Introduction to digital image Representation

Page No.:

* Digital image representation :-

- An image may be defined as two dimensional function f(x, y) where x and y are spatial co-ordinates and f is the intensity (amplitude) at any pair of co-ordinates x, y.
- The term gray scale or gray level is used to refer to the intensity of monochrome images.
- An image may be continous with respect to the x and y co-ordinates and also in amplitude.
- To convert such image into digital form needs to convert co-ordinates as well as intensity of image.
- Digitalizing the co-ordinates value is called sampling and digitalizing amplitude value is called as quantization.
- Thus, when x, y and amplitude values of f are finite, discrete., continuous then we called the image a digital image.

* Co-ordinate Conventions :-

- The result of sampling and quantization is a matrix of real numbers.
- Assume that an image f(x, y) is sampled so that resulting image has m rows and N colums.
- We can say that the image is of size MXN. The co-ordinates convention is used in the image processing toolbox is given below.

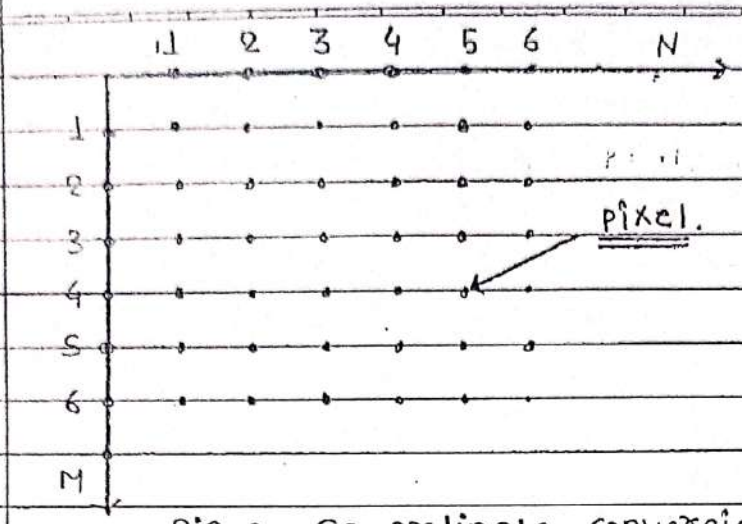


fig :- Co-ordinate conversions used in matlab toolbox.

- The co-ordinate system in above figure represents a matlab matrix as.

$$f = \begin{bmatrix} f(1,1)\ f(1,2) \cdots f(1,n) \\ f(2,1)\ f(2,2) \cdots f(2,n) \\ f(3,1)\ f(3,2) \cdots f(3,n) \\ \vdots \\ \vdots \\ f(m,1)\ f(m,2) \cdots f(M,N) \end{bmatrix}$$

- Matrix in matlab are stored in variables with names such as a, f, xfh etc.
- variables must begin with a letter and may contain only letters, numerals and underscores.

Page No.:

* Reading, Writting and displaying Images :-

1) Reading Images :-

- Images are read into matlab environment using function imread.
- The syntax of imread is imread ('filename');
- Here filename is the string containing the complete name of the image file containing extension.

Ex :-

```
>> f = imread ('coins.png');
```

- Reads the png file into image array f. The semicolon at the end of the statement is used for supressing the output.
- If a semicolon is not used, matlab displays the result of operation on screen.
- Following are some of the image/ graphics formats supported by imread and imwrite.

Format name	Description	ref recognized extensions.
TIFF	Tagged Image file format	.tif, .tiff.
JPEG	Joint Photographic Experts Group.	.jpg, .jpeg.
PNG	Portable Network Graphics.	.png.
BMP	BitMap picture	.bmp.

- Function size gives the row and column dimensions of an images.

Ex :

```
>> size (f)
    246   300
>> [R, C] = size (f)
    R = 246
    C = 300.
```

2) Writing Images :-

- Images are written onto the disk using function imwrite.
- The basic syntax of imwrite is imwrite (f, 'filename') ;
- Where f is the image array containing the filename to be copied and file name specifies the destination file name.
- By using imwrite function, you may change the format of source file.

Ex :

```
>> f = imread ('coins.png') ;
>> imwrite (f, 'mycoins.jpg') ;
>> imfinfo conins.png
>> imfinfo mycoins.jpg
```

3) Displaying Images :-

- Images are displayed on the matlab desktop using function imshow, which has the basic syntax :-

imshow (f, g).

- Where f is an image array and G is the number of intensity levels used to display it.

- If G is omitted, it defaults to 256 using syntax:

imshow(f, [low, high])

- Displays all values less than low as blank and all values greater high as white.

- The values inbetween are displayed as intermediate intensity values using the default gray levels.

- Finally, the syntax imshow(f,[]) sets the variable low to the minimum value of array f and high to the maximum value of f.

- This form of imshow is useful for displaying images that have a low dynamic range.

Ex:

```
>> f = imread('rice.png');
>> imshow(f)
>> imshow(f, [50, 150])
>> imshow(f, [])
>> imshow(f), figure, imshow(f, [50, 150]), figure, imshow(f, []);
```

or

```
>> f = imread('rice.png');
>> subplot(1,3,1); imshow(f);
>> subplot(1,3,2), imshow(f, [50, 150])
>> subplot(1,3,3), imshow(f, []).
```

Page No.:

★ Data classes and Image types :-

Data classes :-

- matlab supports 10 data classes that are listed in the following table.

Data class	Description.
double	- It is a double pricision floating point numbers whose approximate range is 10^{-308} to 10^{307}. - This data class assigns 8 bytes per element.
uint8	- It is a unsigned 8 bit integer with range [0, 255]. It allocate 1 byte per element. The defaut dataclass of an image is uint8.
uint16	- It is unsigned 16 bit integers with range [0, 65535]. It allocate 2 bytes per element.
uint32	- unsigned 32 bit integers in range [0, 4294967295]. It allocate 4 bytes per element.
int8	- It is signed 8 bit integers in the range [-128, 127]. It assign 1 byte per element.

int16	- It is signed 16 bit integers in the range [-32768, 32767]. It assigns 2 bytes per element.
int32	- It is signed 32 bit integers in the range [-2147483648, 2147483647]. It assigns 4 bytes per element.
single	- It is single prisision floating point numbers with values in approximate range 10^{-38} to 10^{37}. It assigns 4 byte per element.
logical	- This data class allows logical 0's or logical 1's. It assign 1 byte per element..
char.	- It is a character datatype allocating 2 bytes per element.

Examples :-

① double :

a = [12.56 -19 ; 4.5 98]

a = 12.5600 -19.0000
4.5000 98.0000.

② uint8 :-

b = uint8([100 908 ; -25 150])

b = 100 255
0 150

③ uint16 :-

c = uint16([10000 98 ; -250 15.70])

c = 10000 98
0 16.

④ uint32 :-

d = uint32([-30 12.11 ; 500 25000])

d = 0 12
500 25000.

⑤ Single :-

e = single([-45 9.8 ; 122 1000])

e = -0.0045 0.0010
0.0122 1.0000

⑥ int8 :-

f = int8([100 908 ; -30 19.14])

f = 100 127
-30 19.

⑦ int16 :-

G = int16([10000 98 ; -250 15.70])

G = 10000 98
-250 16.

⑧ int32 :-

H = int32([879 6894 ; 50.34 1570])

H = 879 6894
50 1570.

Page No.:

(9) Logical :-

```
I = logical ([25 0 -4 ; 1 7.8 5])
I = 1 0 1
    1 1 1
```

(10) Char :-

```
I = char ['COCSIT']
I = cocsit
```

Page No.

* Image types :-

- The matlab toolbox supports four types of images:

1) Intensity Images.
2) Binary Images.
3) RGB Images.
4) Indexed Images.

1) Intensity Images :-

- An intensity image is a data matrix whose values have been scaled to represent intensities.
- When the elements of an intensity image are of data class uint8 or uint16, they have intensity values in the range [0, 255] or [0, 65535] respectively.
- If the image is of class double, the values are floating point numbers in the range [0, 1].

2) Binary Images :

- Binary Images have a very specific meaning in matlab.
- A binary image is a logical array of 0's and 1's.
- Thus a binary image is of data class logical.
- You can use 'logical' function to convert any numeric array to logical array.

- If 'A' contains elements other than 0's and 1's, the 'logical' function can be used to convert 'A' to logical array.

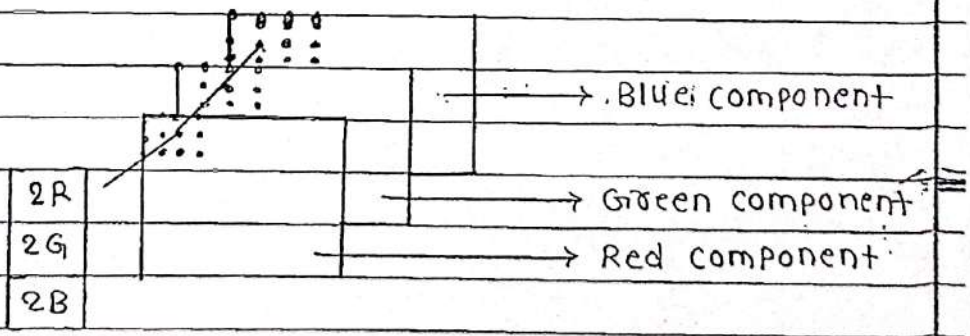
3) RGB Images :-

- An RGB is a mxnx3 array of color pixels, where each ~~other~~ color pixel is a triplet corresponding to the Red, green and blue components of an RGB image at a specific location.

- The three images forming RGB color image are referred to as the Red, green and blue component images.

- The dataclass of the component images determines their range of values.

- A typical representation of pixels of RGB image is shown in following figure.



Three color component of a color pixel.

fig : A typical representation of RGB images.

4) Indexed Images :

- An indexed image has two components - A data matrix of integers (x) and a color map matrix (map).
- A matrix map is m*3 array of class double containing floating point values in the range 0 to 1.
- The length 'm' of the map is equal to the number of colors it defines.
- The color of each pixel is determined by using the corresponding values of integer matrix 'x'.

★ Practicle :

% Study of Image Types :

```
>> f = imread ('Cameraman.Png');
>> g = imread ('Paper.Png');
>> h = imread ('Coins.Png');
>> subplot (1, 3, 1), imshow (f)
>> subplot (1, 3, 2), imshow (g)
>> subplot (1, 3, 3), imshow (h)
```

Page No.:

* Converting between Data classes and Image Types :

* Data classes:

- The toolbox provides specific functions that perform the scaling nessecary to convert beth image types & data classes.

- To change the data class of any array the syntax is

B = Datatype (a)

- where a is a variable of any data class and data type represent the type in which conversion is to be done.

For ex :

```
>> a = [-20  10 ; 28.2  40]
>> b = Uint8 (a)
```

```
-20  -10        b =   0   10
28.2  -40             28   40
```

```
>> c = logical (a)
```

```
c =  1   1
     1   1
```

* Converting between Image Types :

- The IPT (Image Processing Toolbox) in matlab provides following functions for converting bet^n Image types.

Function Name	converts input to	valid input image dataclass.
im2uint8	uint8	logical, uint8, uint16, double.
im2uint16	uint16	uint8, logical, uint16, double.
im2double	double	logical, uint8, uint16, double.
mat2gray	double[(0,1)]	double
im2bw	logical	uint8, uint16, logical.

1) im2uint8 :

- Function im2uint8 converts an input to uint8.
- If the input is already of uint8 dataclass, this function returns the same result.

Ex :

```
>> f = imread ('coin.png')
>> g = im2uint8 (f)
>> imshow (f, []), figure, imshow (g, [])
```

Page No.:

e) im2uint16 :

- Function im2uint16 converts an input to uint16 data class.
- values in the output image are in range [0, 65535].
- If the input is same class no changes are done.

ex :

```
>> f = imread ('rice.png');
>> g = im2uint16 (f);
>> imshow [f, []), figure, imshow (g, [])
```

5) im2double :

- Function im2double converts an input to class double.
- The values in the output image are adjusted in the range [0, 1].

Ex :

```
>> f = imread ('rice.png');
>> g = im2double (f);
>> imshow (g)
```

4) mat2gray :

- An arbitrary array of class double can be converted to class double scaled in the range [0, 1] can be accomplished with the function mat2gray.
- The basic syntax of: mat2gray is

```
g = mat2gray (f, [fmin fmax]);
```

- where g is the output image whose values are scaled in the range [0, 1].

- The parameters fmin and fmax are such that values less than fmin in f because 0 in g and values

ex:

```
>> f = imshow imread ('rice.png');
>> g = mat2gray (f);
>> imshow (g)
```

5> im2bw:

- Finally the function im2bw produces a binary image from an intensity image.

- The syntax is

g = im2bw (f, T);

- where T is the threshold value.

- The output image g has values 0 for all pixels with intensity values less than T and values 1 for all other pixels.

- Thus the output image will have either 0 or 1 value.

- The default value of T is 0.5.

Ex:

```
>> f = imread ('rice.png');
>> g = im2bw (f);
>> imshow (g)
>> g = im2bw (f, 0.4);
>> imshow (g)
>> g = im2bw (f, 0.7);
>> imshow (g)
```

Page No.:

* Introduction to m-function Programming :

- In matlab m-files are created using a texteditor and are saved with a name in form 'filename.m'. such as multi.m or sumprod.m.

- m-functions are also written in texteditor and follow some rules like functions should start with a keyword function and should be saved with function name.

- Following are the components of m-function file.

1) The function defn line.
2) The H1 line.
3) Help Text.
4) function body
5) Comments.

1) The function defn line :

- The function defn line has the form.

function [output] = functionname (Inputs)

Ex :

function [x, y] = Sumprod (a, b)

- where a, b are the inputs.
- x is used to store sum of (a and b) while y is used to store product of a and b.
- The function defn line should always start with keyword function.

Page No.:

2) The H1 line:

- The H1 line is the first textline.
- It is a single comment line that immediatly follows the function defn line.
- Blank lines are not allowed in betn H1 line and function defn line

3) Help Text:

- The help text is a text block that follows the H1 line without any blank lines in between.
- Help text is used to provide comments and online help for the function.
- when the user types

>> help functionname at the command prompt matlab displays all comment lines that appear after function defn line.

4) Function Body:

- The function body contains all the matlab statements that perform computation and assigns value to the output variable.

5) Comments:

- Comments are the lines Precided by %. which can be added at the end of the m-function file.

Page No.:

Ex :

```
function [x,y] = sumprod (a,b)
%. Function to compute sum and product.
%. ________ (Multiple comments):
%. ________
x = a+b ;
y = a*b ;
disp (x)
disp (y)
```

Note : save this m-function with name sumprod.

* Operators in matlab :-

- matlab operators are grouped into three main categories.

1) Arithmetic operators.
2) Relational operators
3) Logical operators.

1) Arithmetic operators :-

- matlab has two different types of arithemetic operations - matrix arithmetic operation and

- matrix operations are done by following the rules of linear algebra while array operations are done on an element by element basis.

- The dot operator distinguishes array operations from matrix operations.

Ex:

a*b is a matrix multiplication.

a.*b is a array multiplication.

- Following table list array and matrix arithmetic operators.

operator	Name	matlab function	Example
+	Array & matrix addn	plus (a,b)	>> c=a+b; or >> c = plus(a,b)
−	Array & matrix subtraction	minus (a,b)	>> c = a-b; or >> c = minus(a,b)
.*	Array multiplication	times (a,b)	>> c = a.*b; or >> c = times(a,b)
*	matrix multiplication	mtimes (a,b)	>> c = a*b; or >> c = mtimes(a,b)
./	Array right division	edivide (a,b)	>> c = a./b; or >> c = edivide(a,b)
.\	Array left division	ldivide (a,b)	>> c = a.\b; or >> c = ldivide(a,b)