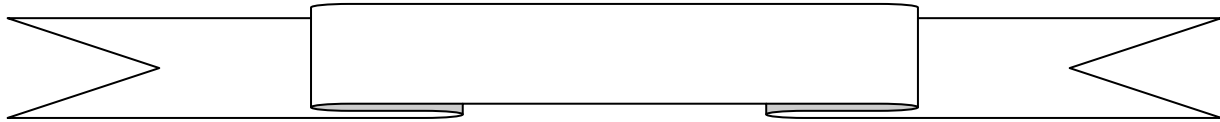




CHAPTER 2

Digital Image Fundamentals

Part 1



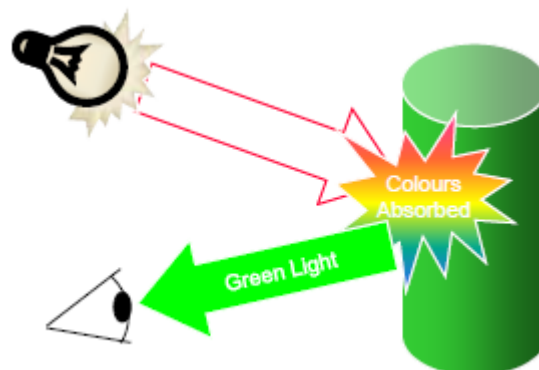
1. Theoretical

➤ Brightness Adaptation

- The ability of the eye to discriminate between changes in light intensity at any specific adaptation level.
- The eye's ability to discriminate between different intensity levels
- For example, when you enter a dark theater on a bright day, it takes an appreciable interval of time before you can see well enough to find an empty seat.

➤ How do we see colors?

- The colors that we perceive are determined by the nature of the light reflected from an object.
- For example, if white light is shone onto a green object most wavelengths are absorbed, while green light is reflected from the object.

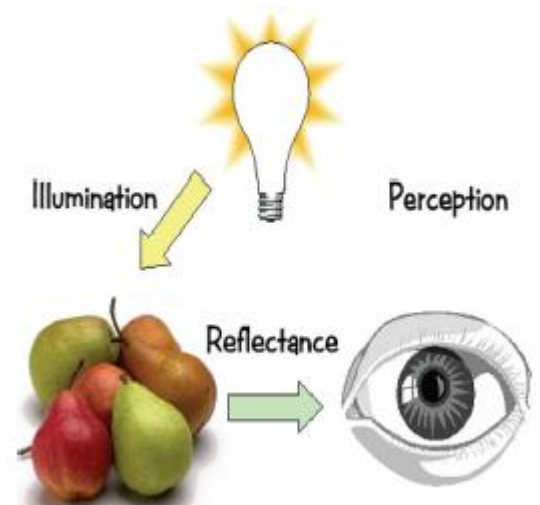


- **White** is the color which contains all the wavelengths of visible light without absorption, has maximum brightness.
- **Black** is the darkest color, the result of the absence of or complete absorption of light.
- **Green** objects reflect light with wavelengths primarily in the 500 to 570 nm range (green color range) while absorbing most of the energy at other wavelength

➤ A Simple Image Formation Model

$$f(x,y) = i(x,y)r(x,y)$$

- $f(x,y)$: intensity at the point (x,y)
- $i(x,y)$: illumination at the point (x,y) (determined by illumination Source)
 - $0 < i(x,y) < \infty$
- $r(x,y)$: reflectance (determined by imaged object)
 - $0 < r(x,y) < 1$
- In real situation:
 - $L_{\min} \leq L = f(x,y) \leq L_{\max}$
 - $L_{\min} = i_{\min} * r_{\min}$
 - $L_{\max} = i_{\max} * r_{\max}$
 - L : gray level



➤ Sampling & Quantization

- Digitalization of an analog signal involves two operations:
 - **Sampling**: Digitizing the x- and y-coordinates.
 - **Quantization**: Digitizing the amplitude values.

➤ Representing Digital Image

- Images can easily be represented as matrices.
- Pixel values are most often grey levels in the range 0-255(black-white).
- Discrete intensity interval $[0, L-1]$, $L=2^k$
- The number b of bits required to store a $M \times N$ digitized image

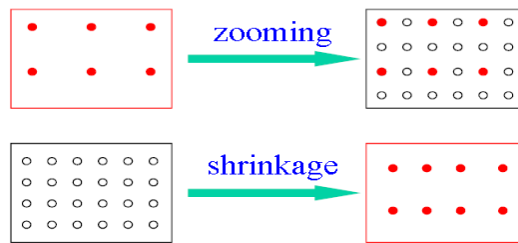
$$b = M \times N \times k$$

➤ Spatial and Intensity Resolution

✚ Spatial Resolution

- ✘ The spatial resolution of an image is determined by how sampling was carried out
- ✘ Spatial resolution simply refers to the smallest discernable detail in an image

- ✘ Changing the spatial resolution of a digital image, by zooming or shrinking. Simply, zooming and shrinking are the operations of oversampling (increase resolution) and undersampling (decrease resolution) a digital image, respectively.



- **Interpolation**

Process of using known data to estimate unknown values

- **Interpolation** (sometimes called resampling)

an imaging method to increase (or decrease) the number of pixels in a digital image.

- Zooming a digital image requires two steps:

1. The creation of new pixel locations,
2. And assignment of gray levels to those new locations.

The assignment of gray levels to the new pixel locations is an operation of great challenge. It can be performed using three approaches:

- 1) **Nearest Neighbor Interpolation (Pixel Replication)**: each pixel in the zoomed image is assigned the gray level value of its closest pixel in the original image.
- 2) **Bilinear Interpolation**: the value of each pixel in the zoomed image is a weighted average of the gray level values of the pixels in the nearest 2-by-2 neighborhood, in the original image.

$$V(x,y) = ax + by + cxy + d$$

- 3) **Bicubic Interpolation**: The intensity value assigned to point (x,y) is obtained by the following equation

$$v(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j$$

The sixteen coefficients are determined by using the sixteen nearest neighbors.

In MATLAB, this can be performed using the *imresize* function. This function accepts a factor greater than 1.0 for zooming and smaller than 1.0 for shrinking. Also, it accepts the required method of interpolation, either nearest neighbor or bilinear. Cubic interpolation is the default method.

$$imresize(A, SCALE, METHOD)$$

- Shrinking is done in the same manner as just described for zooming

✚ Intensity (gray level) resolution

- ✘ Gray-level resolution is the smallest discernible change in gray level.

2. Practical

Example1: Reducing the Spatial Resolution of an Image

```
% Reading the image and converting it to a gray-level image.
I=imread('Fig','jpg');
s=size(I); %256*256*3
I=rgb2gray(I);
s1=size(I); %256*256
% Reducing the Size of I using bilinear Interpolation
I128=imresize(I,0.5,'bilinear'); imshow(I128),pause
I64=imresize(I,0.25,'bilinear');close,imshow(I64),pause
I32=imresize(I,0.125,'bilinear');close,imshow(I32),pause
I16=imresize(I,0.0625,'bilinear');close,imshow(I16),pause

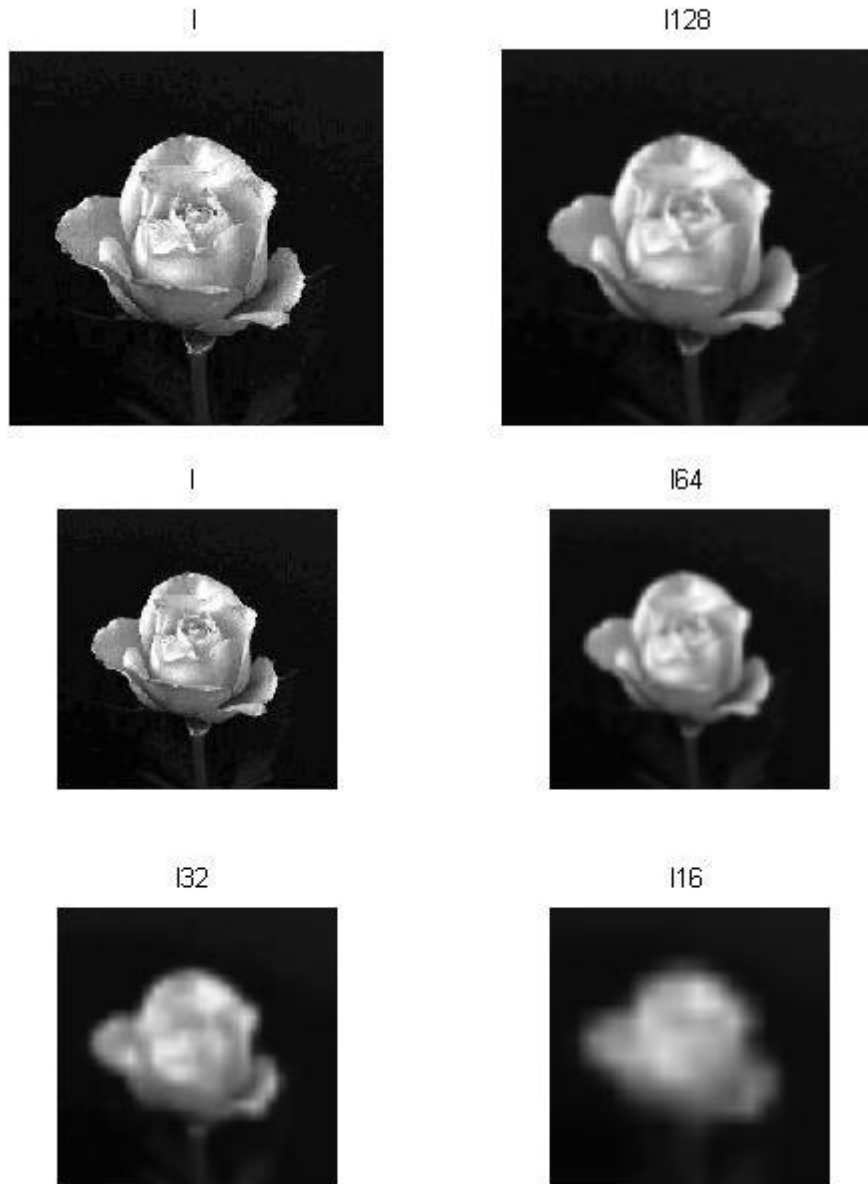
% Resizing the Reduced Images to the Original Size (256 X 256) and
Comapre
% them:
I16=imresize(I16,16,'bilinear');
I32=imresize(I32,8,'bilinear');
I64=imresize(I64,4,'bilinear');
I128=imresize(I128,2,'bilinear');

close
figure
subplot(121),imshow(I),title('I')
subplot(122),imshow(I128),title('I128')
pause,close

figure
subplot(221),imshow(I),title('I')
subplot(222),imshow(I64),title('I64')
subplot(223),imshow(I32),title('I32')
subplot(224),imshow(I16),title('I16')
pause,close
```

Output:





Example2: Reducing the Number of Gray Levels of an Image

```

% Reading the image and converting it to a gray-level image.
I=imread('Fig.jpg');
I=rgb2gray(I);

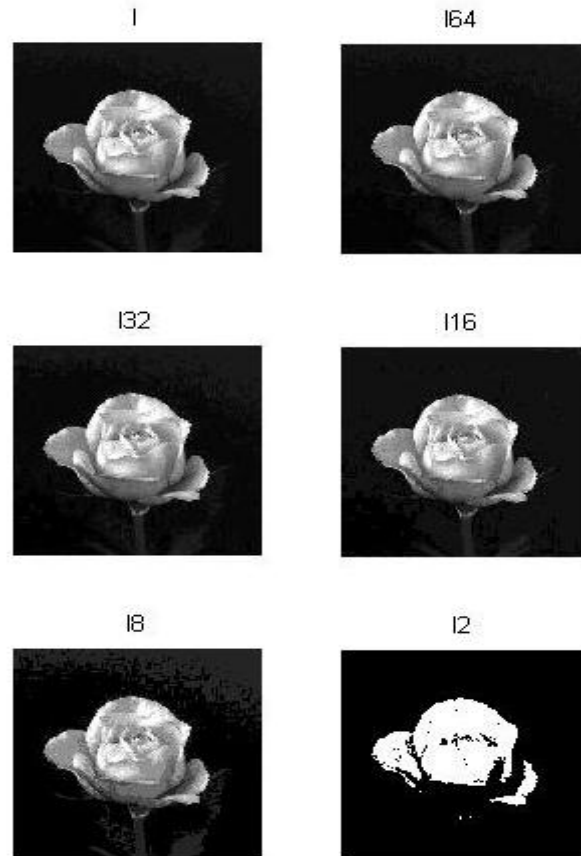
% A 64 gray-level image:
[I64,map64]=gray2ind(I,64);
% A 32 gray-level image:
[I32,map32]=gray2ind(I,32);
% A 16 gray-level image:
[I16,map16]=gray2ind(I,16);
% A 8 gray-level image:
[I8,map8]=gray2ind(I,8);
% A 2 gray-level image:
[I2,map2]=gray2ind(I,2);

```

```

figure(1)
subplot(321),subimage(I),title('I'),axis off
subplot(322),subimage(I64,map64),title('I64'),axis off
subplot(323),subimage(I32,map32),title('I32'),axis off
subplot(324),subimage(I16,map16),title('I16'),axis off
subplot(325),subimage(I8,map8),title('I8'),axis off
subplot(326),subimage(I2,map2),title('I2'),axis off

```



3. Homework:

1. Write a MATLAB code like example1 to reduce the size of image using nearest Interpolation. Which is better in quality of image, nearest or bilinear interpolation?
2. Write a MATLAB code that reads a gray scale image and generates the vertically flipped image of original image. (Typically like the following result)

