Image Analysis

Inel 5046
Prof. Vidya Manian

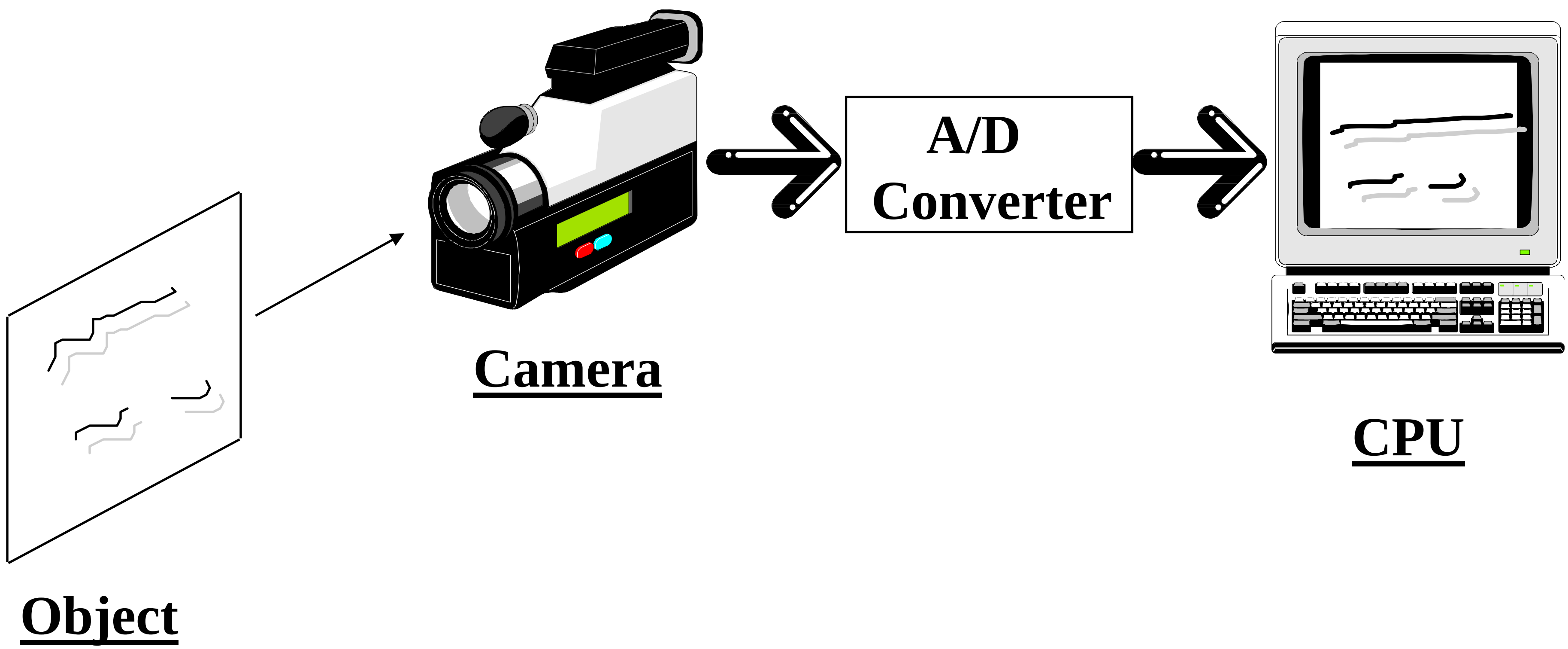
What is an image?

References:

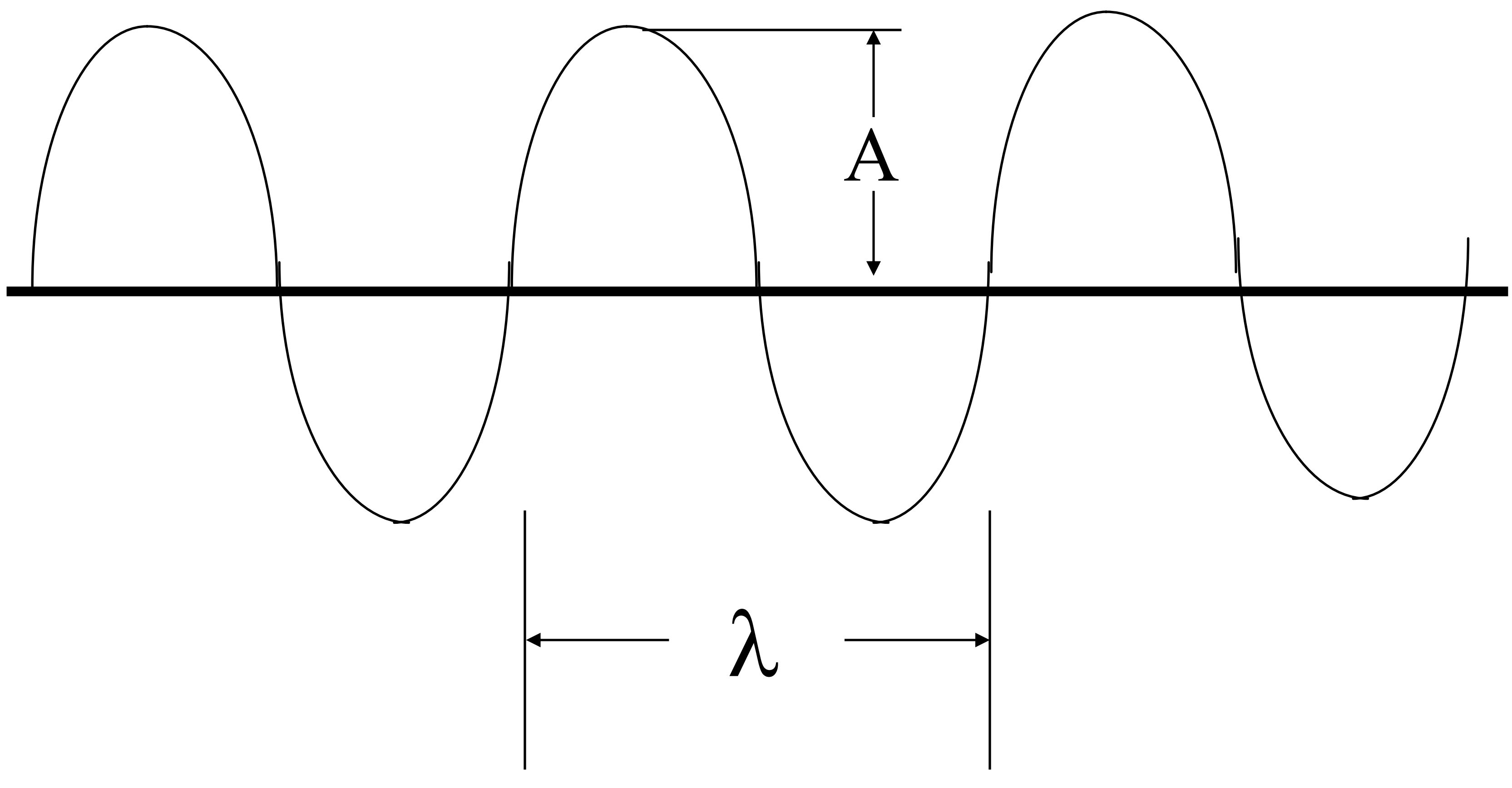
Pattern Recognition and Image Analysis, Gose, Johnsonbaugh, Jost.
Digital Image Processing, R. C. Gonzalez, R. E. Woodsd.

Matlab: Image Processing Toolbox.

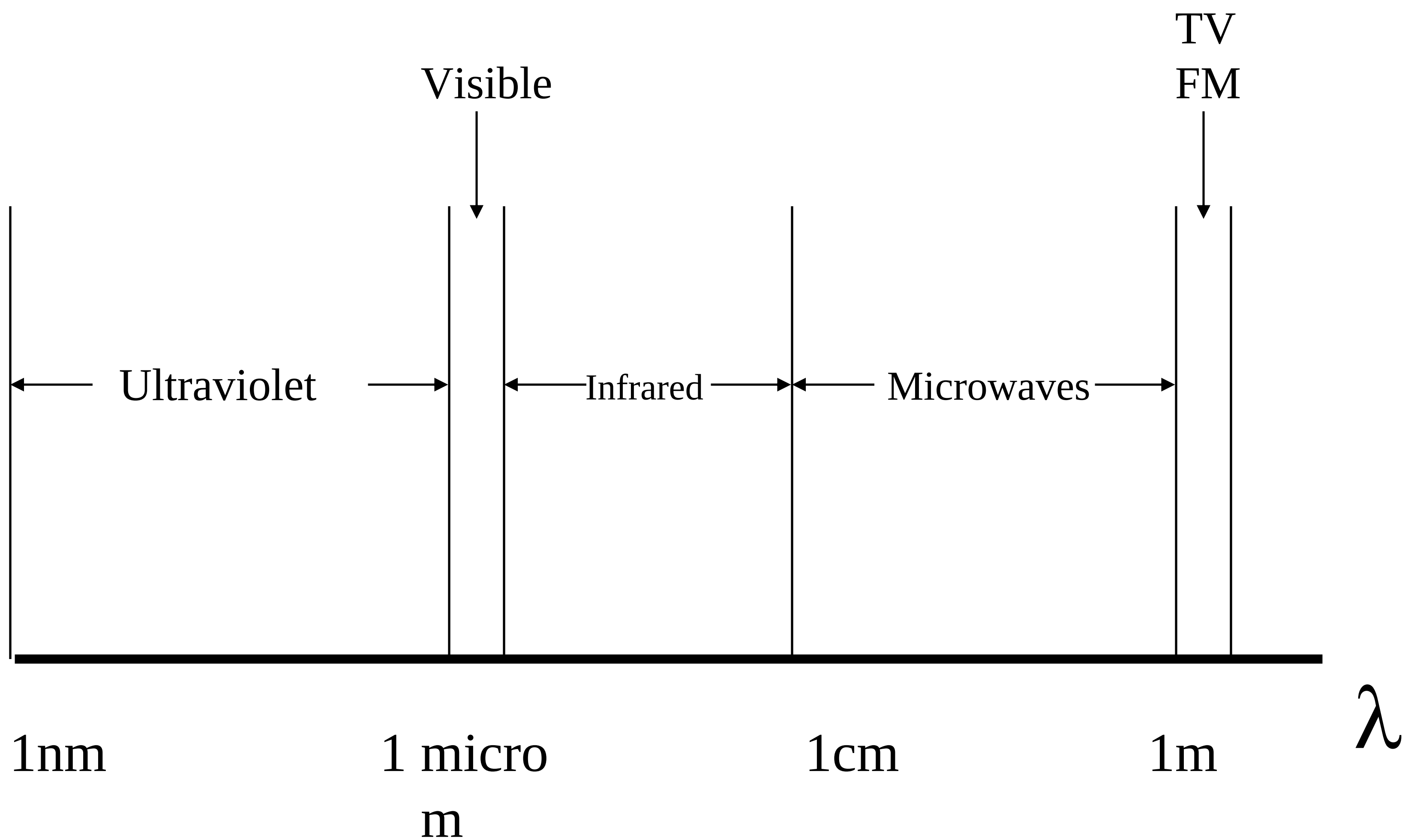
Computer Vision



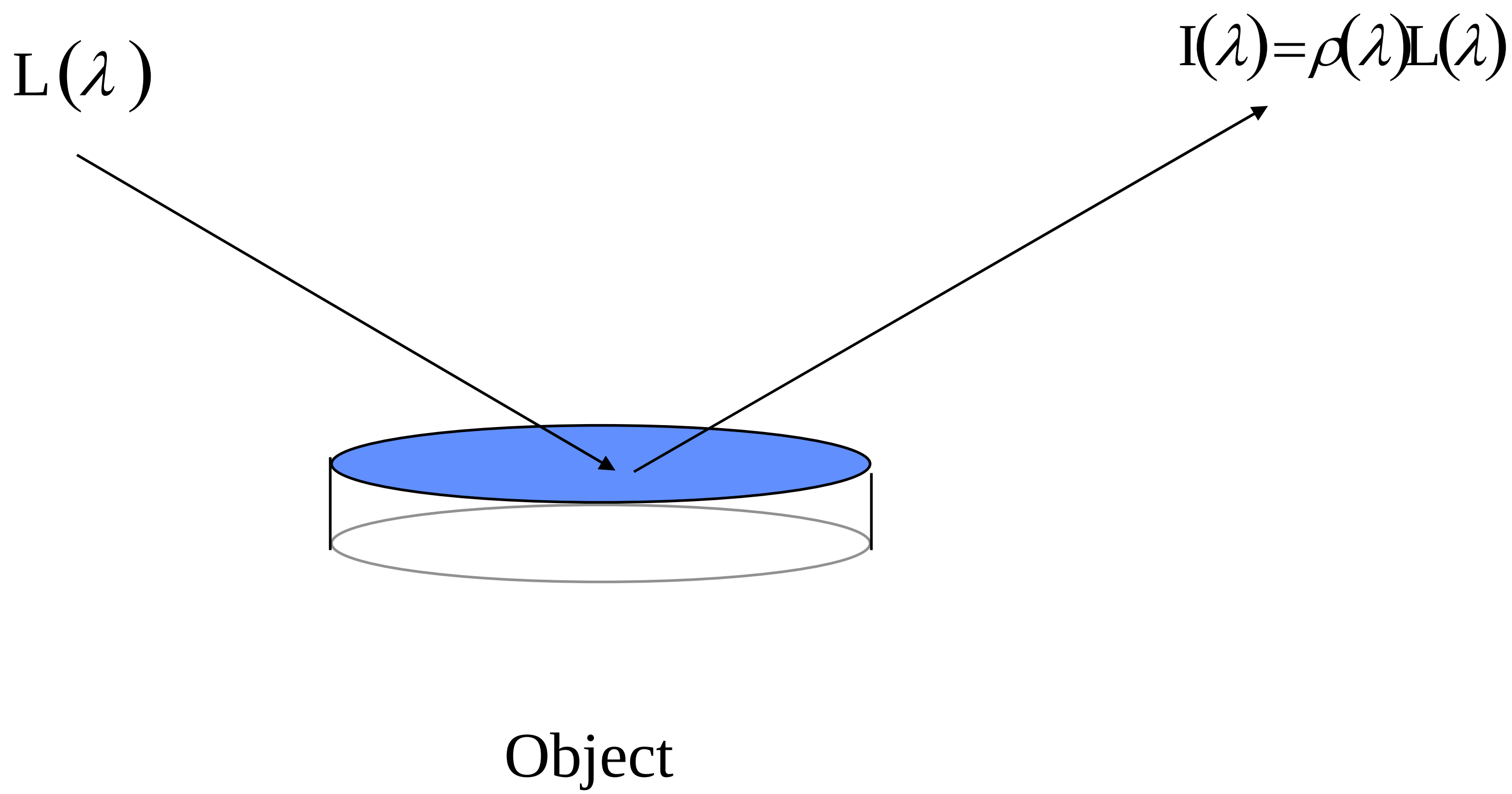
Electromagnetic Wavelength (Light)



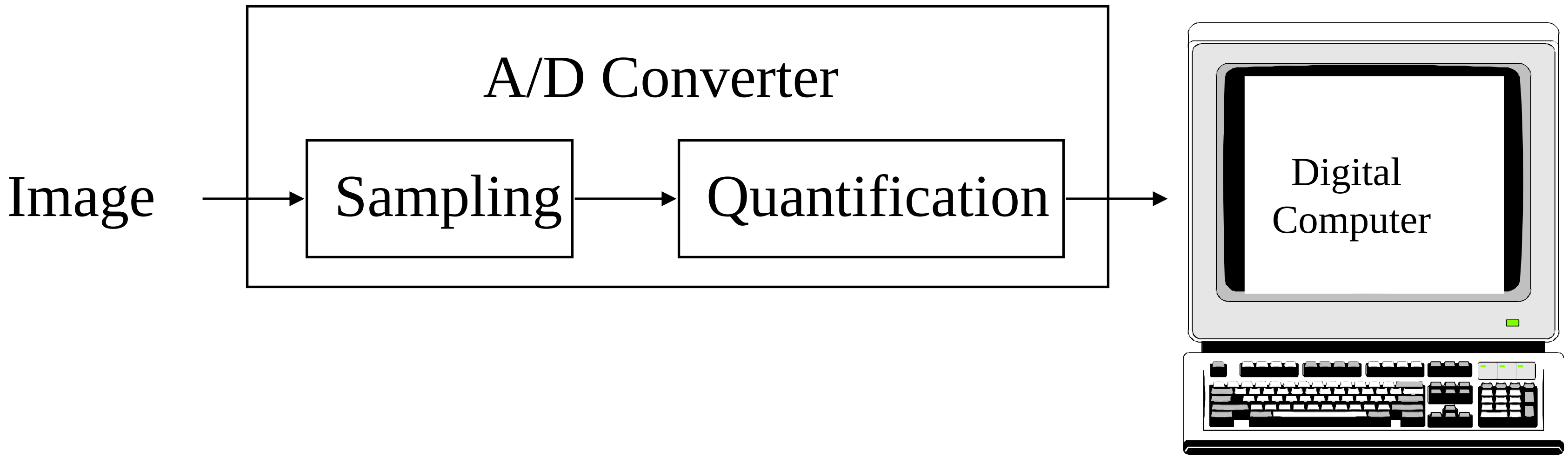
Electromagnetic Spectrum



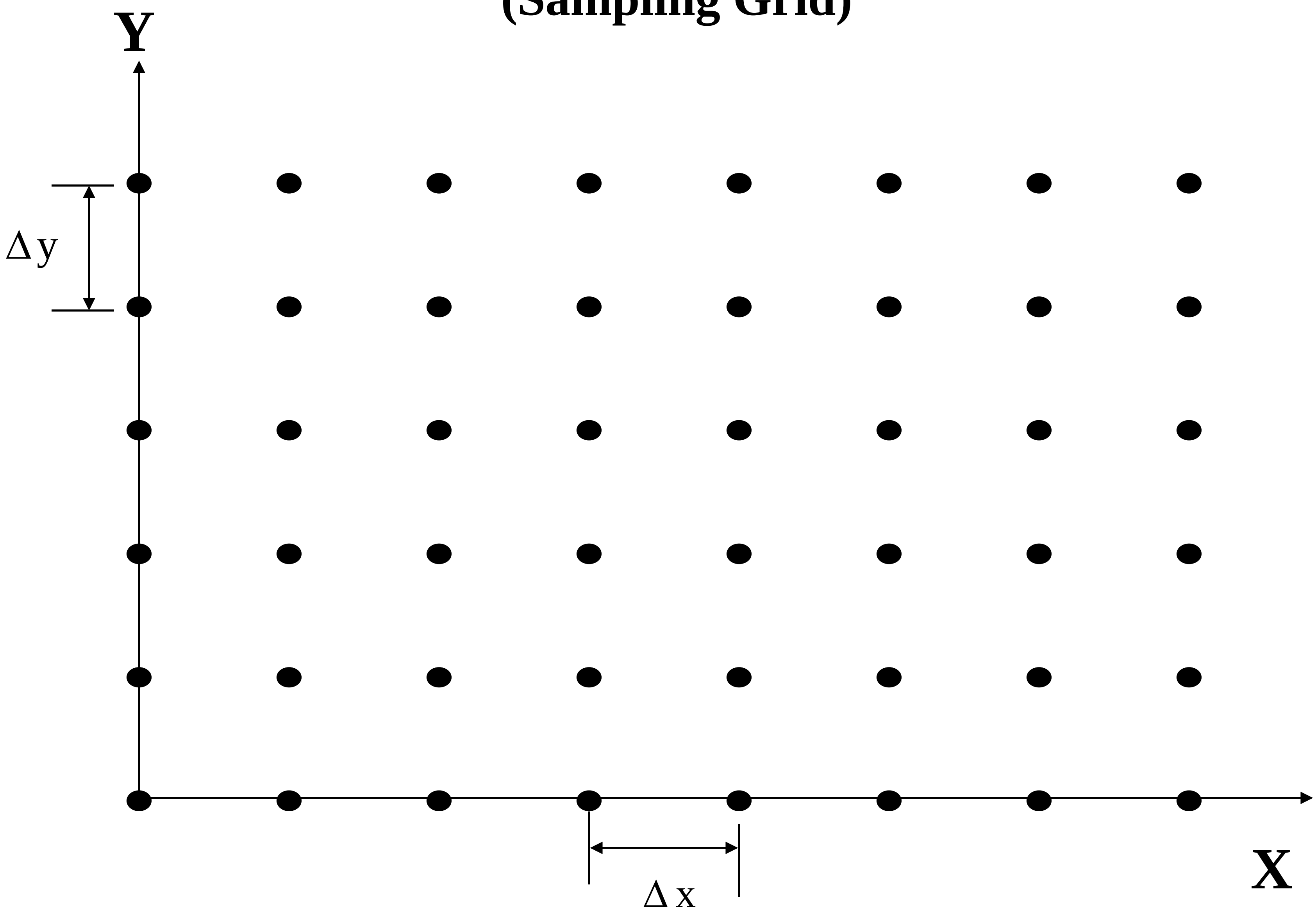
Illumination and Reflection



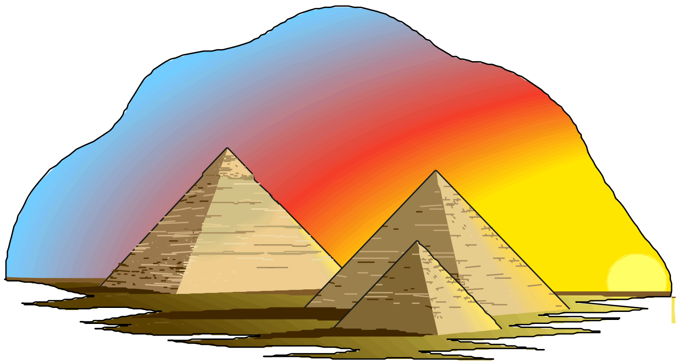
Digitazation of an Image



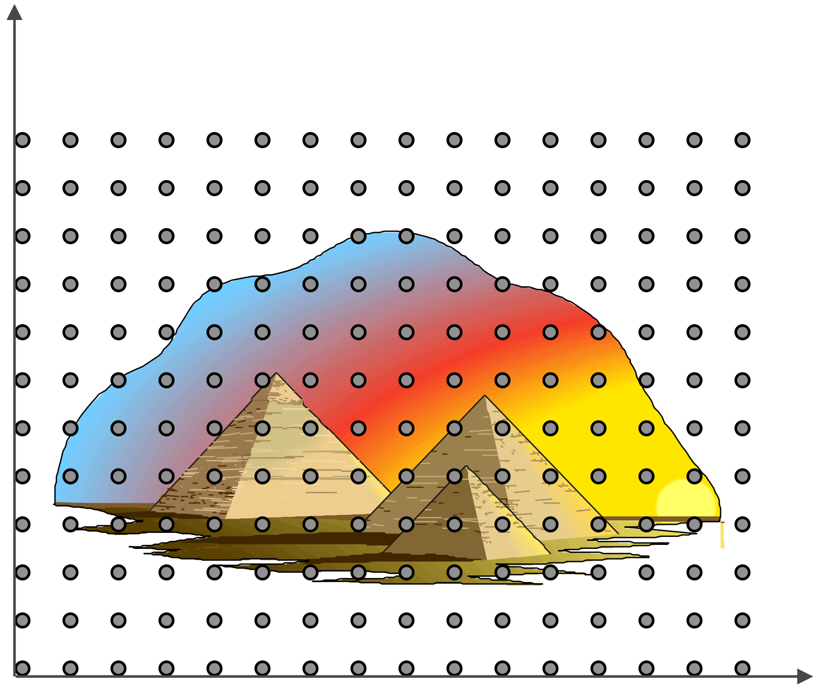
Sampling
(Sampling Grid)



Sampling

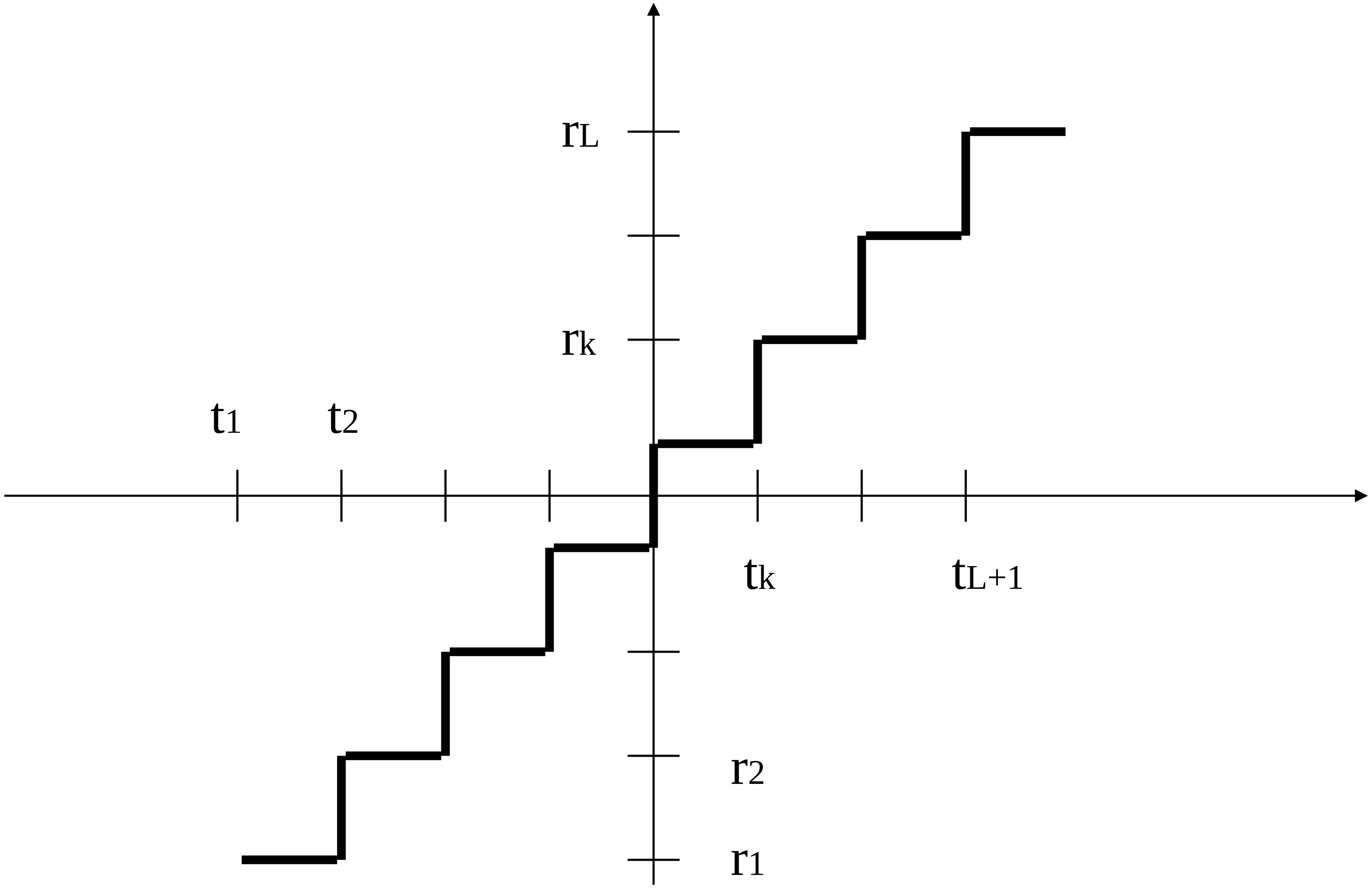


Analog Image



Samples of the Image

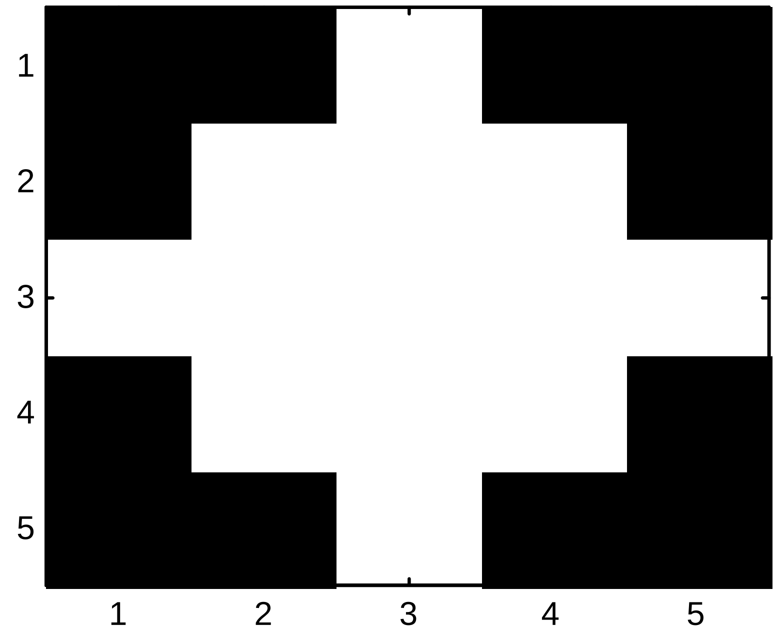
Quantizer



Black and White Image

BW256 =

0	0	256	0	0
0	256	256	256	0
256	256	256	256	256
0	256	256	256	0
0	0	256	0	0

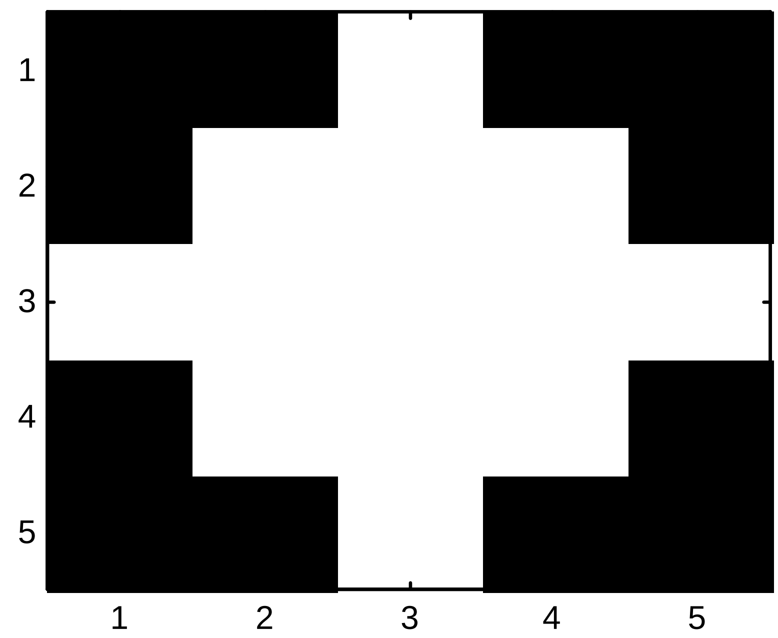



```
imagesc(BW256)
colormap(gray(256))
```

Black and White Image

BW =

0	0	1	0	0
0	1	1	1	0
1	1	1	1	1
0	1	1	1	0
0	0	1	0	0

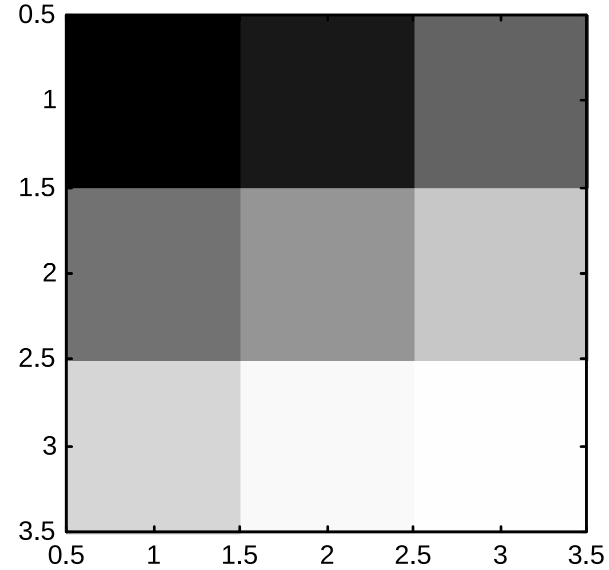



```
imagesc(BW)
colormap(gray(256))
```

Image Grayscale
0-255

X =

0	25	100
115	150	200
215	250	255




```
n = 3
m = 3
figure('Position', [100 100 n m])
image(X)
colormap(gray(256))
```

```
imagesc(X)
colormap(gray(256))
```

Gray Scale Image

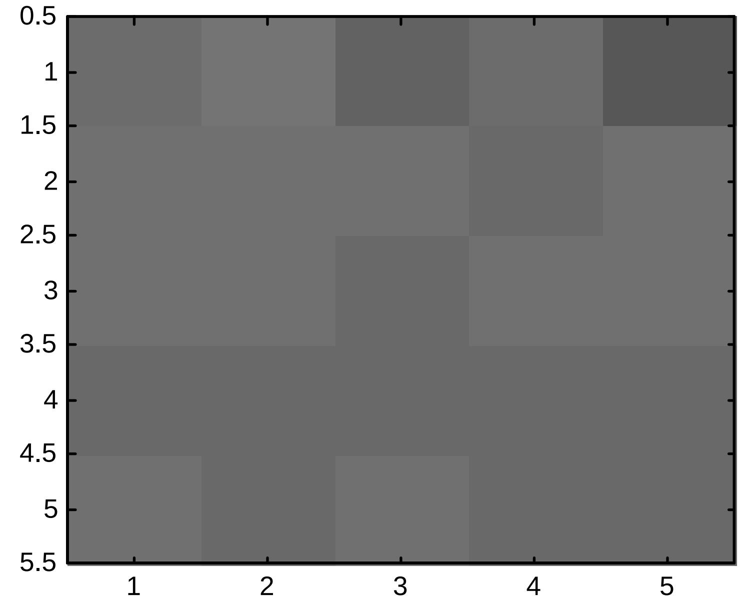
load trees

Image (indexed Image)

First 5 columns and rows

Xsub =

109	117	99	109	88
113	113	113	106	113
113	113	106	113	113
106	106	106	106	106
113	106	113	106	106




```
n = 5
m = 5
figure('Position', [100 100 n m])
image(Xsub)
colormap(gray(256))
```

```
imagesc(X)
colormap(gray(256))
```

Visualization in MATLAB

```
load trees
I = ind2gray(X,map);
imshow(I)
```

What is I? (Intensity Image)

Isub =

0.7232	0.8245	0.6599	0.7232	0.6003
0.7745	0.7745	0.7745	0.7025	0.7745
0.7745	0.7745	0.7025	0.7745	0.7745
0.7025	0.7025	0.7025	0.7025	0.7025
0.7745	0.7025	0.7745	0.7025	0.7025

Matlab Image classes

- Double- floating number ("a number with decimals") between 0 (black) and 1(white) to each pixel
- Uint8- assigns an integer between 0 and 255
- requires roughly 1/8 of the storage compared to the class double
- Indexed image – 2 matrices, first matrix has the same size as the image and one number for each pixel. second matrix is called the *color map* and its size may be different from the image.
- The numbers in the first matrix is an instruction of what number to use in the color map matrix.

Image format conversions

Operation:	**Matlab command:**
Convert between intensity/indexed/RGB format to binary format.	dither()
Convert between intensity format to indexed format.	gray2ind()
Convert between indexed format to intensity format.	ind2gray()
Convert between indexed format to RGB format.	ind2rgb()
Convert a regular matrix to intensity format by scaling.	mat2gray()
Convert between RGB format to intensity format.	rgb2gray()
Convert between RGB format to indexed format.	rgb2ind()

I=im2double(I); converts an image named I from uint8 to double.
I=im2uint8(I); converts an image named I from double to uint8.

Reading and writing images	**Matlab command**
Read an image (‘filename')	imread()
Write an image to a file (‘filename’,format)	imwrite(,)

Fundamentals of Image Processing

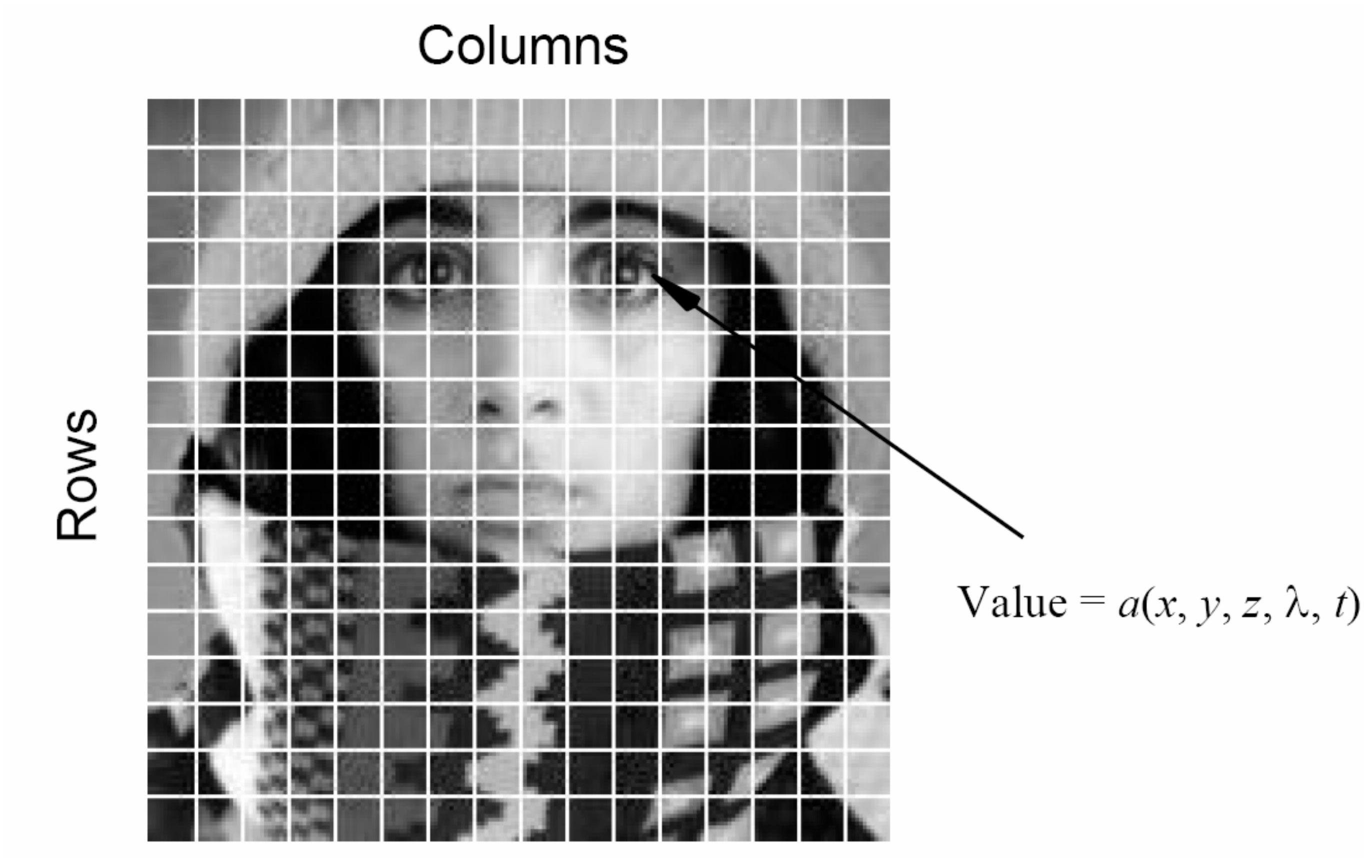
- Image Processing — *image in → image out*
- Image Analysis — *image in → measurements out*
- Image Understanding — *image in → high-level description out*

Parameter	*Symbol*	*Typical values*
Rows	N	256,512,525,625,1024,1035
Columns	M	256,512,768,1024,1320
Gray Levels	L	2,64,256,1024,4096,16384

Table 1: Common values of digital image parameters

Grey levels, $L = 2^B$, where B is the number of bits in the Binary representation of the brightness levels. B>1 grey level Image, B=1 then binary image

Value of pixel at Row=3, column=10 is 110

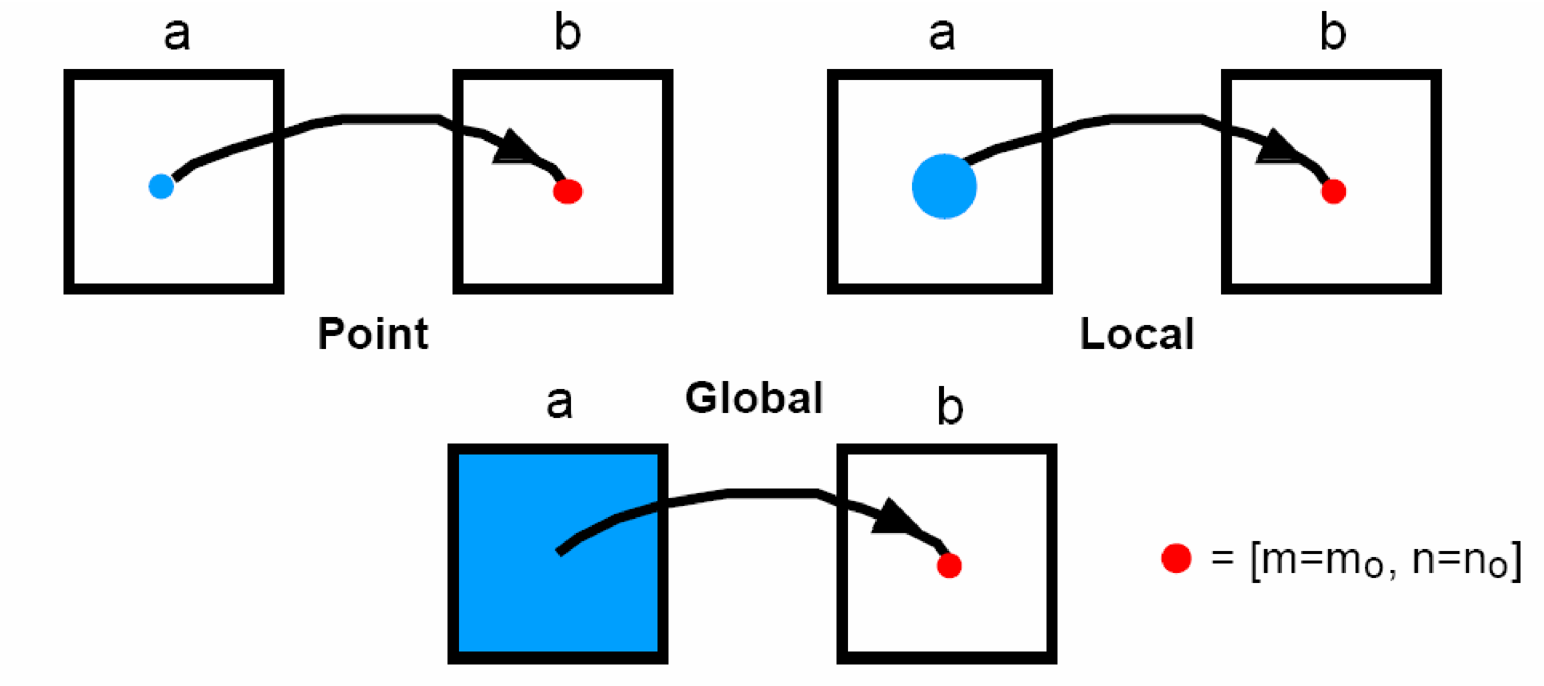


Characteristics of Image Operations

Operation	Characterization	Generic Complexity/Pixel
• *Point*	– the output value at a specific coordinate is dependent only on the input value at that same coordinate.	*constant*
• *Local*	– the output value at a specific coordinate is dependent on the input values in the *neighborhood* of that same coordinate.	P^2
• *Global*	– the output value at a specific coordinate is dependent on all the values in the input image.	N^2

Table 2: Types of image operations. Image size = $N \times N$; neighborhood size = $P \times P$. Note that the complexity is specified in operations *per pixel*.

Illustrutation of various types of image operations



Types of neighborhoods

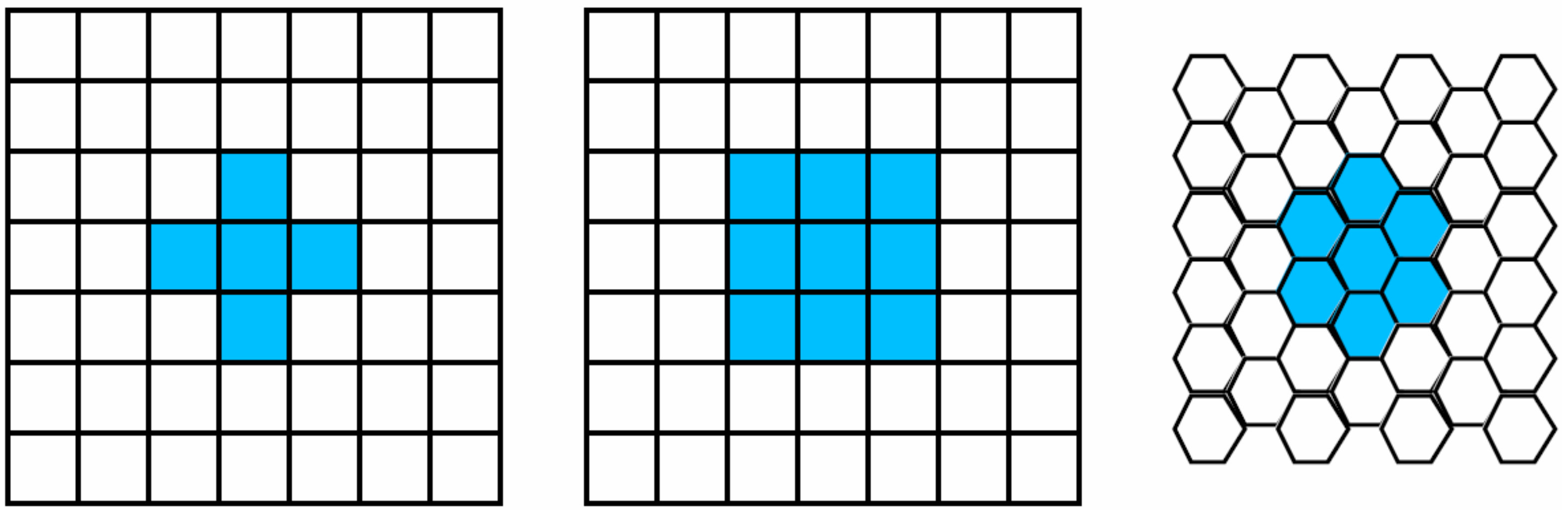


Figure 3a
Rectangular sampling
4-connected

Figure 3b
Rectangular sampling
8-connected

Figure 3c
Hexagonal sampling
6-connected

-restrict to rectangular sampling, due to hardware/software considerations

Tools-convolution

$$c = a \otimes b = a * b$$

In 2D continuous space:

$$c(x,y) = a(x,y) \otimes b(x,y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} a(\chi,\zeta) b(x-\chi, y-\zeta) d\chi d\zeta$$

In 2D discrete space:

$$c[m,n] = a[m,n] \otimes b[m,n] = \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} a[j,k] b[m-j, n-k]$$

Properties of convolution: commutative, associative and distributive

Tools-Fourier transform

•FT represents 2D signal as a weighted sum of sines and cosines.

$$e^{jq} = \cos(q) + j\sin(q)$$

Forward FT $$A = F\{a\}$$

Inverse FT $$a = F^{-1}\{A\}$$

Fourier transform is a unique and invertible operations

$$a = F^{-1}\{F\{a\}\} \quad and \quad A = F\{F^{-1}\{A\}\}$$

In 2D discrete space:

Forward –

$$A(\Omega,\Psi) = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} a[m,n] e^{-j(\Omega m + \Psi n)}$$

Inverse –

$$a[m,n] = \frac{1}{4\pi^2} \int_{-\pi}^{+\pi} \int_{-\pi}^{+\pi} A(\Omega,\Psi) e^{+j(\Omega m + \Psi n)} d\Omega d\Psi$$

FT can be written in terms of magnitude and phase

$$a[m,n] = |a[m,n]| e^{j\vartheta[m,n]}$$

- If a 2D signal is real, then the Fourier transform has certain symmetries.

$$A(u,v) = A^*(-u,-v) \qquad A(\Omega,\Psi) = A^*(-\Omega,-\Psi)$$

• If a 2D signal is real and even, then the Fourier transform is real and even.

$$A(u,v) = A(-u,-v) \qquad\qquad A(\Omega,\Psi) = A(-\Omega,-\Psi)$$

• The Fourier and the inverse Fourier transforms are linear operations.

$$F\{w_1 a + w_2 b\} = F\{w_1 a\} + F\{w_2 b\} = w_1 A + w_2 B$$

$$F^{-1}\{w_1 A + w_2 B\} = F^{-1}\{w_1 A\} + F^{-1}\{w_2 B\} = w_1 a + w_2 b$$

• FT is periodic, with period 2π

$$A(\Omega + 2\pi j, \Psi + 2\pi k) = A(\Omega,\Psi) \qquad j,k \text{ integers}$$

• Convolution in the spatial domain is equivalent to multiplication in the frequency domain

$$c = a \otimes b \quad \overset{F}{\leftrightarrow} \quad C = A \bullet B$$

$$c = a \bullet b \quad \overset{F}{\leftrightarrow} \quad C = \frac{1}{4\pi^2} A \otimes B$$

• If a two-dimensional signal $a(x,y)$ is scaled in its spatial coordinates then:

$$\begin{aligned} &If \quad a(x,y) \rightarrow a\left(M_x \bullet x, M_y \bullet y\right) \\ &Then \quad A(u,v) \rightarrow A\left(u/M_x, v/M_y \right) \Big/ \left| M_x \bullet M_y \right| \end{aligned}$$

• If a two-dimensional signal $a(x,y)$ has Fourier spectrum $A(u,v)$ then:

$$A(u=0, v=0) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} a(x,y)\,dxdy$$

$$a(x=0, y=0) = \frac{1}{4\pi^2}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} A(u,v)\,dxdy$$

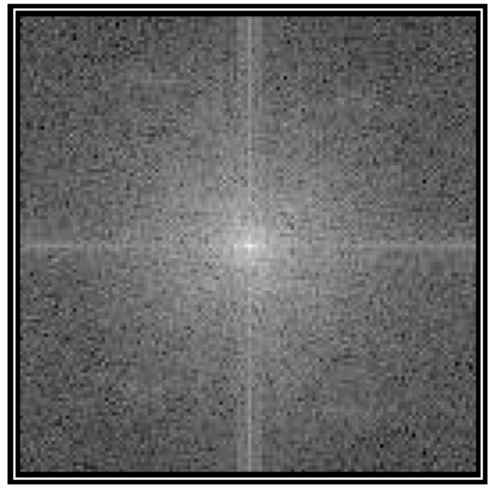
• If a two-dimensional signal $a(x,y)$ has Fourier spectrum $A(u,v)$ then:

$$\frac{\partial a(x,y)}{\partial x} \overset{F}{\leftrightarrow} juA(u,v) \qquad \frac{\partial a(x,y)}{\partial y} \overset{F}{\leftrightarrow} jvA(u,v)$$

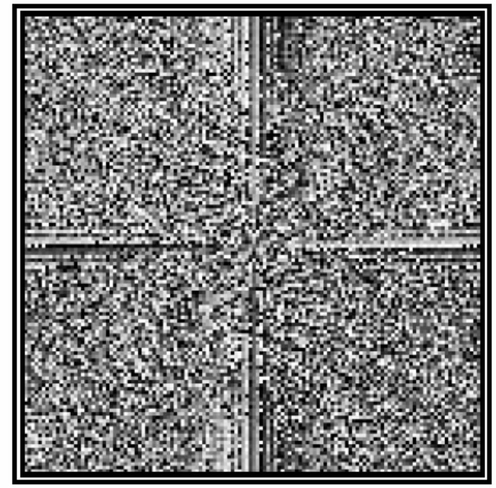
$$\frac{\partial^2 a(x,y)}{\partial x^2} \overset{F}{\leftrightarrow} -u^2 A(u,v) \qquad \frac{\partial^2 a(x,y)}{\partial y^2} \overset{F}{\leftrightarrow} -v^2 A(u,v)$$

Importance of magnitude and phase

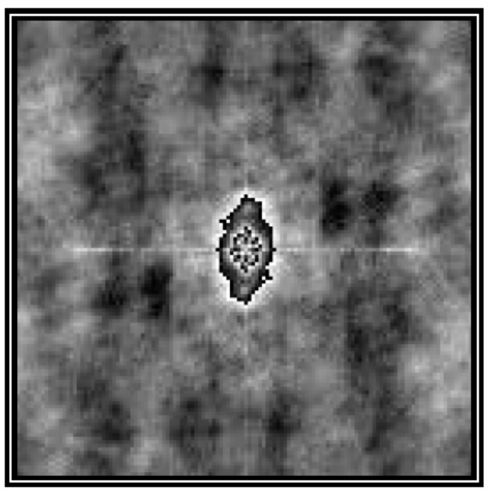
Original



$\log(|A(\Omega,\Psi)|)$



$\varphi(\Omega,\Psi)$



$\varphi(\Omega,\Psi) = 0$

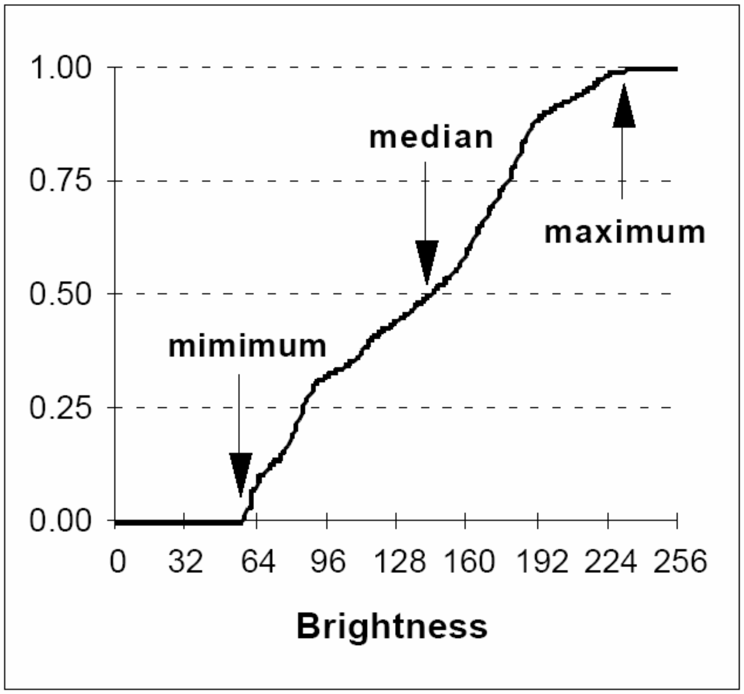
$|A(\Omega,\Psi)| = constant$

Image statistics

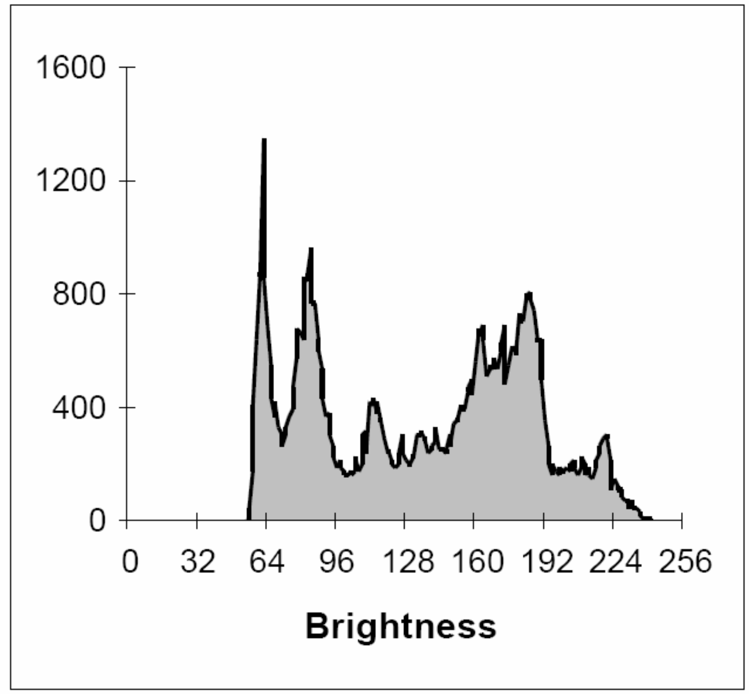
Statistic	*Image*	*ROI*
Average	137.7	219.3
Standard Deviation	49.5	4.0
Minimum	56	202
Median	141	220
Maximum	241	226
Mode	62	220
SNR (db)	*NA*	33.3

Region is the interior of the circle

Algorithms
Histogram based operations



Brightness distribution function



Brightness histogram

Contrast stretching

$$b[m,n] = \left(2^B - 1\right) \bullet \frac{a[m,n] - \text{minimum}}{\text{maximum} - \text{minimum}}$$

Sensitive to outliers

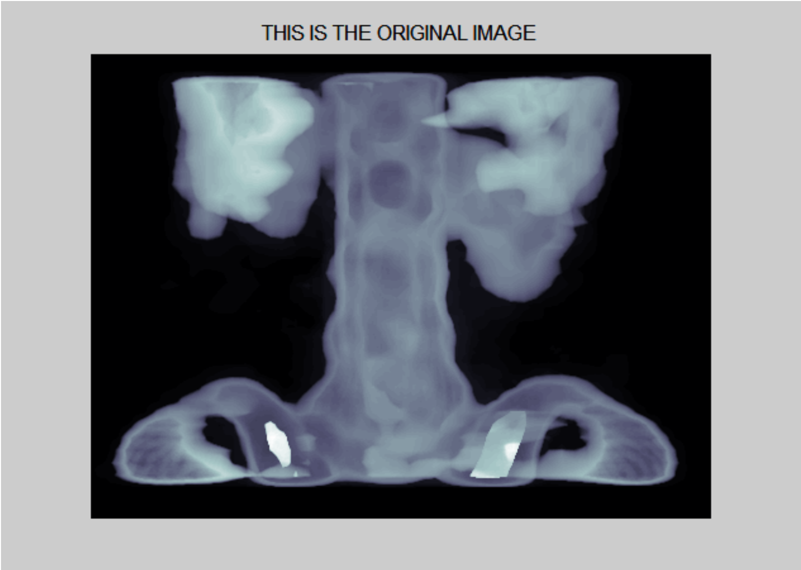
$$b[m,n] = \begin{cases} 0 & a[m,n] \le p_{\text{low}}\% \\ \left(2^B - 1\right) \bullet \dfrac{a[m,n] - p_{\text{low}}\%}{p_{\text{high}}\% - p_{\text{low}}\%} & p_{\text{low}}\% < a[m,n] < p_{\text{high}}\% \\ \left(2^B - 1\right) & a[m,n] \ge p_{\text{high}}\% \end{cases}$$

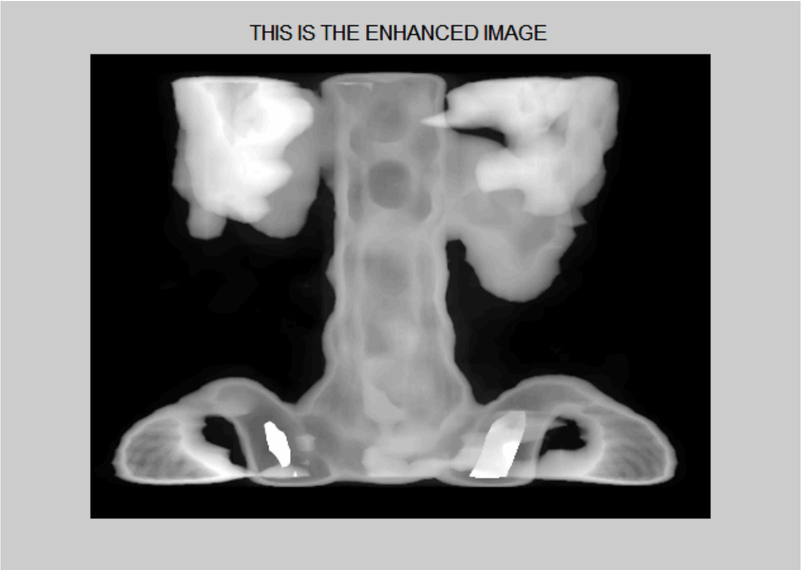
Instead of 0% and 100% use Plow=1% and Phigh=99%

In the above suppress 2^B-1 and

normalize the brightness range to $0 \le b[m,n] \le 1$

Adjust contrast to the optimum level






```
load spine;
img=X;
[m1 n1 r1]=size(img);
img2=double(img);
%calculation of vmin and vmax-
for(k=1:r1)
arr=sort(reshape(img2(:,:,k),m1*n1,1));
vmin(k)=arr(ceil(0.008*m1*n1));
vmax(k)=arr(ceil(0.992*m1*n1));
end
```

```
v_min=vmin;
   v_max=vmax;
for(i=1:m1)
    for(j=1:n1)
       for(k=1:r1)
            img2(i,j,k)=255*(img2(i,j,k)-v_min(1))/(v_max(1)-
v_min(1));
       end
    end
end
%-------------------
img2=uint8(img2);
figure,imshow(img),title('THIS IS THE ORIGINAL IMAGE');
figure,imshow(img2),title('THIS IS THE ENHANCED IMAGE');
```

Mathematics based operations

- Binary operations

NOT	$c = \overline{a}$
OR	$c = a + b$
AND	$c = a \bullet b$
XOR	$c = a \oplus b = a \bullet \overline{b} + \overline{a} \bullet b$
SUB	$c = a \setminus b = a - b = a \bullet \overline{b}$

Matlab >>im2bw %converts image to binary

NOT	
a	
0	1
1	0

↑ input ↑ output

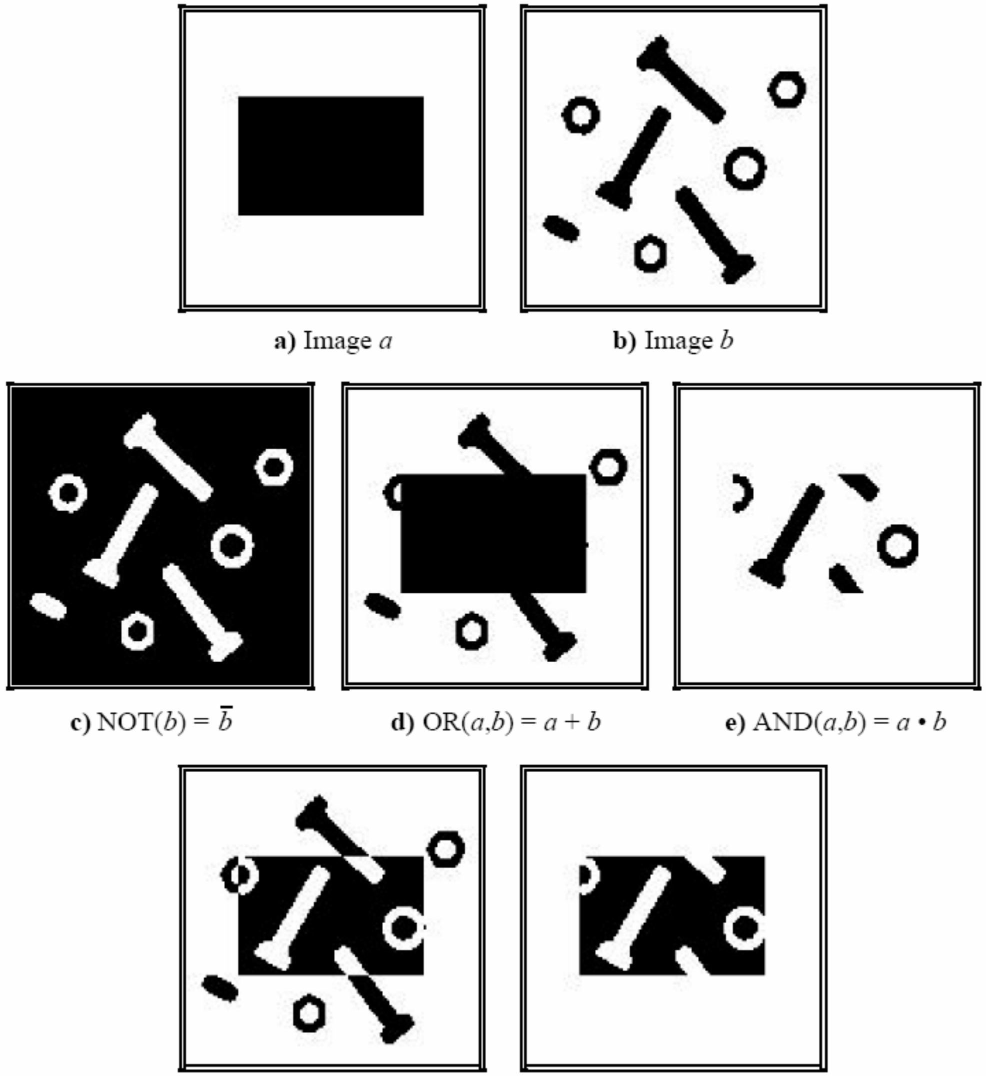
OR	**b**	
a	0	1
0	0	1
1	1	1

AND	**b**	
a	0	1
0	0	0
1	0	1

XOR	**b**	
a	0	1
0	0	1
1	1	0

SUB	**b**	
a	0	1
0	0	0
1	1	0

Examples of various binary operations



a) Image *a*

b) Image *b*

c) NOT(b) = $\bar{b}$

d) OR(a,b) = $a + b$

e) AND(a,b) = $a \bullet b$

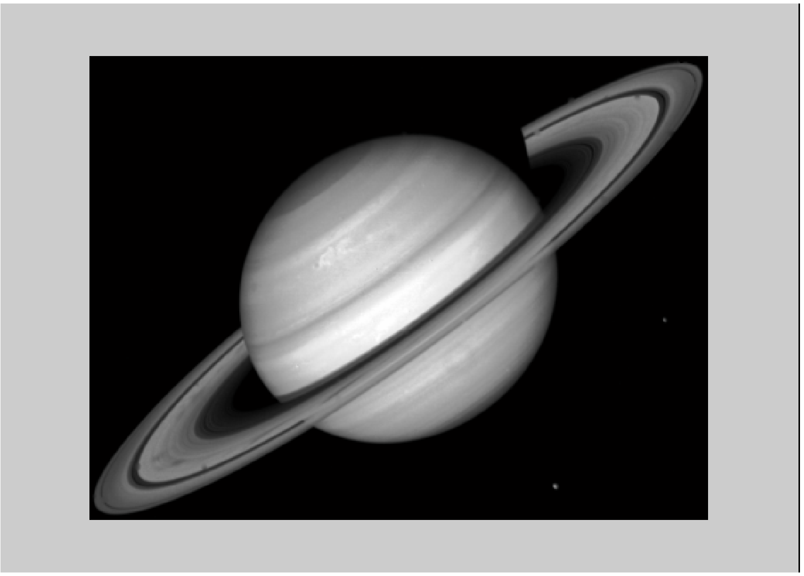
f) XOR(a,b) = $a \oplus b$

g) SUB(a,b) = $a \setminus b$

Arithmetic Based Operations

Operation	*Definition*	*preferred data type*
ADD	$c = a + b$	integer
SUB	$c = a - b$	integer
MUL	$c = a \bullet b$	integer or floating point
DIV	$c = a / b$	floating point
LOG	$c = \log(a)$	floating point
EXP	$c = \exp(a)$	floating point
SQRT	$c = \mathrm{sqrt}(a)$	floating point
TRIG.	$c = \sin/\cos/\tan(a)$	floating point
INVERT	$c = (2^B - 1) - a$	integer

Image negation




```
[x,map]=imread('saturn.tif');
imshow(x,map);
[nr,nc]=size(x);
% image is of type uint8, so convert it to double
x=double(x)+1;
for p=1:nr
for q=1:nc
xnew(p,q)=255-x(p,q);
end
imshow(xnew/max(max(xnew)));
```

Histogram Equalization

- Histogram normalization(linearization)-compare images on a specific basis
- Probability of occurrence of gray level r_k in an image, n – total number of pixels in image, n_k – number of pixels that have grey level r_k, s_k – grey level of output pixel
- L-total number of grey levels

$$p_r(r_k) = \frac{n_k}{k}, \text{ k=0,1,2,...,L-1}$$

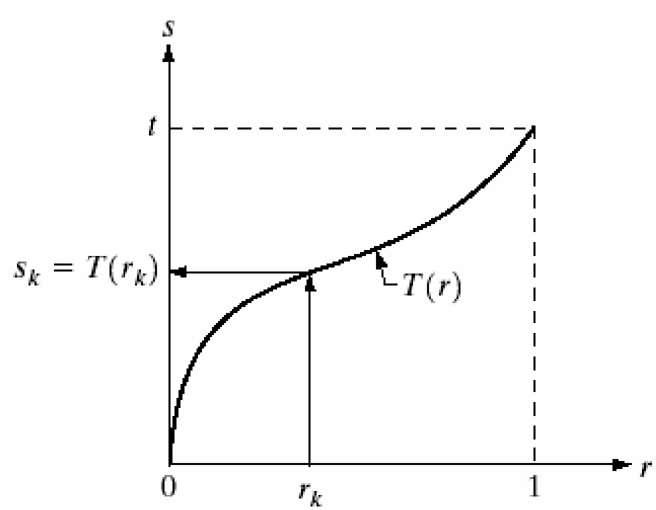
- Transformation function is

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j)$$

$$= \sum_{j=0}^{k} \frac{n_j}{n} \quad \text{k=0,1,2,...,L-1}$$

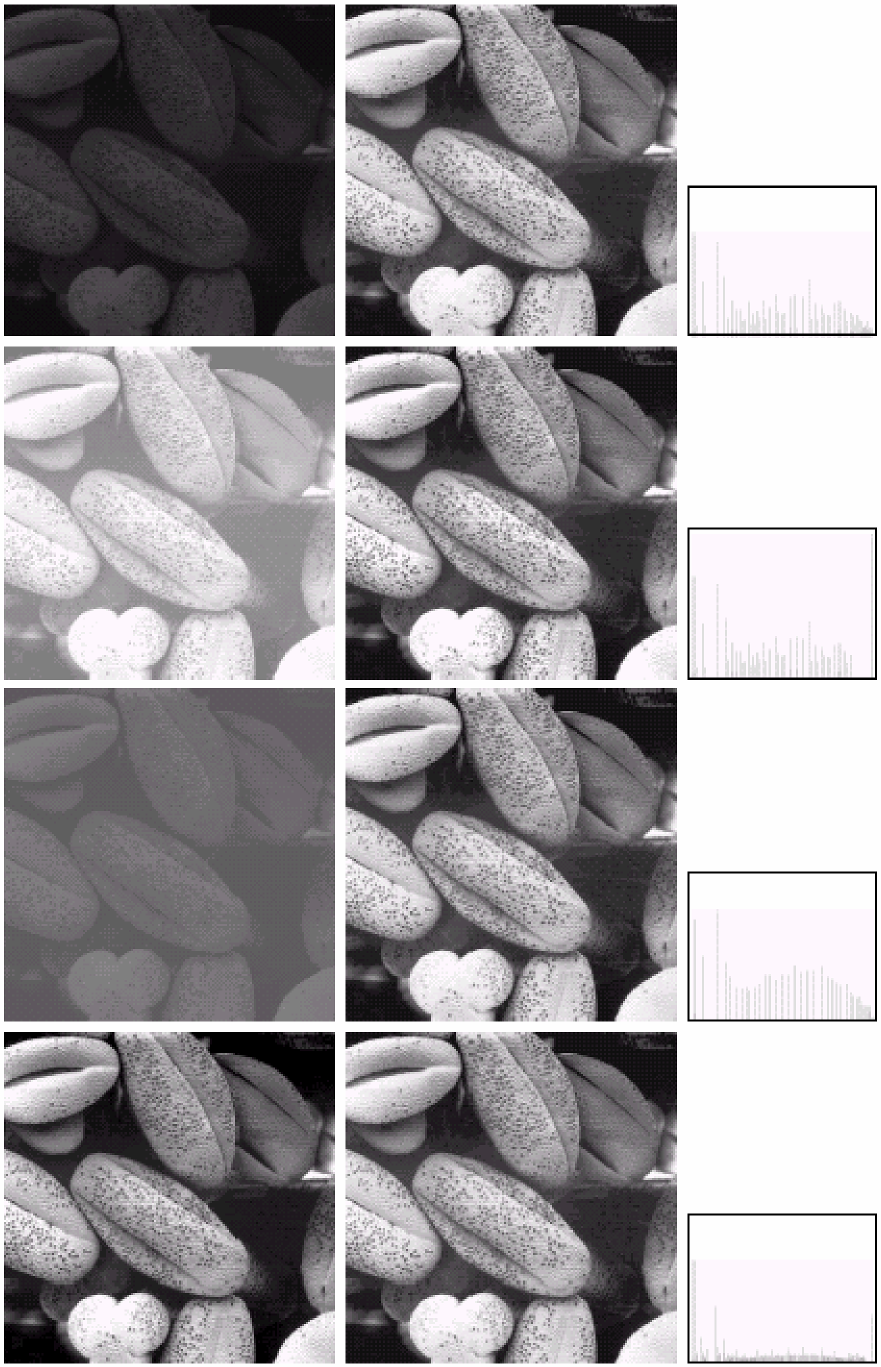
Histogram Equalization

- Transformation function satisfies
  - (a) T(r) is single valued and monotonically increasing in the interval 0 ≤r≤1
  - (b) 0≤T(r) ≤1 for 0≤r≤1



gray-level transformation function that is both single valued and monotonically increasing.

Original images, Results of histogram equalization, corresponding histogram



Enhancement by Image Averaging

- Consider a noisy image,

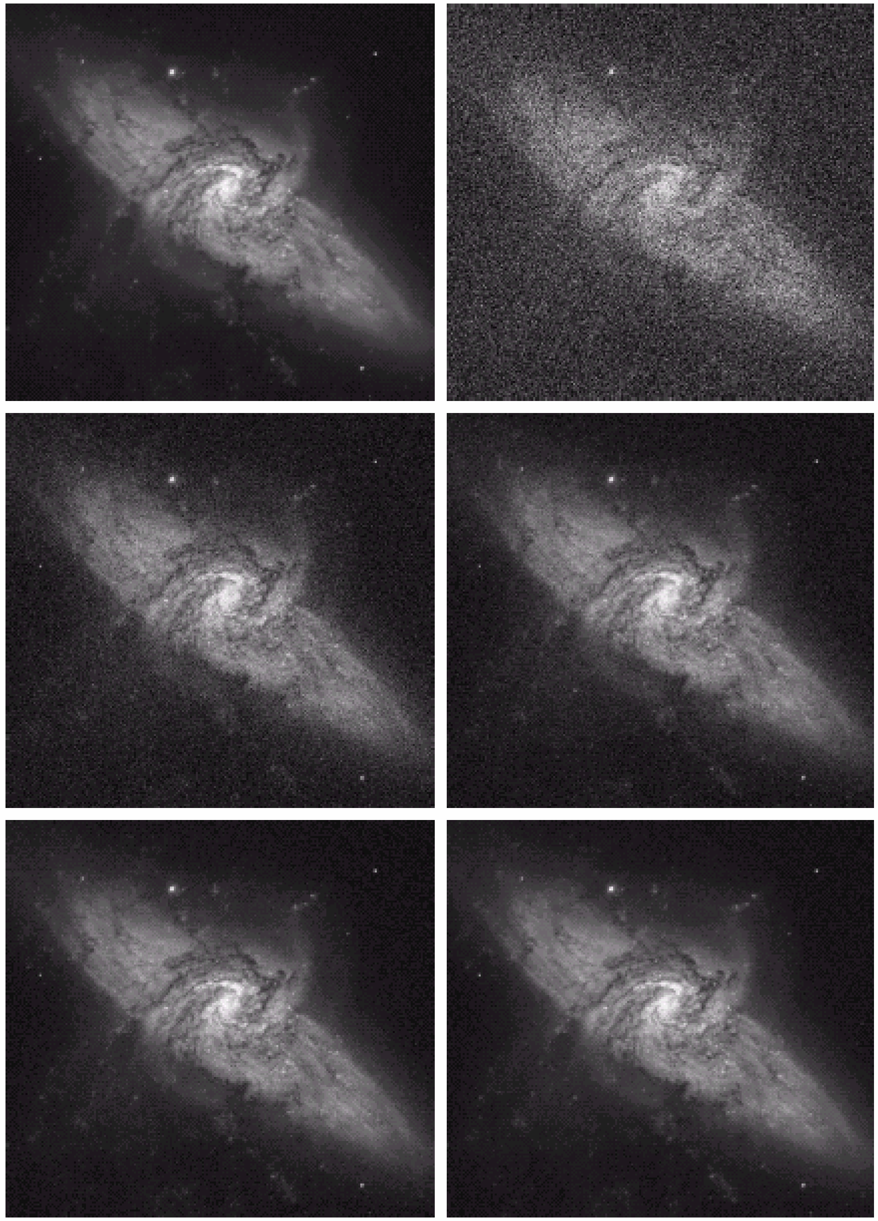
$$g(x,y)=f(x,y)+ \eta(x,y)$$

- Reduce the noise y adding a set of noisy images, $\{g_i(x,y)\}$
- Averaged image,

$$\overline{g}(x,y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x,y)$$

$$E\{\overline{g}(x,y)\} = f(x,y)$$

$$\sigma^2 \overline{g}(x,y) = \frac{1}{K}\sigma^2 \eta(x,y)$$

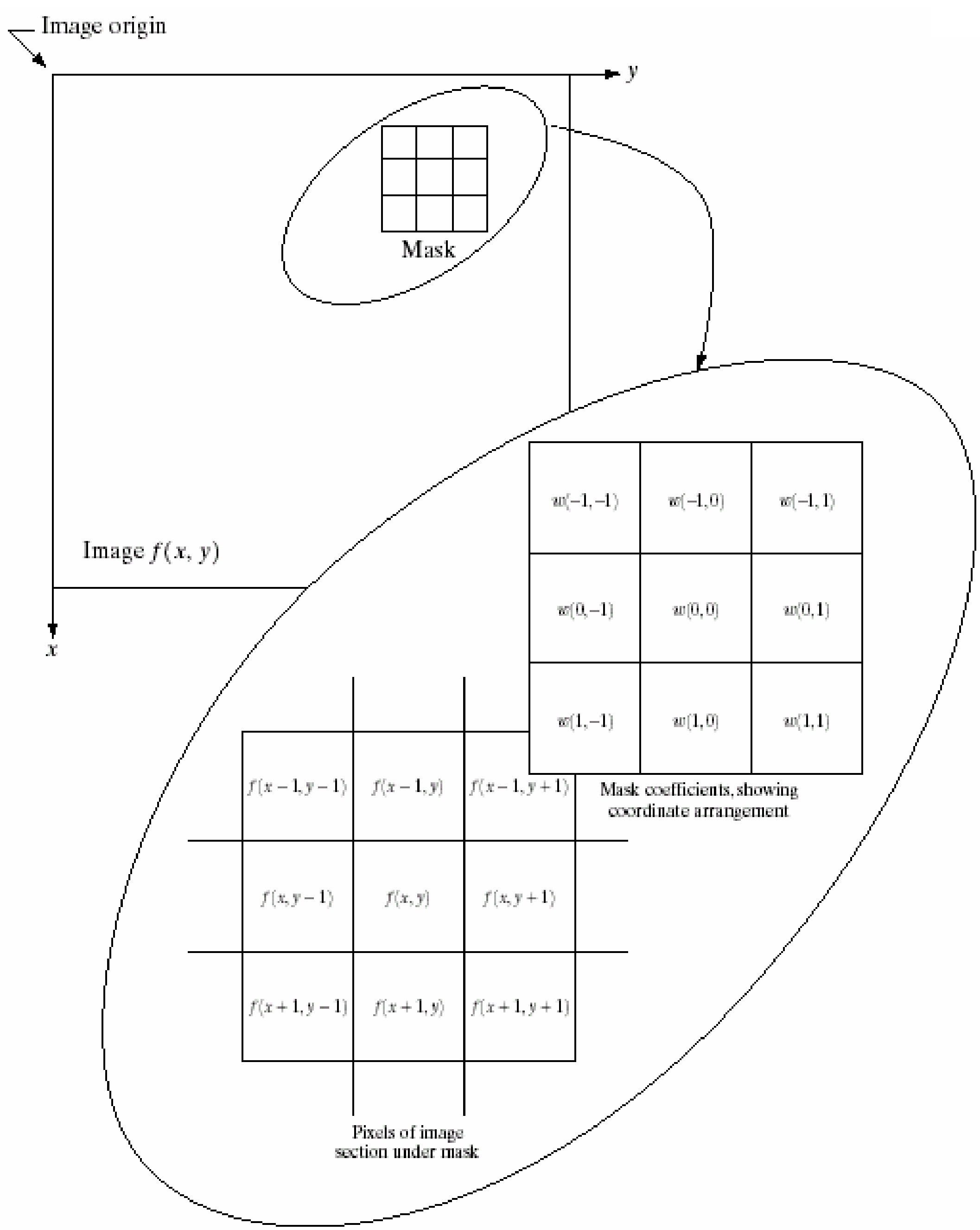


a b
c d
e f

(a) Image of Galaxy Pair NGC 3314. (b) Image corrupted by additive Gaussian noise with zero mean and a standard deviation of 64 gray levels. (c)–(f) Results of averaging $K = 8, 16, 64,$ and 128 noisy images. (Original image courtesy of NASA.)

Basics of Spatial Filtering

- Neighborhood subimage-filter, mask, kernel, template, or window
- Values in mask –coefficients
- Response R of linear filtering with the filter mask at a point (x,y) in the image is:
- R=w(-1,-1)f(x-1,y-1)+w(-1,0)f(x-1,y)+…
- +w(0,0)f(x,y)+…+w(1,0)f(x+1,y)+w(1,1)f(x+1,y+1)



The mechanics of spatial filtering. The magnified drawing shows a 3×3 mask and the image section directly under it; the image section is shown displaced out from under the mask for ease of readability.

Smoothing spatial filters

$\frac{1}{9} \times$

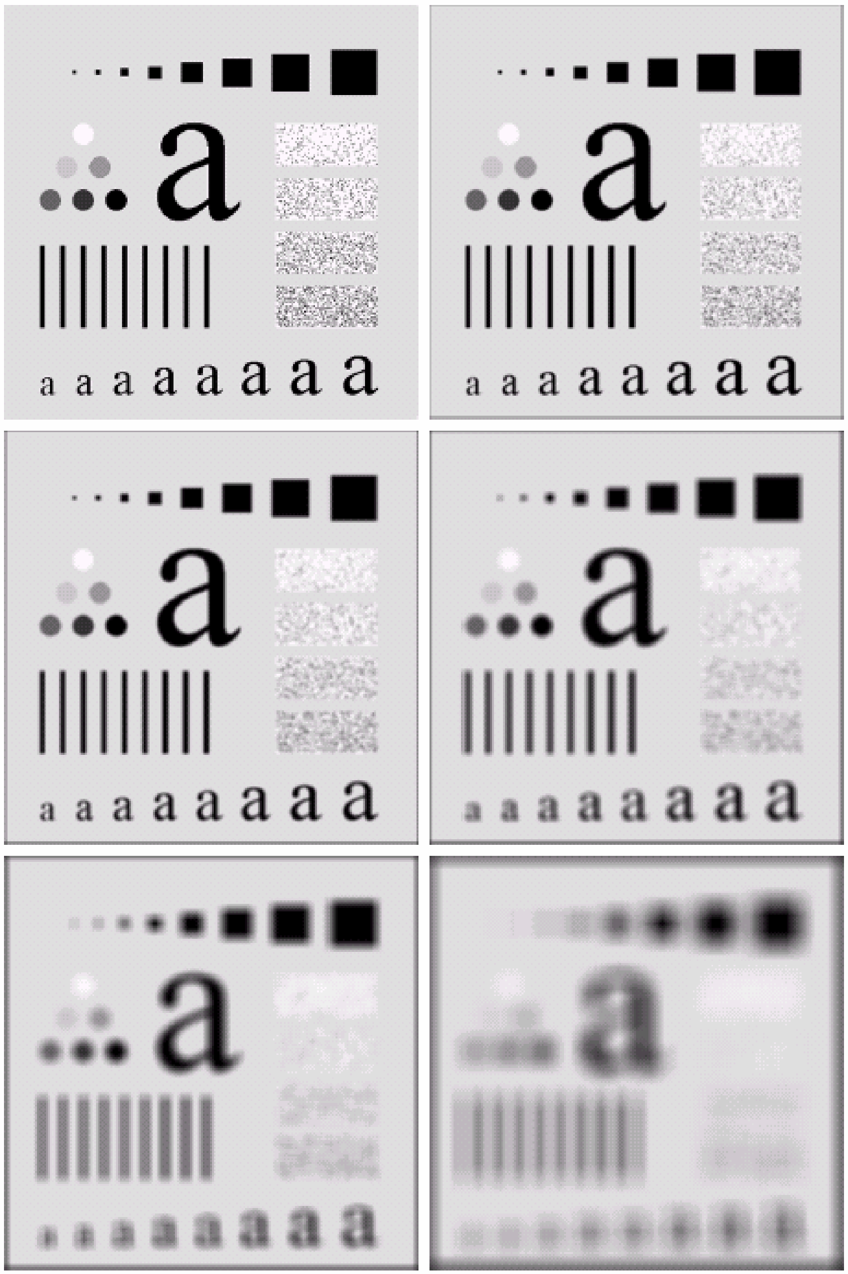
1	1	1
1	1	1
1	1	1

$\frac{1}{16} \times$

1	2	1
2	4	2
1	2	1

a b

Two 3×3 smoothing (averaging) filter masks. The constant multipli er in front of each mask is equal to the sum of the values of its coefficients, as is required to compute an average.



a b
c d
e f

(a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $n = 3, 5, 9, 15$, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their gray levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50 × 120 pixels.

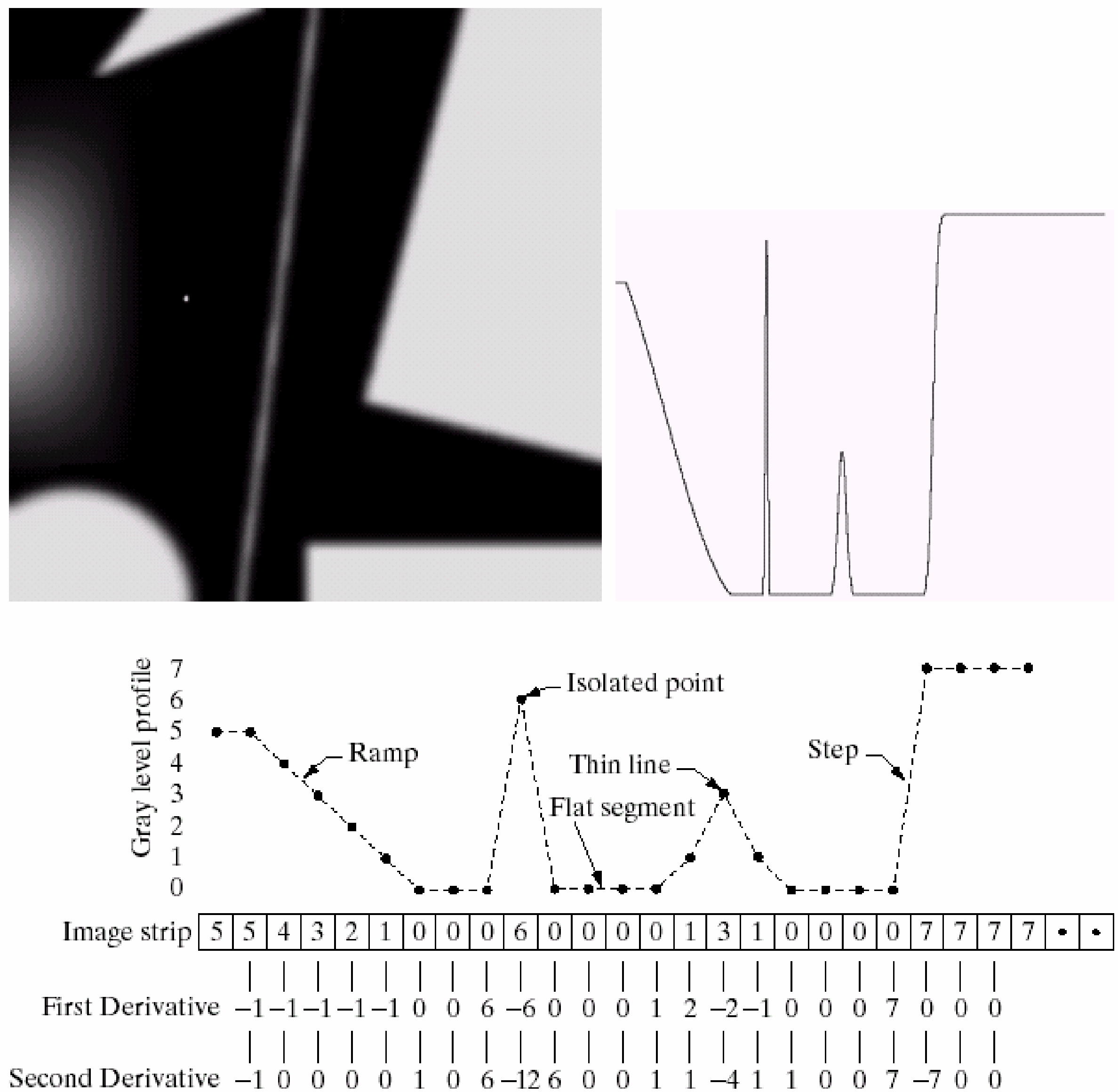
Sharpening spatial filters

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

Change between adjacent pixels
First and second derivative
(1) Must be zero in flat areas;
(2) Non zero at the onset of gray-level step or ramp
(3) First derivative-non zero along ramps; second derivative-Zero along ramps of cnstant slope

a b
c



(a) A simple image. (b) 1-D horizontal gray-level profile along the center of the image and including the isolated noise point. (c) Simplified profile (the points are joined by dashed lines to simplify interpretation).

The Laplacian for enhancement (the second derivative)

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

In the x direction

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

In the y direction

$$\frac{\partial^2 f}{\partial x^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

Sum min *g*

$$\nabla^2 f = [f(x, y+1) + f(x, y-1) + f(x+1, y) + f(x-1, y)] - 4f(x, y)$$

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

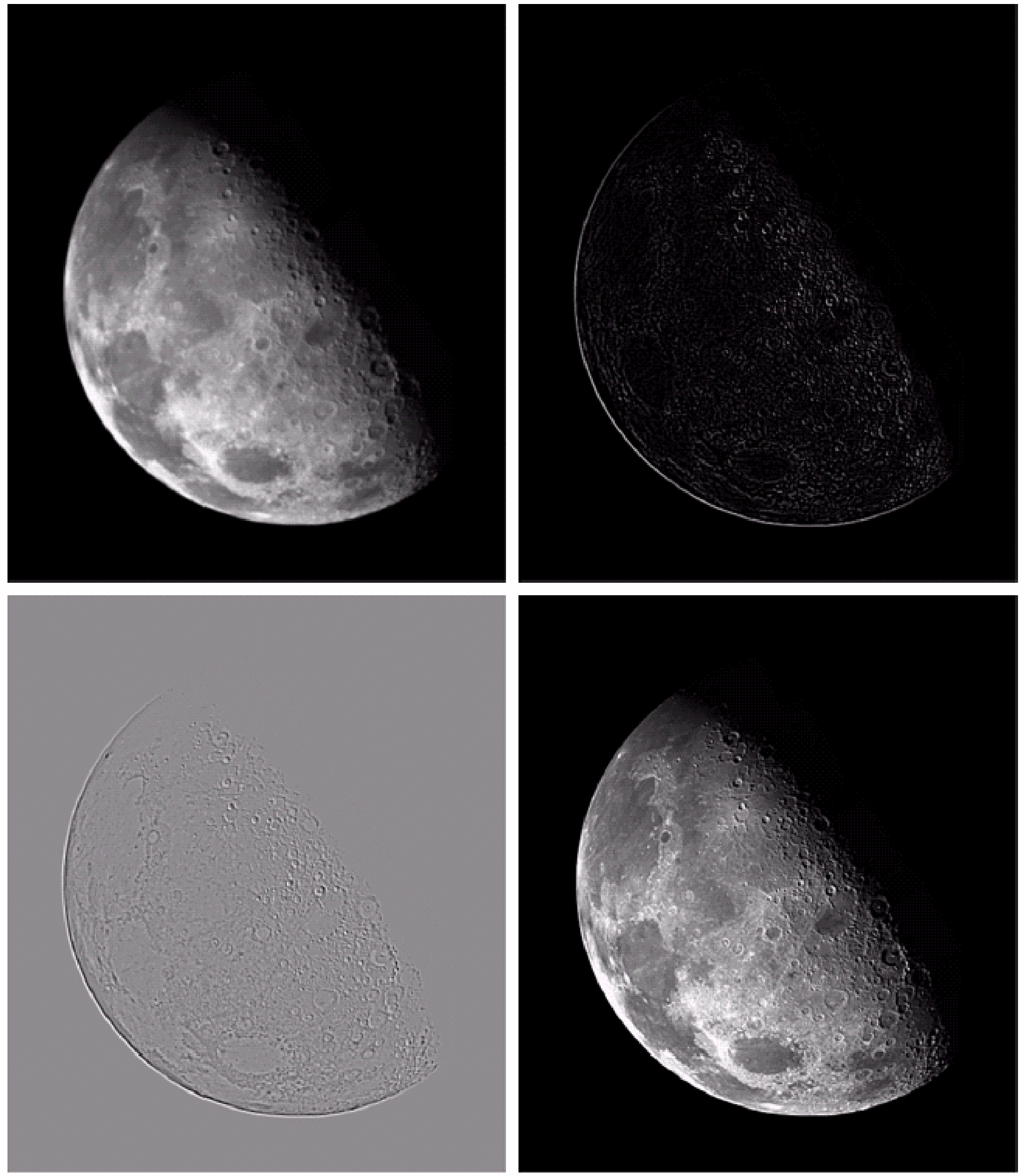
0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

a	b
c	d

(a) Filter mask used to implement the digital Laplacian, as defined in Eq. (3.7-4). (b) Mask used to implement an extension of this equation that includes the diagonal neighbors. (c) and (d) Two other implementations of the Laplacian.

a b
c d

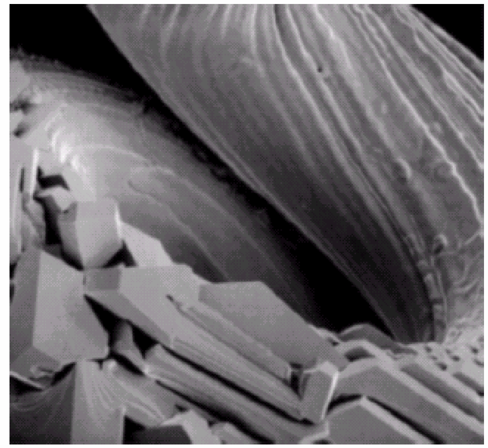


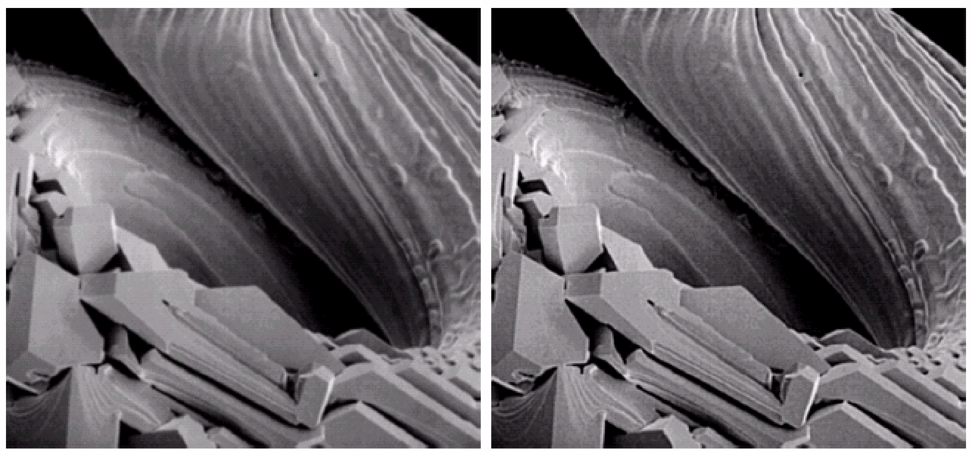
(a) Image of the North Pole of the moon. (b) Laplacian-filtered image. (c) Laplacian image scaled for display purposes. (d) Image enhanced by using Eq. (3.7-5). (Original image courtesy of NASA.)

Composite Laplacian mask

0	–1	0
–1	5	–1
0	–1	0

–1	–1	–1
–1	9	–1
–1	–1	–1





a b c
d e

(a) Composite Laplacian mask. (b) A second composite mask. (c) Scanning electron microscope image. (d) and (e) Results of filtering with the masks in (a) and (b), respectively. Note how much sharper (e) is than (d). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

The Gradient for enhancement (the first derivative)

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

$$\nabla f = mag(\nabla f)$$

$$= [G_x{}^2 + G_y{}^2]^{1/2}$$

$$= \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2}$$

$$\nabla f \approx \left| (z_7 + 2z_8 + z_9) - (z_1 + 2z_1 + z_3) \right| + \left| (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \right|$$

a

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

b

−1	0
0	1

c

0	−1
1	0

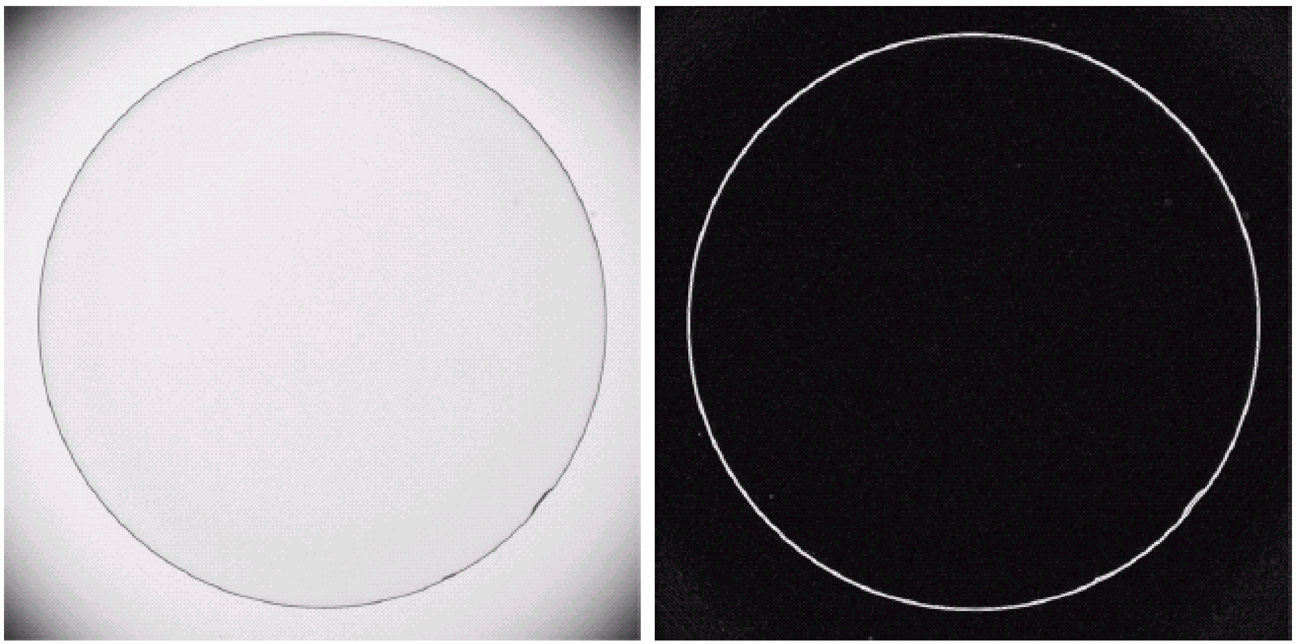
d

−1	−2	−1
0	0	0
1	2	1

e

−1	0	1
−2	0	2
−1	0	1

A 3 × 3 region of an image (the z's are gray-level values) and masks used to compute the gradient at point labeled z_5. All masks coefficients sum to zero, as expected of a derivative operator.



a b

Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock).
(b) Sobel gradient.
(Original image courtesy of Mr. Pete Sites, Perceptics Corporation.)