Introduction to Mathematical Image Processing
Image Processing Toolbox: Part 1
(image reading, display, and storage)

Suh-Yuh Yang (楊肅煜)

Department of Mathematics, National Central University
Jhongli District, Taoyuan City 32001, Taiwan

http://www.math.ncu.edu.tw/∼syyang/

Outline of MATLAB toolbox for image processing

In this series of lectures, we will give a brief introduction to the basic image processing operations in MATLAB, including several parts:

- **Part 1:** image reading, display, and storage back into the disk
- **Part 2:** image quality, histogram, and image sharpening
- **Part 3:** spatial filtering and image denoising
- **Part 4:** image deblurring and Hough transform

The material of these lectures are based on a series of four articles released by Dr. Anil Kumar Maini at

`https://www.electronicsforu.com/?s=matlab+image+processing`

Image processing (IP)

- IP is the technique to convert an image into digital format and perform operations on it to get an enhanced image or extract some useful information from it.
- IP is a multidisciplinary field. It overlaps with other areas such as *pattern recognition, machine learning, artificial intelligence, and human vision research.*
- Different steps involved in IP include importing the image with an optical scanner or from a digital camera, analyzing and manipulating the image *(e.g., data compression, image enhancement and filtering),* and generating the desired output image.
- The need to extract information from images and interpret their content has been the driving factor in the development of IP.

Applications of image processing

IP finds use in numerous sectors:

- *Medicine:* digital radiography (數位X線攝影), positron emission tomography (PET, 正子斷層掃描), computerized axial tomography (CAT, 電腦軸向斷層掃描), magnetic resonance imaging (MRI, 核磁共振成像), and functional magnetic resonance imaging (fMRI).
- *Industry:* safety systems, quality control and automated guided vehicle control.
- *Military:* detection of soldiers or vehicles, to missile guidance, and object recognition and reconnaissance.
- *Consumer electronics:* digital cameras and camcorders, high-definition TVs, monitors, DVD players, personal video recorders, and cell phones.

MATLAB toolbox for IP

- MATLAB: *(matrix laboratory)* is a platform for solving mathematical and scientific problems. It is a proprietary programming language developed by *MathWorks (Cleve Moler).*
- IP Toolbox is a collection of functions that extends the capability of the MATLAB numeric computing environment for image processing, analysis, visualization and algorithm development.
- IP Toolbox can be used to perform image segmentation, image enhancement, noise reduction, geometric transformations, image registration, and 3D image processing operations.

What is a digital image?

- An image may be defined as a two-dimensional real-valued function $f : \overline{\Omega} \to \mathbb{R}$.
- In an image f, the amplitude $f(x,y)$ is called the *intensity* of the image at point (x,y) and x and y are spatial coordinates. When x and y and amplitude values of f are all finite discrete quantities, the image is referred to as a *digital image.*
- Digitizing the coordinate values is referred to as *sampling,* while digitizing the amplitude values is called *quantization. The result of sampling and quantization is a matrix of real numbers.*
- A digitized image is represented as:

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

Each element in the array is referred to as a pixel or an image element.

Basic IP commands in MATLAB

- In MATLAB, a digital image is represented as:

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

- Images are read in MATLAB environment using `imread`.

  The `imread` function reads pixel values from an image file and returns a matrix of all pixel values.

grayscale and RGB images of penguins

Commands (functions) in MATLAB environment

- Read image named `filename.xxx` from the disk and then set as a variable (matrix) `G` in MATLAB:

  ```
  >>G=imread('filename.xxx');
  ```

- Display the variable `G` in an image form: `>>imshow(G)`
- Writes the variable `G` into the disk as an image in the new specified filename `newname.xxx` in the current directory:

  ```
  >>imwrite(G,'newname.xxx');
  ```

- Example: Read a RGB (color) image of penguins:

  ```
  >>G=imread('Penguins_RGB.jpg');
  ```

Examples

```
>>H=rgb2gray(G);
>>imwrite(H, 'Penguins_gray.jpg');
>>F=imread('Penguins_gray.jpg');
>>imshow(F)
```

$\leftarrow 0.2989 * R + 0.5870 * G + 0.1140 * B$

```
>>size(F)      ← or [m,n] = size(F)
ans =     554   636
>>size(G)     ← or [m,n,p] = size(G)
ans =     554   636   3
>>whos F
Name  Size     Bytes  Class  Attributes
F     554x636  352344  uint8
>>whos G
Name  Size       Bytes    Class  Attributes
G     554x636x3  1057032  uint8
```

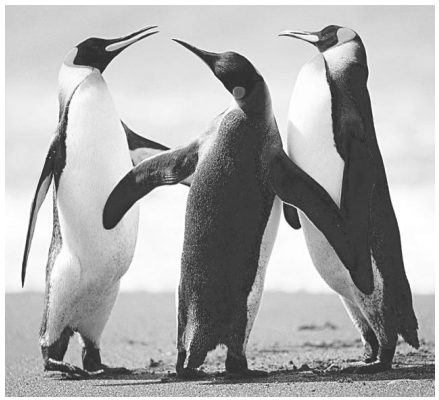
Image display

- Images are displayed on the MATLAB desktop using `imshow`.
- `imshow(F)`: `F` is an image array of data type uint8 or double.

  uint8: restricts the values of integers between 0 and 255.

  double: values between 0 and 1, 0 black and 1 white. Any value between 0 and 1 is displayed as grayscale.

  Any value greater than 1 is displayed as white, and a value less than zero is displayed as black.
- `imshow(F,[low high])`: displays as black if $F(m,n) \leq$ `low` and as white if $F(m,n) \geq$ `high`.



`imshow(F)`

`imshow(F, [50 100])`

Image display (continued)

The image tool in the IP toolbox provides a more interactive environment for viewing and navigating within images, displaying detailed information about pixel values, measuring distances and other useful operations.

```
>>F=imread('Penguins_gray.jpg');
>>imtool(F)
```

The status text at the bottom of the main window shows the column/row location and the value of the pixel lying under the mouse cursor.

Image display (continued)

Multiple images can be displayed within one figure using `subplot`.

```
>>A=imread('Penguins_gray.jpg');
>>B=imread('Penguins_RGB.jpg');
>>figure
>>subplot(1,2,1),imshow(A)
>>subplot(1,2,2),imshow(B)
```

Note: `subplot(3,2,3)` *tells* MATLAB *to divide the figure into three rows and two columns and set the third cell as active.*

Writing images

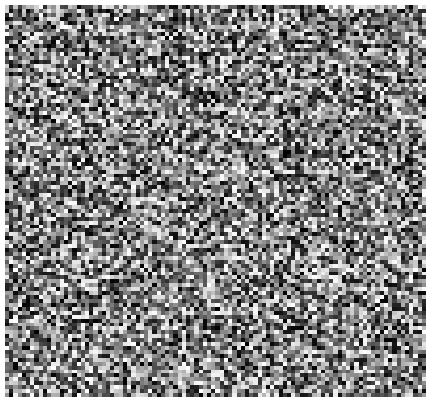
Images are written to the current directory using `imwrite`.

```
imwrite(f,'filename');
```

Writes image data `f` to the file `'filename'` in current folder.

Supports graphic file formats including `gif`, `hdf`, `jpeg`, `jpg`, `pbm`, `bmp`, `pgm`, `png`, `pnm`, `ppm`, `tif`, `tiff`, etc.

```
>>F=rand(100,100); % 100x100 matrix with values in [0,1]
>>imwrite(F,'random.png')
>>FF=imread('random.png');
>>imshow(FF)
```



Writing images (continued)

```
F=imread('Penguins_gray.jpg');

% save it in jpg format with quality parameter 10 (poor quality)
imwrite(F,'Penguins_gray_10.jpg','quality',10);

% save it in jpg format with quality parameter 75 (default)
imwrite(F,'Penguins_gray_75.jpg','quality',75);

% save it in jpg format with quality parameter 10 (high quality)
imwrite(F,'Penguins_gray_90.jpg','quality',90);
```

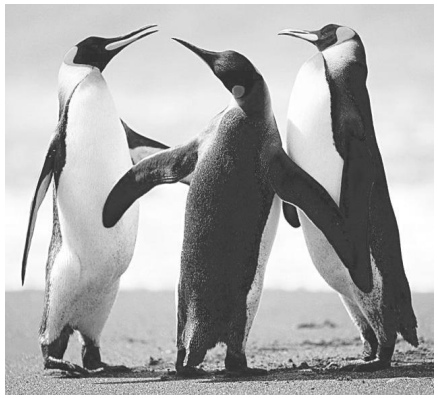
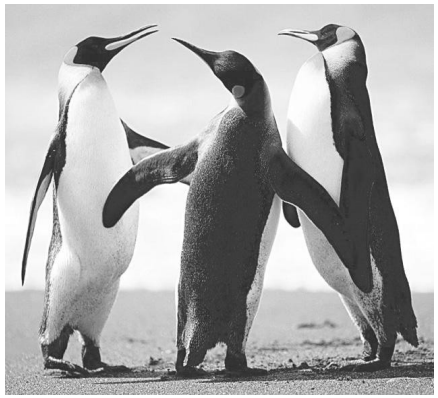




Image information

```
>>imfinfo('Penguins_gray_75.jpg')
ans=
    :
    FileSize:  49327
    Format:  'jpg'
    Width:  636
    Height:  554
    ColorType:  'grayscale'
    :
```

```
>>K=imfinfo('Penguins_gray_75.jpg');
>>[K.FileSize,K.Height,K.Width]
ans=
    49327  554  636
```