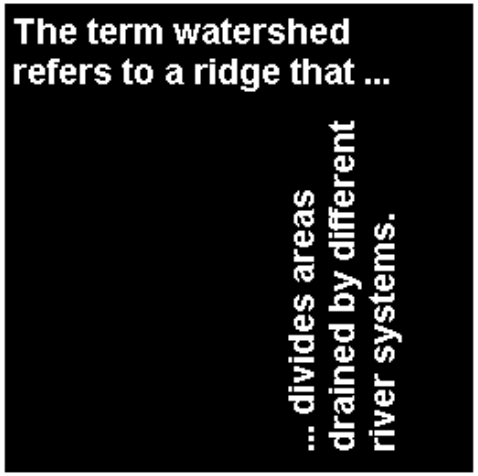
1. Binary images.
2. Intensity (gray-level) images.
3. Indexed images.
4. Color images.

I – Binary Images: (1-Bit)

A binary image is a logical array of 0s and 1s. An array of 0s and 1s whose values are of data class, say, uint8 is not considered a binary image in Matlab. Function *logical* is used to convert a numeric binary array to binary in Matlab. This function is used to convert all nonzero quantities to logical 1s and all entries with value 0 to logical 1s to test an array if it is logical we use the function *islogical*.

Example:

1. Read the image ('**text.png**') ?
2. Show the matrix?
3. What is the Min and Max value of pixels? **(Notice binary)**
4. Display the image?
5. What is the storage space?



text.png

2 – Intensity Images (Gray-Scale): 8-Bits

When the elements of an image are of class uint8 and uint16, they have integer values in the range [0-255] and [0 65535] respectively.
Values of called, class *double* intensity images are in the range [0-1] by convention

Example:

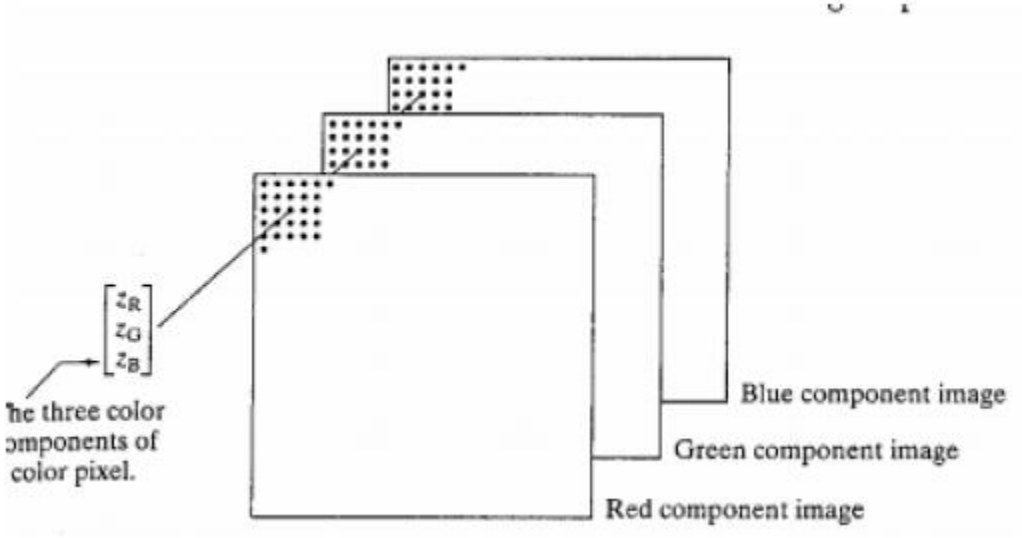
1. Read the image ('**cameraman.tif**') ?
2. Show the matrix?
3. What is the Min and Max value of pixels?
4. Display the image?
5. What is the storage space?

cameraman.tif

3 – Color Images: (24-Bits)

Image Processing toolbox handles colour images either as indexed images or RGB (Red, green and blue).

An RGB image is an MxNx3 array of colour pixels where each colour pixel is a triplet corresponding to the red, green and blue components of any RGB image at a specific spatial location.



Example:

1. Read the image (‘**onion.png**’) ?
2. Show the matrix?
3. Display the image?
4. What is the storage space?

onion.png

Extracting each matrix of RGB:

redChannel = rgbImage(:,:,1); % Red channel

In order to form an RGB image from the three image components RGB, we use the function “cat”

A=cat(3,Fr,Fg,Fb);

Exercise:

1. Read the image (‘**onion.png**’) ?
2. Extract each color channel from that photo?
3. Display the four images (‘onion.png’) and each channel (red, green and blue) on one figure and give them titles? **(What do you understand?)**
4. Delete the blue matrix then merge the photo again and display it?

4 – Indexed Images: (8-bits)

An indexed image has two components: Data matrix of integers and color map matrix which is an mx3 array of class double [0-1].

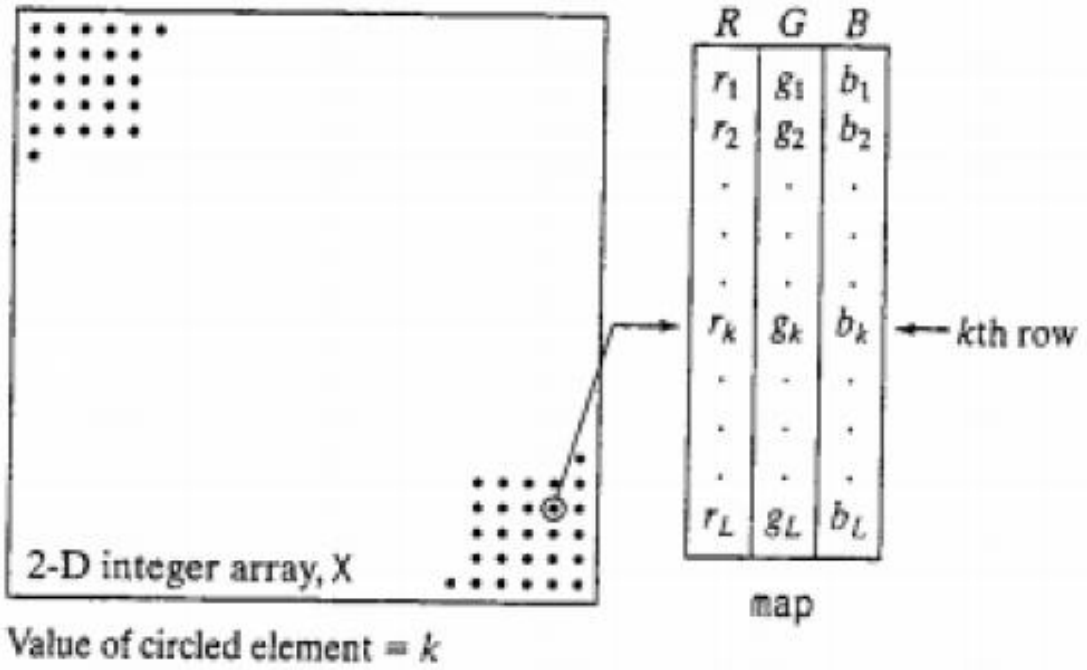
The length, m, of the map is equal to the number of colours it defines.

The relationship between the values in the image matrix and the colormap depends on the class of the image matrix.

If the image matrix is of class single or double, it normally contains integer values 1 through p, where p is the length of the colormap. the value 1 points to the first row in the colormap, the value 2 points to the second row, and so on.

If the image matrix is of class logical, uint8 or uint16, the value 0 points to the first row in the colormap, the value 1 points to the second row, and so on.

Suppose we are dealing with 8-bit colour image. This let each integer pixel value (pointer) of the image points to an entry of 256 levels of the colormap. For any given image, this does not mean that the all 256 levels are used. Only the colours defined by the image are used. It might be 222 or 200.



To display an indexed image, we write

imshow (X,map)

or

image (I); colormap(map)

Example:

1. Read the image (‘**trees.tif**’) ?
2. Display it?
3. How many colors are there in the image? (Use ***colorbar***)?
4. What is the size of the image?
5. What is the value of pixel in (30,100)?
6. Display the image without map matrix? **What do you understand?**

trees.tif

Example:
(Another way to read image)

1. Read the image (‘**clown**’) ?
   USE
   (load clown; image(X); colormap(map);)
2. How many colors are there in the image?
   (Use ***colorbar***)?
3. What is the size of the image?
4. What is the value of pixel in (30,100)?

clown

Adjust colors in intensity images:

```
load clown
image(X); colormap(1-map); colorbar
figure; image(X); colormap(abs(0.5-map)); colorbar
```

Converting between Data Classes:

Matlab expects operands in numerical computations to be double-precision floating-point numbers. If C is an array of class double in which all values are in the range [0-255], it can be converted to an uint8 array with the command D=uint8(C). Values that are less than 0 are converted to 0 and values that are greater than 255 are converted to 255. Numbers in between are converted to integers by discarding their fractional parts.

Converting between Image Types:

Name	Converts Input to:	Valid Input Image Data Classes
im2uint8	uint8	logical, uint8, uint16, and double
im2uint16	uint16	logical, uint8, uint16, and double
mat2gray	double (in range [0, 1])	double
im2double	double	logical, uint8, uint16, and double
im2bw	logical	uint8, uint16, and double

Example 1:

Given an image f of class double

f=

-0.5000 0.5000

 0.7500 1.5000

g=im2uint8(f)

Im2uint8 is different from uint8

mat2gray:

Converts an arbitrary array of class double to an array of class double scaled to the range [0-1].
If the input image is of class uint8m uint16 or logical, function im2double converts it to class double with values in the range [0-1].
If an array of class **double** results from computations that yield values outside the range [0-1] , **im2double** will have no effect on it. In this case **mat2gray** should be used.

Example 2:

```
h=uint8([25 50; 128 200])
im2double(h)
```

Example 3:

```
h=[25 50; 128 200]
im2double(h)
```

Example 4:

1. Read the image (‘**cameraman.tif**’) ?
2. Convert it into black and white colors (Logical)?
3. Display the two pictures on one figure?

Example 5:

1. Read the image (‘**trees.tif**’) ?
2. Convert it into black and white colors (Logical)?
3. How many colors are used in each image?
4. Display the two image on one figure?

Example 6:

1. Read the image (‘**clown**’)?
2. Convert it into **gray-scale** by using ***ind2gray*** ?
3. Convert the gray-scale image into black and white image?
4. Display the three images and title them?