

An Introduction to Image Processing using Scilab

Ketan Kotwal
Department of Electrical Engineering
IIT Bombay
www.ee.iitb.ac.in/student/~ketan

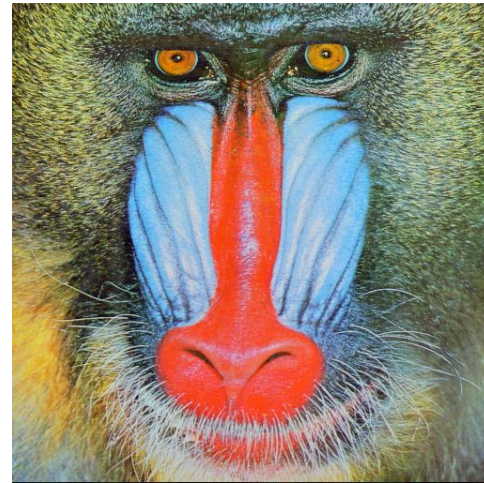
What are digital images?

- Image can be grayscale or color (RGB)
- Can be treated as a matrix
- Dimension of image (matrix):
 - $M \times N$ for grayscale
 - $M \times N \times 3$ for RGB. (Red, Green, Blue)

Image Examples



Grayscale image



Color (RGB) image

Reading and Display

- To read image: *imread(filename)*;
--> `imread('baboon.png') ;`
- To display image: *imshow(filename)*;
--> `I=imread('peppers.png') ;`
--> `imshow(I) ;`

Data conversion

- Data type: 8-bit unsigned integer (*uint8*)
(Sometimes 16-bit unsigned integer- *uint16*)
- Most of the operations require floating point data.
- Conversion functions:
 - *im2double, double*
 - *im2uint8, uint8*
 - *im2int8, int8*

Image conversion

- To convert RGB image into grayscale:

rgb2gray

```
-->rgb2gray(I);
```

- To convert image into binary: *im2bw*

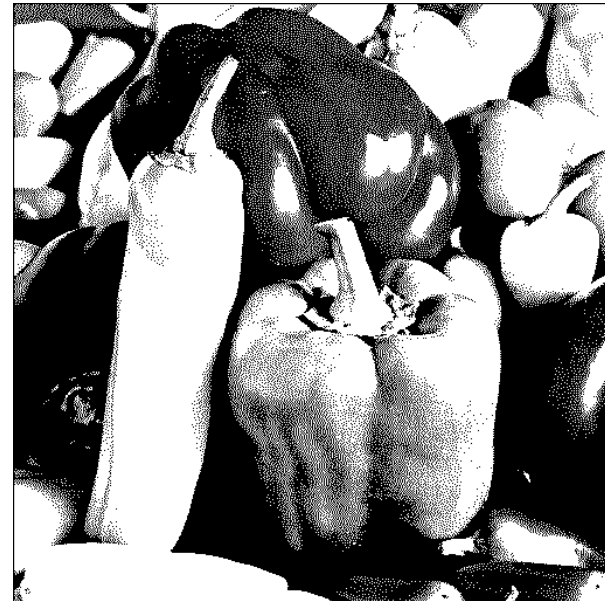
```
-->im2bw(I);
```

```
-->im2bw(I, 0.5);
```

Image Conversion



`rgb2gray(peppers)`



`im2bw(peppers)`

Color space conversion

- From RGB to HSV:

`-->rgb2hsv(I) ;`

- From RGB to YCbCr:

`-->rgb2ycbcr(I) ;`

- From RGB to YIQ:

`-->rgb2ntsc(I) ;`

- Similar conversions:

`hsv2rgb; ycbcr2rgb; ntsc2rgb;`

Image Conversion (RGB -> HSV)



Hue component



Saturation
component



Value component

Writing Images

- To write image into the disk:

imwrite(data, filename);

Filename should contain a valid extension.

```
-->imwrite(Z, 'test.png' );
```

- Supported file types:

- bmp, jpeg, jpg

- png, ppm

- tif, tiff

Basic Operations

- Cropping images:

```
-->cropped=imcrop(I,[20,20,100,100]);  
-->imshow(cropped);
```

- Resizing images:

```
-->resized=imresize(I,0.75);  
-->imshow(resized);
```

- Complementing images:

```
-->complemented=imcomplement(I);  
-->imshow(complemented);
```

Basic Operations



cropped



resized



complemented

Noise: Adding Noise

- Adding noise:

```
-->I=imread('lena.png');  
-->J=imnoise(I,'salt & pepper',  
0.02);
```

- Noise types:

- salt & pepper
- gaussian
- Speckle

Amplitude 1 is maximum image intensity.

Noise Addition



salt & pepper



gaussian



speckle

Spatial Domain Image Filtering

- Use of predefined filters:

$H = fspecial(type);$

- Type:

- average-[hsize]
- gaussian-[hsize, sigma]
- laplacian-[hsize]
- log-[hsize, sigma]
- prewitt-[horizontal]
- sobel-[horizontal]
- unsharp-[alpha]

Image Filtering

```
-->filtered = imfilter(I,H);
```



A: average
B: Gaussian
C: Laplacian
D: LoG
E: Prewitt
F: Unsharp

Edge Detection

- Detect edge map:

- > `BW_edge=edge(I, method) ;`

- Method:

- sobel-[thresh, dir]
 - prewitt- [thresh, dir]
 - log- [thresh, sigma]
 - fftderiv- [thresh, dir, sigma]
 - canny-[thesh, sigma]

Edge Detection



A: Sobel
B: Prewitt
C: LoG
E: Canny

Multi-Resolution Analysis

- Generate image pyramid:

```
-->I1=impyramid(I, method);  
-->I2=impyramid(I1, method);
```

- Method: expand or reduce

Image Compression

- Calculate SVD of the image I.

```
--> [u, s, v] = svd(I);  
--> s_val = diag(s);  
--> compressed = u(:, [1:n])  
* diag(s_val(1:n)) * v([1:n], :);  
--> imwrite(compressed, 'I_n.png');
```

- What is missing here?

Advanced Topics

- Hough transform
 - Detection of lines
- Fourier transform
 - Filtering in frequency domain

Recap

- Reading/ writing/ displaying image
- Basic image operations
- Noise removal
- Image filtering
- Edge detection

Thanks

- Scilab help:
--> `help command`

ketank@iitb.ac.in 😊