Image Processing ToolBox (IPT)
" MATLAB"

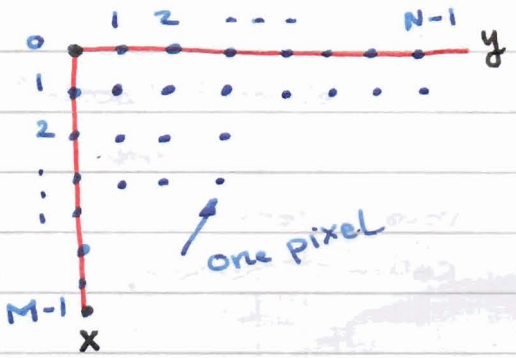
IPT is a Collection of functions that extend the Capablity of the Matlab numeric computing enviroment. and support a wide range of image processing, including.

- Spatial image transformation
- Morphological operation
  Neighborhood and block operations.
- linear filtering and Filter design
- Transform
- Image enhancement & analysis
- Deblurring
- Region of interest operations

Coordinate Conventions:

The result of sampling and Quantization is a matrix of real numbers, Two principal ways are used to represent digital image:

1. first Convention: (used in many image processing books)

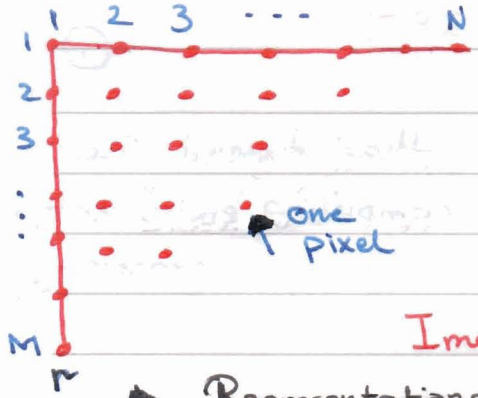


- Image origin is defined to be at $(x,y) = (0,0)$, the next Coordinate values along the first row of the image are $(x,y) = (0,1)$
- x ranges from 0 to M-1
  y ranges from 0 to N-1
  in integer increment.

2- Second Convention:

in IPT Matlab toolbox, another Convention is used, the notation (r,c) to indicate rows and Columns

- the origin of the coordinate system is at $(r,c) = (1,1)$.
- r ranges from 1 to M
  c ranges from 1 to N in integer increment.



IPT documentation refers to the Coordinates as pixel coordinates, or Spatial Coordinates (less than the first one).

Images as Matrices:

- Representations of digital image functions:

1.
$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix}$$

2. Matlab matrix representation:

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

The two representations are equal except for the shift in origin:

The notation $f(p,q)$ denotes the element located in row p and column q.

$1 \times N$ matrix is called a row vector.

$M \times 1$ matrix is called a Column Vector.

1×1 matrix is a scalar.

① "Reading Images"

To read an image, use the imread command, whose syntax is:

`I = imread('filename')`: read and store the image in an array named I.

"filename" - string containing the complete name of the image file include extensi

For example :

f = imread('chestxray.jpg');

reads the jpg image chestxray into image array

I = imread('pout.tif') ⇒ read and store the image in an array named I.

whos Command :

whos Command is used to g information about variable in the workspace :

```
>> whos
```

Name	Size	Bytes	class	Attributes
I	291 x240	69840	uint8	—

size Command :

function size gives the row and Column dimensions of an image :

```
>> size(f);

ans =

     1024
```

```
>> [M, N] = size(f);
```

This syntax return the number of rows (M) and Column (N) in the image.

② Displaying Images

To display an image, the imshow function is used, which has the basic syntax

* imshow(f, G)

f - an image, G - the number of intensity levels used to display the image, if G is omitted, it defaults to 256 levels.

* imshow(f, [low high])

. displays as black all values less than or equal to low.

. displays as white all values greater than or equal to high.

. The values in between are displayed using the ~~default~~ default number of Levels.

* imshow (f, [ ])

sets variable Low to the minimum value of array f and high to its maximum value.

This form is useful for images with low dynamic range.

pixval command:

pixval is used to display the intensity values of individual pixel interactively.

Moving the cursor, the coordinates of the cursor position and the cooresponding intensity values are shown.

⊛ Color image: pixval displays Red, Green, blue components, when the image is color.

if the left button on the mouse is clicked and then held Pressed, pixval displays the Euclidean distance between the intial and current cursor location, The syntax is:

pixval.

Example①

```
f= imread ('rose_512.tif'); % read from disk an image
whos f % extract basic information about the image
imshow(f) % display the image.
```

if another image, g, is displayed using imshow, MatLab ~~display~~ replaces the image in the screen with the new image.

```
>> figure, imshow(g) % Keep the first image
```

and output a second image.

» imshow(f), figure, imshow(g) % display both images.

Example ②:

Suppose that we read an image h and find that using imshow(h) produces the image that has a low dynamic range, to correct:

imshow(h, []) % Improve the image h.

③ Writing Images:

Images are written to disk using function imwrite,

Syntax:

* imwrite(f, 'filename')

filename must include the extension of the file.

* imwrite(f, 'filename', 'extension').

examples:-

» imwrite(f, 'patient10_run1', 'tif') or alternatively

» imwrite(f, 'patient10_run1.tif')

* if filename contains no path information, then imwrite saves the file in the current working directory.

Special cases for imwrite Syntax:

A- JPEG images:

forimage with JPEG format, the syntax can have additional parameters:

imwrite(f, 'filename.jpg', 'quality', q)

q: is an integer between 0 and 100 (the lower the number the higher the degradation due to JPEG Compression).

example: to reduce storage and transmission time, it is important that the images be compressed as much as possible while not degrading their visual appearance beyond a reasonable level (false contouring, for example).

» imwrite(f, 'bubble25.jpg', 25) → 'quality'

» imwrite(f, 'bubble15.jpg', 15) → 'quality'

```
>> imwrite(f, 'bubble0.jpg', 'quality', 0)
>> imwrite(f, 'bubble5.jpg', 'quality', 5)
>> imwrite(f, 'bubble50.jpg', 'quality', 50)
>> imwrite(f, 'bubble.jpg')
```

q = 15 :⇒ false contouring visible.

q = 5 & q = 0 ⇒ quite visible

q = 25 ⇒ acceptable solution with some error.

* Some of the image/graphics formats supported by imread and imwrite, starting with Matlab 6.5,

TIFF, JPEG, GIF, BMP, PNG, XWD

* GIF is supported by imread, but not by imwrite.

Imfinfo function:

The imfinfo function returns information about the image in the file, such as size, width, height, the compression achieved and other image file details.

Syntax:

1) imfinfo filename

```
>> imfinfo bubble25.jpg
```

from the output of this command, we remark that:

Image

Filesize (in bytes) ⇒ the number of bytes in the original image is computed:

File size = (width * Height * Bit Depth) / 8 . ①
(original image).

Compression ratio = $\frac{\text{original image File size}}{\text{File size in imfinfo result}}$.

2) The information fields displayed by imfinfo can be captured into a structure variable that can be used for computations.

```
>> K = imfinfo('bubble25.jpg');
```

K - structure name; K - contains fields, that can be accessed by dot operator.

for example, the image height and width are stored in structure fields K.Heigh and K.width.

example Using structure:

Compute the compression ratio for bubble25.jpg:

```
» K = imfinfo ('bubble.jpg');
» image_bytes = K.Width * K.Heigh * K.BitDepth /8 ;
» Compressed_bytes = K.FileSize ;
» Compression_ratio = image_bytes/ Compressed_bytes
    Compression_ratio = 35.1612.
```

B- TIF images with imwrite

The general syntax (imwrite) applicable for tif image;

```
imwrite (g, 'filename.tif', 'compression', 'parameter', ...
        'resolution', [colres rowres])
```

↓ continue on new line.

Parameter can have one of the following values:

'none' – no compression,

'packbits' – packbits compression (the default for nonbinary image).

'ccitt' – ccitt compression (the default for binary image).

The 1x2 array [colres rowres] –(two integer), contains the Column resolution and row resolution in dots-per-Unit (the default values are [72 72]. for example

dpi – dots (pixels) per inch.

Colres – dpi in vertical direction, rowres – dpi in horizontal dir.

* specifying the resolution by a single scalar, res, is equivalent to writing [res res].

Example :

Suppose an 8-bit X-ray image of a circuit board generated during quality inspection. it is in jpg format, at 200 dpi, The image is of size 450 x 450 pixels, so its dimensions are 2.25 x 2.25 inches.

Q: store the image in tif format, with no compression, under the name sf, in addition we want to reduce the size of the image to 1.5 x 1.5 inches while keeping the pixel count at 450 x 450.

```
» res = round (200 * 2.25/1.5);
» imwrite (f, 'sf.tif', 'compression', 'none', 'resolution', res)
```

The the vector [colres rowres] were determined multiplying roo dpi by the ratio 2.25/1.5.

* round function rounds its argument to the nearest integen.