CSE6706:

Advanced Digital Image Processing

Dr. Md. Monirul Islam

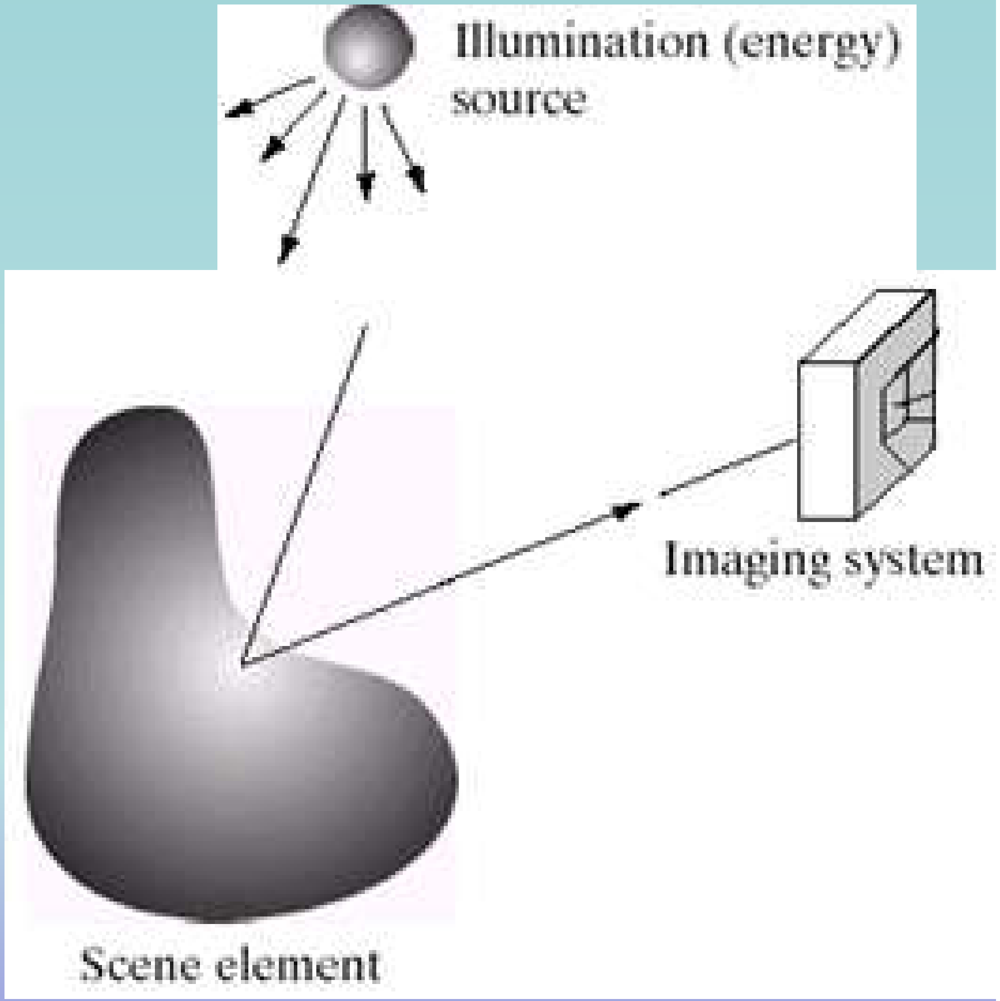
CSE-BUET

Topics

- Image Processing Basics
- Data Structure
- Image Enhancement

Image Acquisition

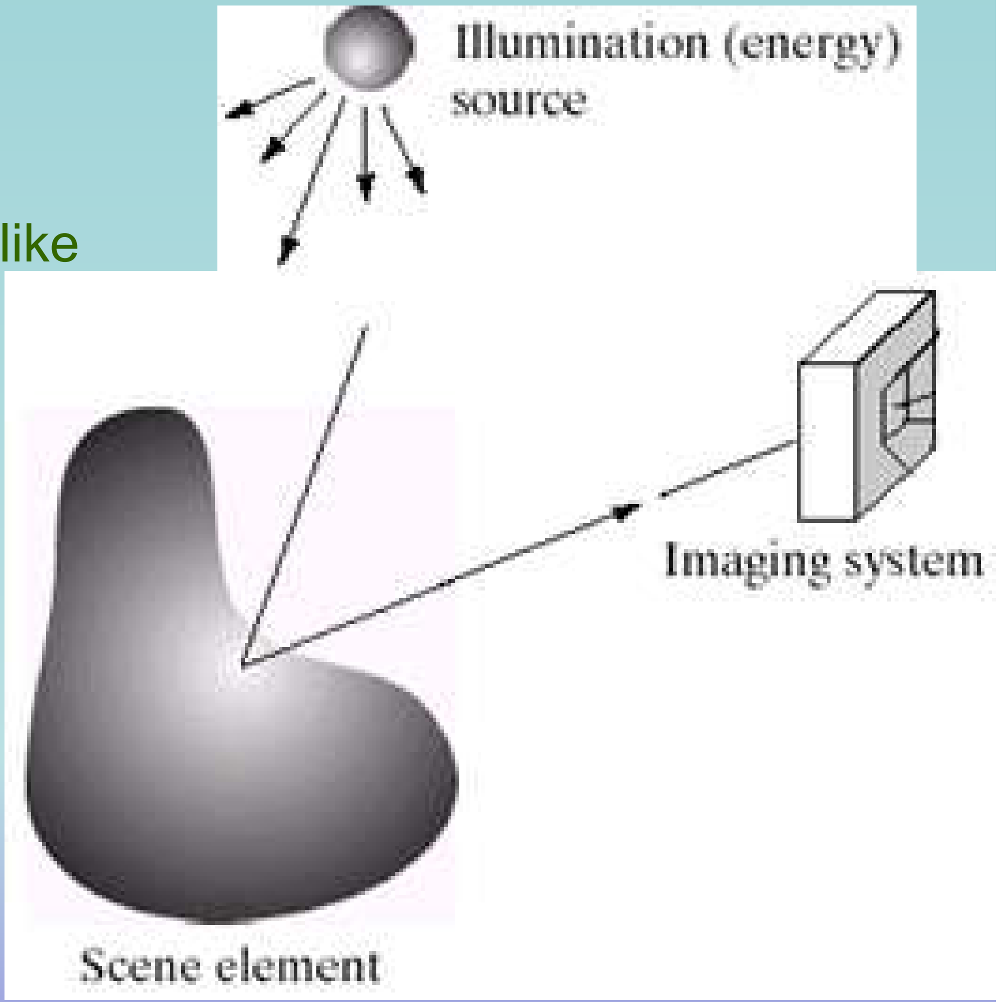
- Three main elements
  - Illumination source
  - Scene
  - Sensor (imaging system)



CSE-BUET

Image Acquisition

- Illumination source:
  - Can be light energy or
  - EM spectrum
  - Even less tradition sources like
    - Sound, heat

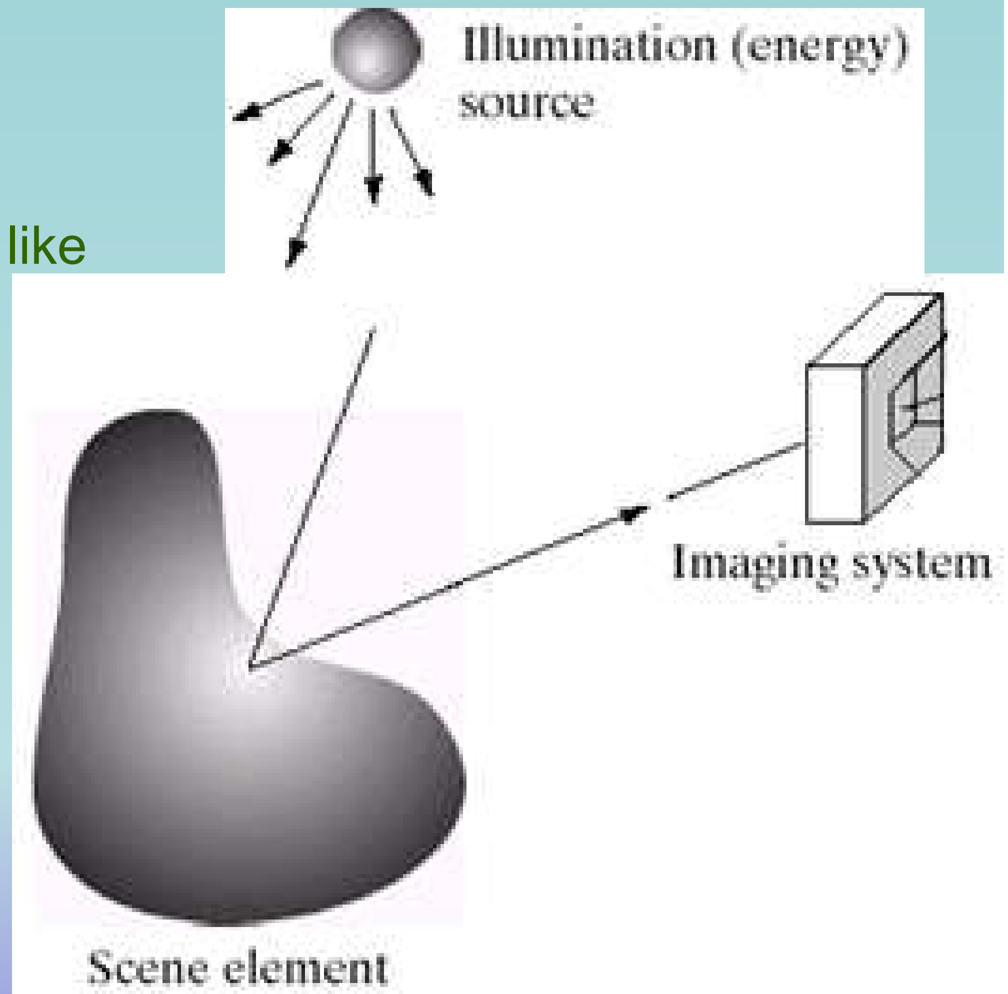


CSE-BUET

Image Acquisition

- Illumination source:
  - Can be light energy or
  - EM spectrum
  - Even less tradition sources like
    - Sound, heat

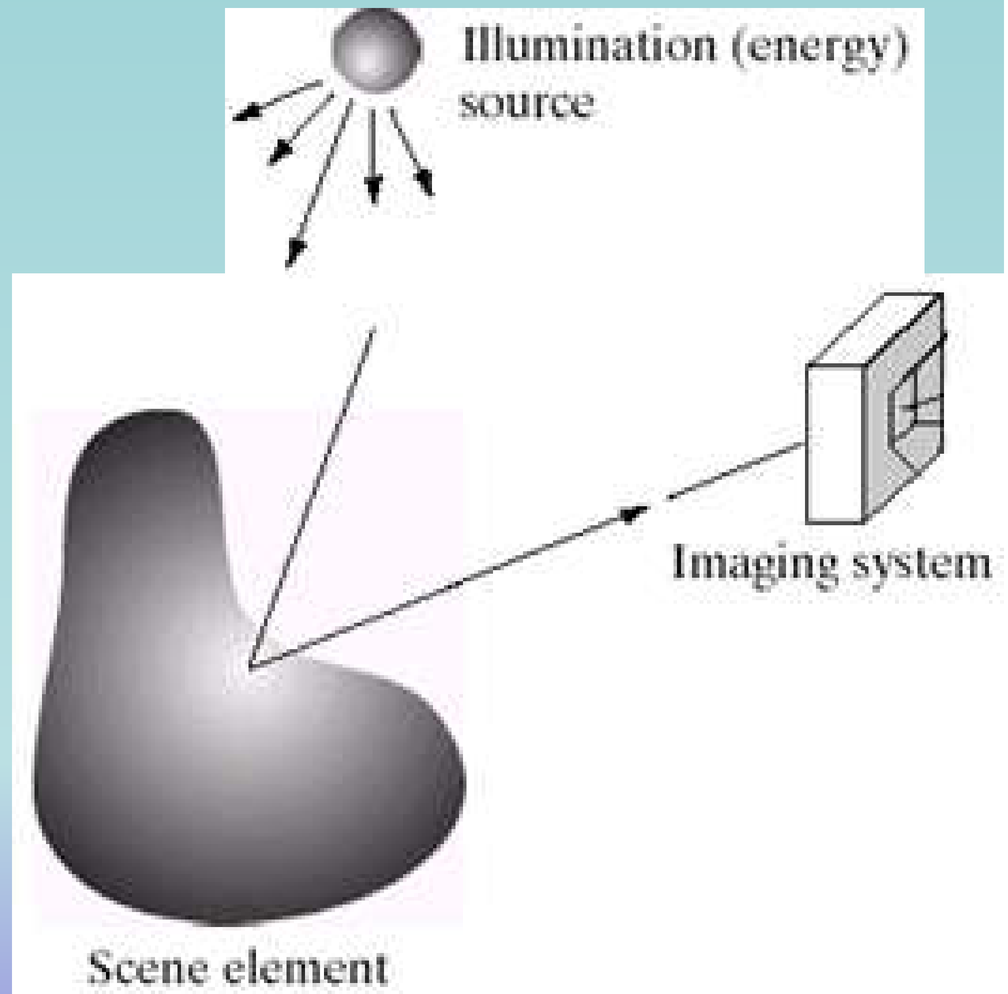
- Scene:
  - Any object: visible or hidden
  - Source itself



CSE-BUET

Image Acquisition

- Sensor:
  - Should be capable of sensing the energy



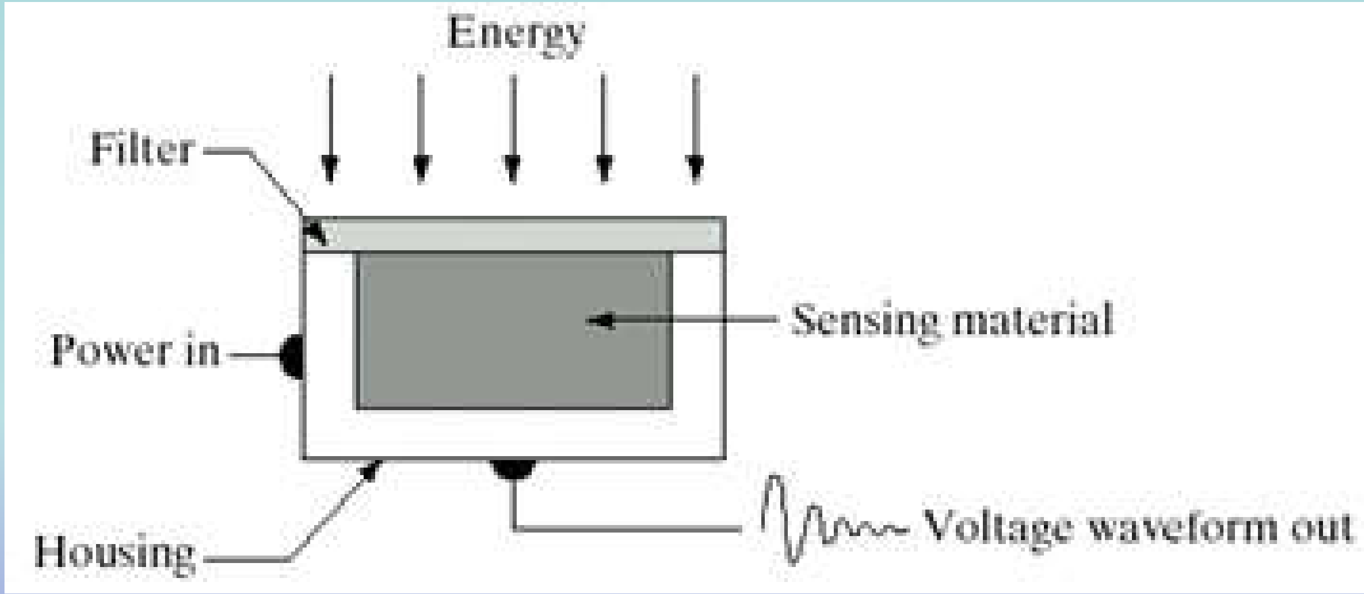
CSE-BUET

Image Sensors

- 3 main sensor arrangements:
  - Single sensor
  - Line sensor
  - Array sensor

Image Sensors

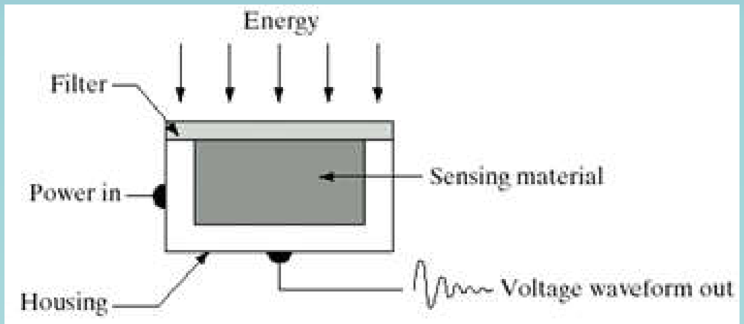
- Single sensor
  - Sensing element can be a photodiode
  - Filter absorbs extra energy or acts as pass-band



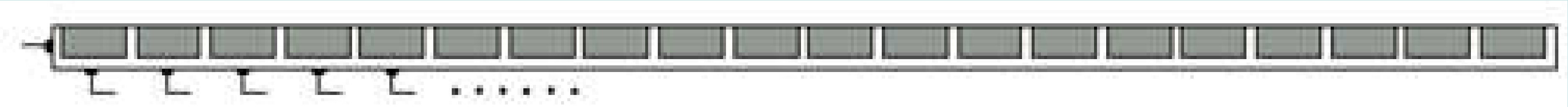
CSE-BUET

Image Sensors

- Line sensor



Single sensor

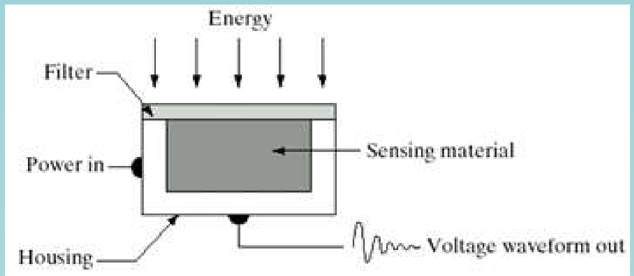


Line sensor

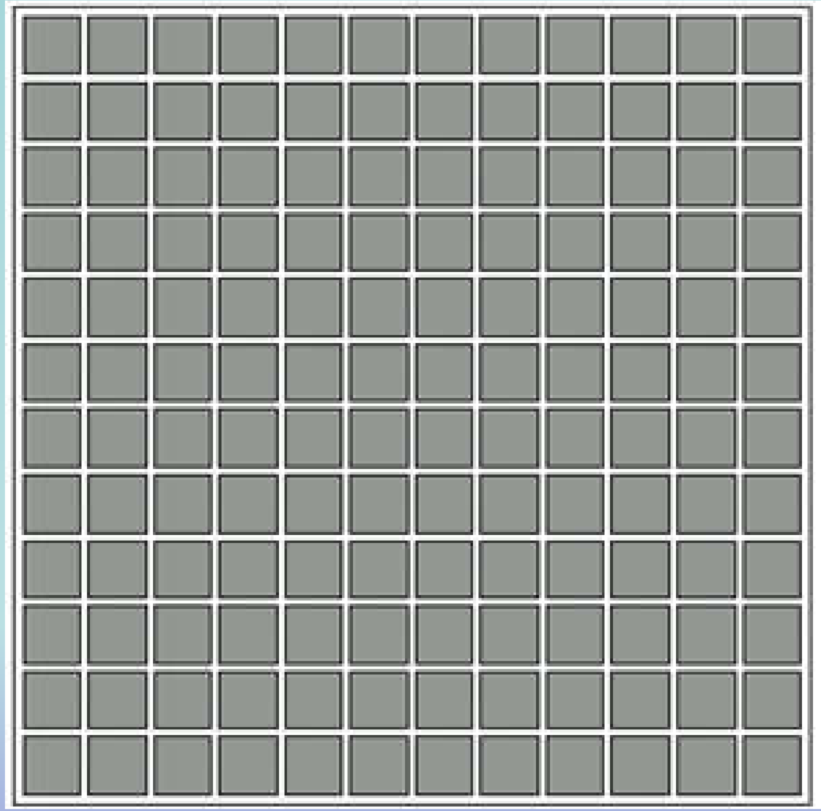
CSE-BUET

Image Sensors

- Array sensor



Single sensor

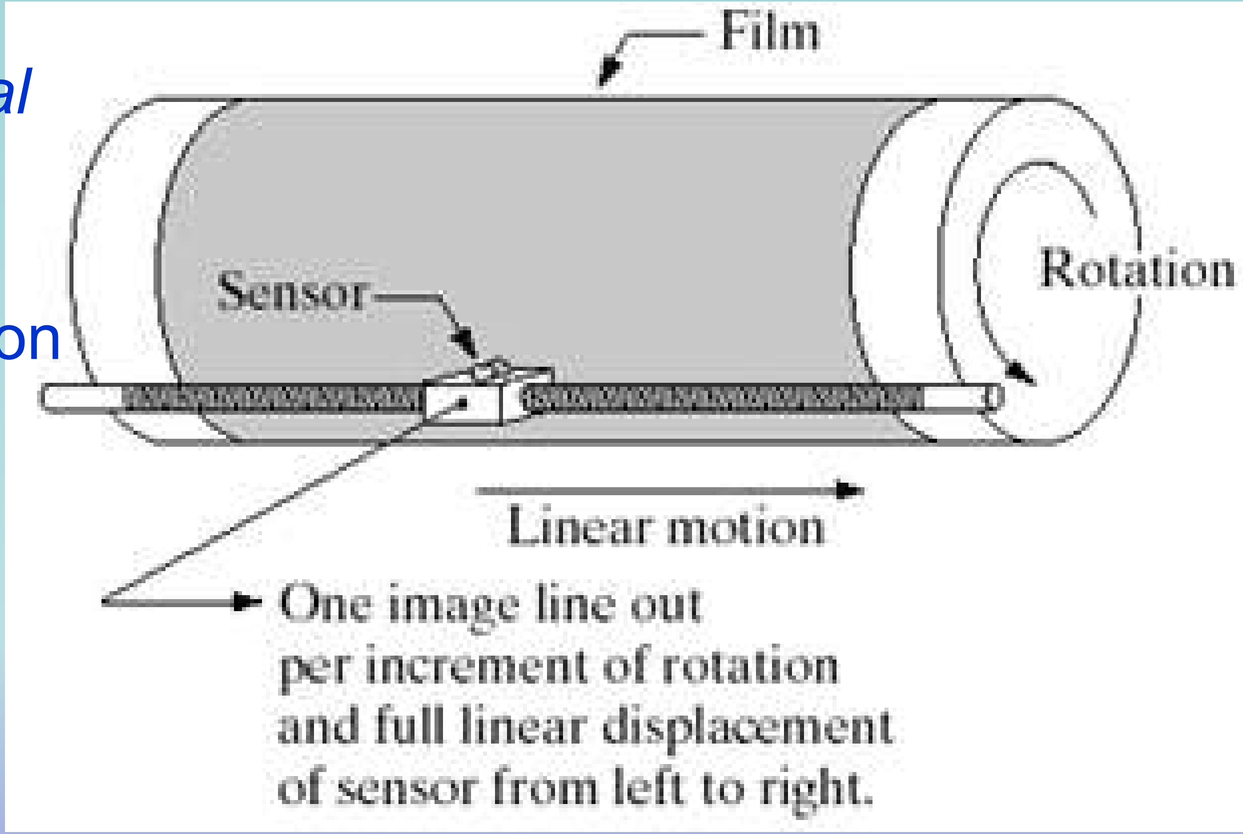


Array sensor

CSE-BUET

Image Acquisition using Single Sensor

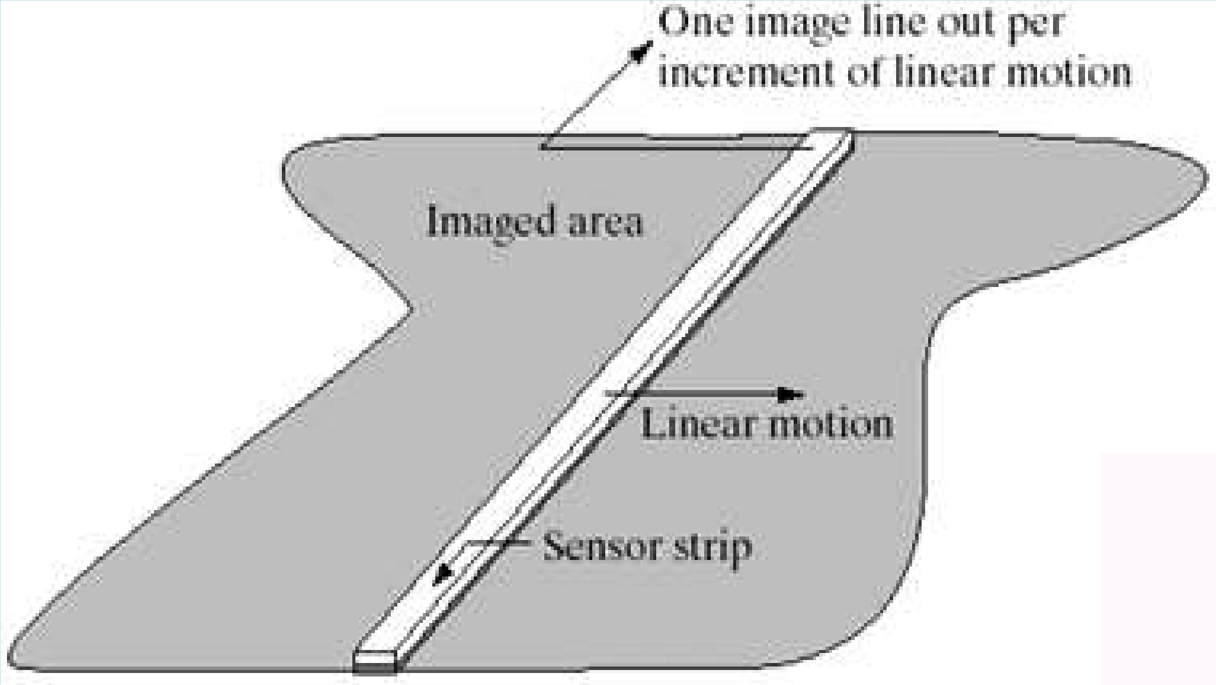
- can image only a particular point
- Requires *mechanical movement* in both direction
- High spatial resolution is possible in both direction



CSE-BUET

Image Acquisition using Line Sensors (or Sensor Strips)

- Two possible arrangements
  - Linear arrangement
    - Flatbed scanner
    - Airborne imaging



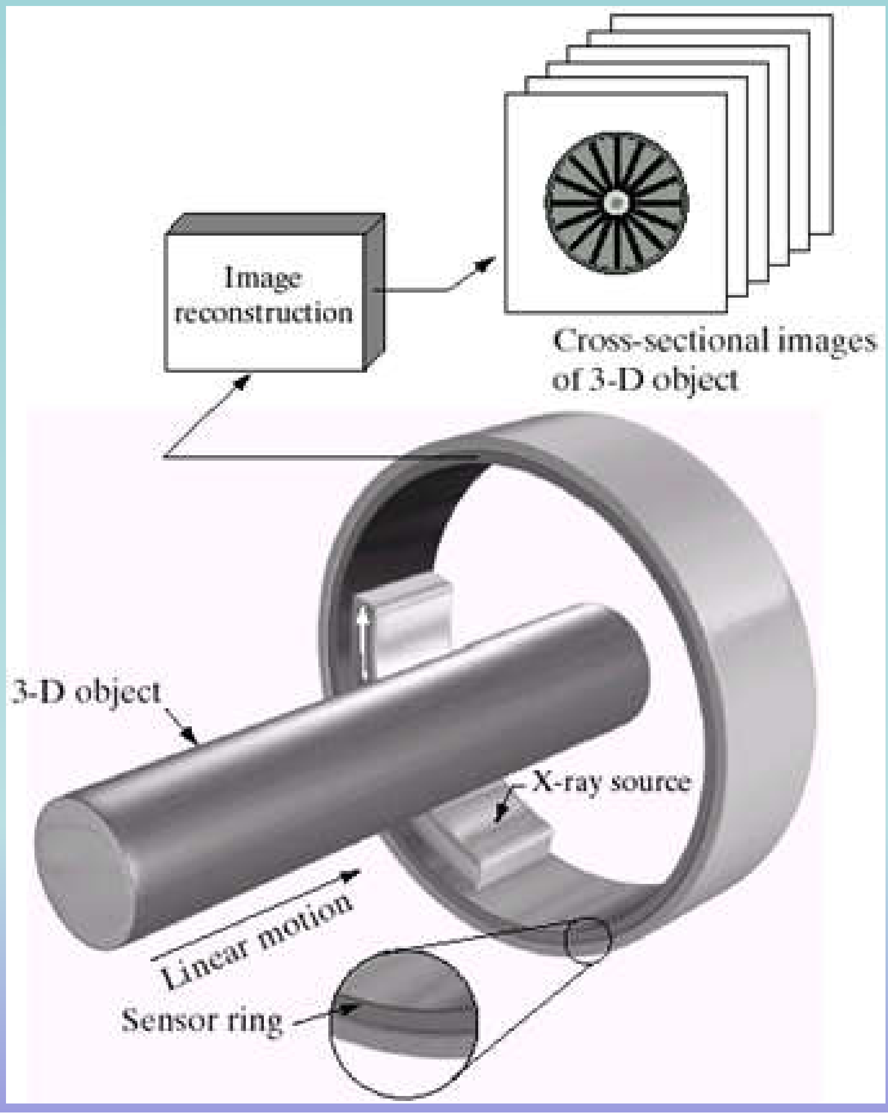
Linear arrangement

CSE-BUET

Image Acquisition using Line Sensors (or Sensor Strips)

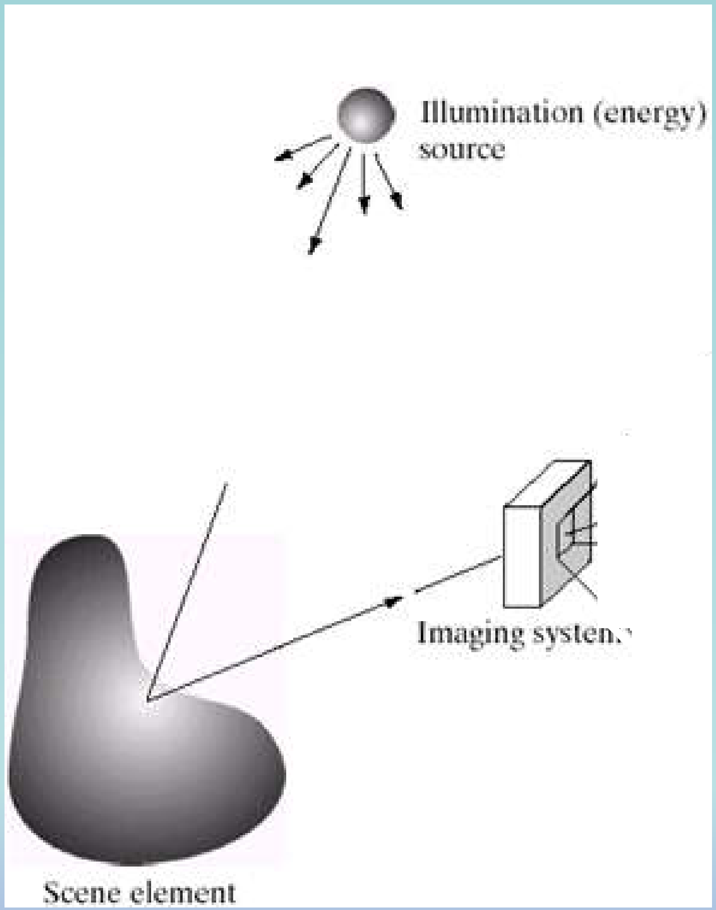
- Two possible arrangements
  - Ring or Circular arrangement
    - Used in CT, PET or MRI

Circular arrangement



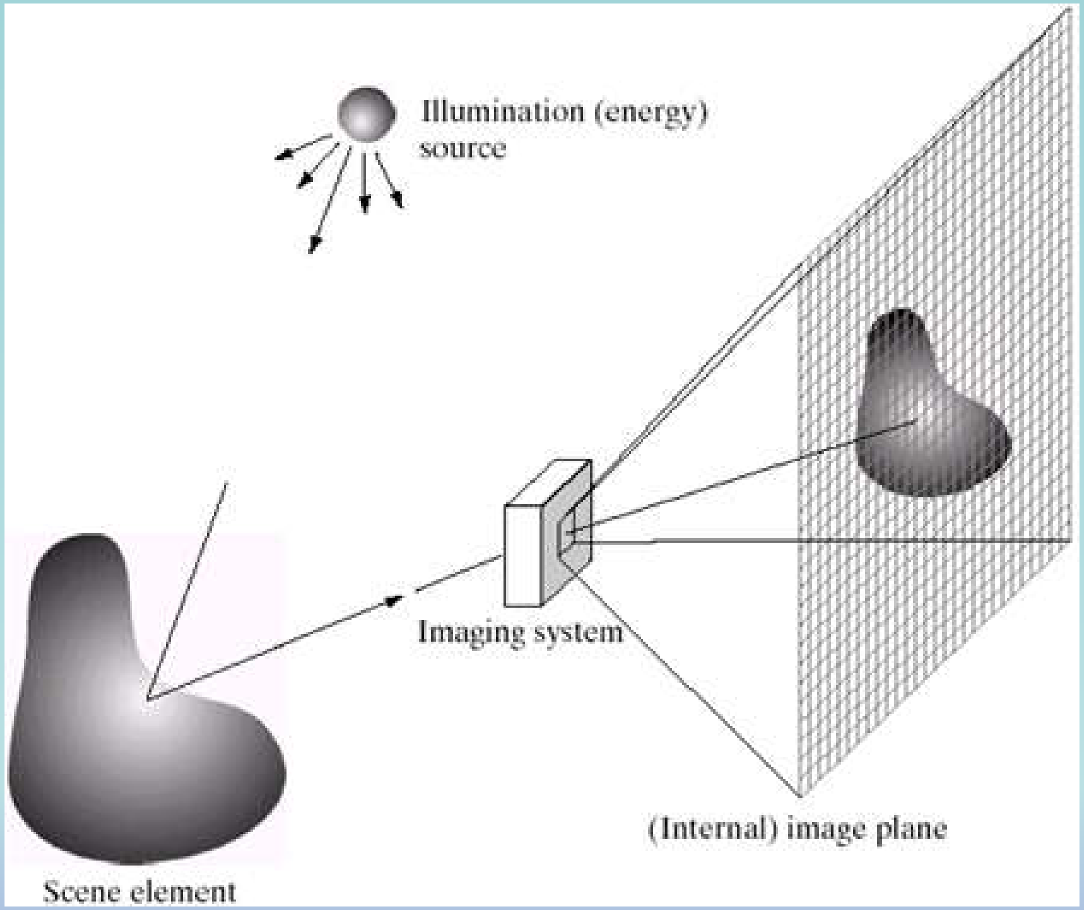
CSE-BUET

Image Acquisition using Sensor Array



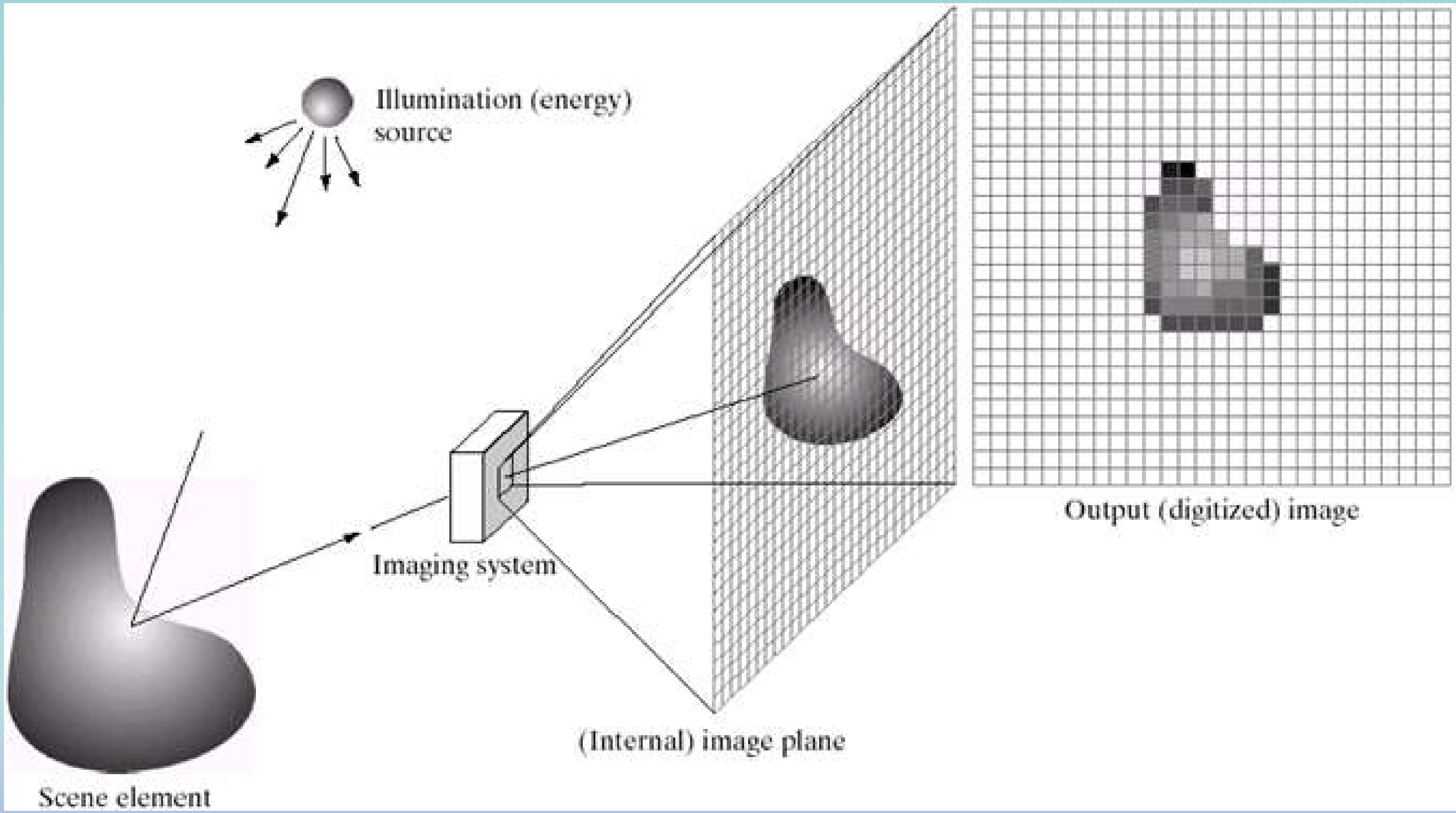
CSE-BUET

Image Acquisition using Sensor Array



CSE-BUET

Image Acquisition using Sensor Array



CSE-BUET

Image Formation Model

- 2 D function $f(x, y)$
- $f(x, y)$ represents intensity which is proportional to energy radiation

$$0 < f(x, y) < \infty$$

Image Formation Model

- 2 components of $f(x, y)$
  - Illumination, $i(x, y)$: $0 < i(x, y) < \infty$
  - Reflection (or transmission), $r(x, y)$: $0 < r(x, y) < 1$

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

Image Formation Model

- Typical values of Illumination: $i(x, y)$
  - Sunlight: 90K lm/sq. m
  - Full moon: 10K lm/sq. m
  - Commercial office: 1K lm/sq. m

- Typical values of Reflection: $r(x, y)$
  - Black velvet: 0.01
  - Stainless still: 0.65
  - Flat-white wall paint: 0.80
  - Silver plated metal: 0.90
  - Snow: 0.93

Image Formation Model

- Typical values of Illumination: $i(x, y)$
  - Commercial office: 1K lm/sq. m
- Typical values of Reflection: $r(x, y)$
  - Black velvet: 0.01
  - Snow: 0.93
- Range of gray levels

$$f_{\min} = L_{\min} = i_{\min} \times r_{\min} \approx 10$$

$$f_{\max} = L_{\max} = i_{\max} \times r_{\max} \approx 1000$$

Image Formation Model

- Range of gray levels

$$f_{\min} = L_{\min} = i_{\min} \times r_{\min} \approx 10$$
$$f_{\max} = L_{\max} = i_{\max} \times r_{\max} \approx 1000$$

- $[L_{min}, L_{max}]$ is gray scale
- Usually, $[L_{min}, L_{max}]$ is normalized to $[0, L\text{-}1]$
  - $l = 0$ is black
  - $l = L - 1$ is White

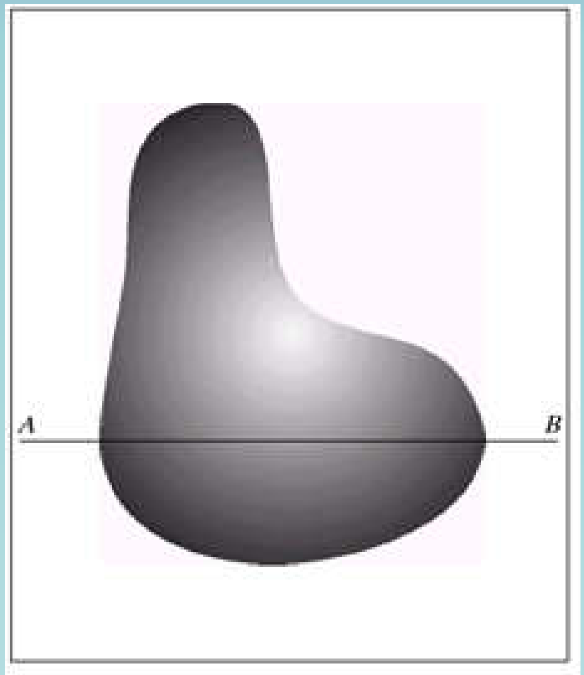
Image Digitization

- Two steps:
  - Sampling
  - Quantization

Image Digitization

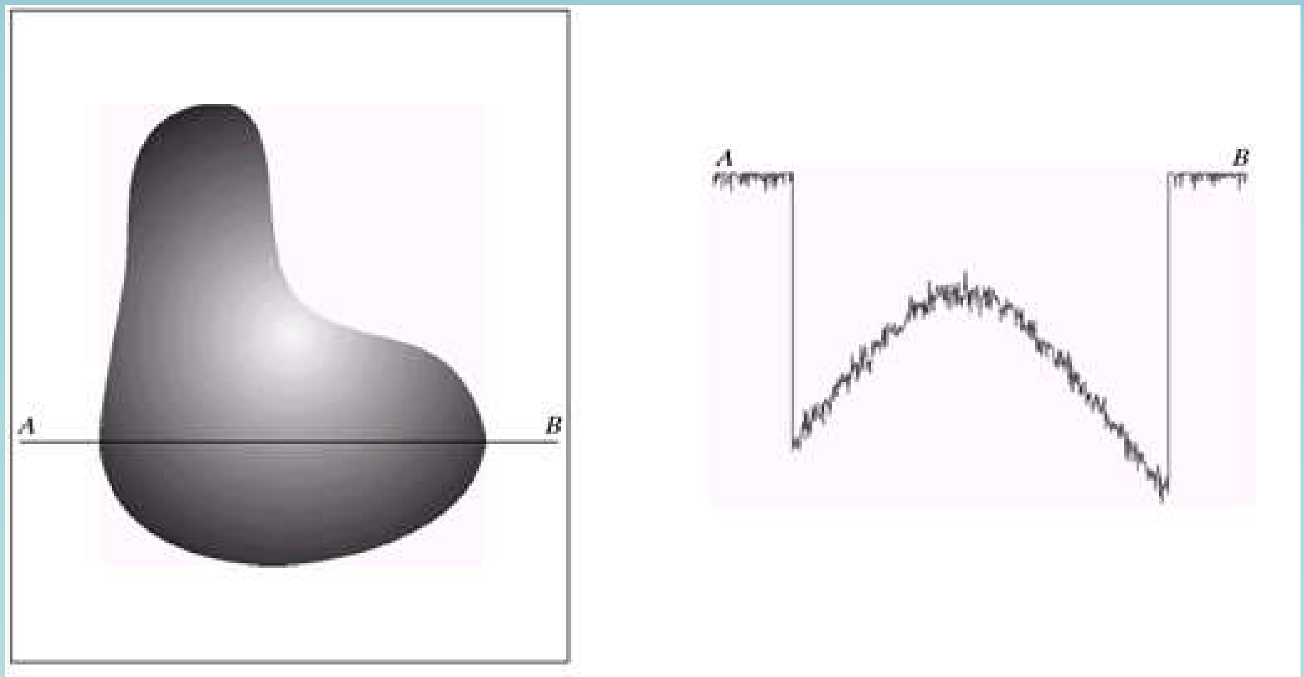
- Sampling
  - Digitizing coordinates
- Quantization
  - Digitizing amplitudes (gray scale values)

Image Digitization



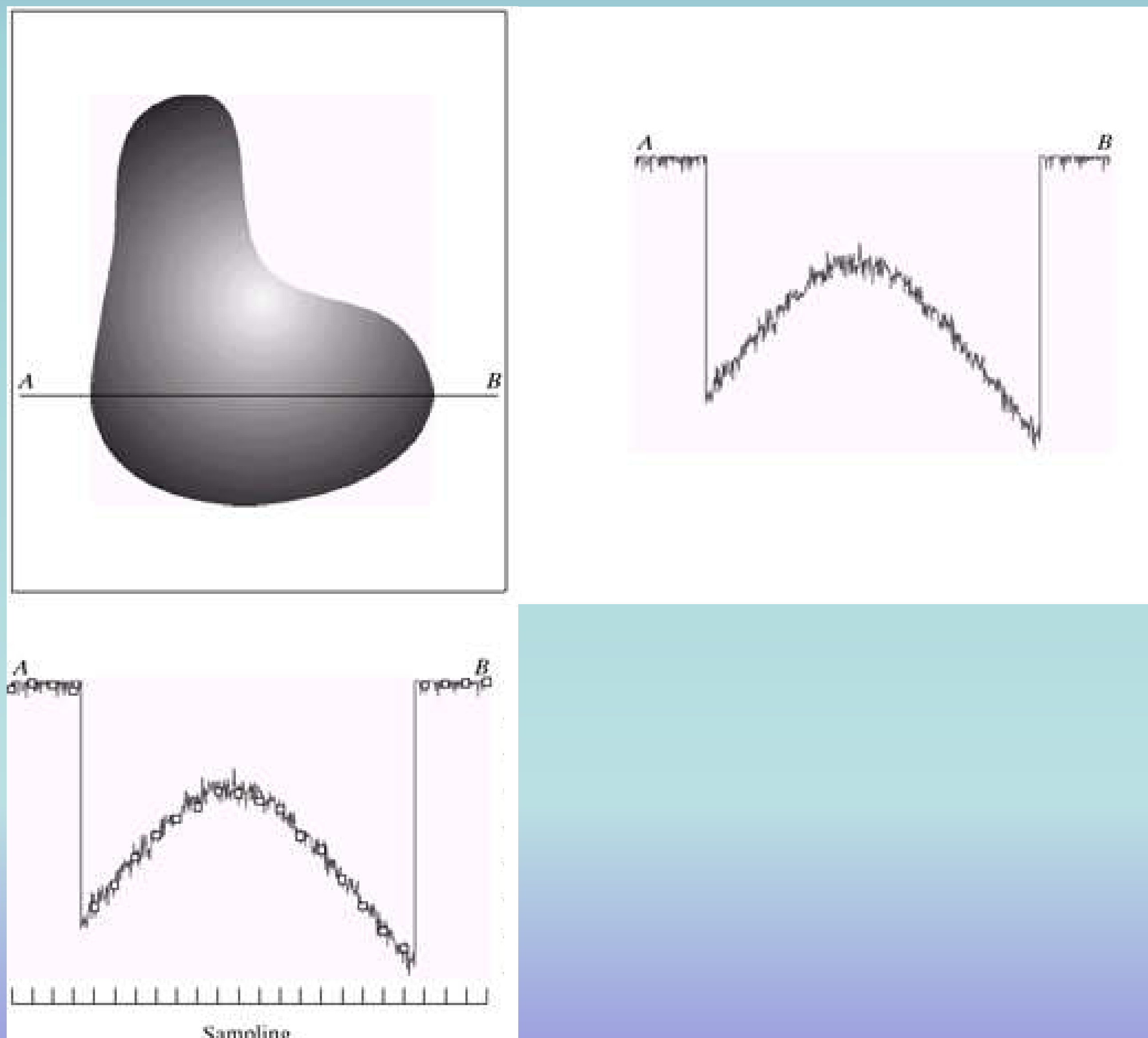
CSE-BUET

Image Digitization



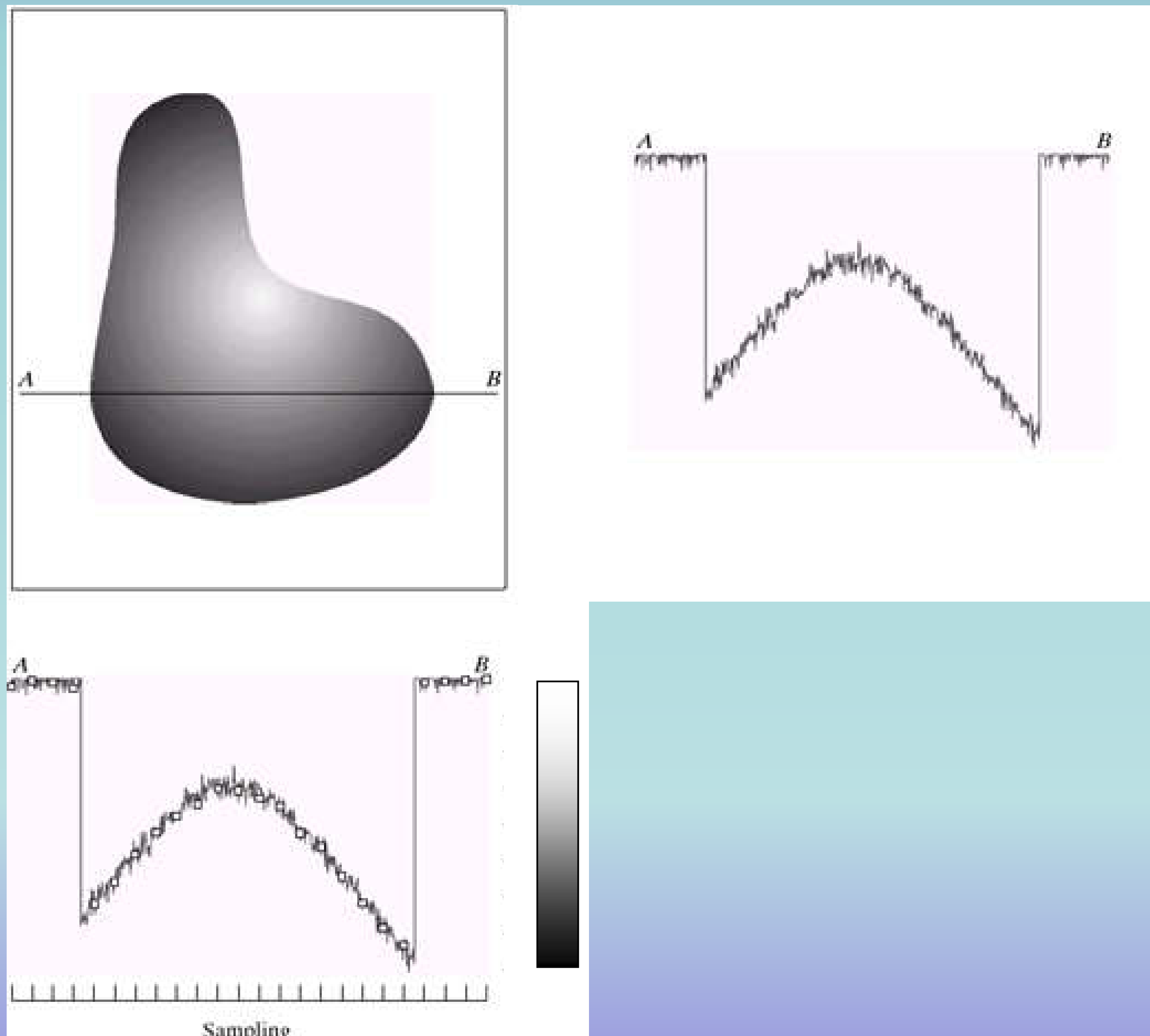
CSE-BUET

Image Digitization



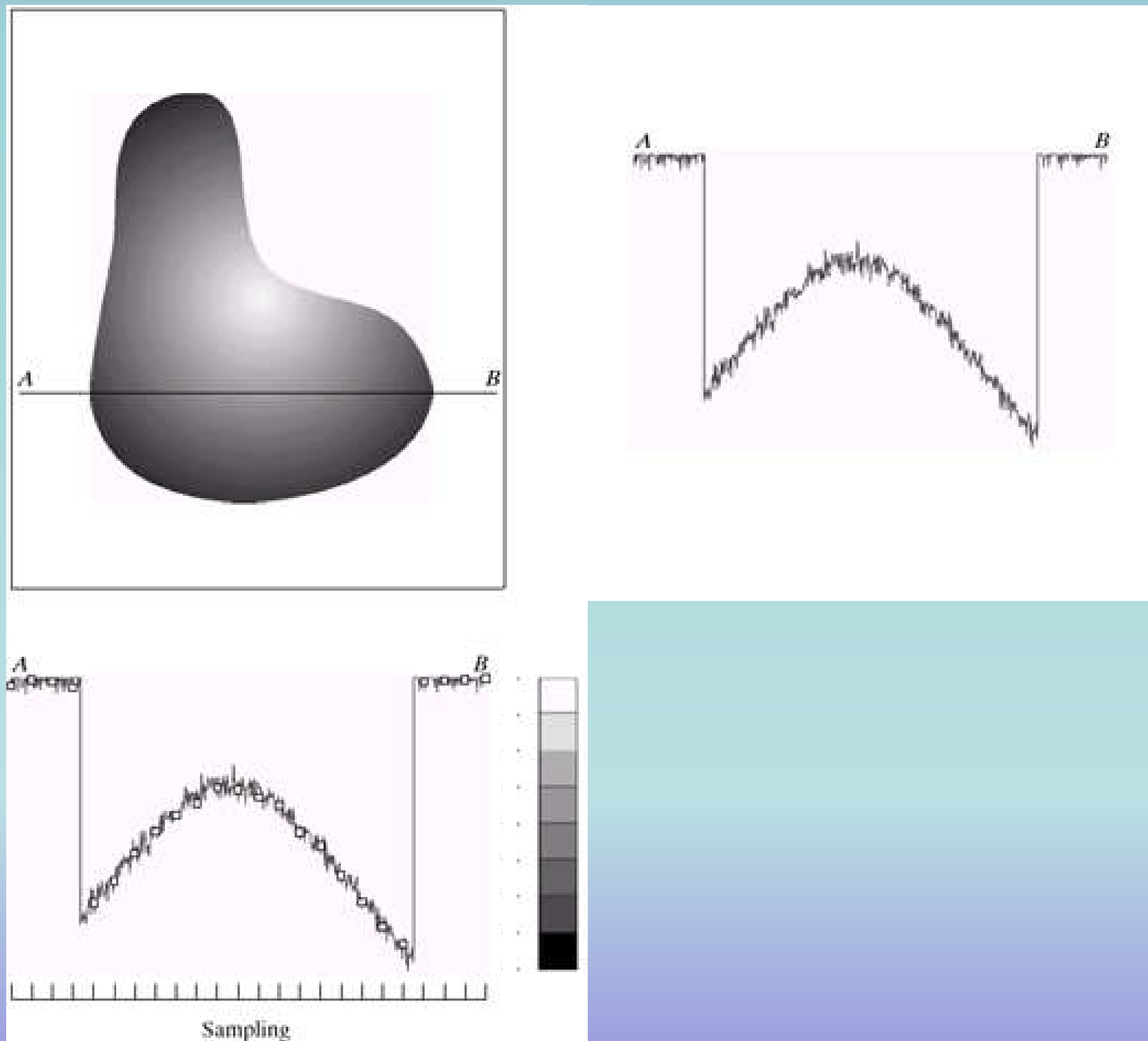
CSE-BUET

Image Digitization



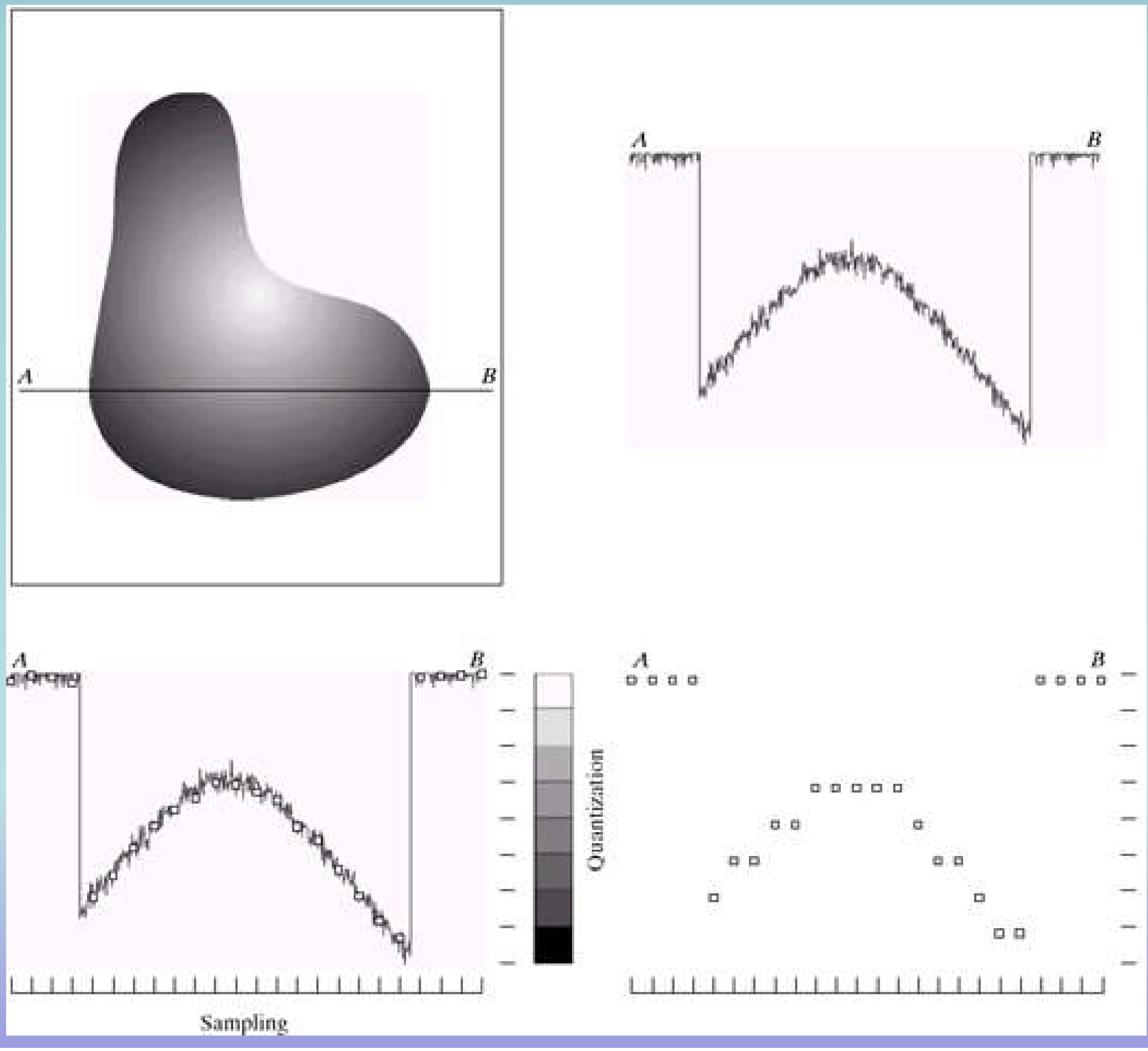
CSE-BUET

Image Digitization



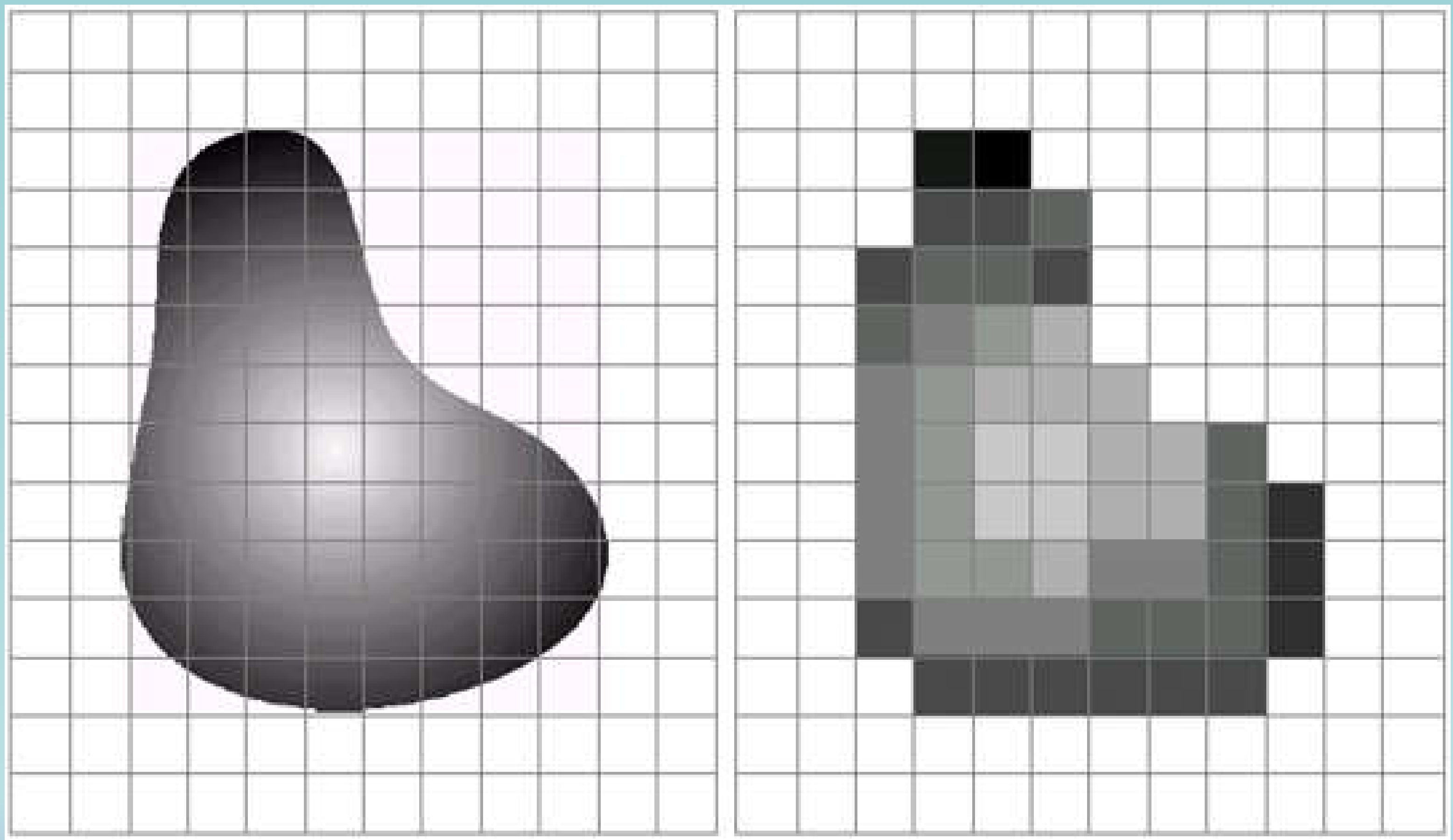
CSE-BUET

Image Digitization



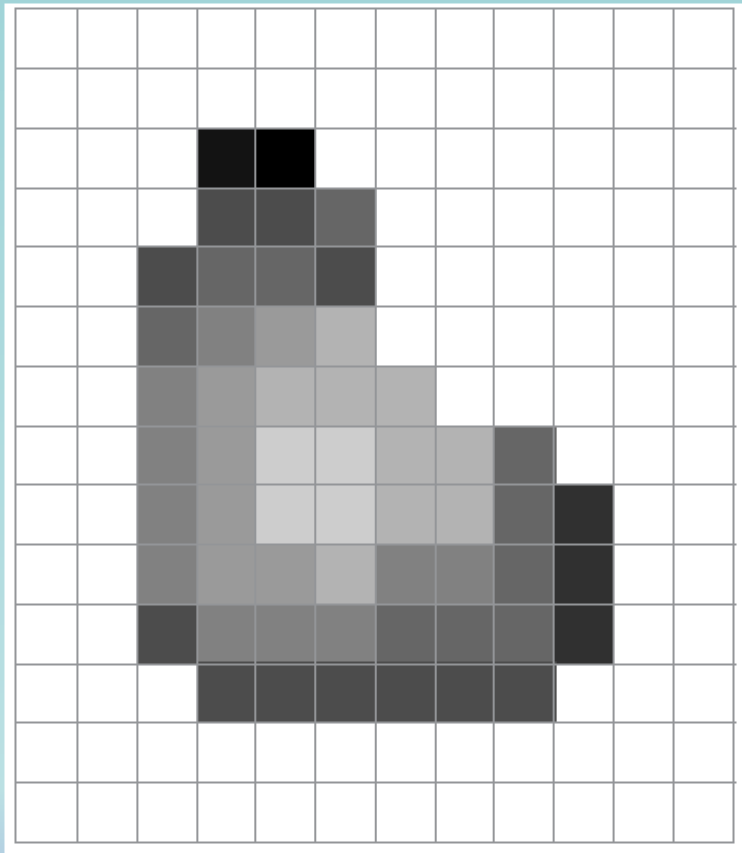
CSE-BUET

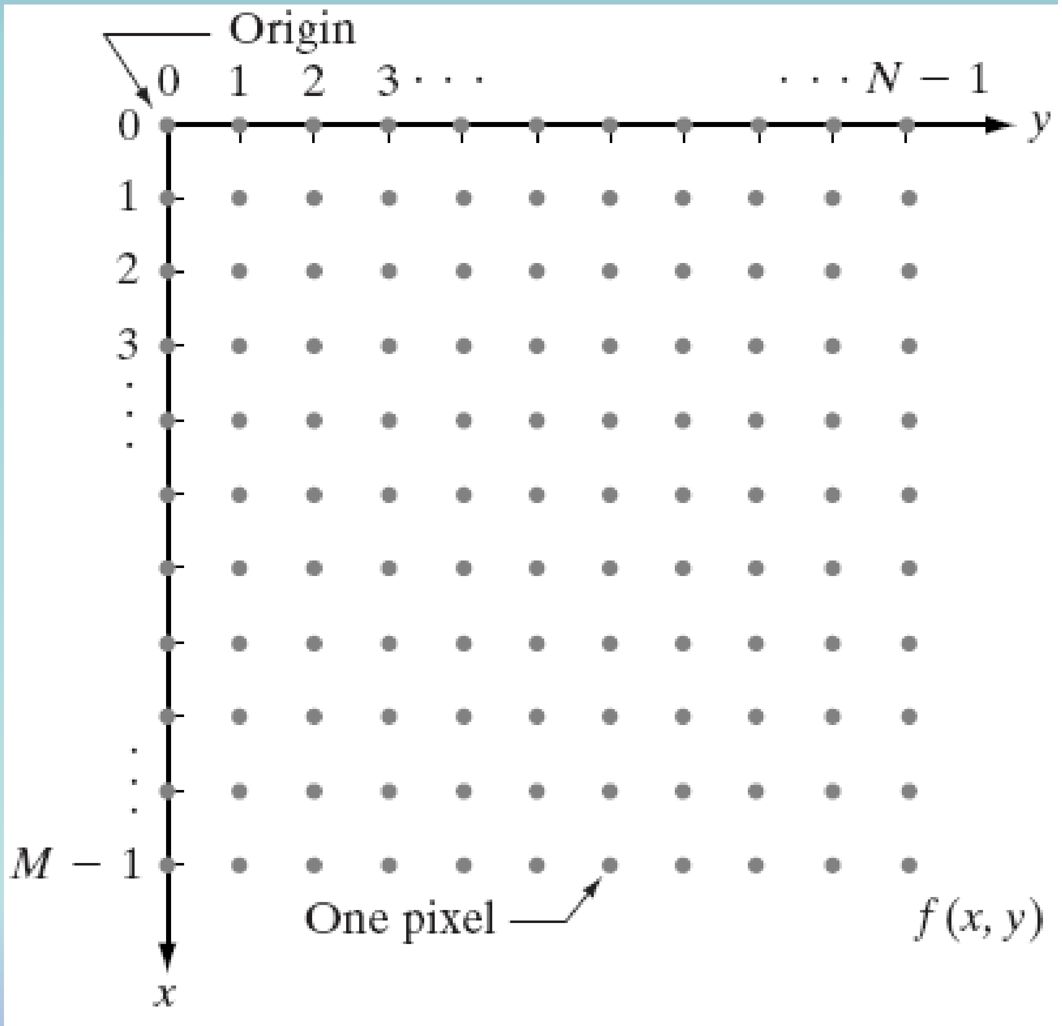
Image Digitization



CSE-BUET

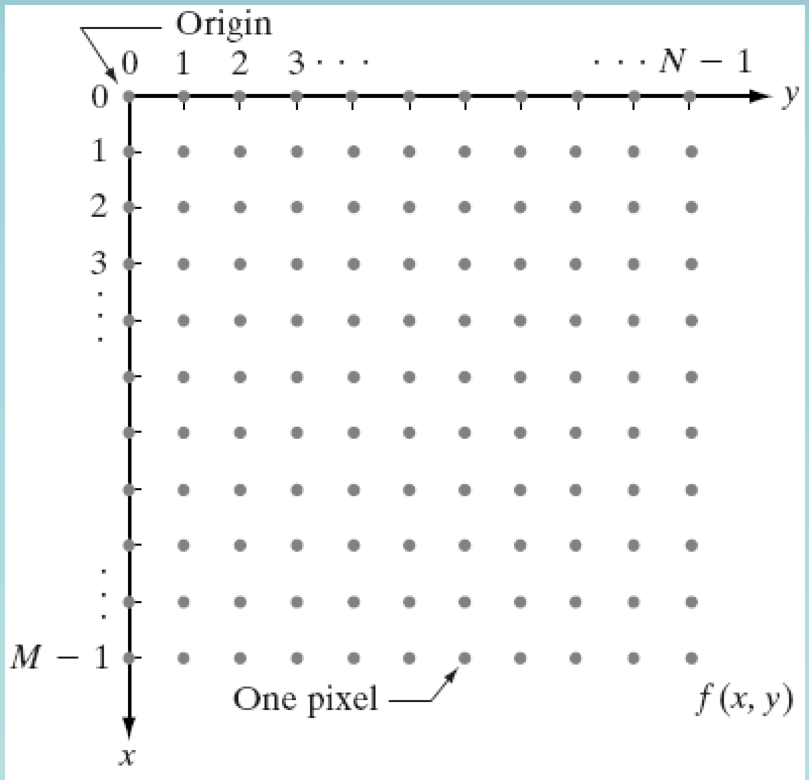
Representation of Digital Image





Matrix Representation

Representation of Digital Image

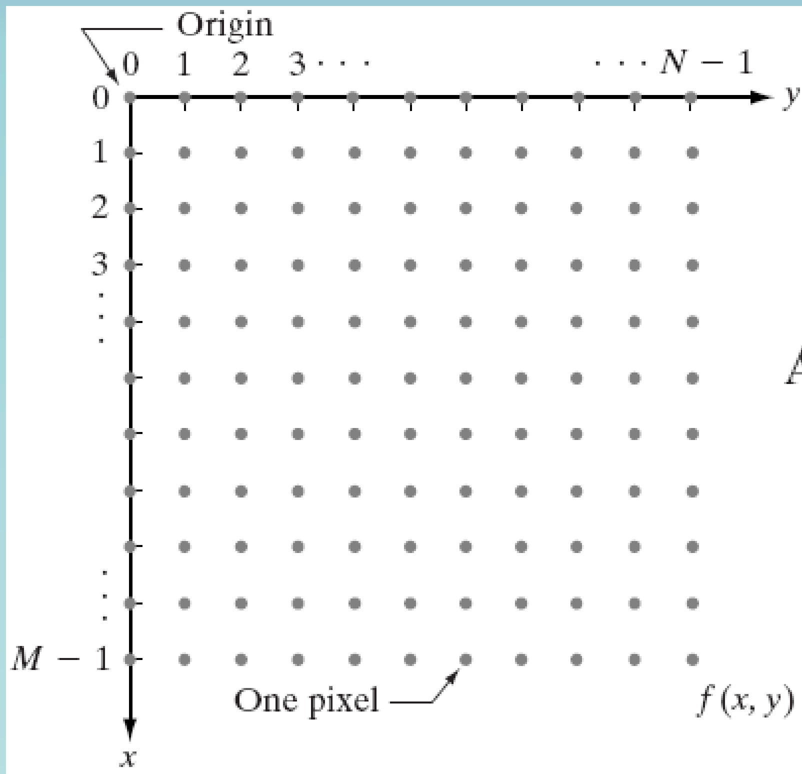


Matrix Representation

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

CSE-BUET

Representation of Digital Image



$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix}$$

Traditional Matrix Representation

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

CSE-BUET

Representation of Digital Image

Formally,

- Sampling:
  - Dividing (x, y) plane into grid with coordinate (z_i, z_j), where,

$$z_i, z_j \in Z$$

- Intermediate:
  - Assign gray level value from R to $f(z_i, z_j)$
- Quantization:
  - Assign gray level value from Z to $f(z_i, z_j)$

CSE-BUET

Representation of Digital Image

- Number of Rows (M) and Columns (N) can be any integer

- Number of gray levels, L, is usually power of 2:

$$L = 2^k$$

- L: Gray level resolution
- M, N: spatial resolution

Representation of Digital Image

- Number of Rows (M) and Columns (N) can be any integer
- Number of gray levels, L, is usually power of 2:

$$L = 2^k$$

- Bit size of an image:

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

- For Square image,

$$b = N^2 k$$

Image Sizes for Different Spatial and Gray Level Resolutions

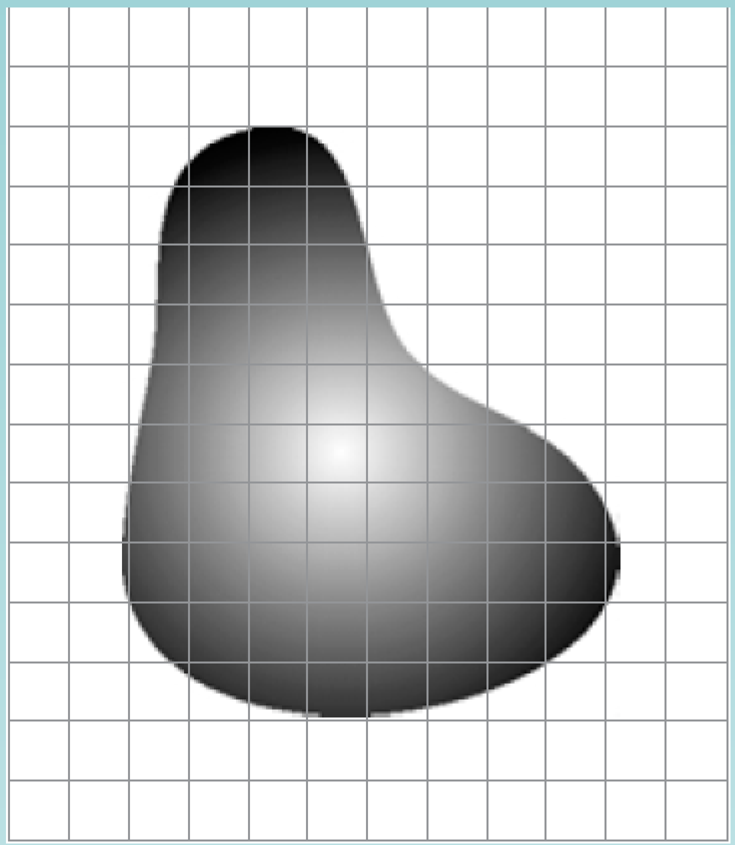
N/k	1 (L = 2)	2 (L = 4)	3 (L = 8)	4 (L = 16)	5 (L = 32)	6 (L = 64)	7 (L = 128)	8 (L = 256)
32	1,024	2,048	3,072	4,096	5,120	6,144	7,168	8,192
64	4,096	8,192	12,288	16,384	20,480	24,576	28,672	32,768
128	16,384	32,768	49,152	65,536	81,920	98,304	114,688	131,072
256	65,536	131,072	196,608	262,144	327,680	393,216	458,752	524,288
512	262,144	524,288	786,432	1,048,576	1,310,720	1,572,864	1,835,008	2,097,152
1024	1,048,576	2,097,152	3,145,728	4,194,304	5,242,880	6,291,456	7,340,032	8,388,608
2048	4,194,304	8,388,608	12,582,912	16,777,216	20,971,520	25,165,824	29,369,128	33,554,432
4096	16,777,216	33,554,432	50,331,648	67,108,864	83,886,080	100,663,296	117,440,512	134,217,728
8192	67,108,864	134,217,728	201,326,592	268,435,456	335,544,320	402,653,184	469,762,048	536,870,912

* The calculation is for square images

CSE-BUET

Effect of Spatial and Gray Level Resolutions

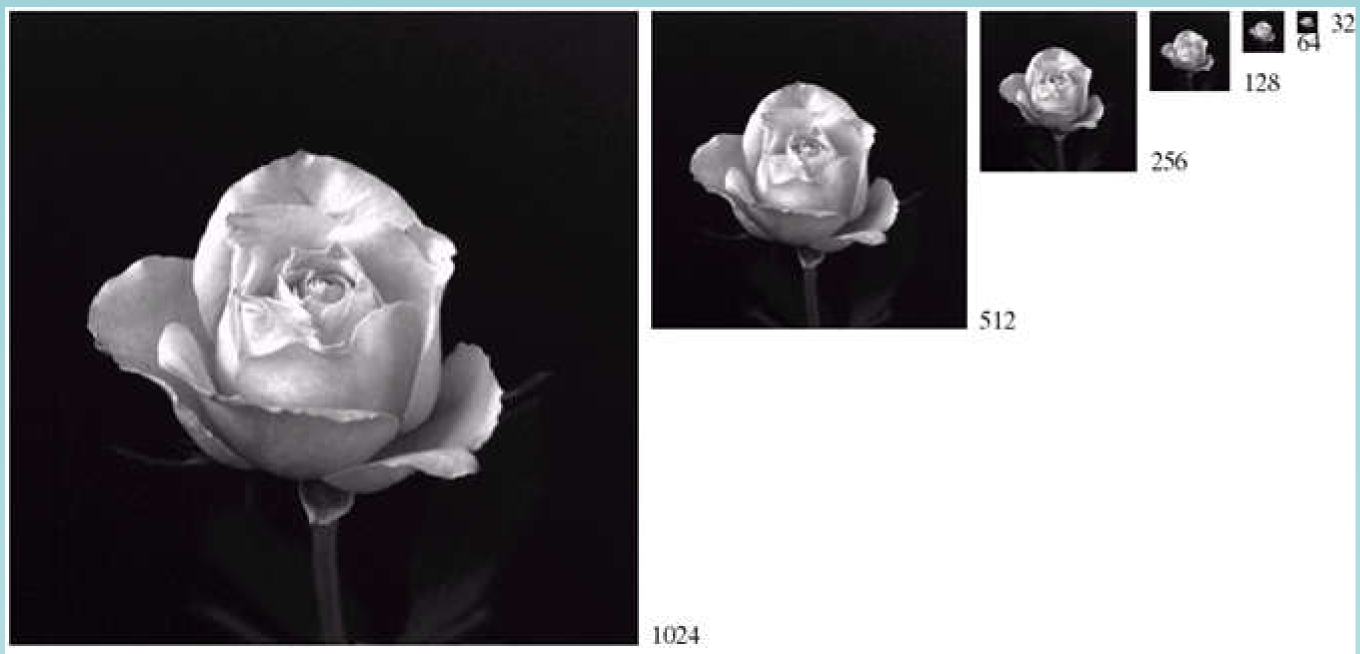
Effect of Spatial Resolutions



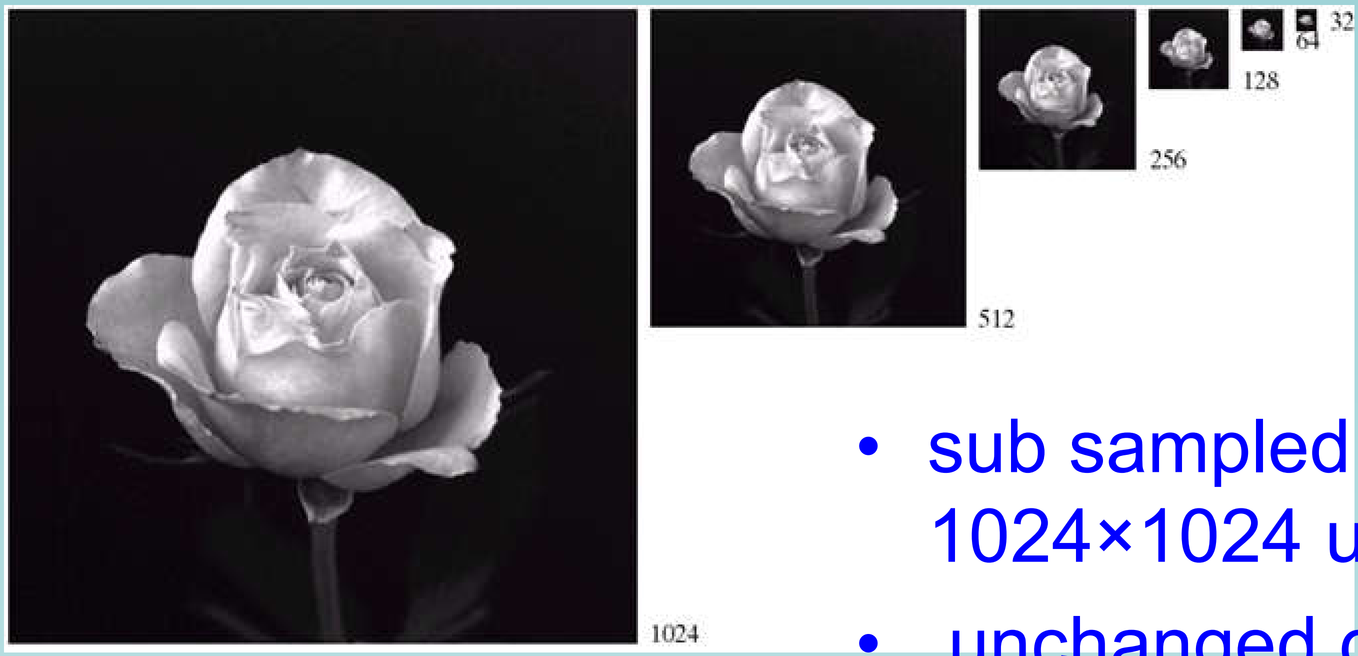
- 14×12 resolution means samples from 14×12 locations
- What happens, if we sample from 8×8 locations?

CSE-BUET

Effect of Spatial Resolutions



Effect of Spatial Resolutions



- sub sampled from 1024×1024 up to 32×32
- unchanged gray level

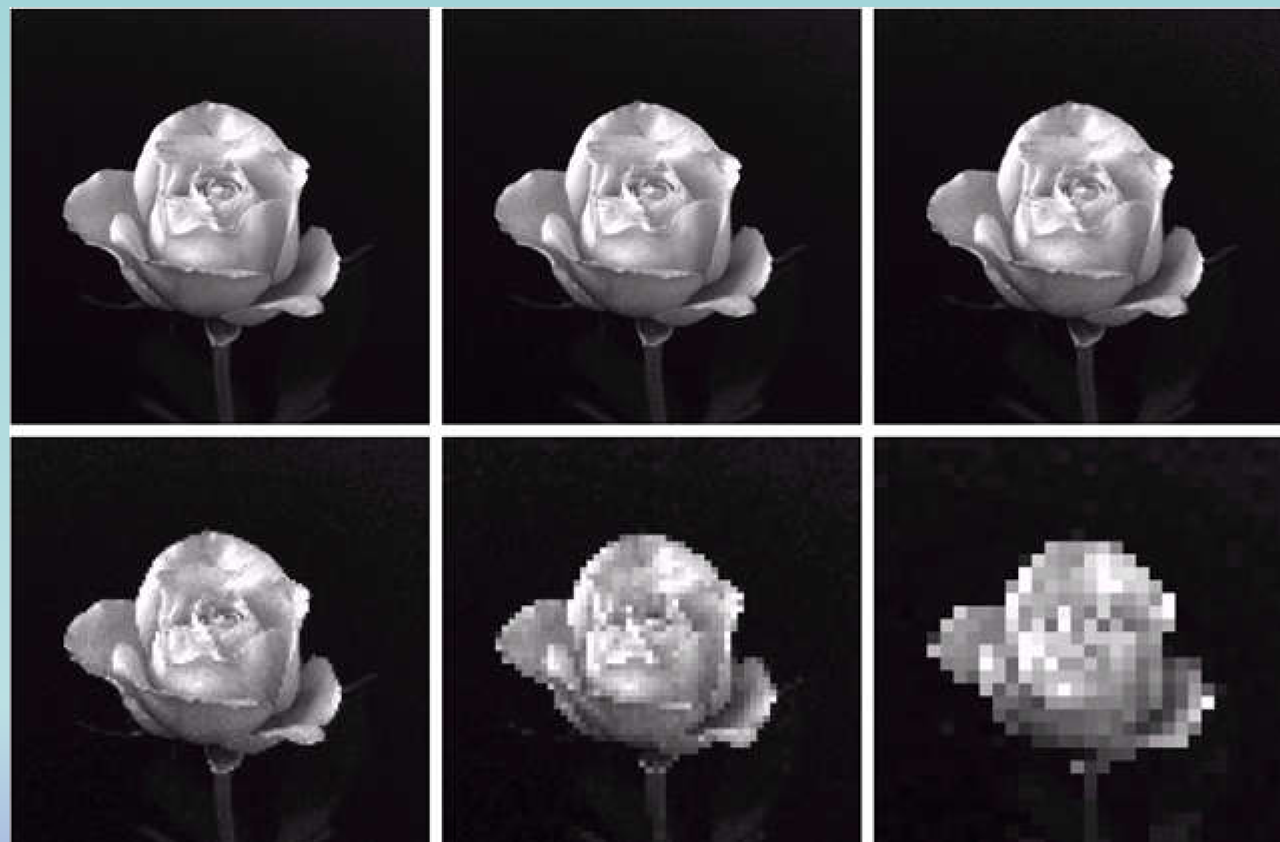
- Delete alternate row and column while sub sampling

CSE-BUET

Effect of Spatial Resolutions

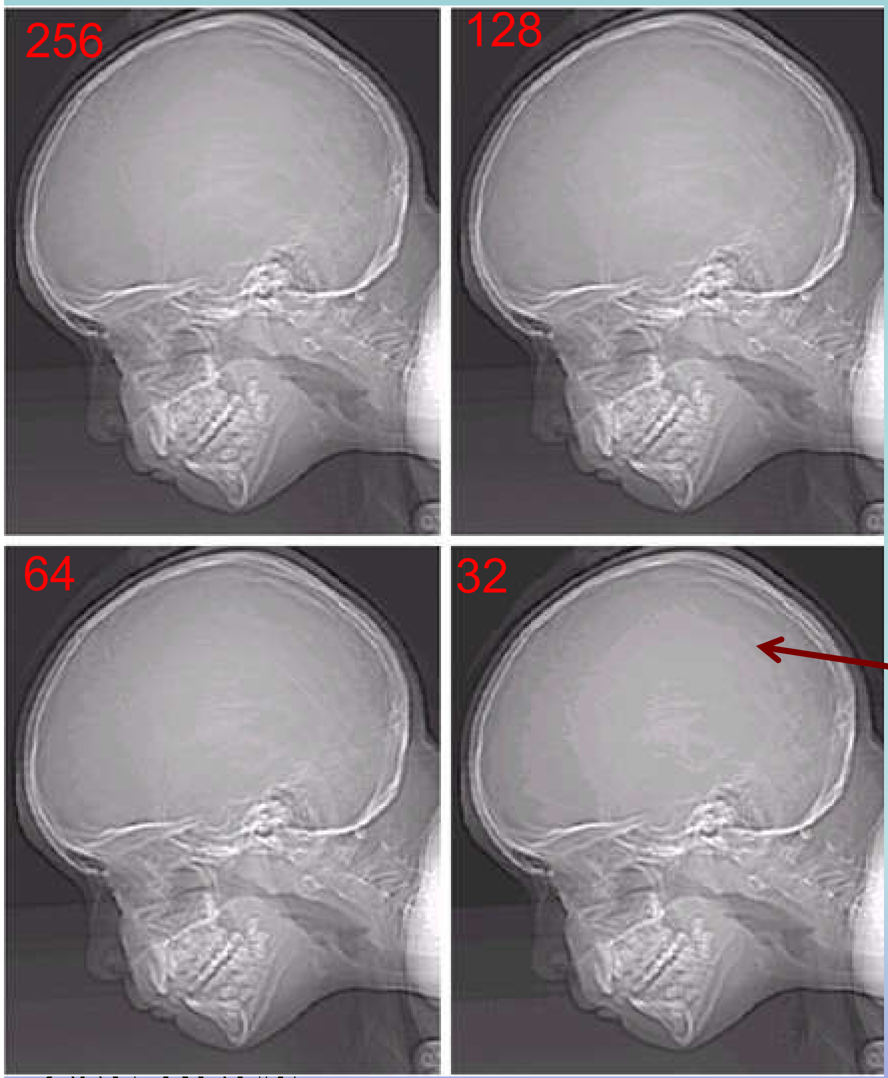
resized to 1024×1024

From → 1024×1024 512×512 256×256



From → 128×128 64×64 32×32

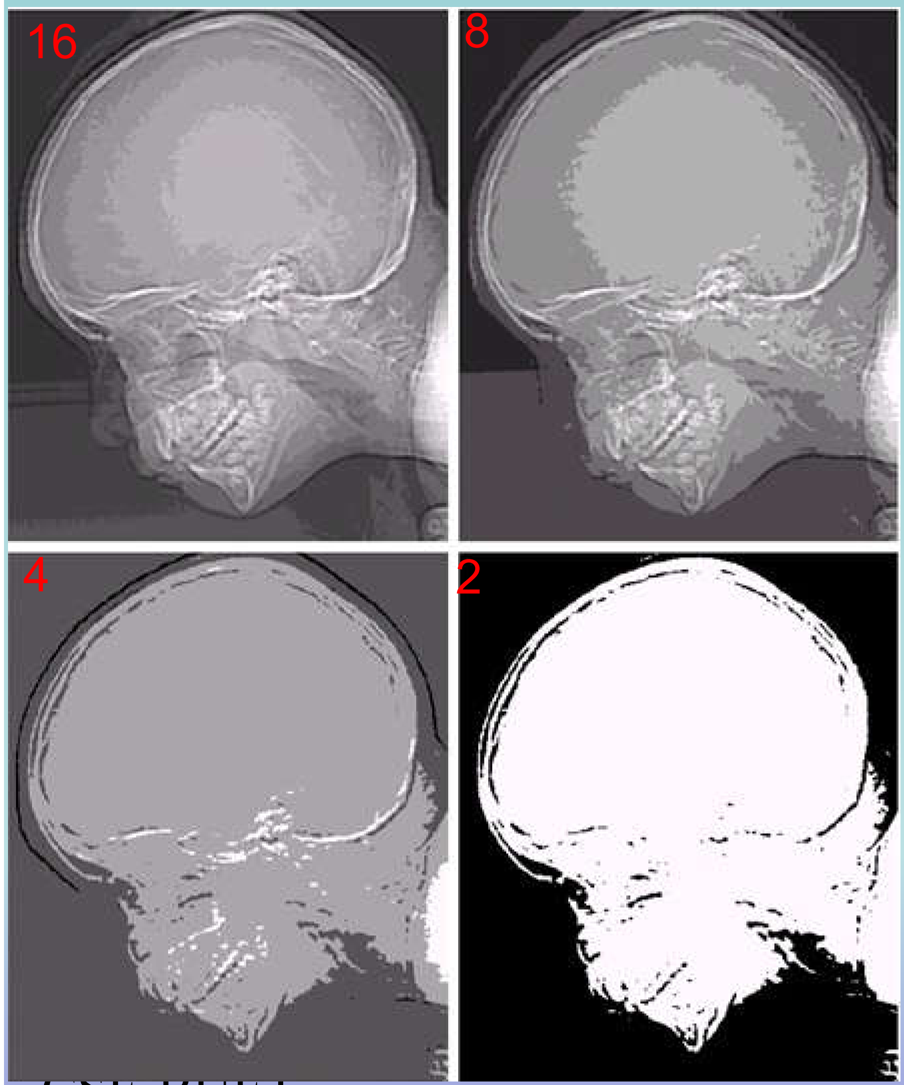
Effect of Gray level Resolutions



- unchanged spatial resolution
- Gray Level Res. changed from 256 to 32

Ridge-like structure in the smooth area

Effect of Gray level Resolutions



- Gray level changed from 16 to 2
  - Ridge-like structure is more prominent
  - Reason: insufficient number of gray levels used

Properties of Digital Images: Zooming and Shrinking

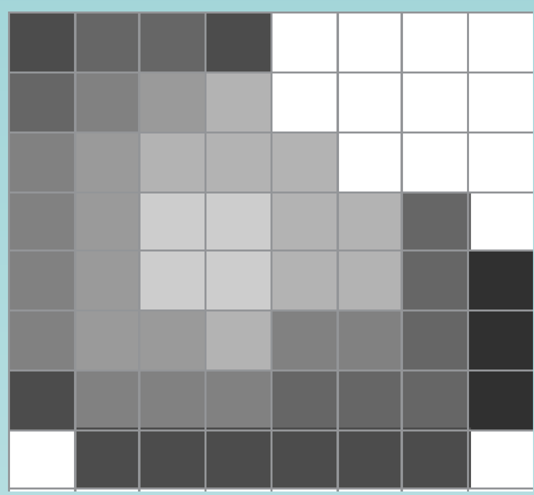
- Related to sampling and quantization, because
  - Zooming: over sampling
  - Shrinking: under sampling
- However, Zooming and Shrinking are done on digital images

Zooming

- Interpolation: *Nearest neighbor, bilinear, bicubic, wavelet based, etc*
- Pixel replication

Zooming

- Nearest neighbor interpolation

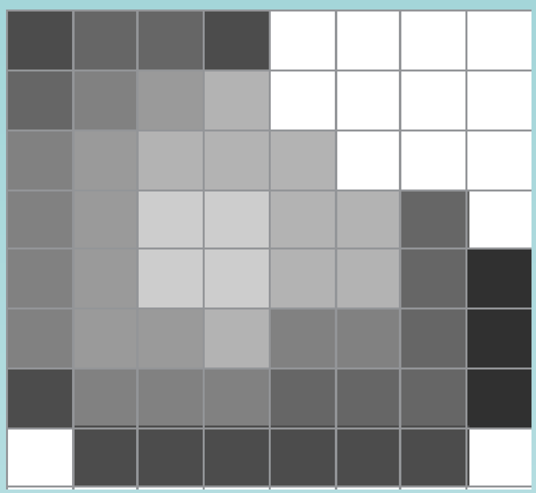


An 8X8 image

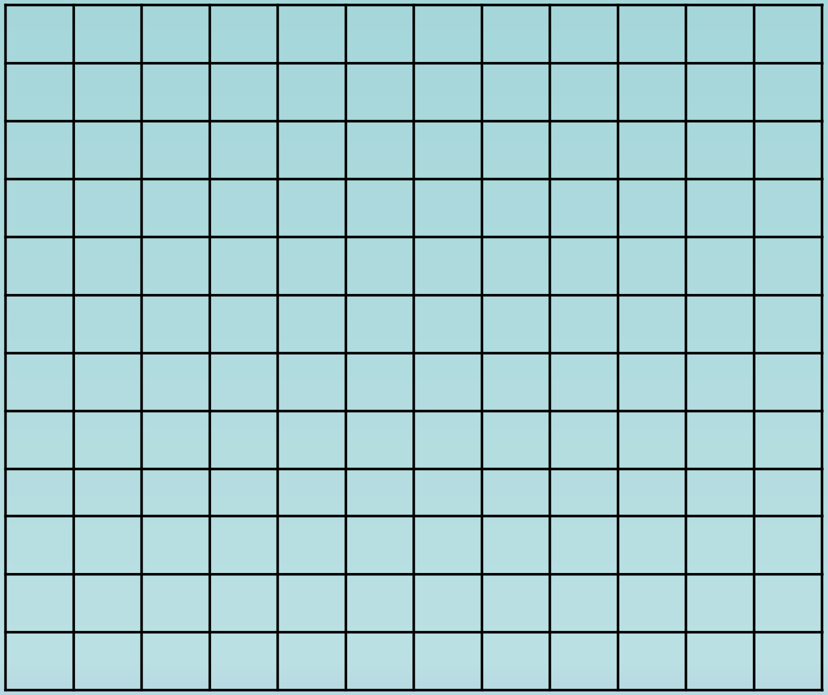
Resize this image to 12X12

Zooming

- Nearest neighbor interpolation



An 8X8 image
Resize this image to
12X12

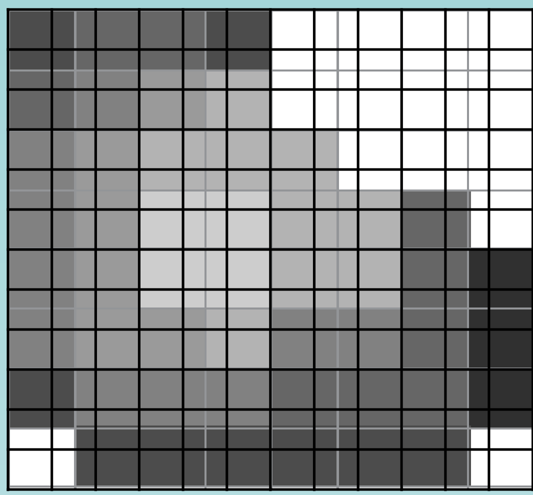


A 12X12 grid to be filled up
Overlay it to 8X8

CSE-BUET

Zooming

- Nearest neighbor interpolation



An 8X8 image
Resize this image to 12X12

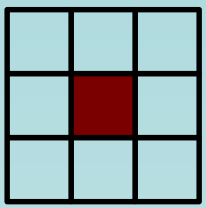
- Each grid of 12X12 fall on some of the 8X8 grid
- Fill the grid of 12X12 from the value of the nearest grid of 8X8
- Then return the 12X12 into its original size

CSE-BUET

Pixels and their Relationships

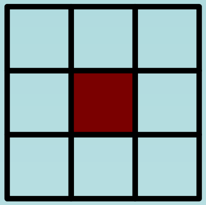
Elements of Images: Pixels and their Relationships

Neighbors of a pixel



Elements of Images: Pixels and their Relationships

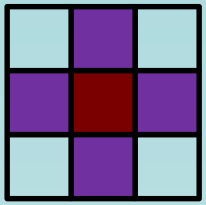
- 3 types of neighbors
  - $N_4(p)$: 4-neighbor
  - $N_D(p)$: diagonal neighbor
  - $N_8(p)$: 8-neighbor



Elements of Images: Pixels and their Relationships

- $N_4(p)$: 4-neighbor

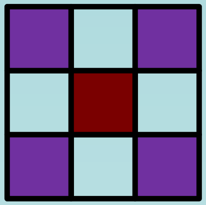
Defined as the pixels at (x+1, y), (x-1, y), (x, y+1), (x, y-1)



Elements of Images: Pixels and their Relationships

- $N_D(p)$: diagonal neighbor

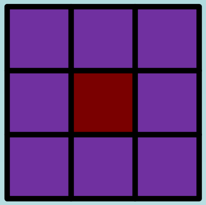
  Defined as the pixels at (x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)



Elements of Images: Pixels and their Relationships

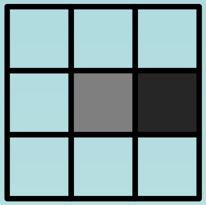
- $N_8(p)$: 8-neighbor

  Defined as the pixels at (x+1, y), (x-1, y), (x, y+1), (x, y-1), (x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)



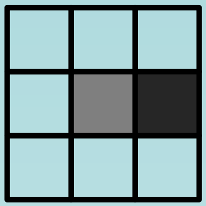
Pixel Adjacencies

- Adjacencies depends on both
  - *neighborhood*
  - *Pixel gray values*



Pixel Adjacencies

- Adjacencies depends on both
  - *neighborhood*
  - *Pixel gray values*

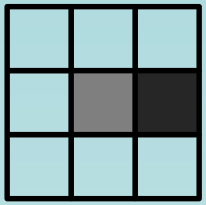


Adjacent pixels must be neighbors and have gray values from the same set, *V*

CSE-BUET

Pixel Adjacencies

- Adjacencies depends on both
  - *neighborhood*
  - *Pixel gray values*



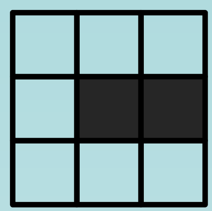
Not adjacent

Adjacent pixels must be neighbors and have gray values from the same set, V

CSE-BUET

Pixel Adjacencies

- Adjacencies depends on both
  - *neighborhood*
  - *Pixel gray values*



Adjacent

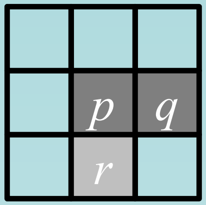
Adjacent pixels must be neighbors and have gray values from the same set, *V*

Pixel Adjacencies

- 3 types of adjacencies
  - 4-adjacency
  - 8-adjacency
  - *m*-adjacency

Pixel Adjacencies

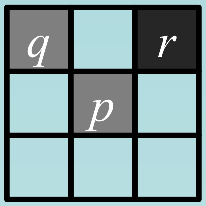
- 4-adjacency
  - Two pixels *p*, *q* are 4-adjacent if
    - *q* in $N_4(p)$
    - *p*, *q* have values from set *V*



CSE-BUET

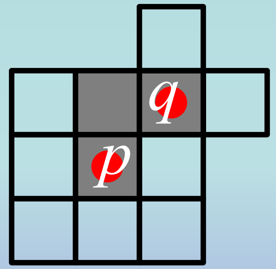
Pixel Adjacencies

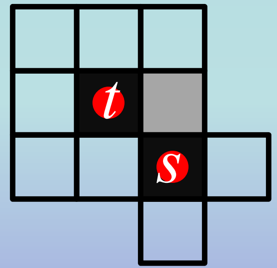
- 8-adjacency
  - Two pixels *p*, *q* are 8-adjacent if
    - *q* in $N_8(p)$
    - *p*, *q* have values from set *V*



Pixel Adjacencies

- *m*-adjacency
  - Two pixels *p*, *q* are *m*-adjacent if
    - *p*, *q* have values from set *V*, and
      - q in $N_4(p)$, or
      - q in $N_D(p)$ and $N_4(p) \cap N_4(q)$ has no pixel with value from V





CSE-BUET

Other graph Theoretic Properties

- (*Digital*) *path*
  - A sequence of pixels

$$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$$

  - where, (x_{i-1}, y_{i-1}) and (x_i, y_i) are adjacent

Other graph Theoretic Properties

- (*Digital*) *path*
  - A sequence of pixels

$$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$$

  - where, (x_{i-1}, y_{i-1}) and (x_i, y_i) are adjacent

- 4-, 8-, *m*-path can be defined based on adjacency

CSE-BUET

Other graph Theoretic Properties

- Closed path

is a path

$$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$$

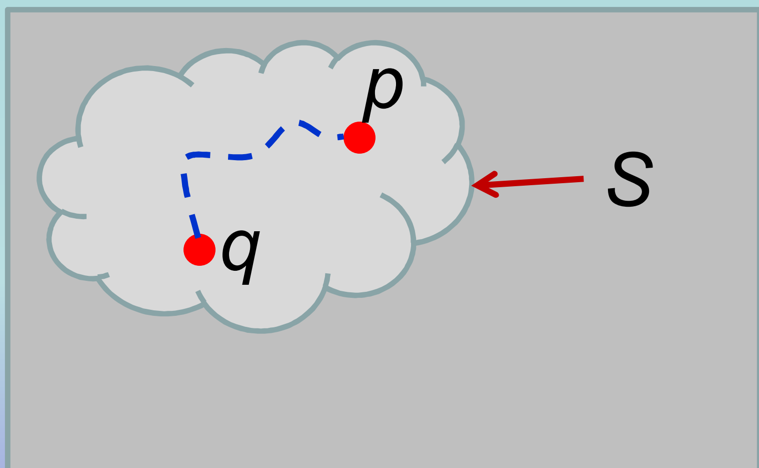
with $(x_0, y_0) = (x_n, y_n)$

Definition of Region

- Connected in a set of pixels, *S*
  - Two pixels *p* and *q* are connected in *S*

    if

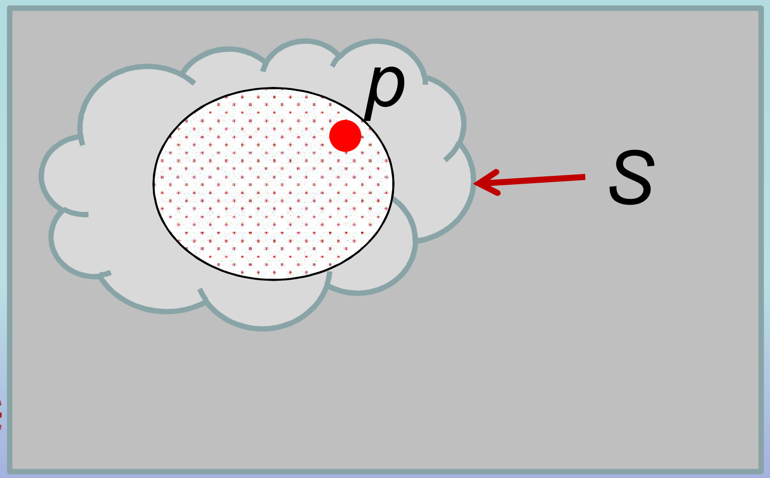
    there is a path between *p* and *q* entirely in *S*



CSE-BUET

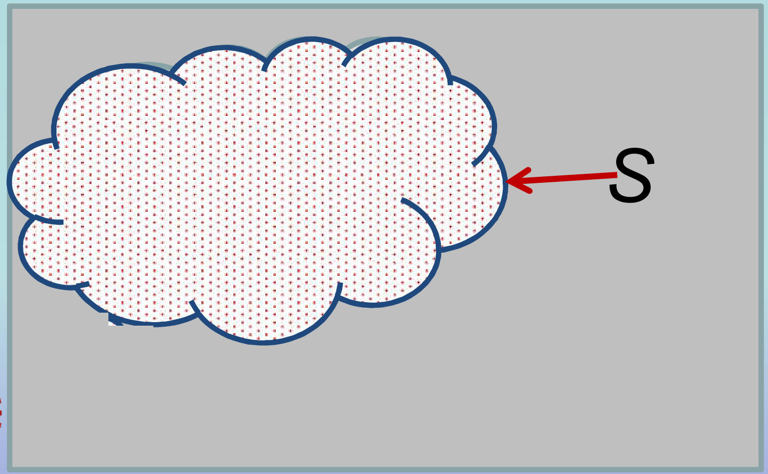
Definition of Region

- Connected Component in a set of pixels, *S*
  - Set of pixels connected to *p*



Definition of Region

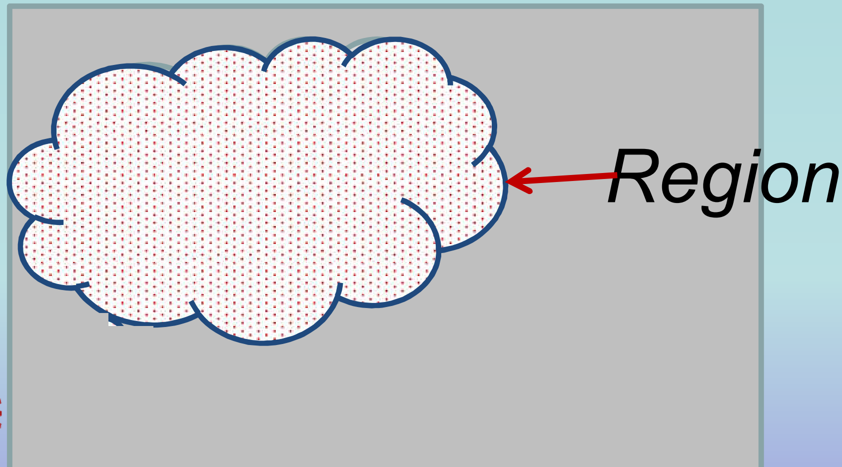
- Connected Set, *S*
  - If all pixels of *S* are connected to each other



CSE-BUET

Definition of Region

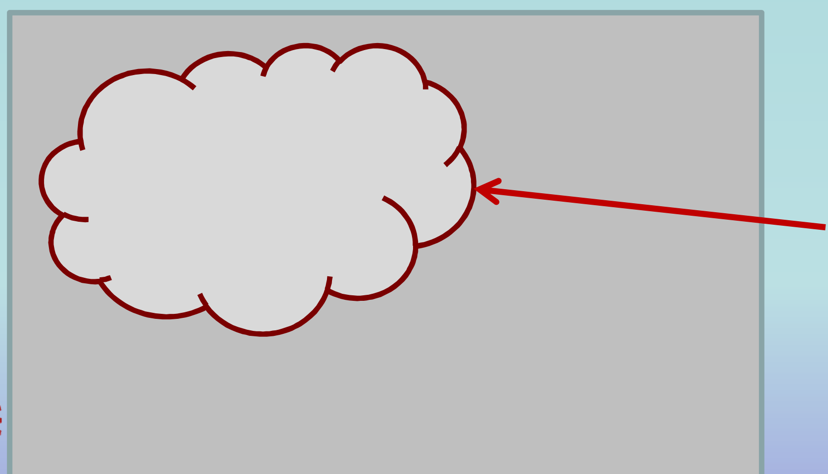
- Region is a Connected Set



CSE-BUET

Definition of Region

- Boundary of a region, *R*
  - Set of pixels whose at least one neighbor is not in *R*



Boundary or contour

CSE-BUET

Identifying a Region: Image Segmentation

- Find edges and link them
- Region growing
- Water shading
- Thresholding
- . . .

Data Structure in Image Processing

CSE-BUET

Levels of Data Structure in DIP

- Matrix
  - Contains original data
    - Gray levels or color values
  - Direct output of *image acquisition devices*

62 63 63 65 66 63 61 63 . .
63 61 59 64 63 60 61 64. .
65 63
63 67
63 62
63 57
62 61
65 66

29 31 34 30 . . .
31 31 32 30. . .
29 30
29 29
.
.

64 64 64 . .
62 64 64 . .
60 62 63 . .
.

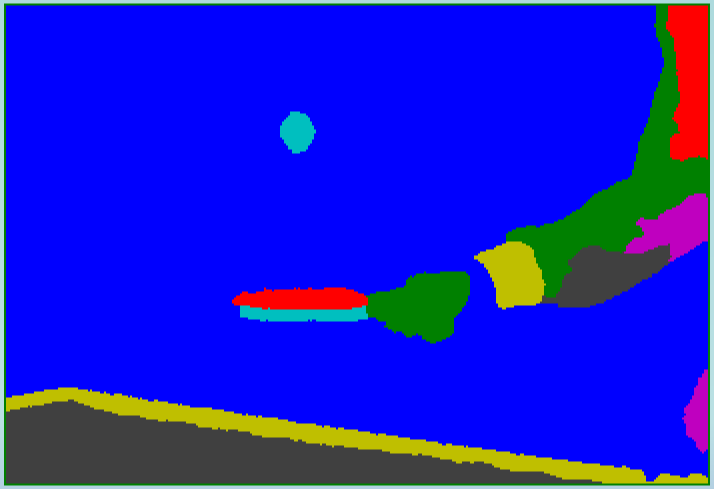
CSE-BUET

Levels of Data Structure in DIP

- Segmented images
  - Parts of image
  - Group of adjacent pixels with certain common attributes
  - Represented by
    - extracted features
    - map

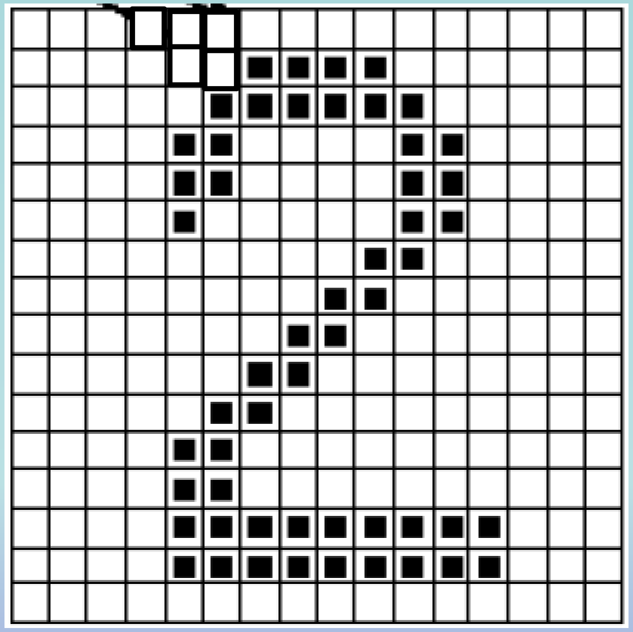
Levels of Data Structure in DIP

- Geometric representation
  - 2D or 3D shapes
  - Sometimes difficult to quantify



Chain Codes

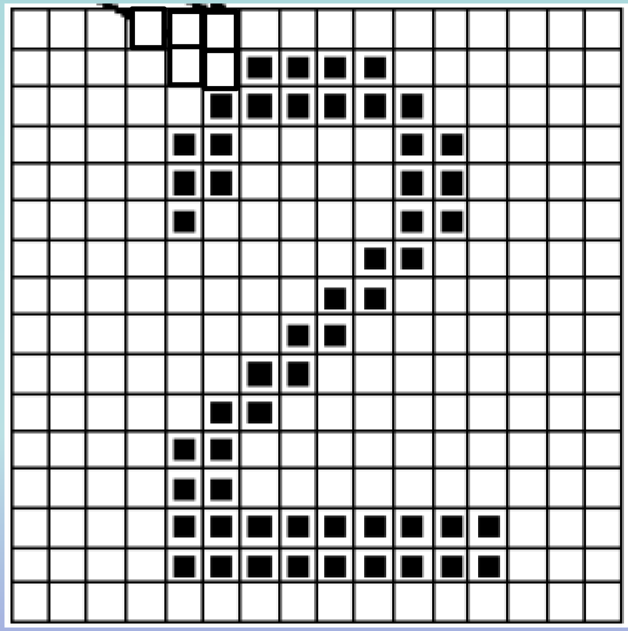
- Represents object's contour
- Symbols represents pixels and its neighborhood
- A reference indicates the starting of the codes

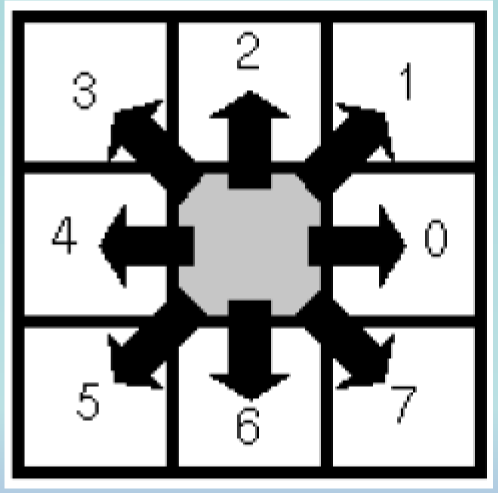


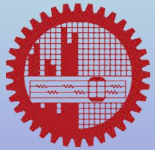
CSE-BUET

Chain Codes

- Represents object's contour
- Symbols represents pixels and its neighborhood
- A reference indicates the starting of the codes

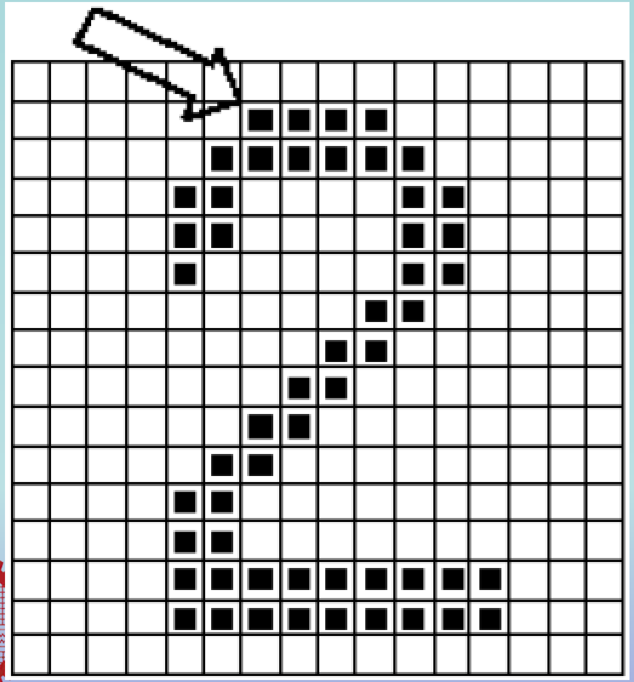


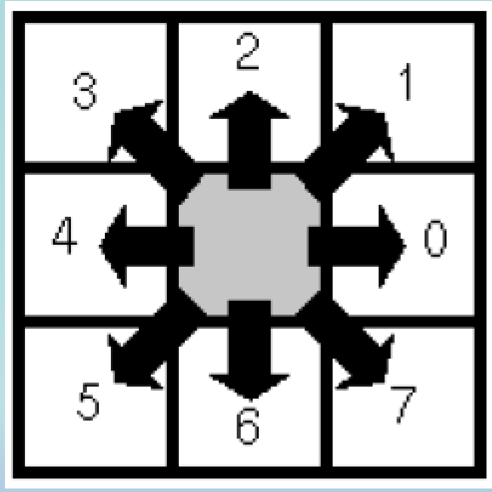




Chain Codes

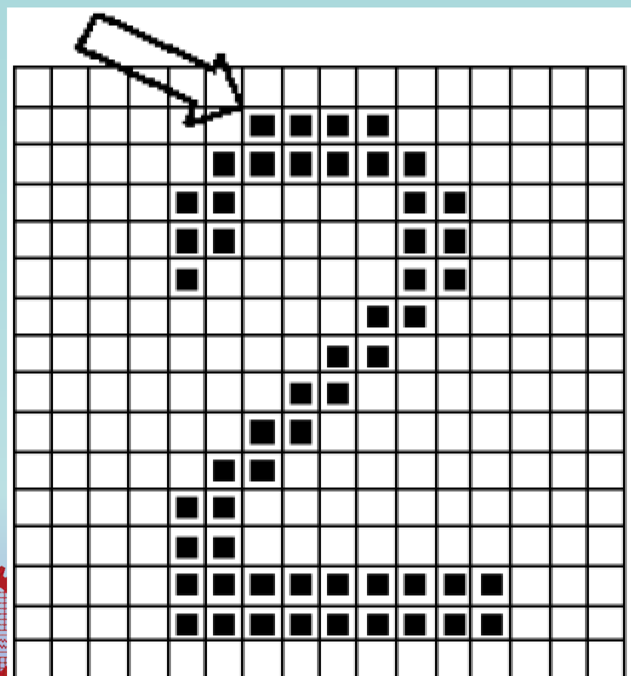
- Represents object's contour
- Symbols represents pixels and its neighborhood
- A reference indicates the starting of the codes

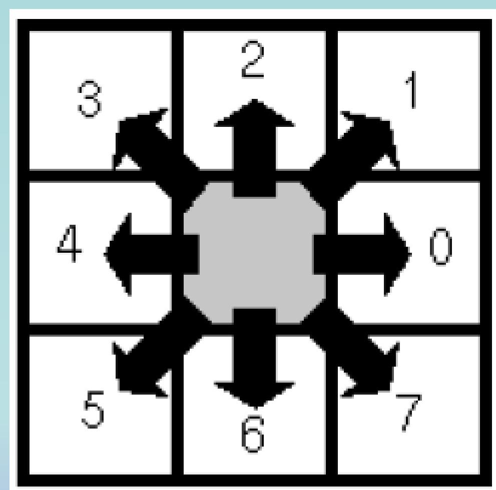




Chain Codes

- Represents object's contour
- Symbols represents pixels and its neighborhood
- A reference indicates the starting of the codes

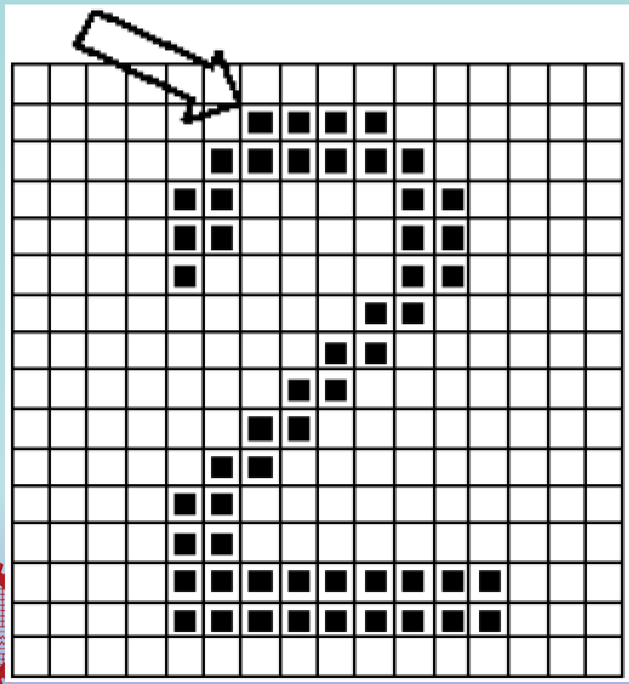


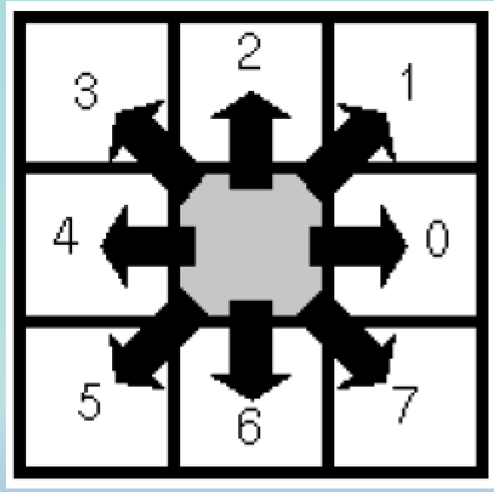


00007766555555660000000644444
444222111112234445652211

Chain Codes

- Can be represented as a 1D array
- Can encode local information



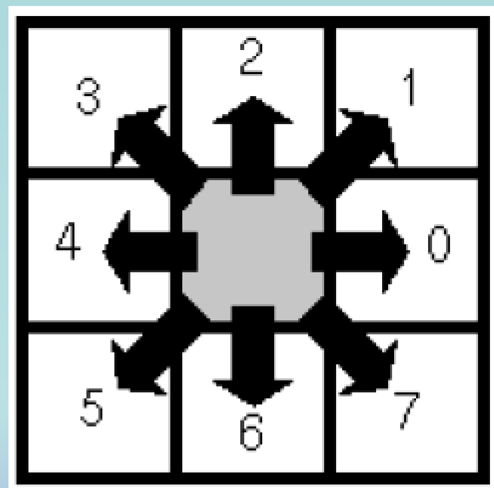


00007766555555660000000644444
444222111112234445652211

Chain Codes

- Difficult to say about global info:
  - How is the structure?

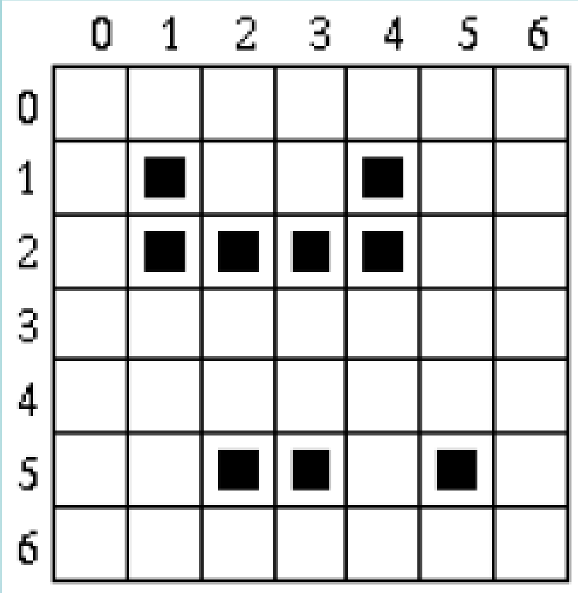
00007766555555660000000644444
4442221111112234445652211 = ?



CSE-BUET

Run Length Coding

- Compact encoding of a sparse matrix
- Used in image compression

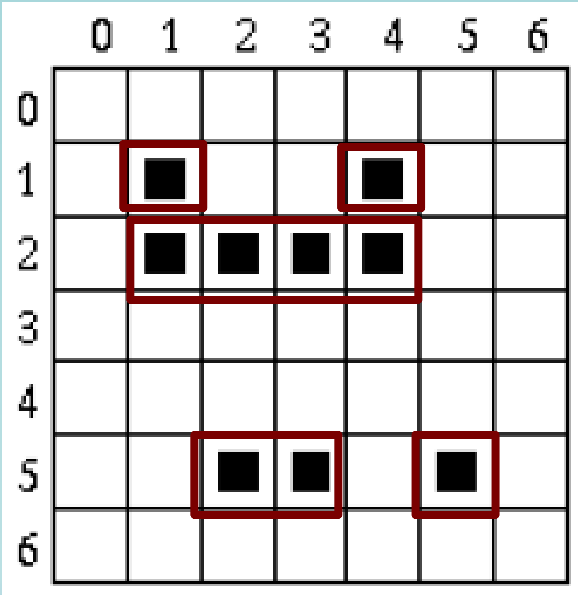


A binary image

Run Length Coding

- Compact encoding of a sparse matrix
- Used in image compression

Identify and encode runs

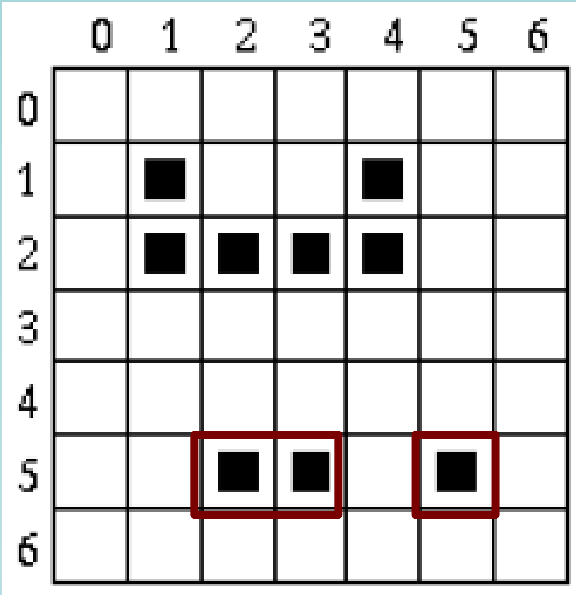


A binary image

CSE-BUET

Run Length Coding

- Encodes only area of object
- Form *list of lists*
- Each list contains
  - A row number
  - multiple pair of 2 integers
    - Staring and beginning of run
  - Example:
  - 5, (2,3), (5,5)

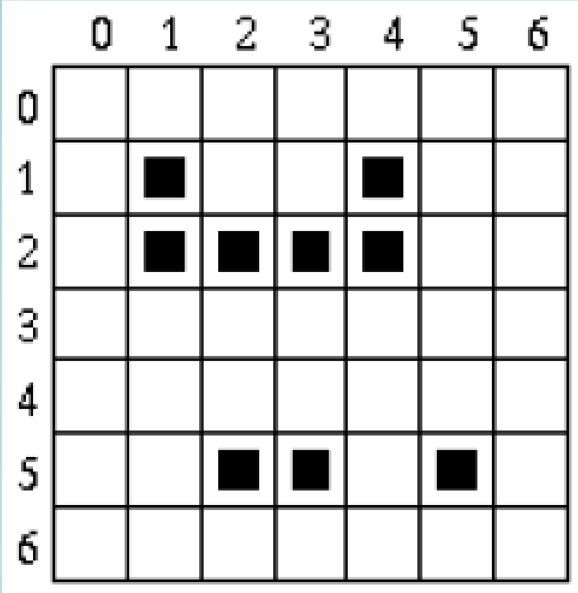


A binary image

CSE-BUET

Run Length Coding

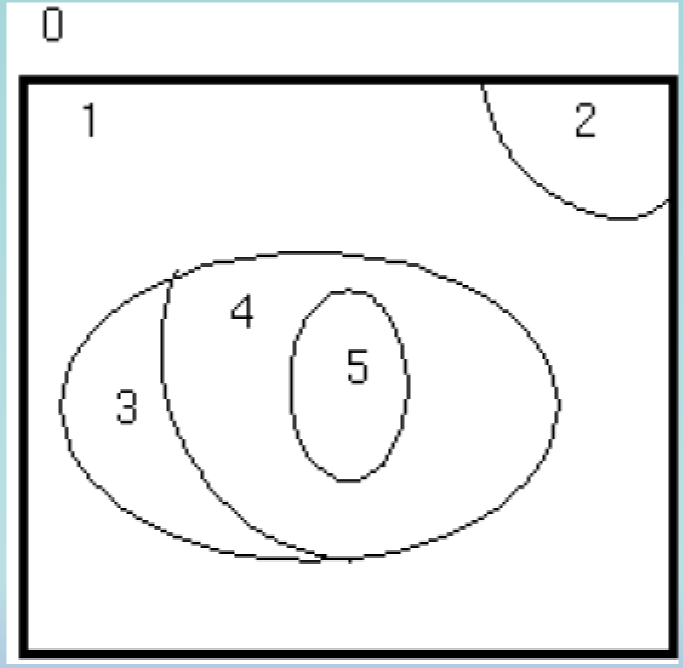
- Encodes only area of object
- Form list of lists
- Each list contains
  - A row number
  - multiple pair of 2 integers
    - Staring and beginning of run
  - Example:
    - 5, (2,3), (5,5)
  - Entire image
    - ((11144)   (214)   (52355)   )

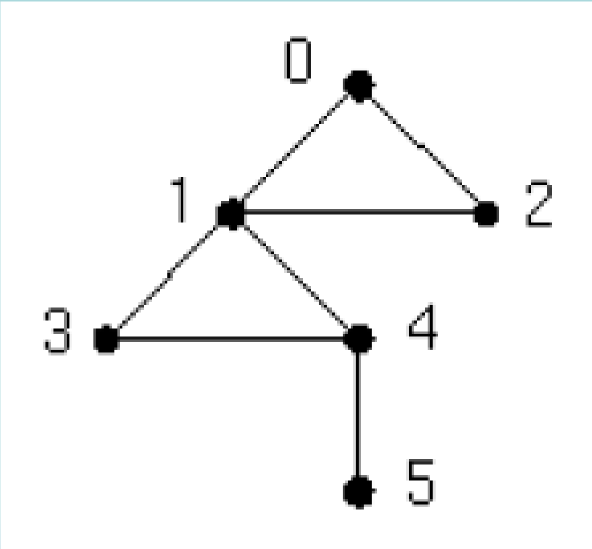


A binary image

Topological Structure

- Uses graph to represent regions





CSE-BUET

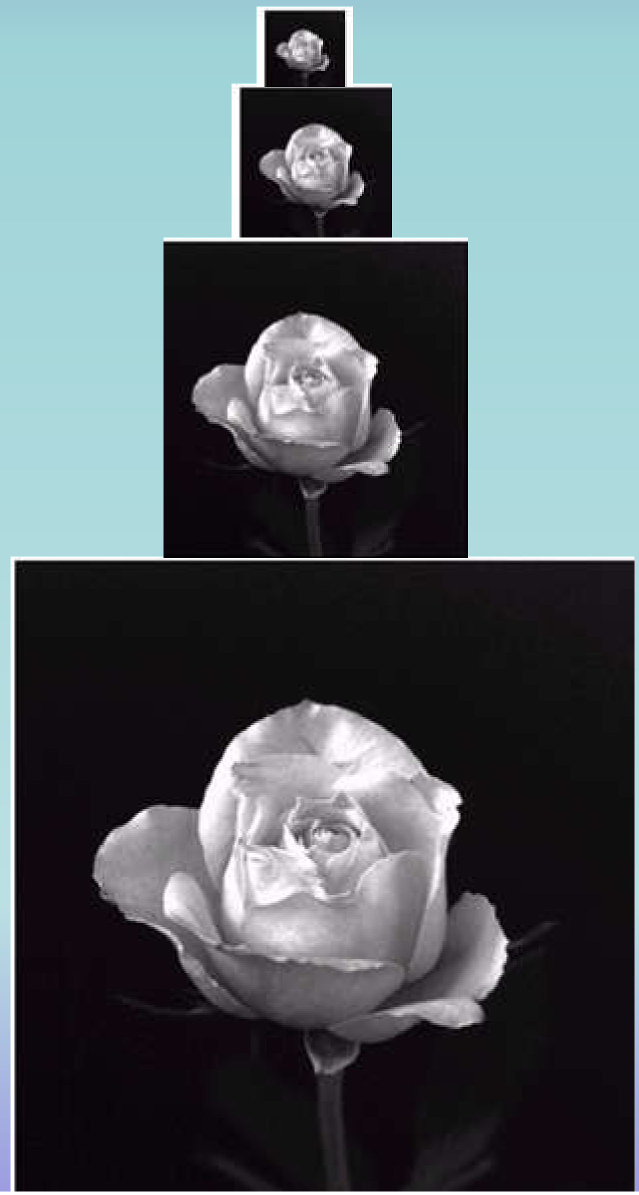
Hierarchical Structure

- Used in multi-resolution image analysis
- Also used to avoid expensive computation
  - Uses only the necessary part of the image
- Several variations exist
  - M-pyramids
  - T-Pyramids

M-Pyramids

- Pyramids of matrix with varying dimensions
- M_L, $M_{L\text{-}1}$, . . ., M_0
- Size reduces as power of 2
- M_L = original size
- M_0 = single pixel
- Total pixels $N^2\,(1 + \frac{1}{4} + \frac{1}{16} + \ldots) \approx 1.33\,N^2$

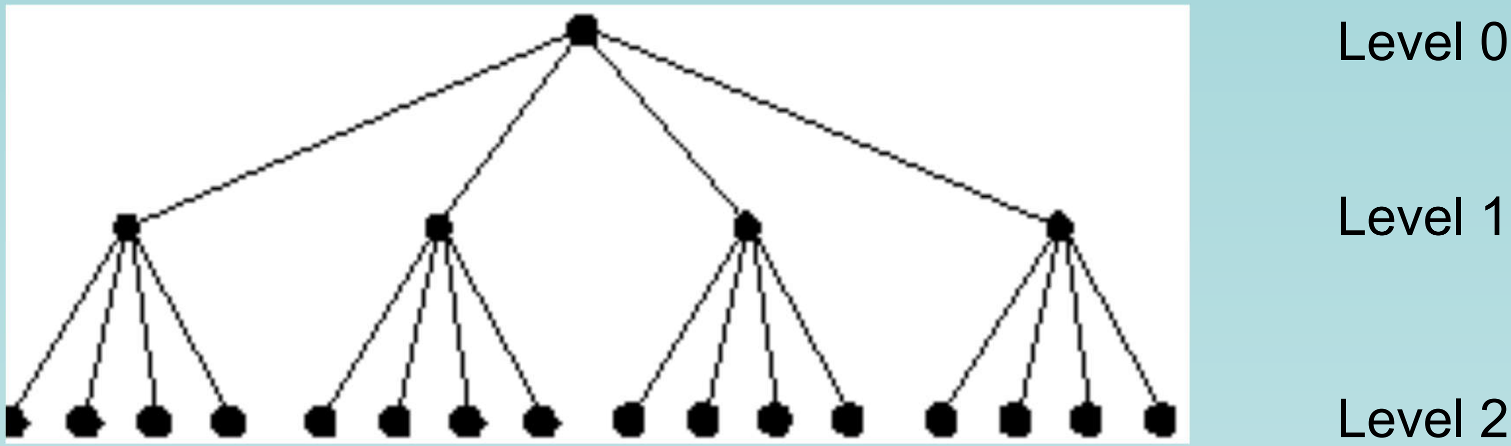
M-Pyramids



CSE-BUET

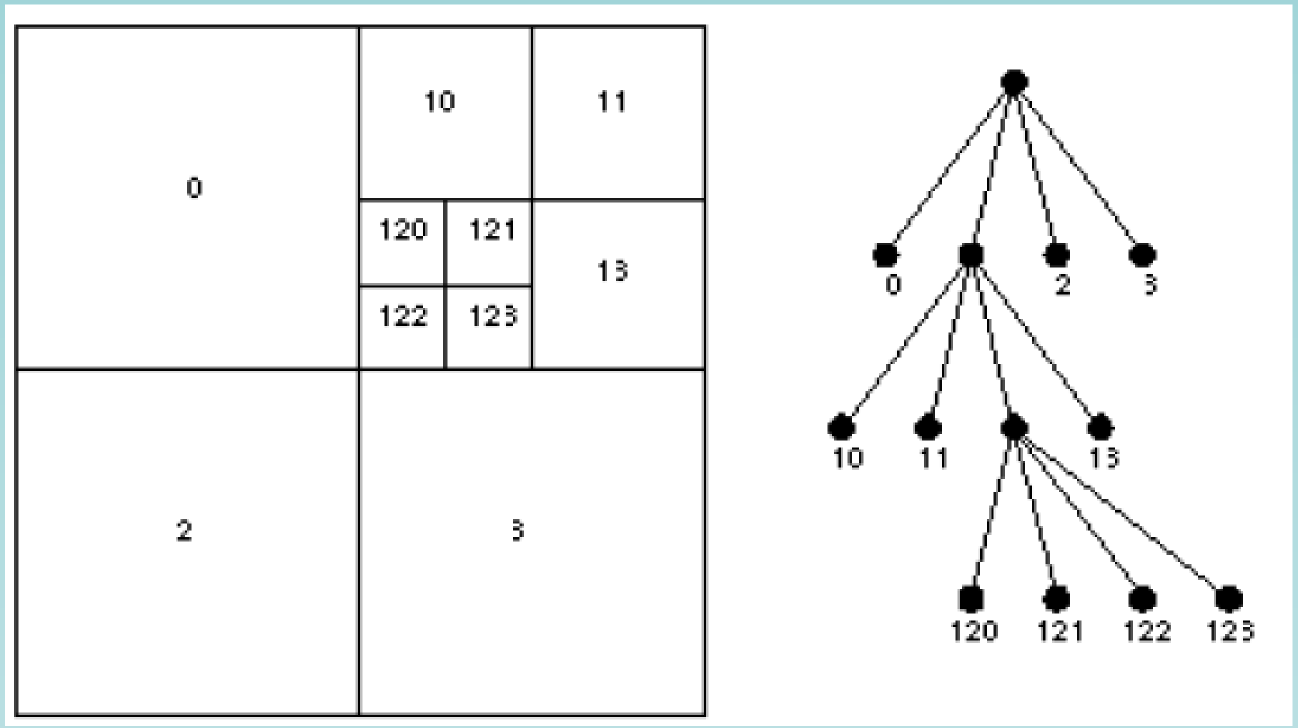
T-Pyramids

- Useful, when we *simultaneously* need multiple resolutions of the same image



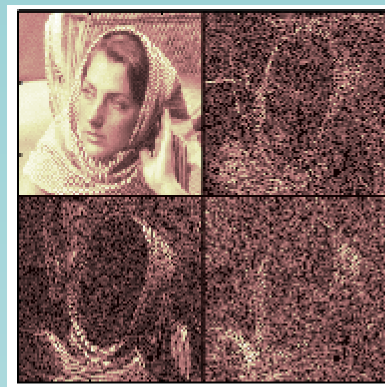
CSE-BUET

T-Pyramids: Quad tree

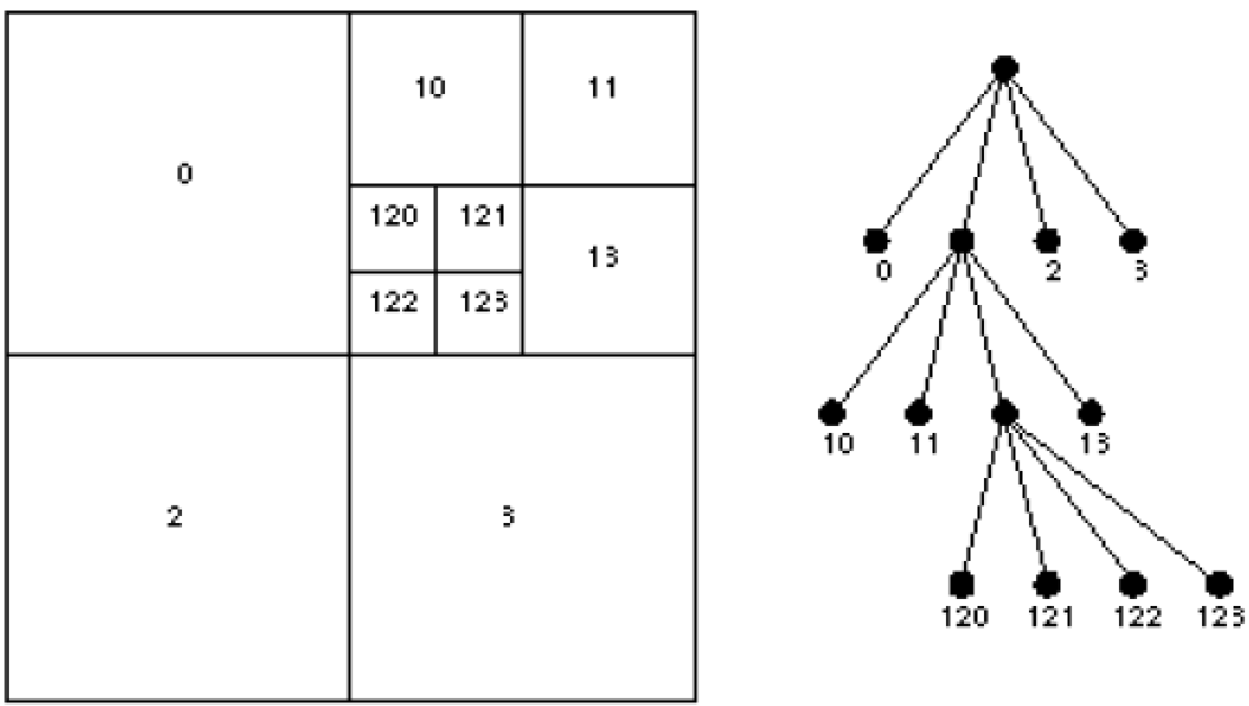


CSE-BUET

T-Pyramids: Quad tree



Wavelet Transform



CSE-BUET

1-D Wavelet Transform

22	24	48	50

1-D Wavelet Transform

22	24	48	50

After 1d Wavelet transform

23	**49**	**-1**	**-1**

CSE-BUET

1-D Wavelet Transform

22	24	48	50

After 1d Wavelet transform

23	**49**	**-1**	**-1**

1-D Wavelet Transform

22	24	48	50

After 1d Wavelet transform

23	**49**	**-1**	**-1**

average **difference**

2-D Wavelet Transform

25	26	26	21	23	30	25	24
30	30	23	27	28	29	25	23
22	28	27	25	25	27	24	22
27	29	29	29	29	26	24	27
22	21	27	22	26	23	20	29
24	28	25	21	28	22	28	23
22	25	24	26	21	29	29	21
25	20	27	20	26	20	22	28

2-D Wavelet Transform

25	26	26	21	23	30	25	24
30	30	23	27	28	29	25	23
22	28	27	25	25	27	24	22
27	29	29	29	29	26	24	27
22	21	27	22	26	23	20	29
24	28	25	21	28	22	28	23
22	25	24	26	21	29	29	21
25	20	27	20	26	20	22	28

After 1d Wavelet transform horizontally

26	24	27	25	-1	3	-4	-1
30	25	29	24	0	-2	-1	-1
25	26	26	23	-3	1	-1	-1
28	29	28	26	-1	0	2	2
22	25	25	25	1	3	2	5
26	23	25	26	-2	2	3	-3
24	25	25	25	-2	-1	-4	-4
23	24	23	25	3	4	3	3

2-D Wavelet Transform

25	26	26	21	23	30	25	24
30	30	23	27	28	29	25	23
22	28	27	25	25	27	24	22
27	29	29	29	29	26	24	27
22	21	27	22	26	23	20	29
24	28	25	21	28	22	28	23
22	25	24	26	21	29	29	21
25	20	27	20	26	20	22	28

After 1d Wavelet transform horizontally

26	24	27	25	-1	3	-4	-1
30	25	29	24	0	-2	-1	-1
25	26	26	23	-3	1	-1	-1
28	29	28	26	-1	0	2	2
22	25	25	25	1	3	2	5
26	23	25	26	-2	2	3	-3
24	25	25	25	-2	-1	-4	-4
23	24	23	25	3	4	3	3

CSE-BUET

2-D Wavelet Transform

We will transform this vertically

26	24	27	25	-1	3	-4	-1
30	25	29	24	0	-2	-1	-1
25	26	26	23	-3	1	-1	-1
28	29	28	26	-1	0	2	2
22	25	25	25	1	3	2	5
26	23	25	26	-2	2	3	-3
24	25	25	25	-2	-1	-4	-4
23	24	23	25	3	4	3	3

2-D Wavelet Transform

26	24	27	25	-1	3	-4	-1
30	25	29	24	0	-2	-1	-1
25	26	26	23	-3	1	-1	-1
28	29	28	26	-1	0	2	2
22	25	25	25	1	3	2	5
26	23	25	26	-2	2	3	-3
24	25	25	25	-2	-1	-4	-4
23	24	23	25	3	4	3	3

After 1d Wavelet transform vertically

28	25	28	25	-1	1	-3	-1
27	28	27	25	-2	1	1	1
24	24	25	26	-1	3	3	1
24	25	24	25	1	2	-1	-1
-2	-1	-1	1	-1	3	-2	0
-2	-2	-1	-2	-1	1	-2	-2
-2	1	0	-1	2	1	-1	4
1	1	1	0	-3	-3	-4	-4

2-D Wavelet Transform

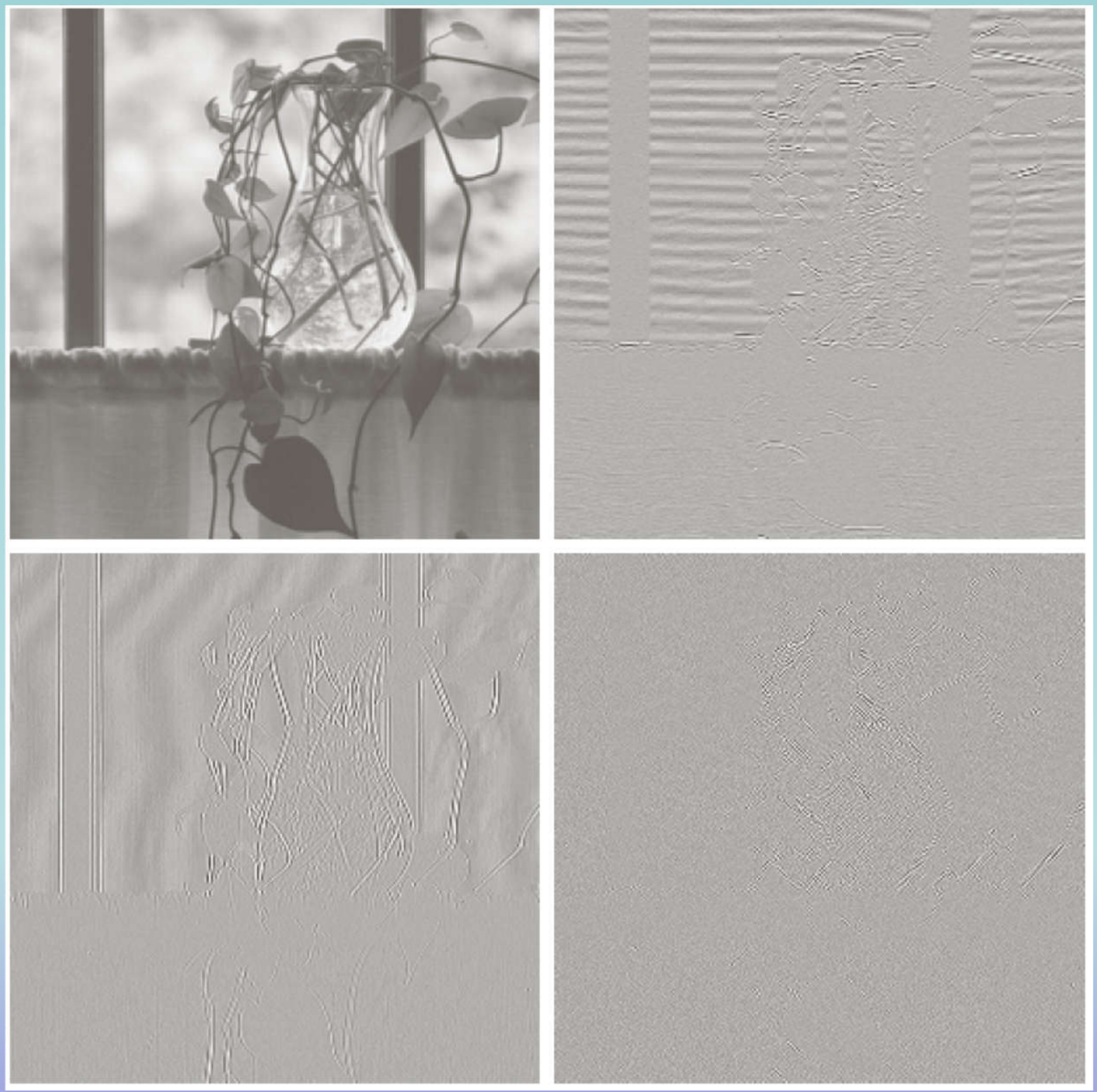
Divided into 4 quadrants

28	25	28	25	-1	1	-3	-1
27	28	27	25	-2	1	1	1
24	24	25	26	-1	3	3	1
24	25	24	25	1	2	-1	-1
-2	-1	-1	1	-1	3	-2	0
-2	-2	-1	-2	-1	1	-2	-2
-2	1	0	-1	2	1	-1	4
1	1	1	0	-3	-3	-4	-4

T-Pyramids: Quad tree

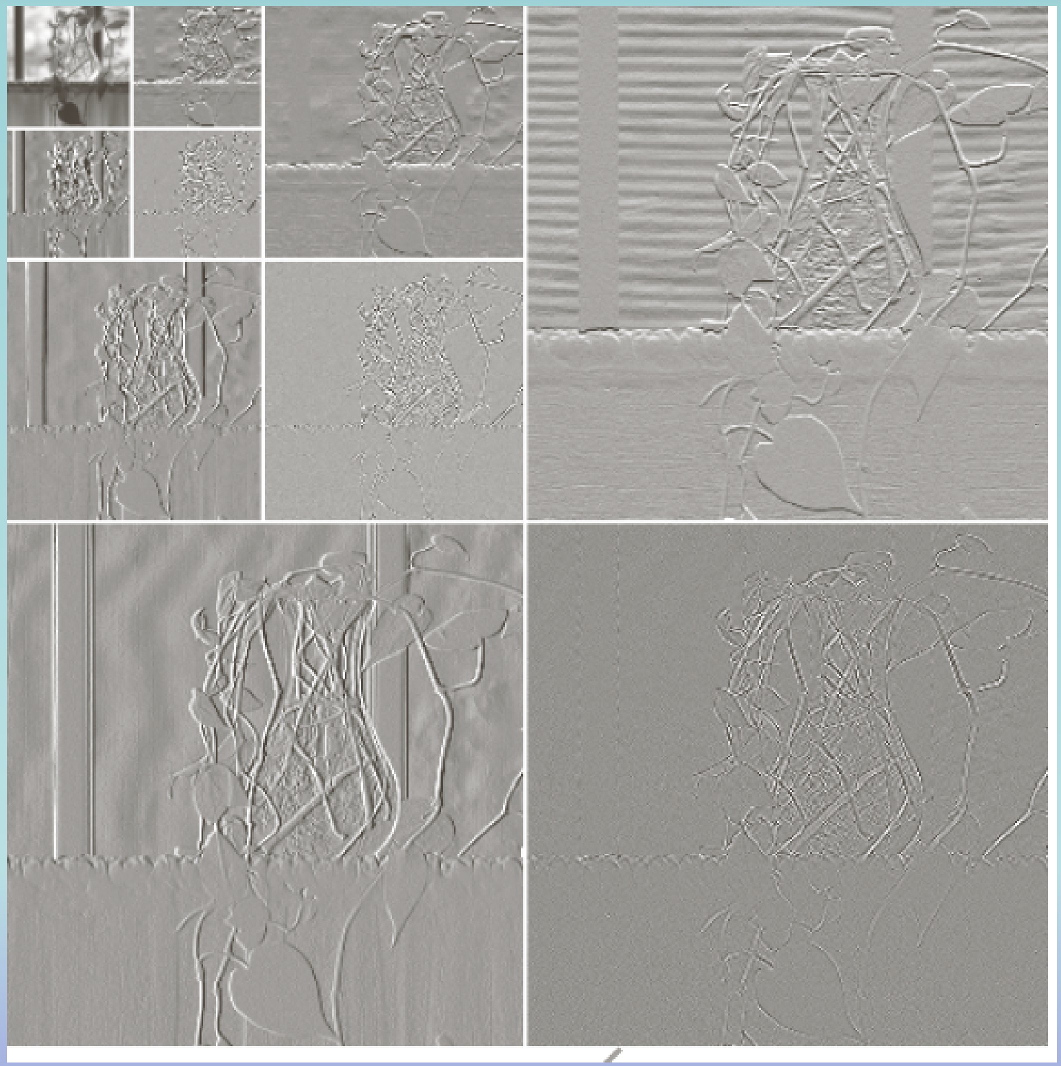
CSE-BUET

T-Pyramids: Quad tree



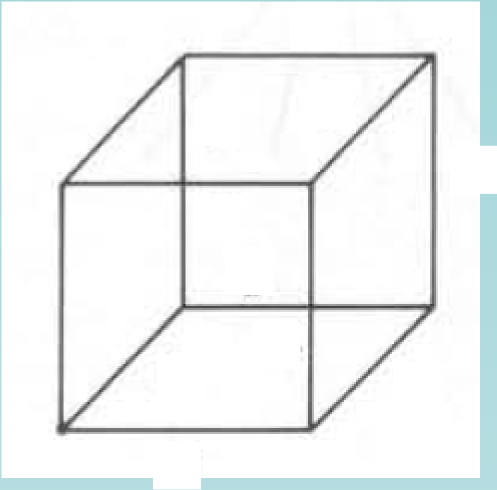
CSE-BUET

T-Pyramids: Quad tree

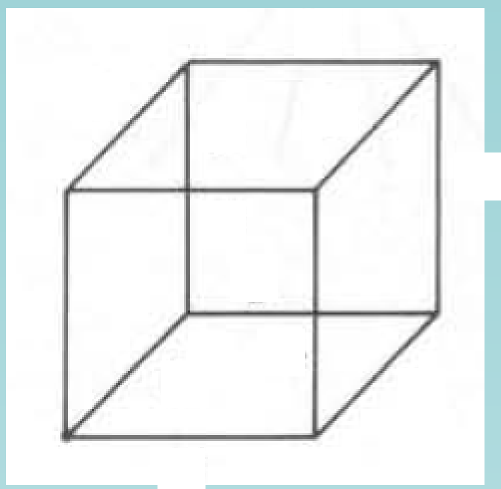


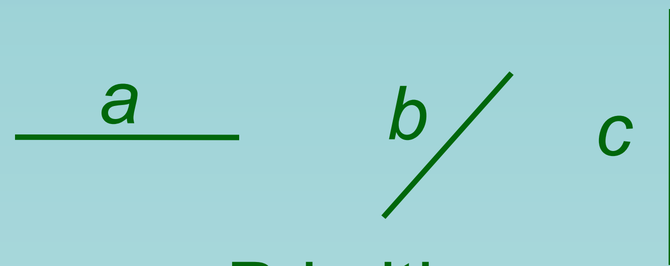
CSE-BUET

Representation using Tree (1)



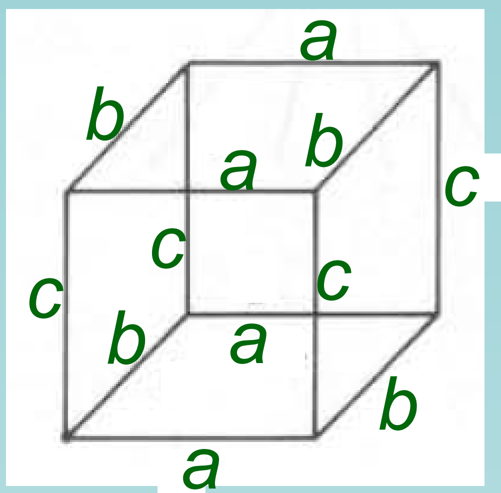
Representation using Tree (1)

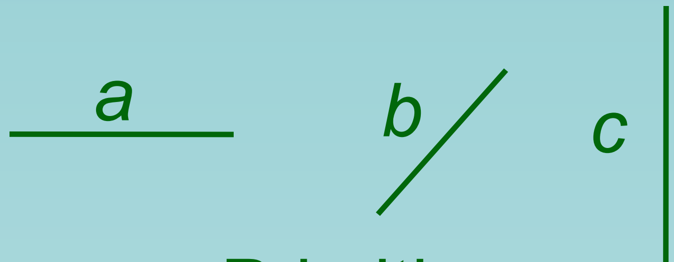




Primitives

Representation using Tree (1)

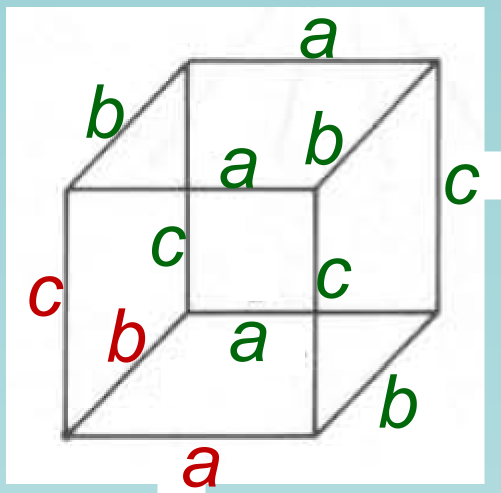


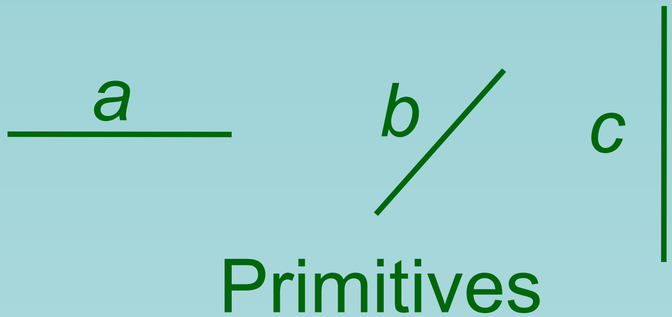


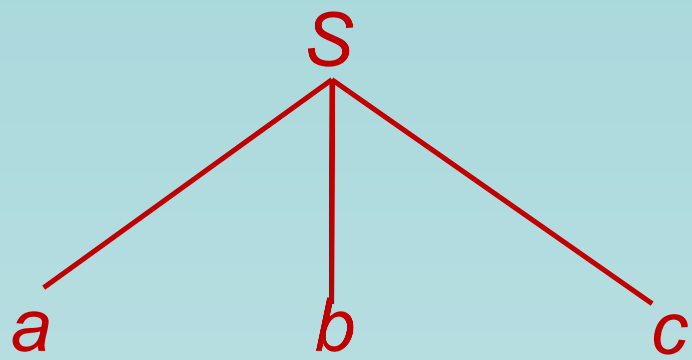
Primitives

CSE-BUET

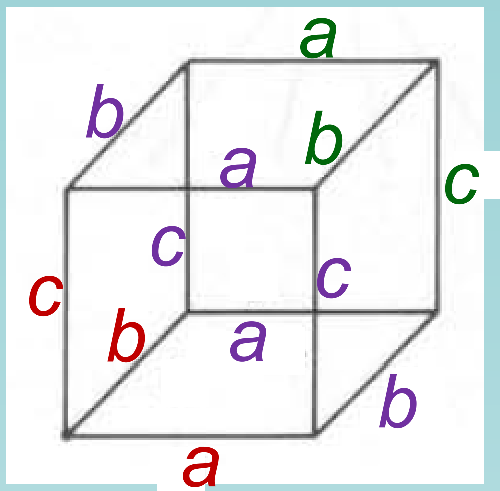
Representation using Tree (1)

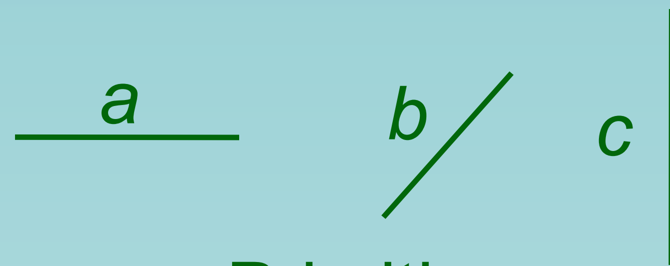




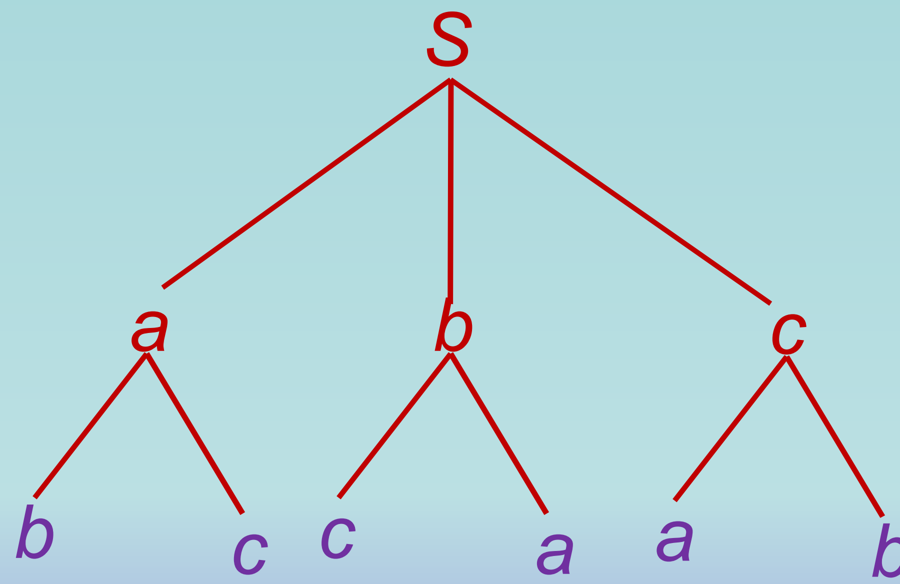


Representation using Tree (1)

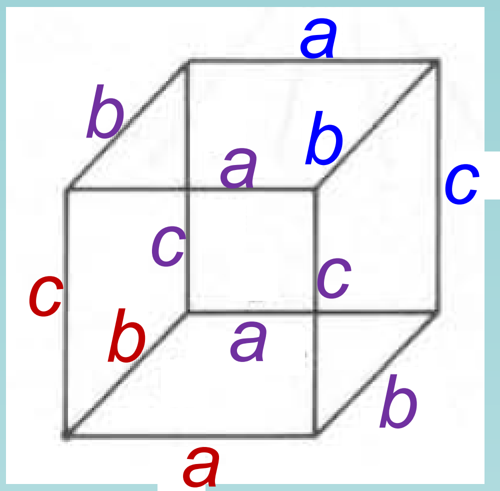


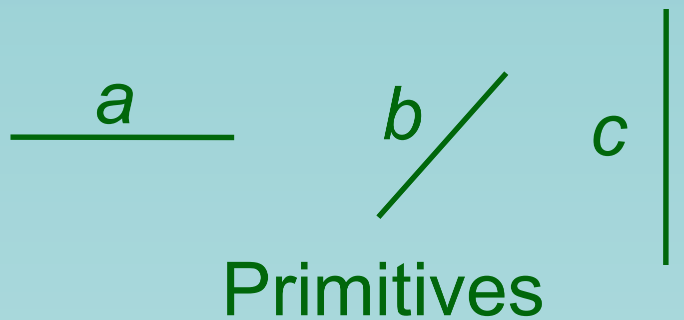


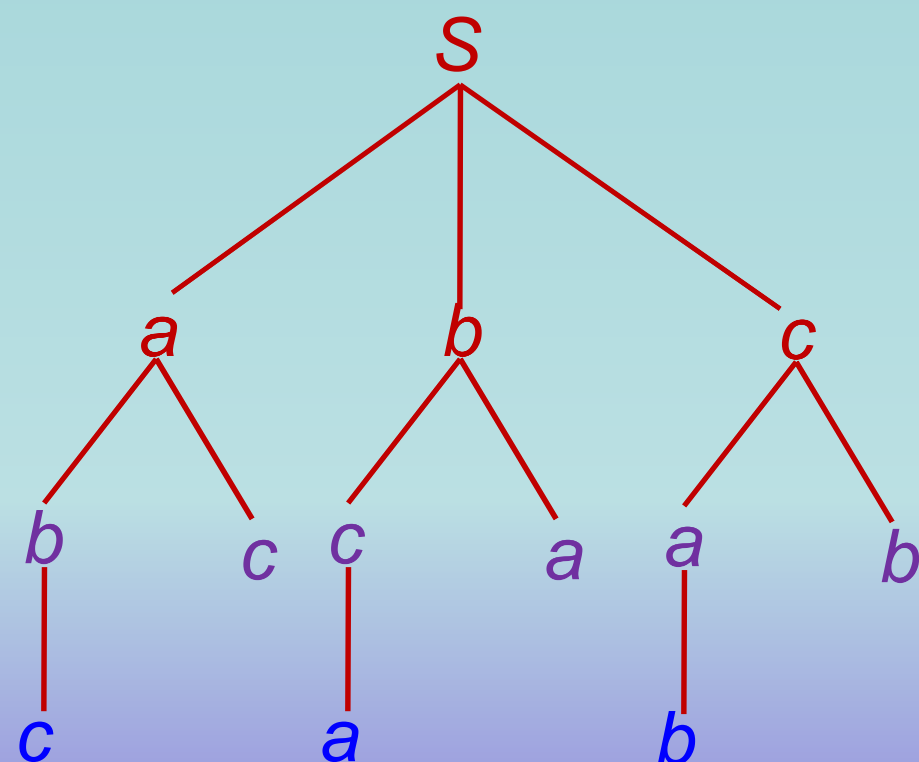
Primitives



Representation using Tree (1)

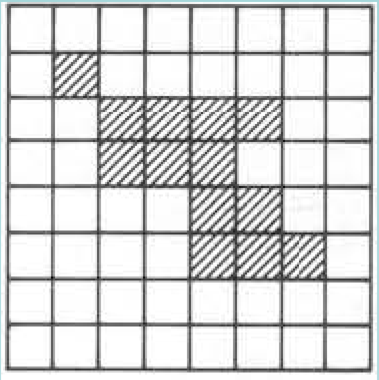






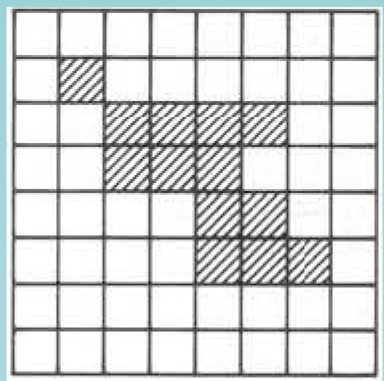
CSE-BUET

Representation using Tree (2)



CSE-BUET

Representation using Tree (2)



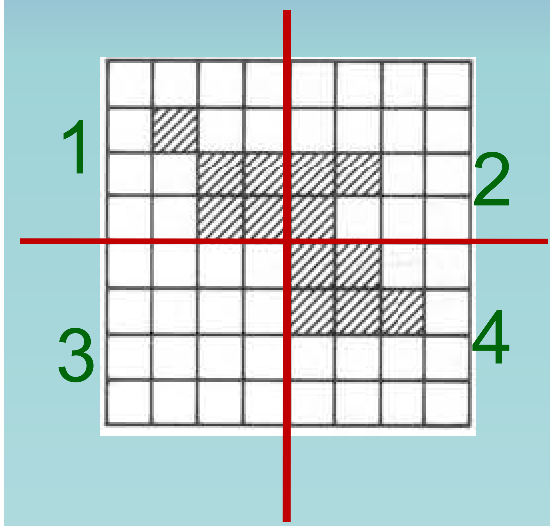
Primitives

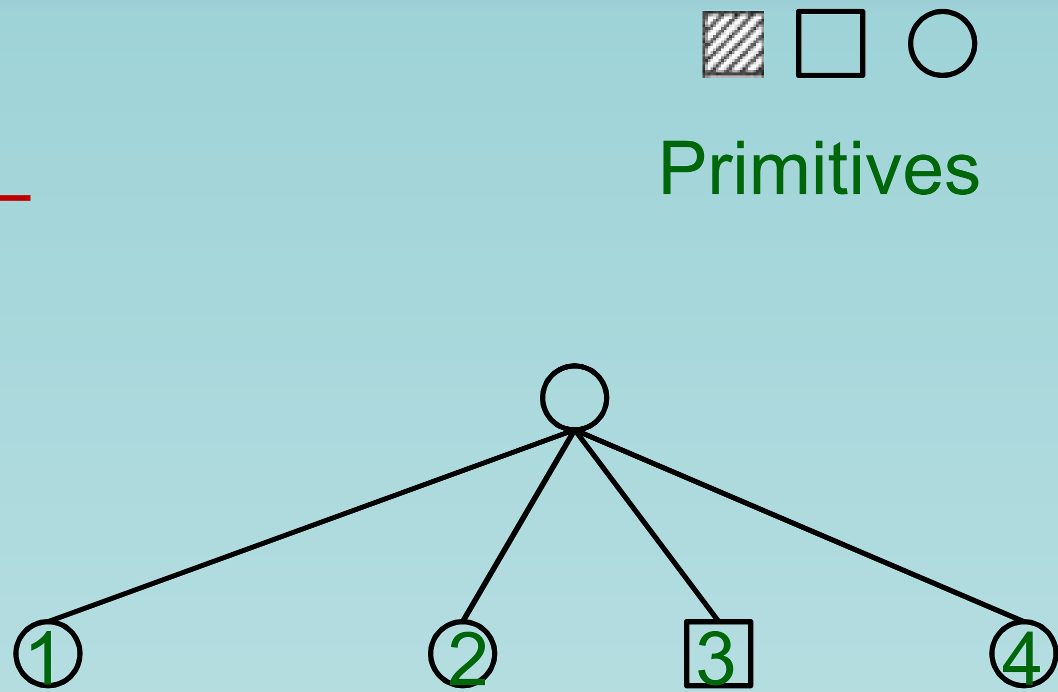
Black region

White region

Gray region

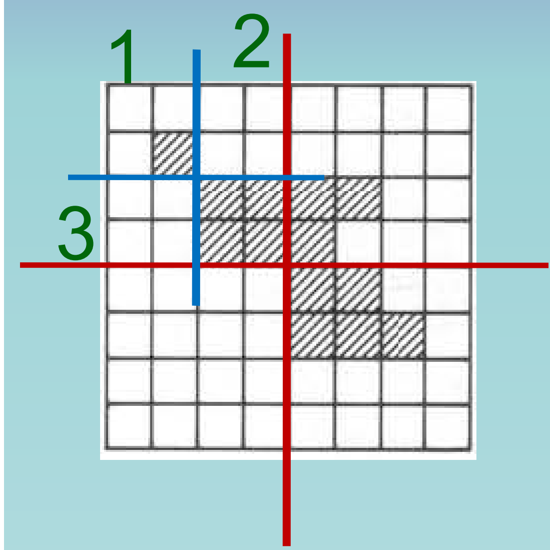
Representation using Tree (2)

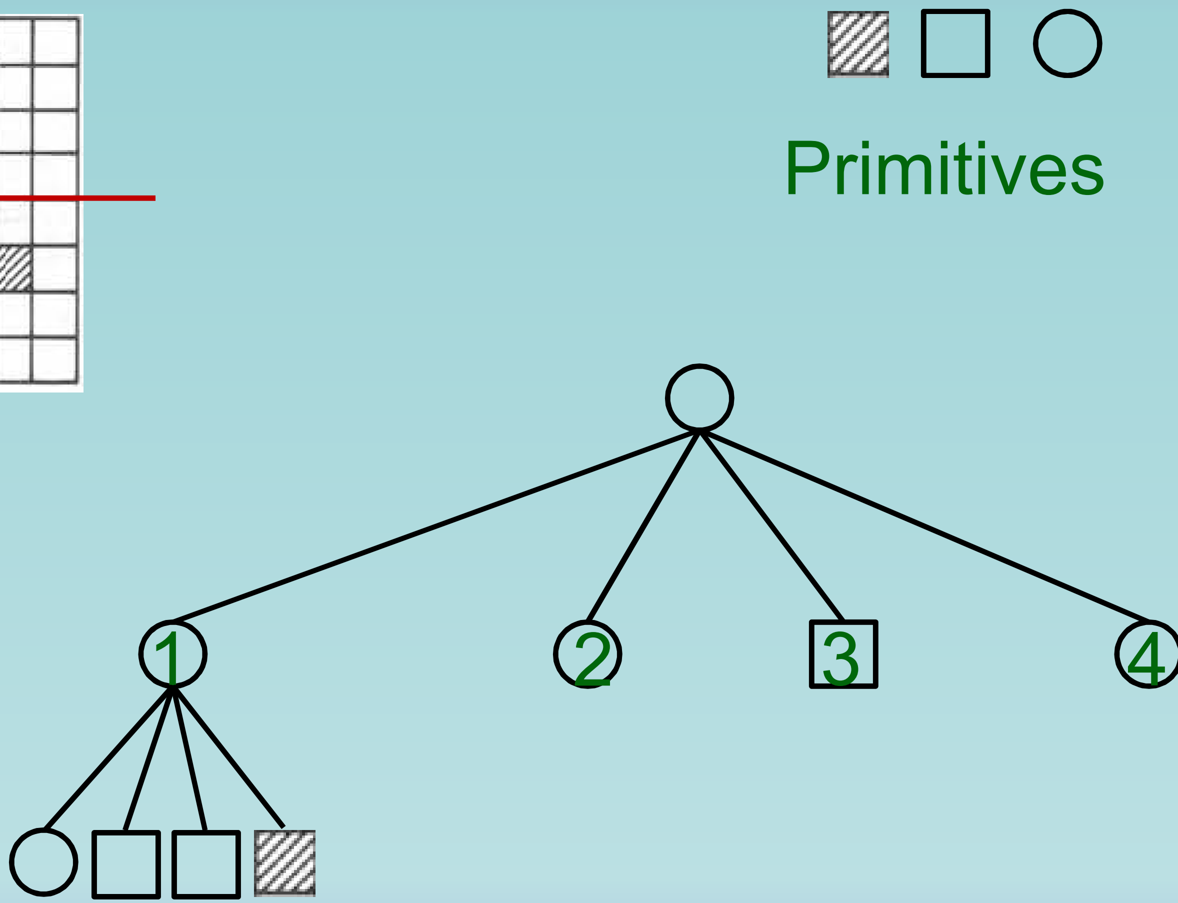




CSE-BUET

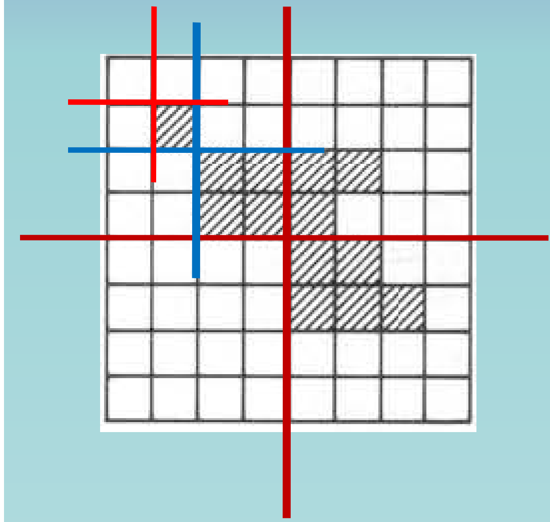
Representation using Tree (2)

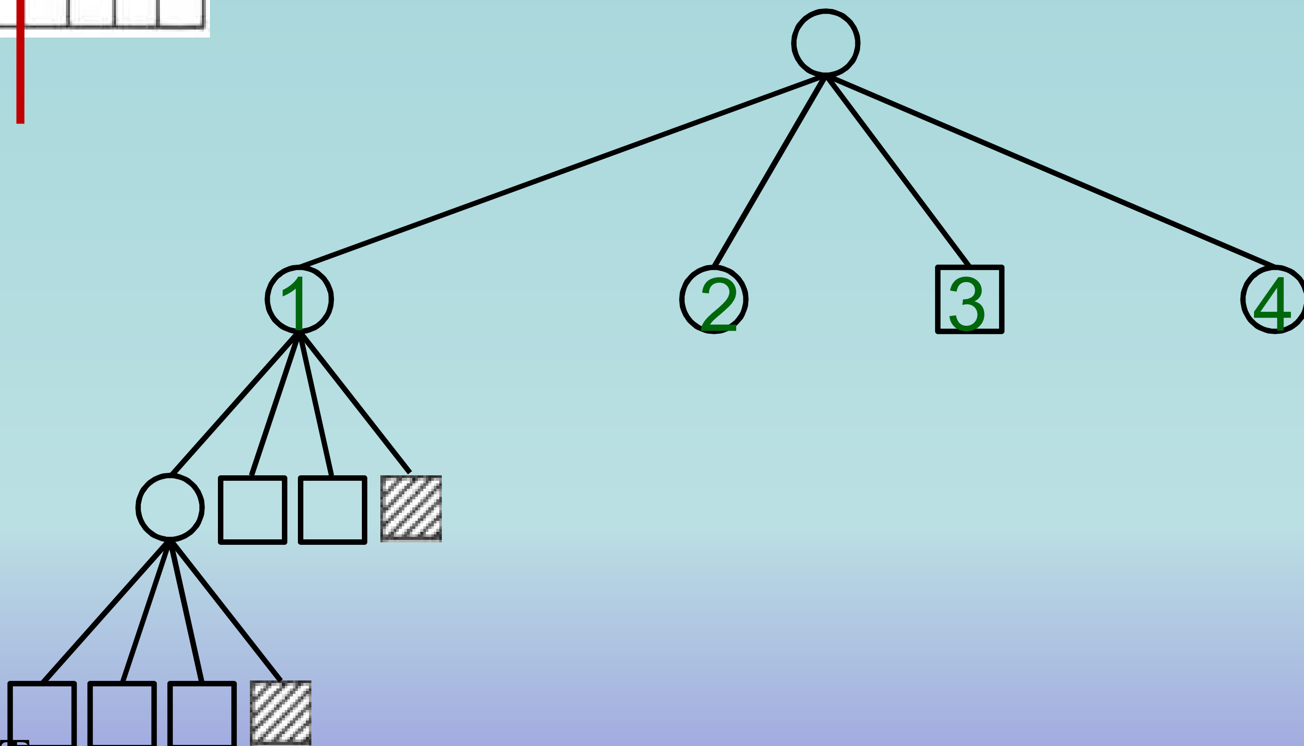




CSE-BUET

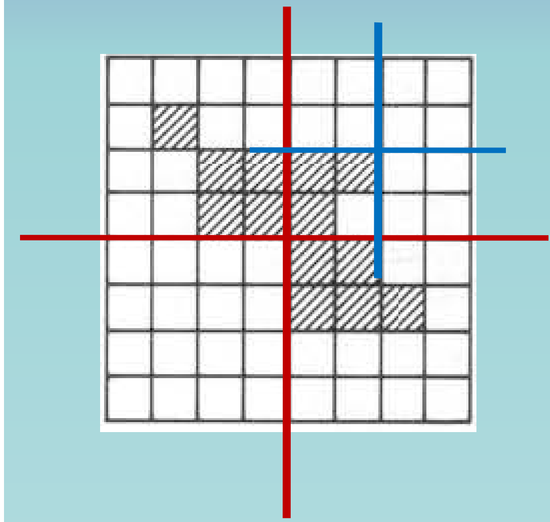
Representation using Tree (2)



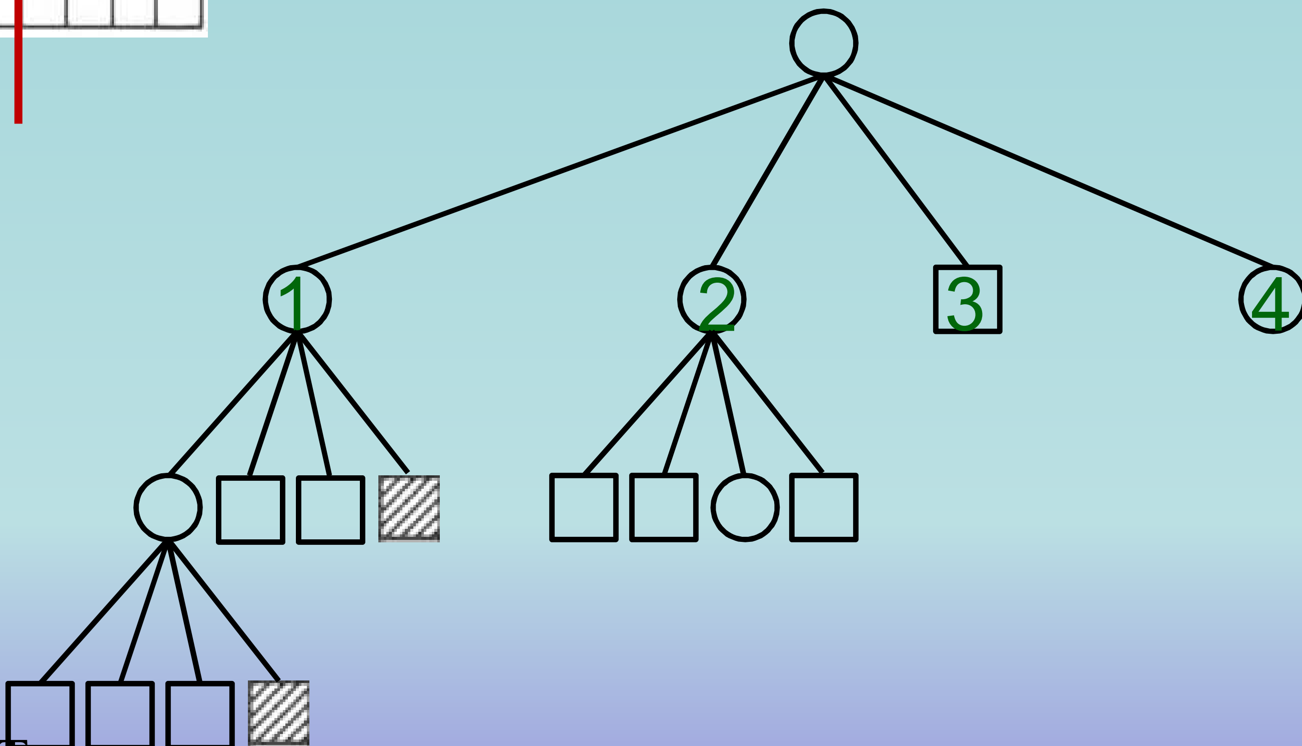


CSE-BUET

Representation using Tree (2)

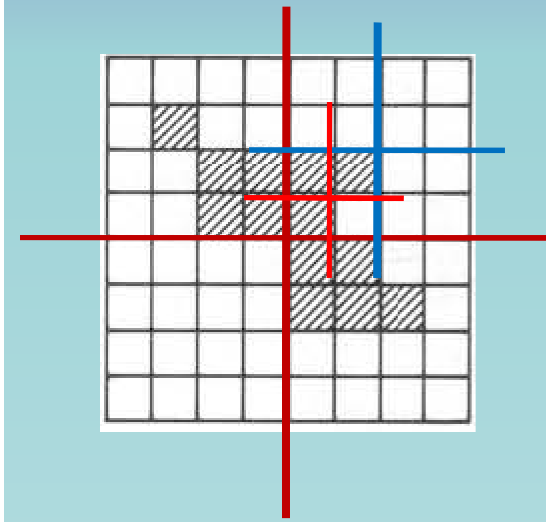


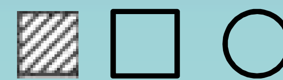
Primitives



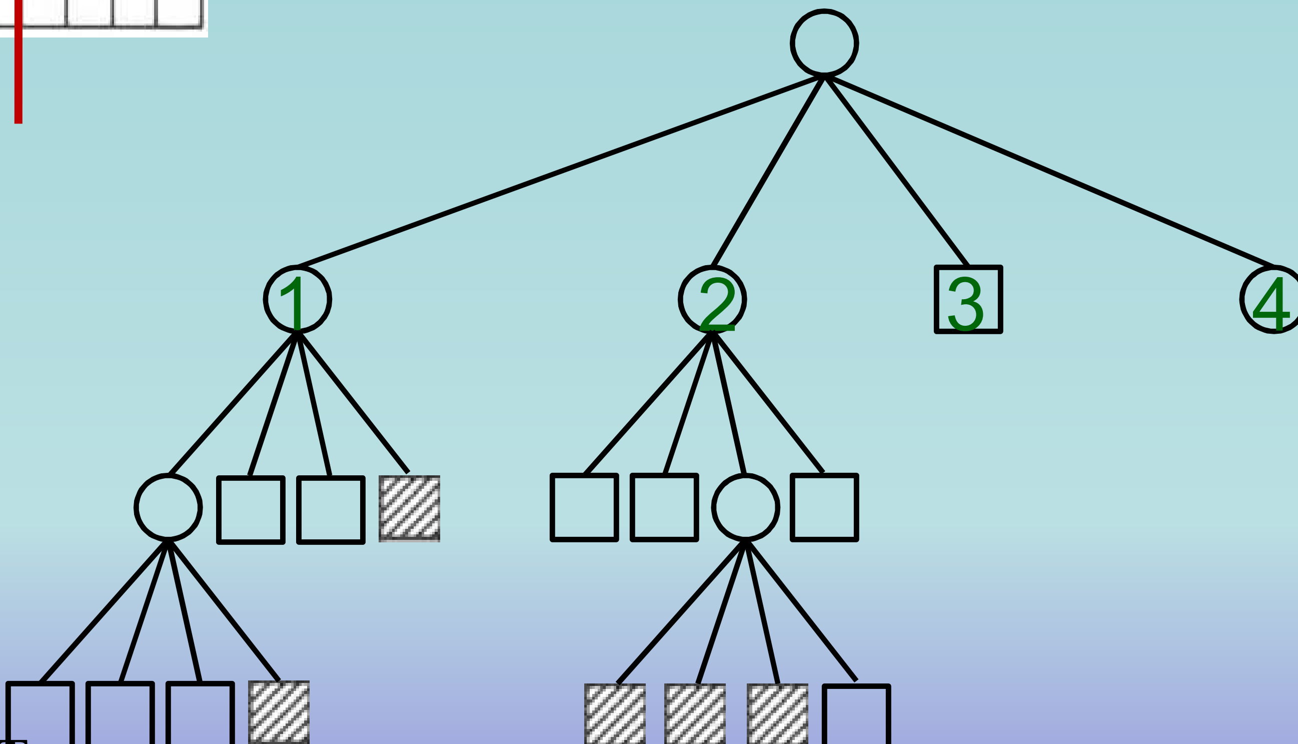
CSE-BUET

Representation using Tree (2)



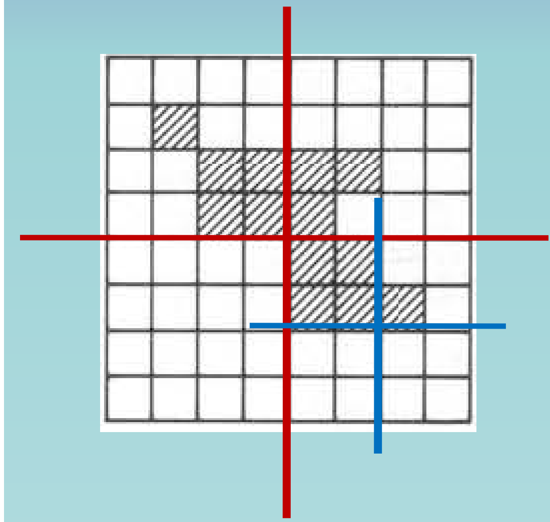


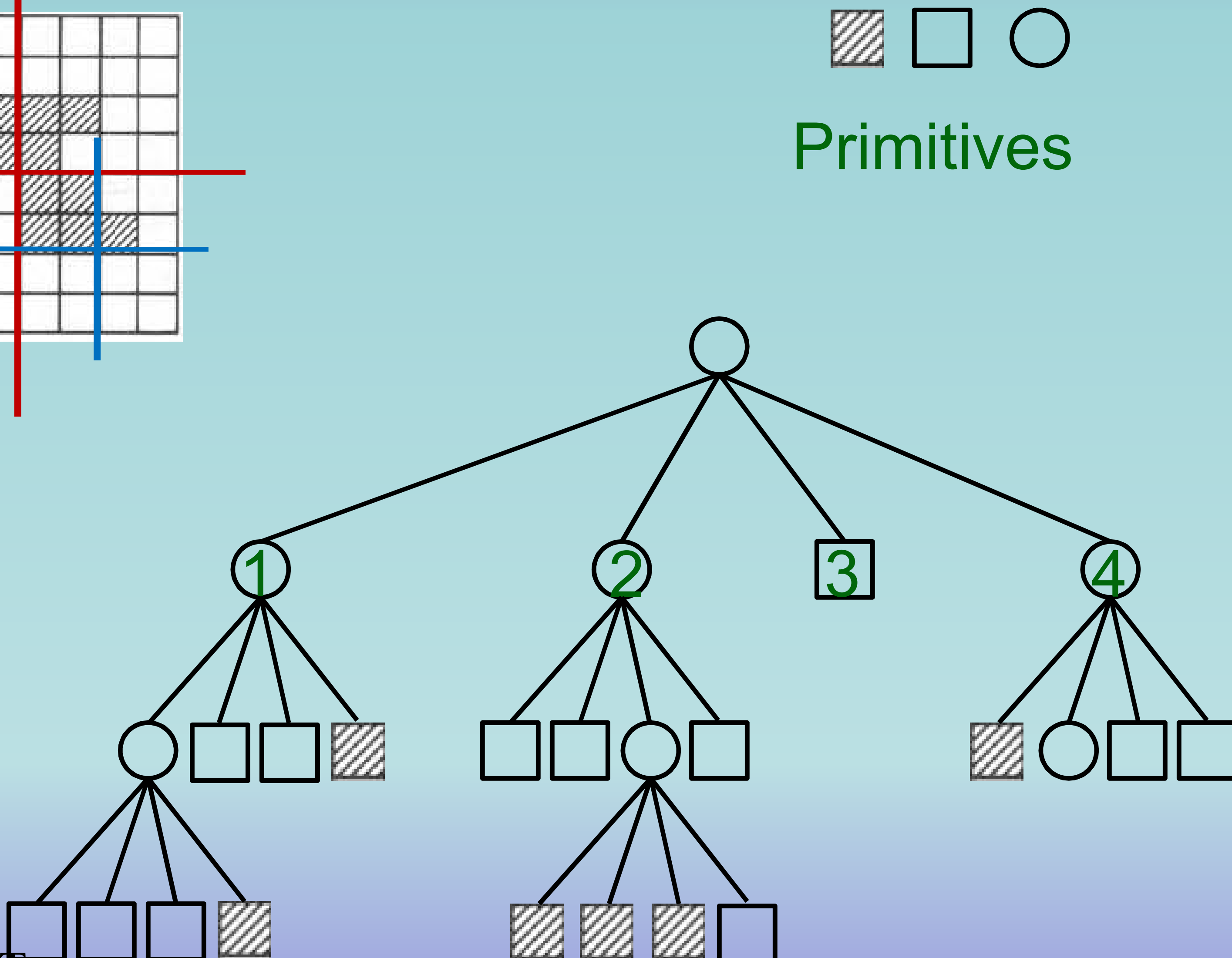
Primitives



CSE-BUET

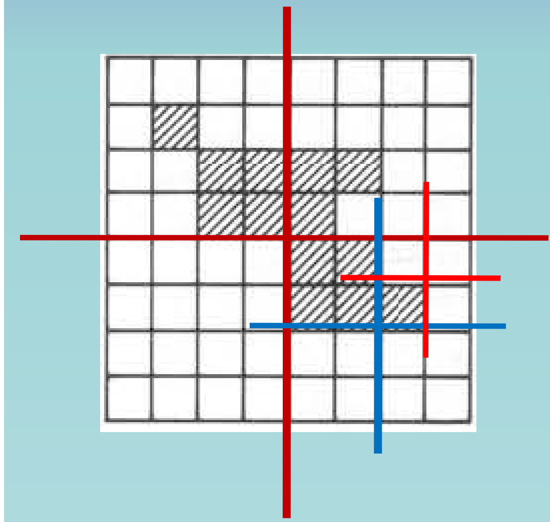
Representation using Tree (2)

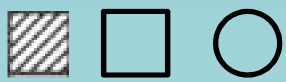


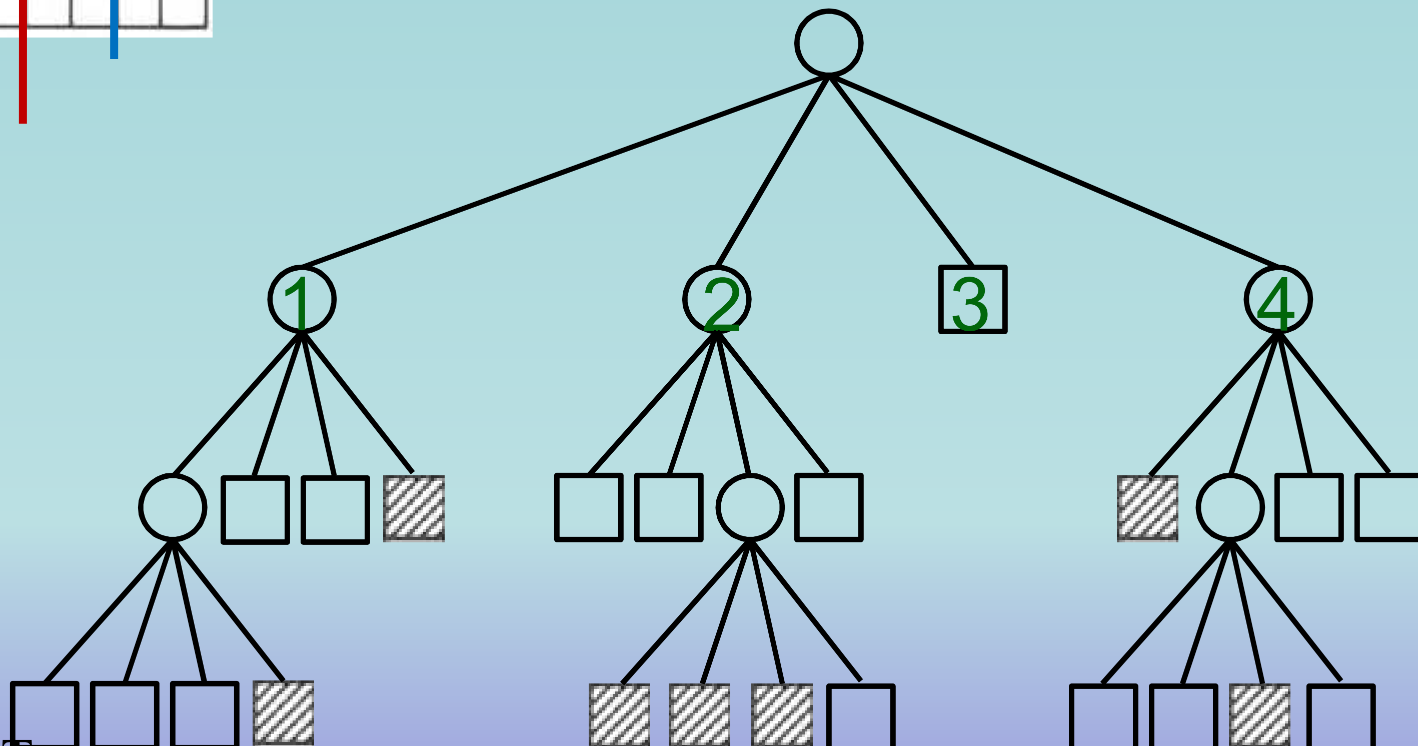


CSE-BUET

Representation using Tree (2)

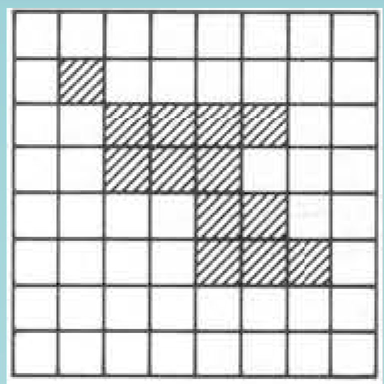


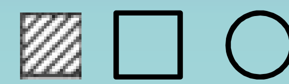




CSE-BUET

Representation using Tree (2)





Primitives

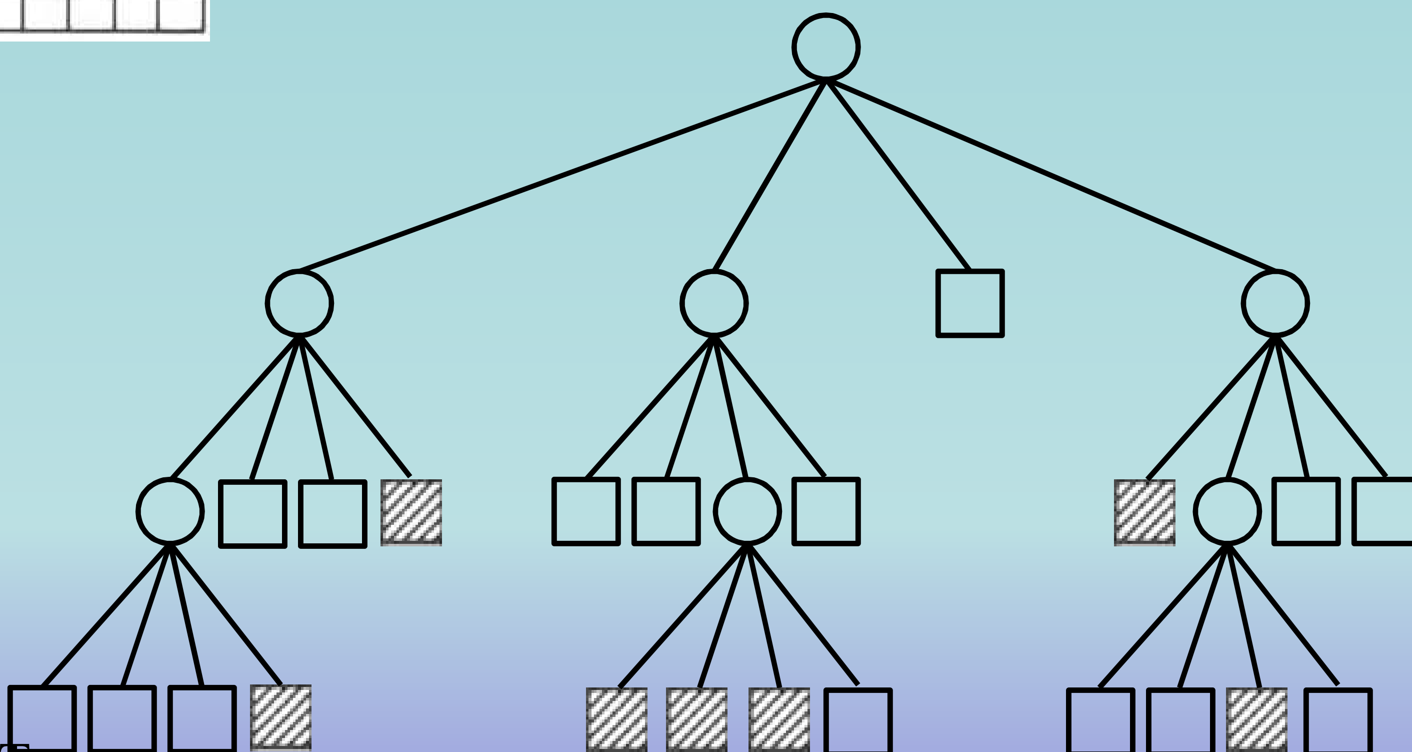


Image Enhancement

CSE-BUET

Image Enhancement

- Objective:
  - Get more *suitable and enhanced* image for *specific* application
- Very subjective, and
- Application dependent
- Viewer is the ultimate judge

Classification of Image Enhancement

- Two categories:
  - Enhancement in spatial domain
  - Enhancement in frequency domain

Classification of Image Enhancement

- Two categories:
  - Enhancement in spatial domain
    - works in image plane
    - manipulates pixel by pixel basis
  - Enhancement in frequency domain

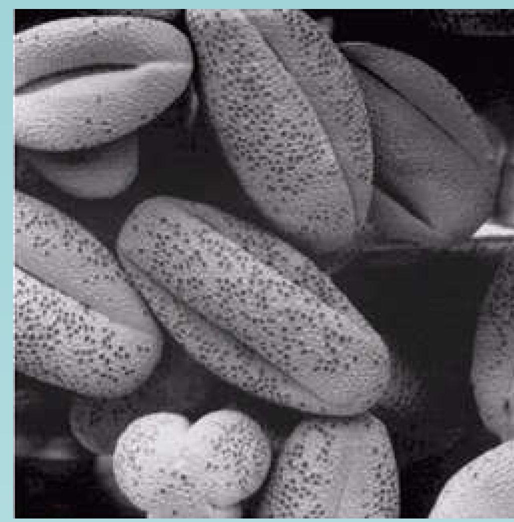
Classification of Image Enhancement

- Two categories:
  - Enhancement in spatial domain
    - works in image plane
    - manipulates pixel by pixel basis
  - Enhancement in frequency domain
    - works in Fourier transformed image
    - usually manipulates the entire transformed image at a time

Classification of Image Enhancement

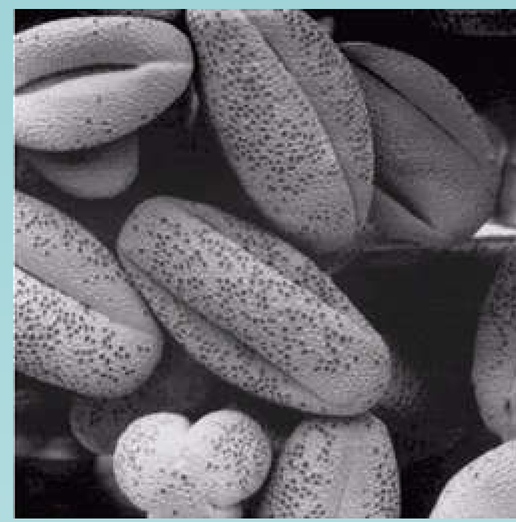
- Two categories:
  - Enhancement in spatial domain
    - works in image plane
    - manipulates pixel by pixel basis
  - Enhancement in frequency domain
    - works in Fourier transformed image
    - usually manipulates the entire transformed image at a time
    - however, element by element operation is also possible

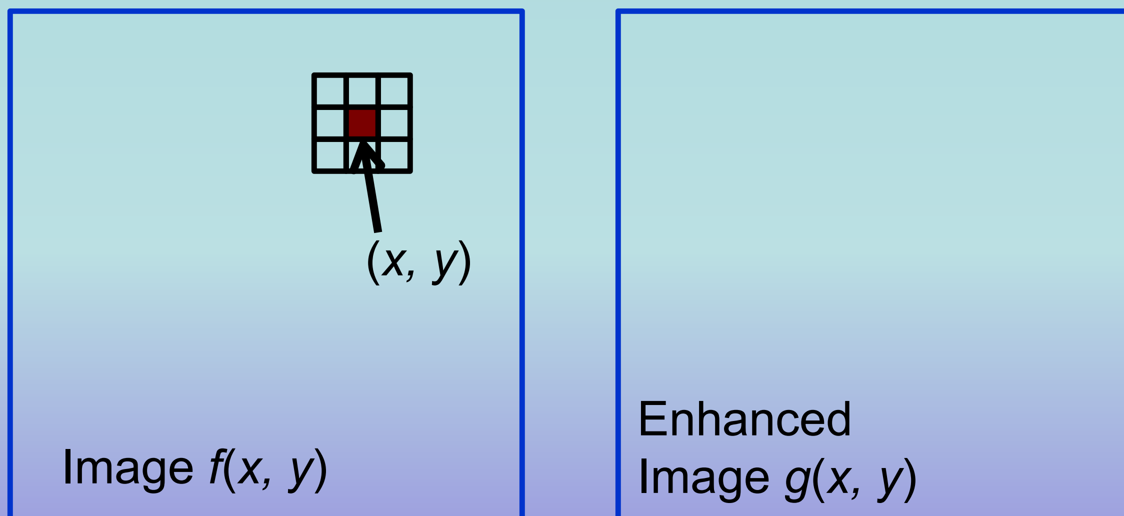
Image Enhancement in Spatial Domain



CSE-BUET

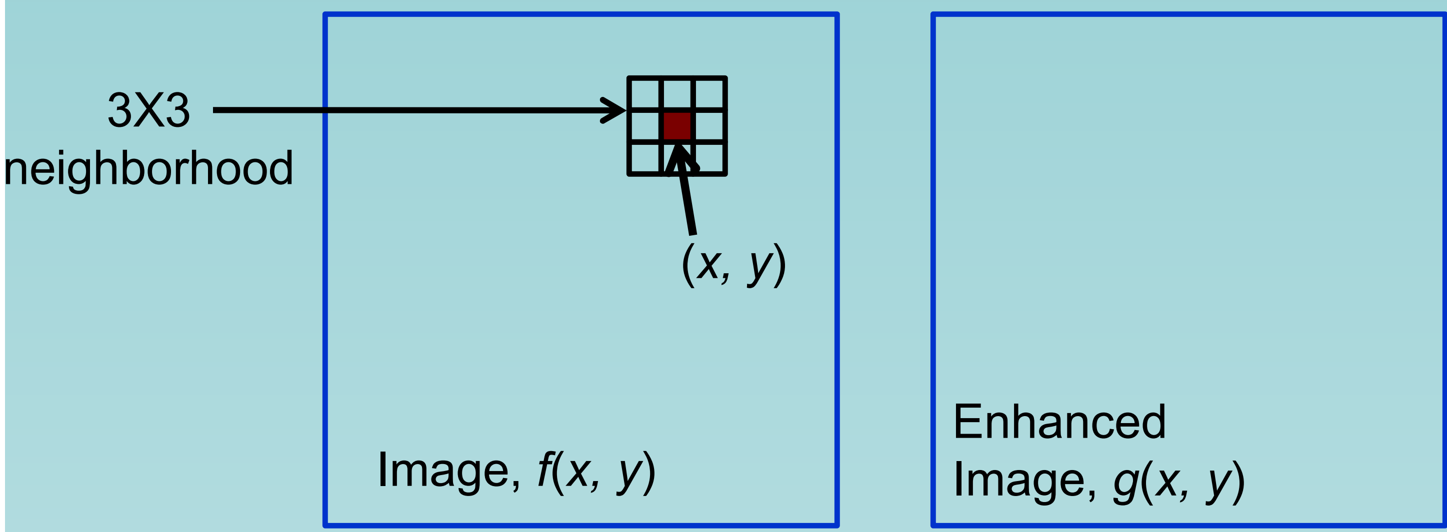
Image Enhancement in Spatial Domain





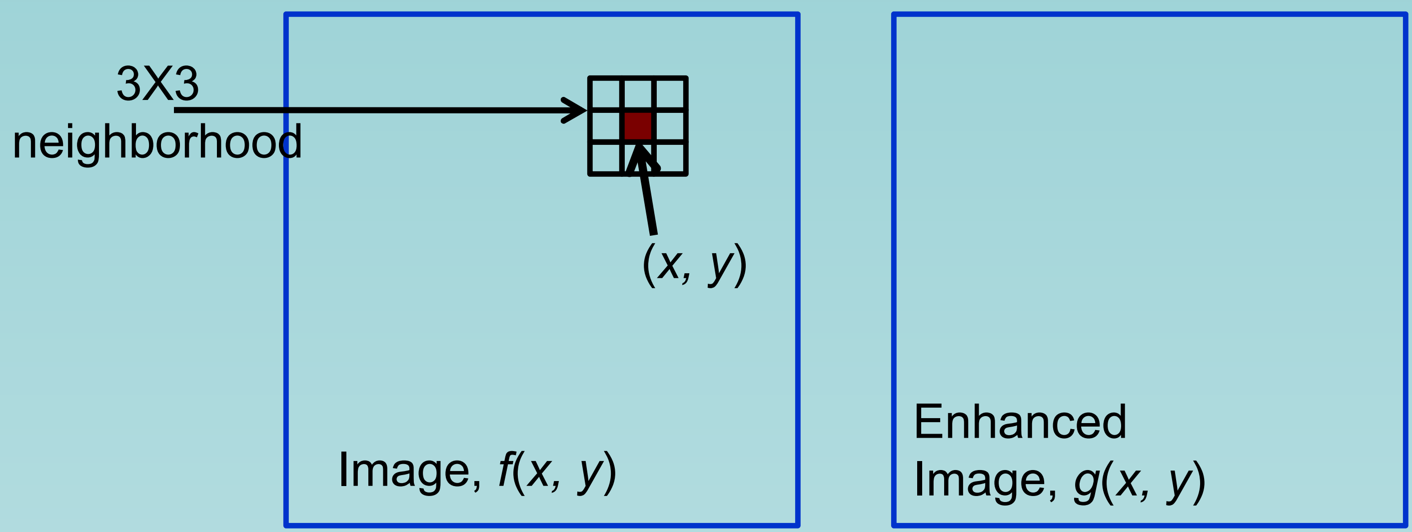
CSE-BUET

Image Enhancement in Spatial Domain



$$g(x,y)=T\left[f(x,y)\right]$$

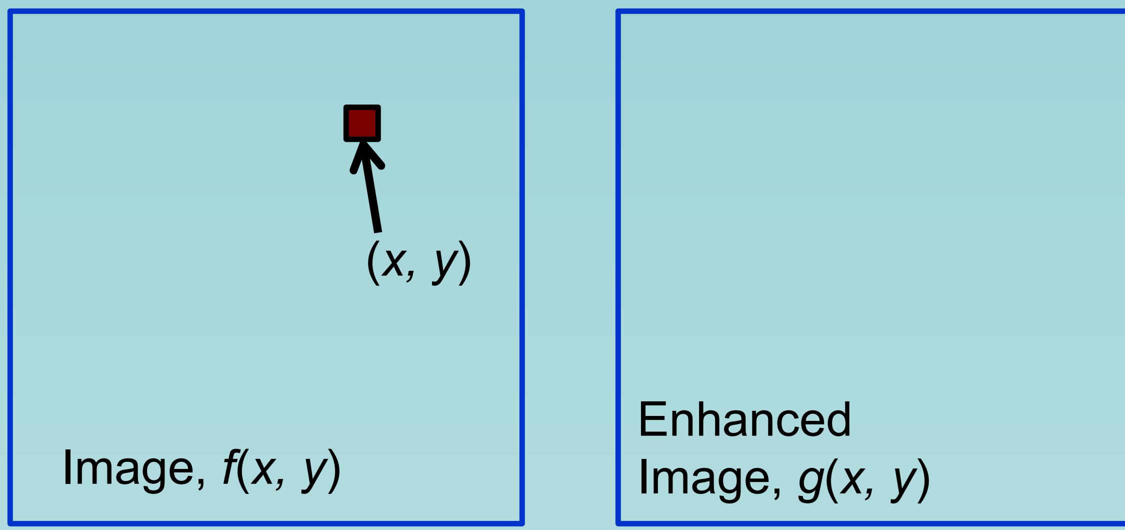
Image Enhancement in Spatial Domain



$$g(x,y)=T\left[f(x,y)\right]$$

- Neighborhood (sub-image) rolls over the entire image, each time centering a different pixel

Image Enhancement in Spatial Domain

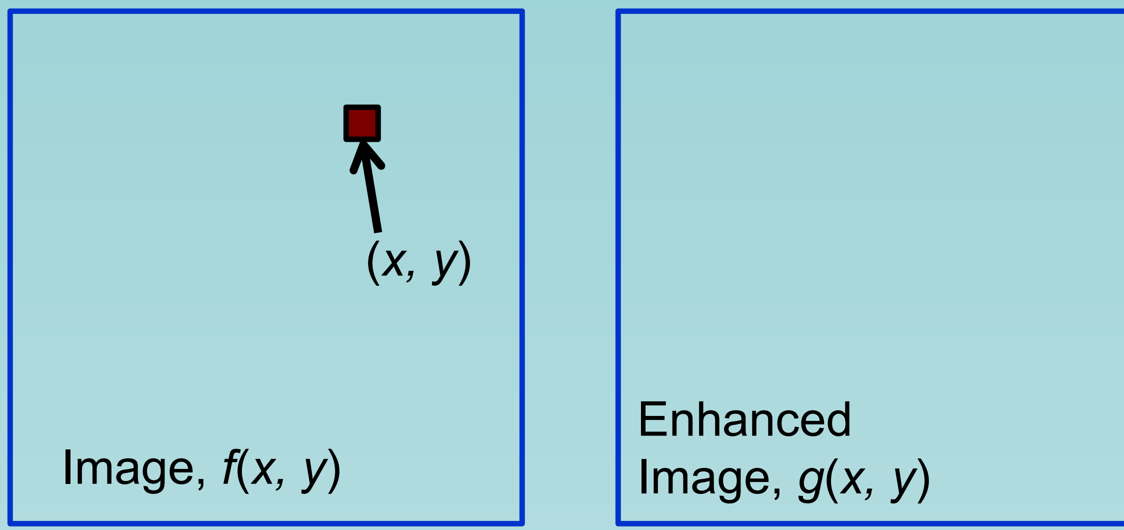


- Neighborhood can be as small as 1X1 sub-image

$$s = T(r)$$

CSE-BUET

Image Enhancement in Spatial Domain



$r = f(x, y)$ $\qquad$ $s = g(x, y)$

- Neighborhood can be as small as 1X1 sub-image

$$s = T(r)$$

Image Enhancement in Spatial Domain: Gray level Transformation

- Contrast stretching
  - Input gray level (*r*):
    - below a threshold (*m*): compressed towards black
    - above *m*: stretched towards white

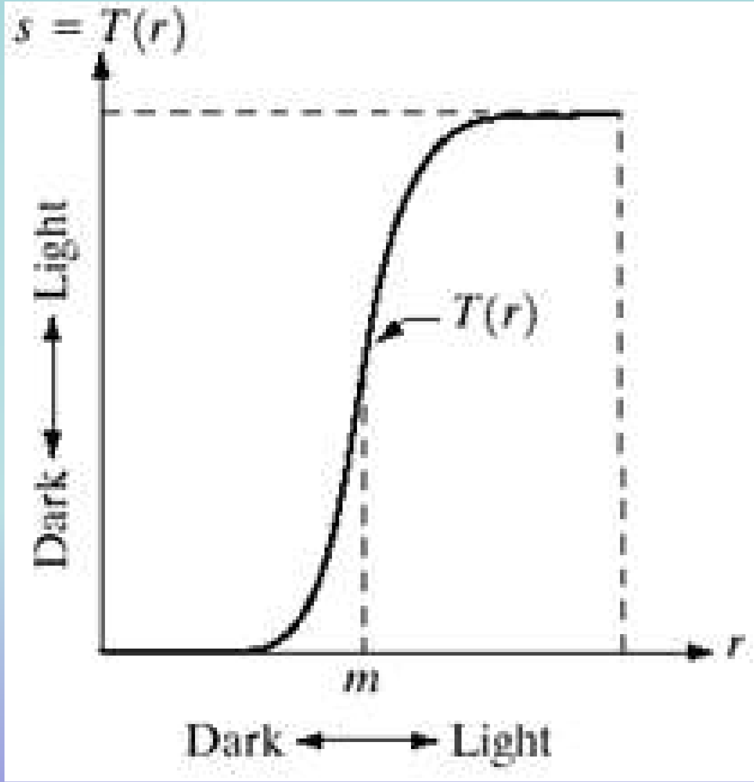
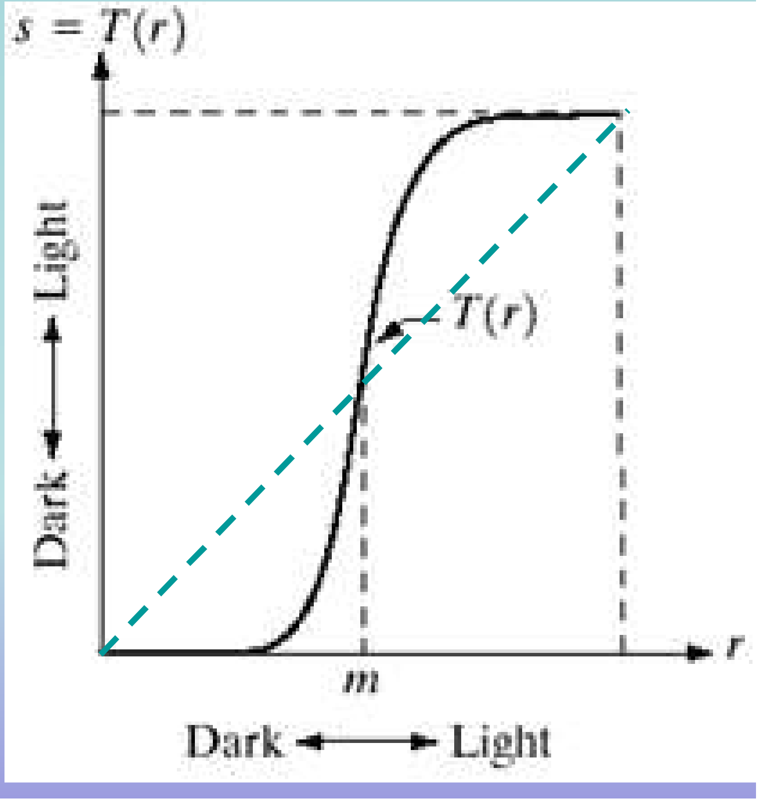


Image Enhancement in Spatial Domain: Gray level Transformation

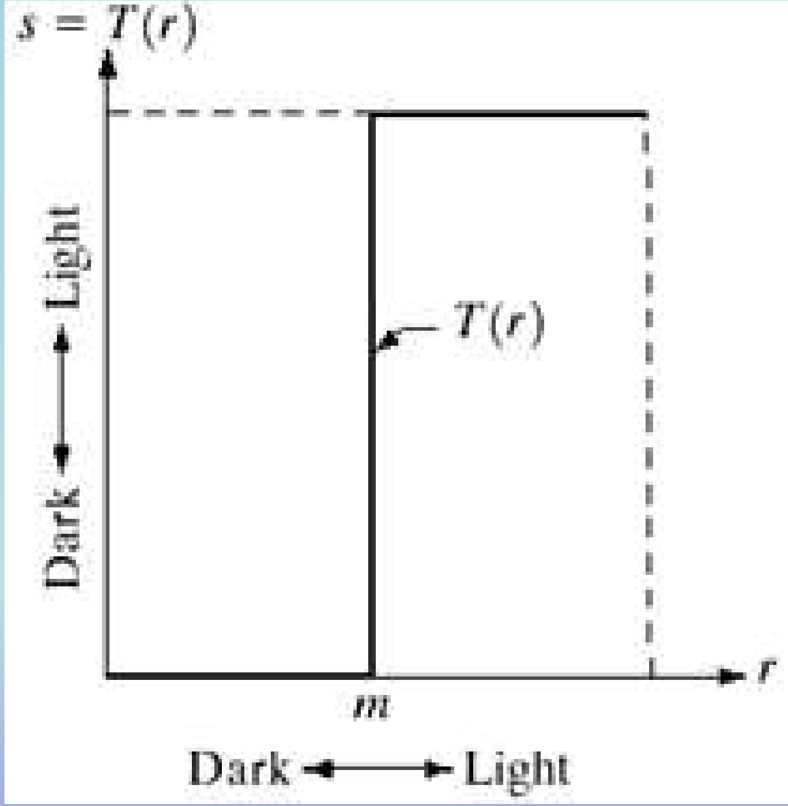
- Contrast stretching
  - Input gray level (*r*):
    - below a threshold (*m*): compressed towards black
    - above *m*: stretched towards white



CSE-BUET

Gray level Transformation

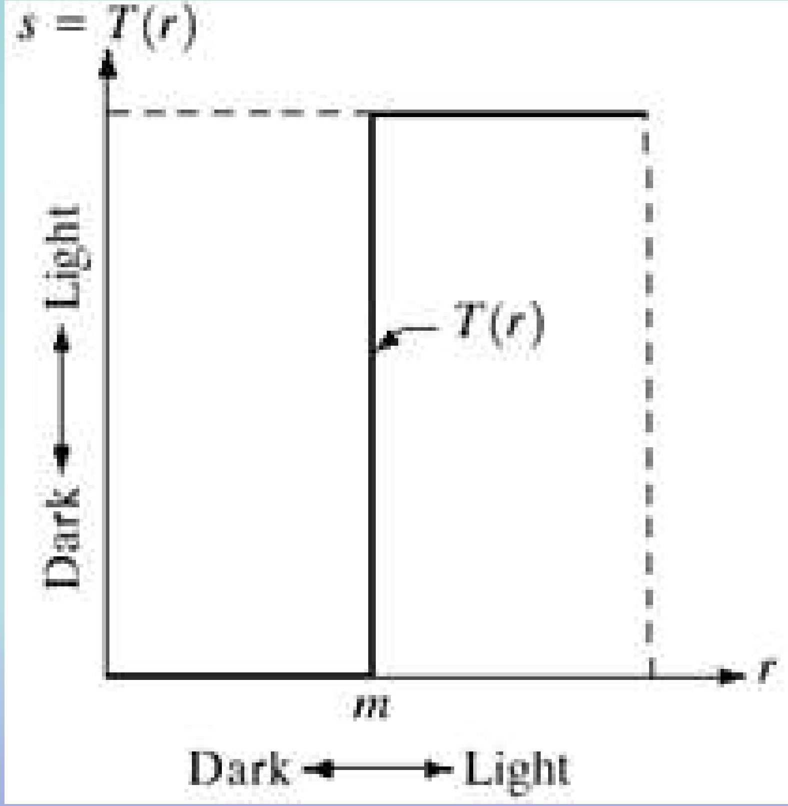
- Thresholding, a special case of contrast stretching
  - Produces a binary image
  - Useful in image segmentation



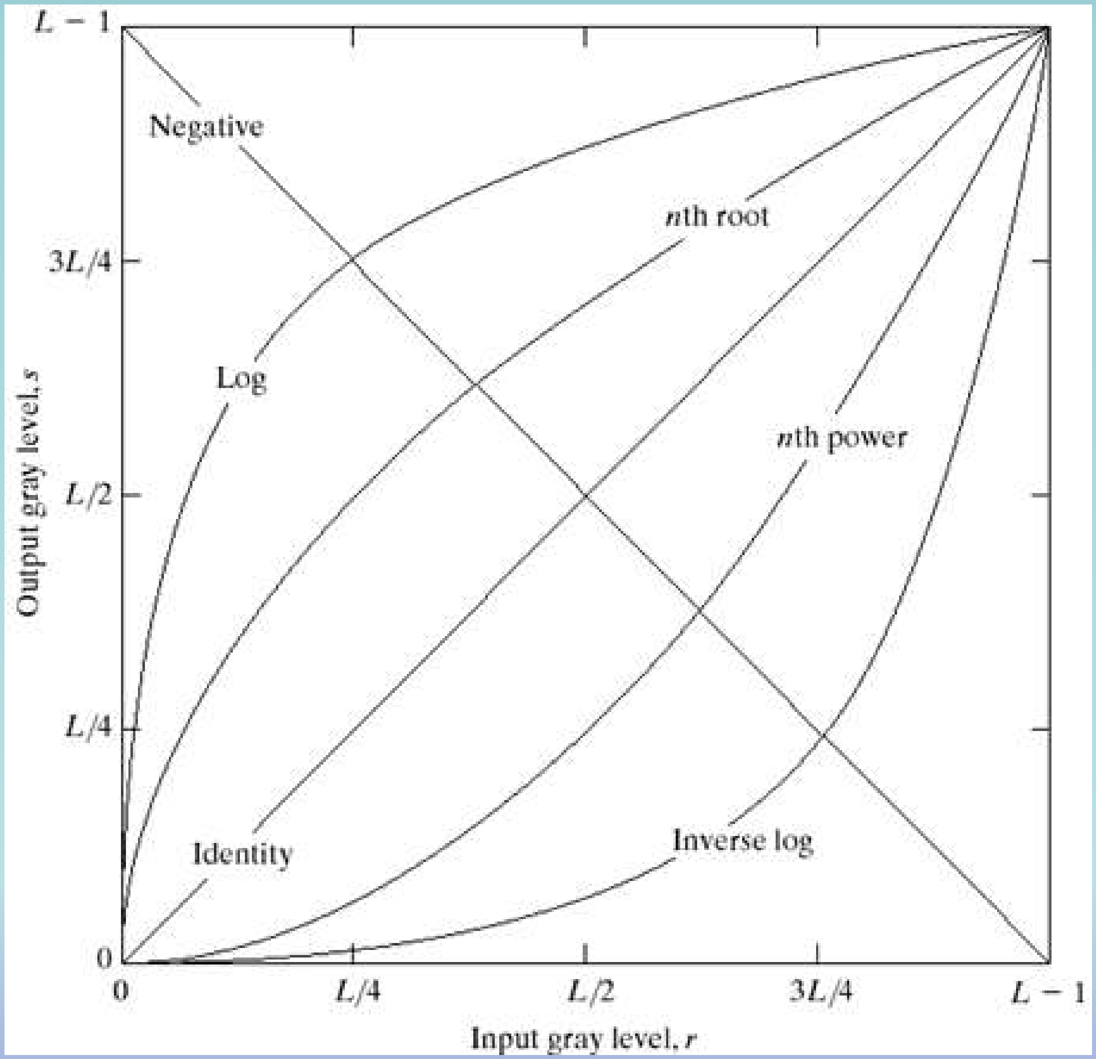
CSE-BUET

Gray level Transformation

- Thresholding, a special case of contrast stretching
  - Produces a binary image
  - Useful in image segmentation



Gray level Transformation



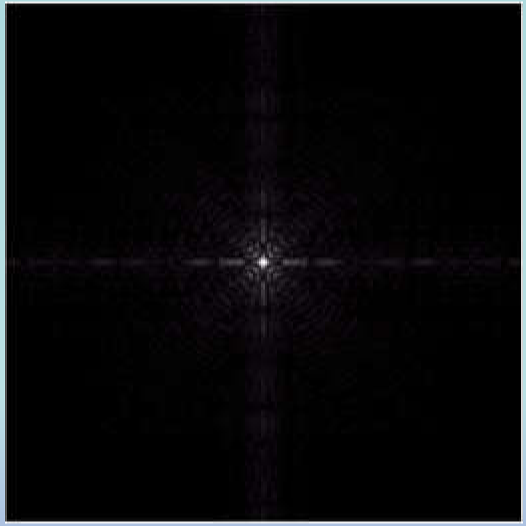
CSE-BUET

Log Transform

- Expands the dark pixels
- Compress the white pixels
- Useful in Fourier transform
  - Values range from 0 to 10^6

Log Transform

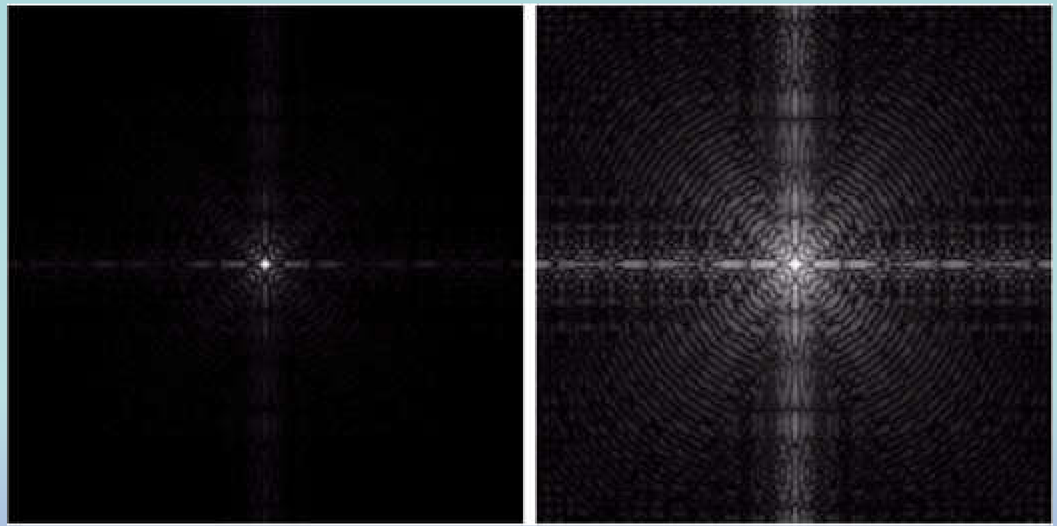
- Expands the dark pixels
- Compress the white pixels
- Useful in Fourier transform
  - Values range from 0 to 10^6



Without Log Xform

Log Transform

- Expands the dark pixels
- Compress the white pixels
- Useful in Fourier transform
  - Values range from 0 to 10^6

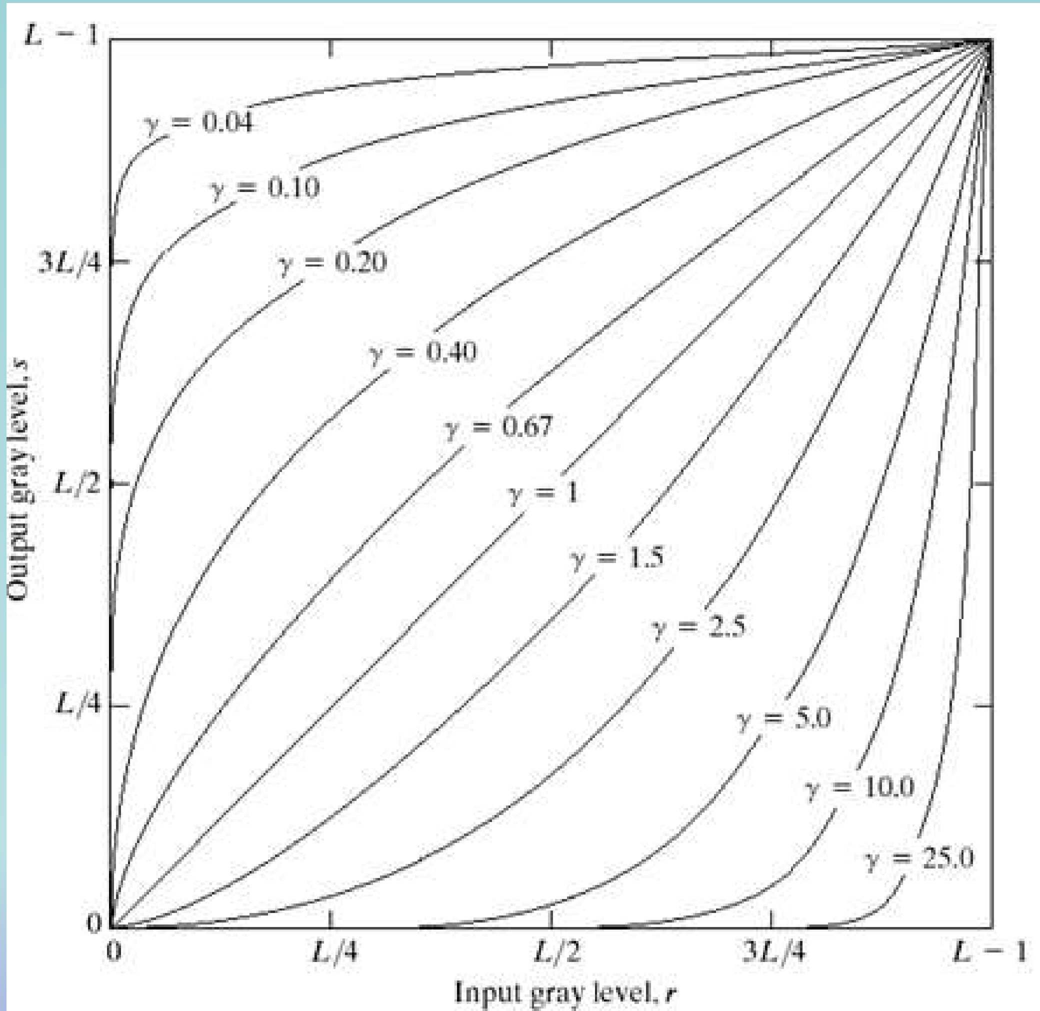


Without Log Xform With Log Xform

CSE-BUET

Power Law Transform: $s = cr^{\gamma}$

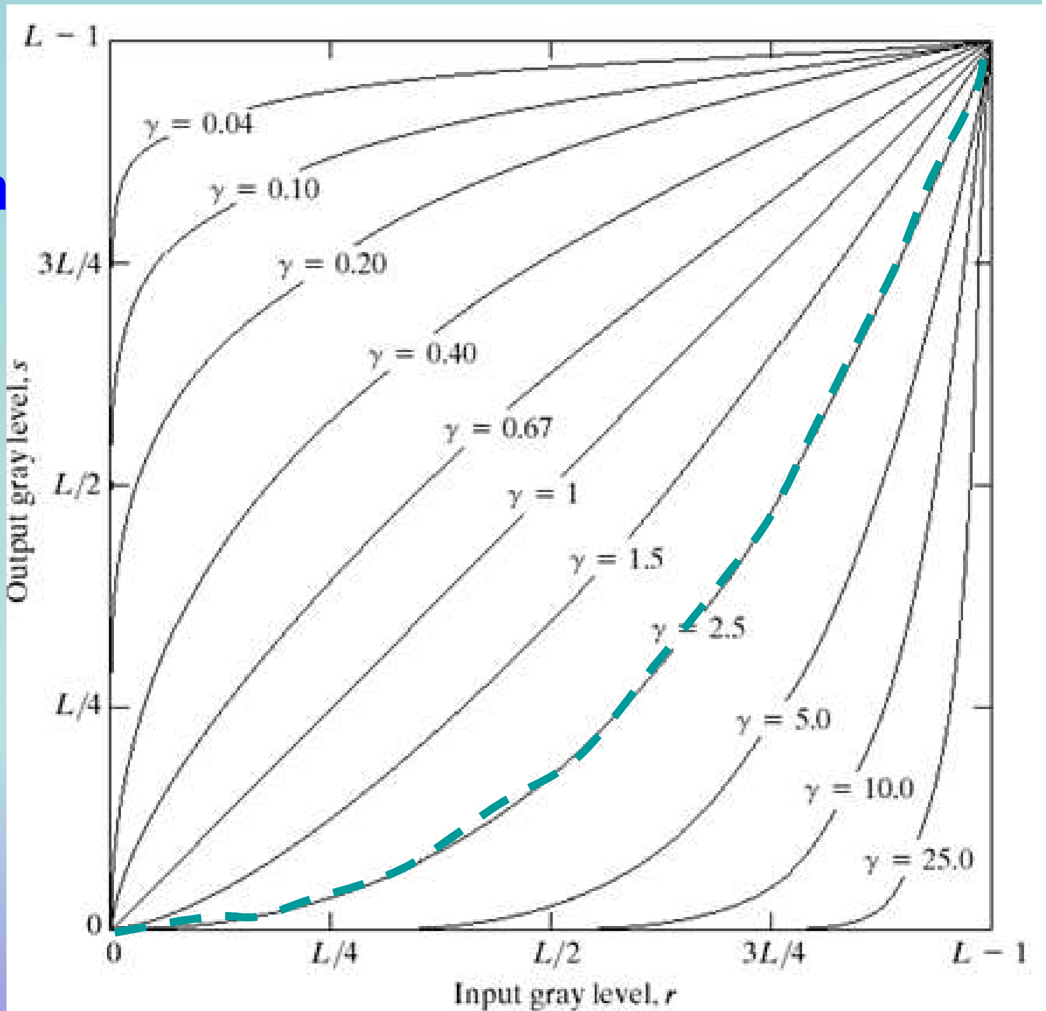
- $\gamma < 1$: changes darker to brighter
- $\gamma > 1$: changes brighter to darker
- used in *gamma* correction of imaging devices



CSE-BUET

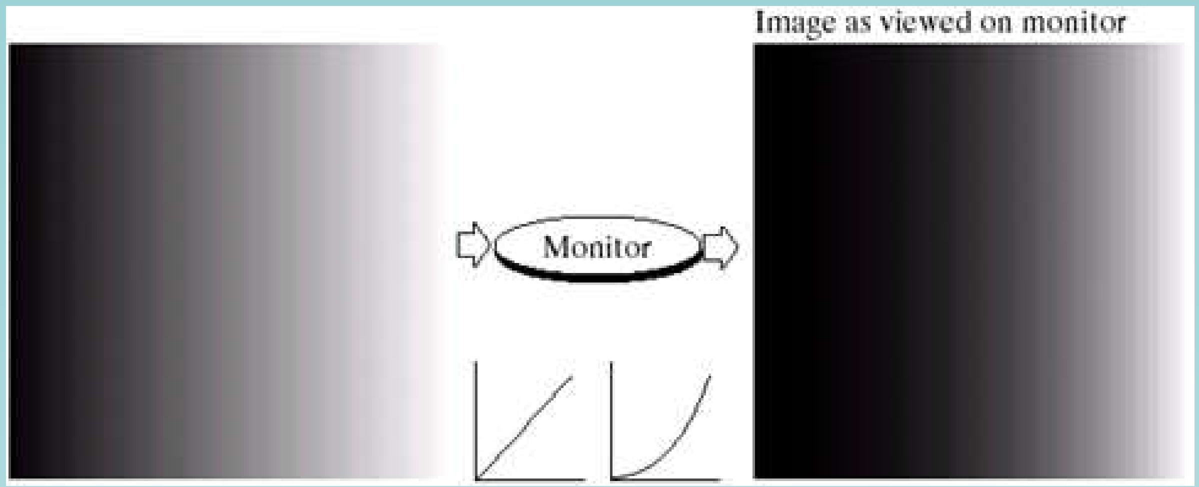
Power Law Transform: $s = cr^{\gamma}$

- Intensity to voltage transform in CRT: a power function with power 1.8 to 2.5
- Makes images darker
- Needs power law transform



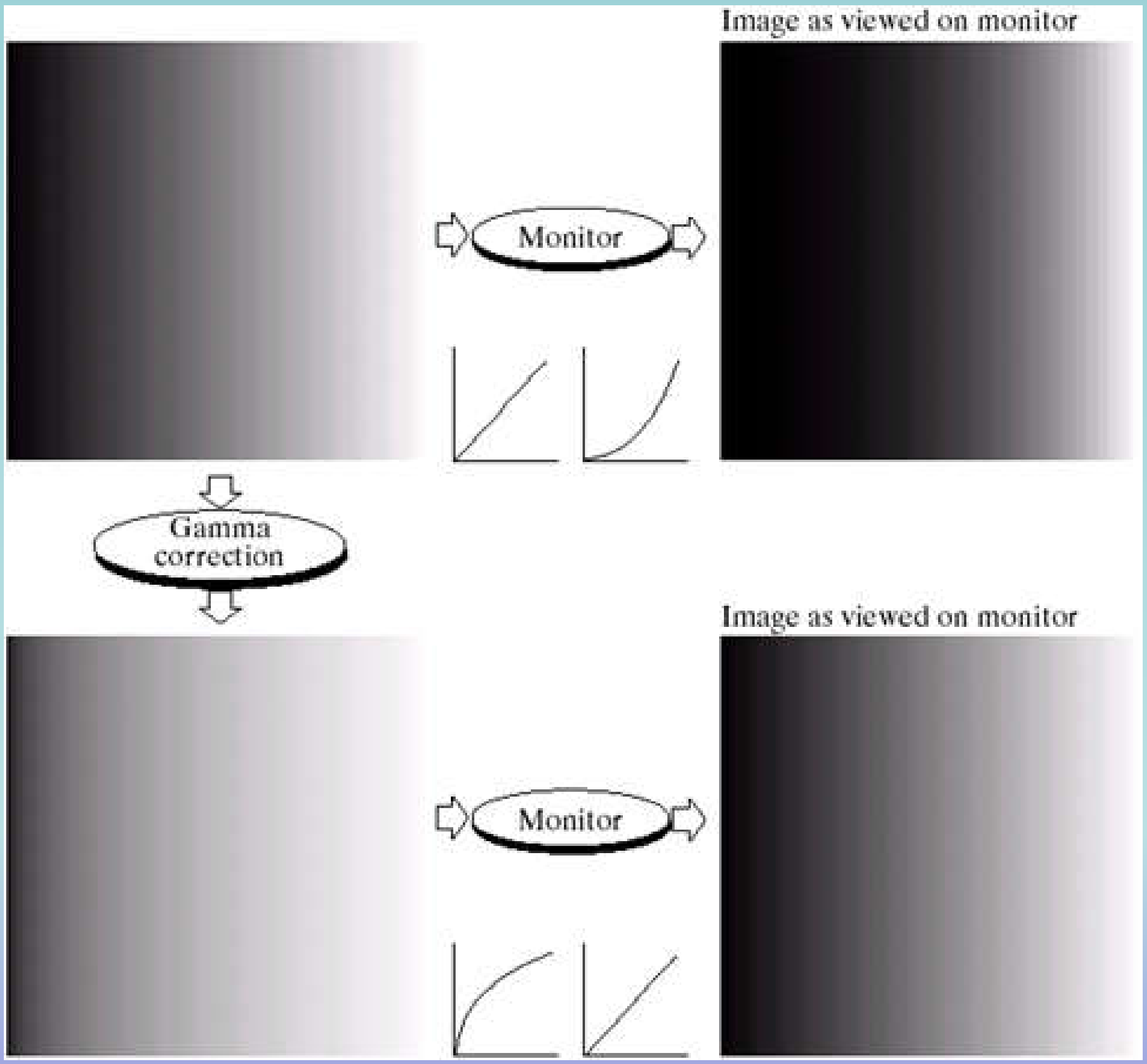
CSE-BUET

Power Law Transform: $s = cr^{\gamma}$



CSE-BUET

Power Law Transform: $s = cr^{\gamma}$



CSE-BUET

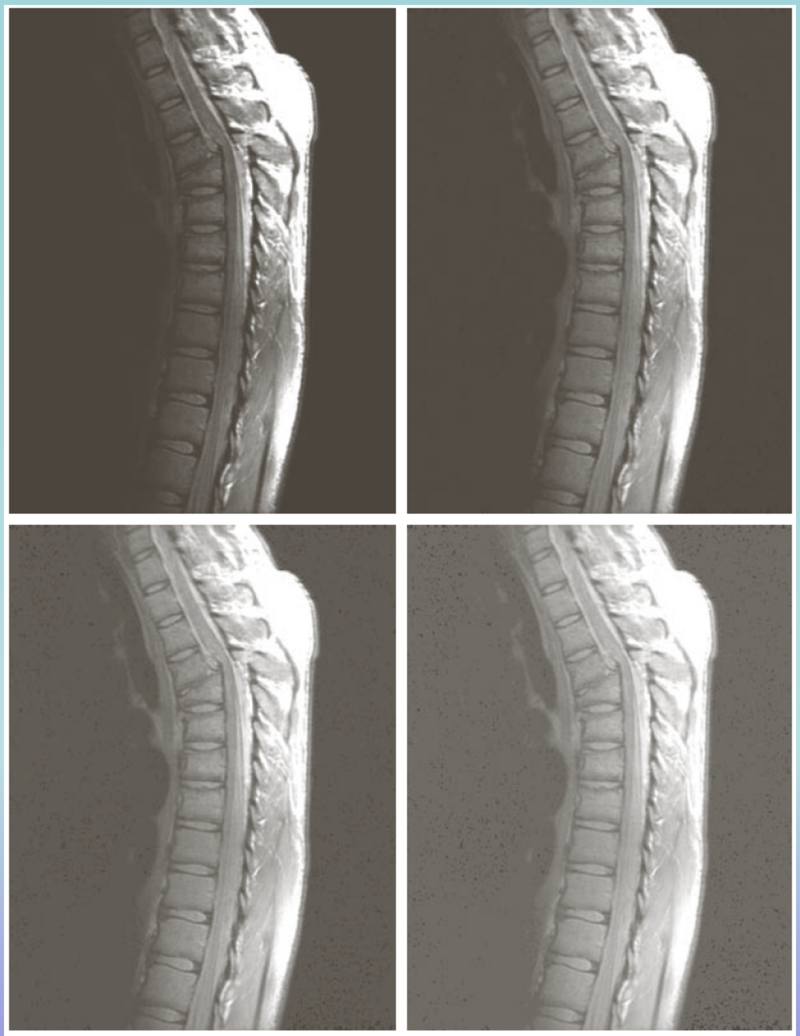
Power Law Transform: $s = cr^{\gamma}$

- $\gamma < 1$: changes darker images to brighter

CSE-BUET

Power Law Transform: $s = cr^{\gamma}$

- $\gamma < 1$: changes darker images to brighter



$\gamma = .6$

$\gamma = .4$

$\gamma = .3$

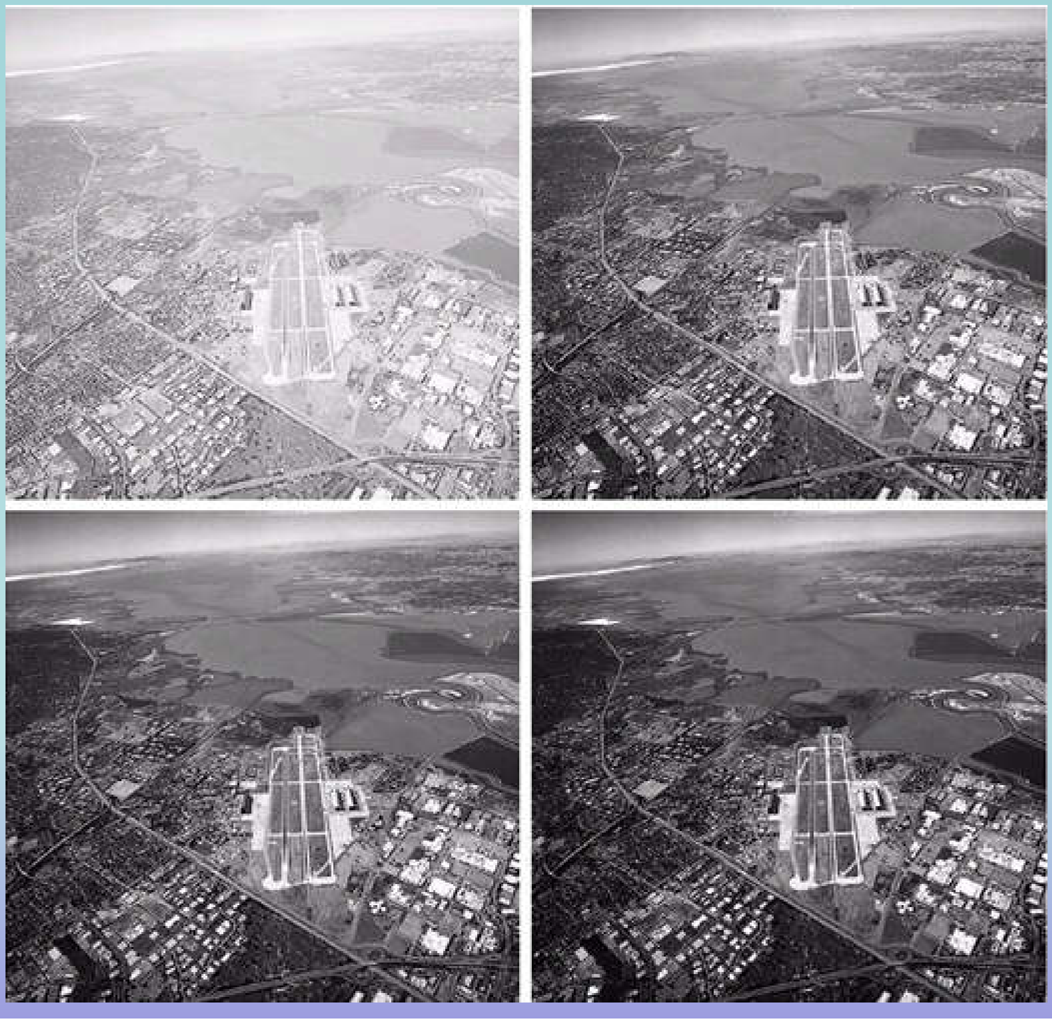
CSE-BUET

Power Law Transform: $s = cr^{\gamma}$

- $\gamma > 1$: changes brighter images to darker

Power Law Transform: $s = cr^{\gamma}$

- $\gamma > 1$: changes brighter images to darker



γ= 3

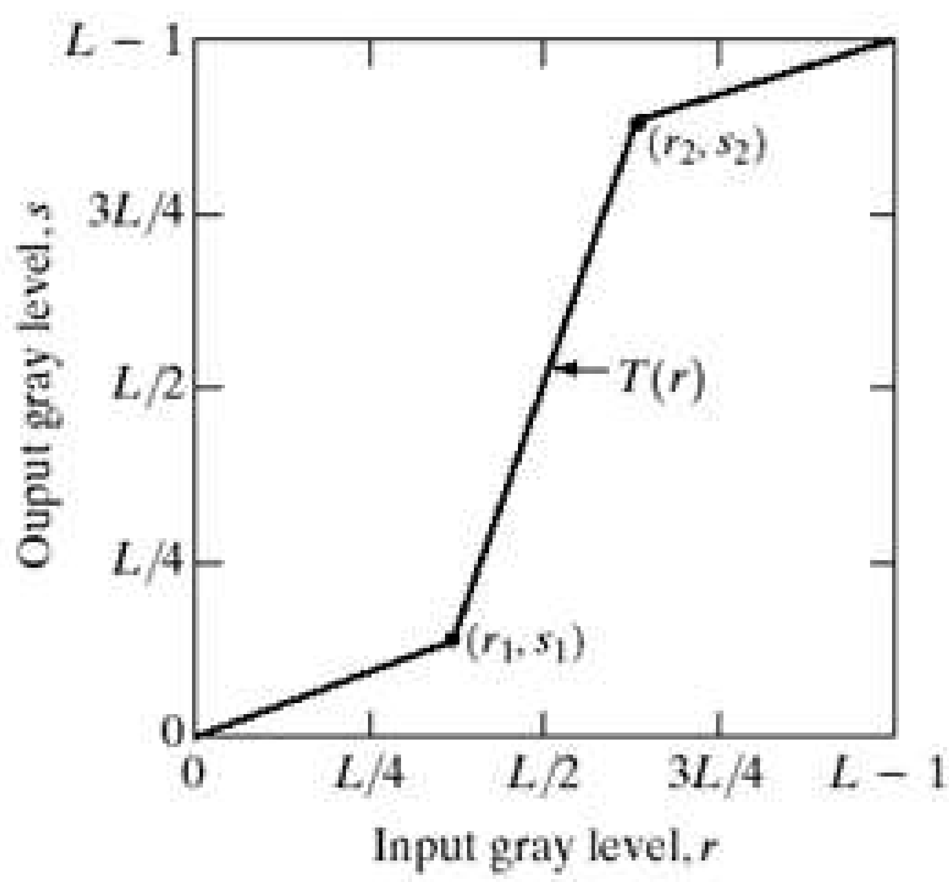
γ= 4

γ= 5

CSE-BUET

Piecewise-Linear Transformation

- Can be made arbitrarily complex
- Idea: bifocal lenses
- Similar to contrast stretching or thresholding

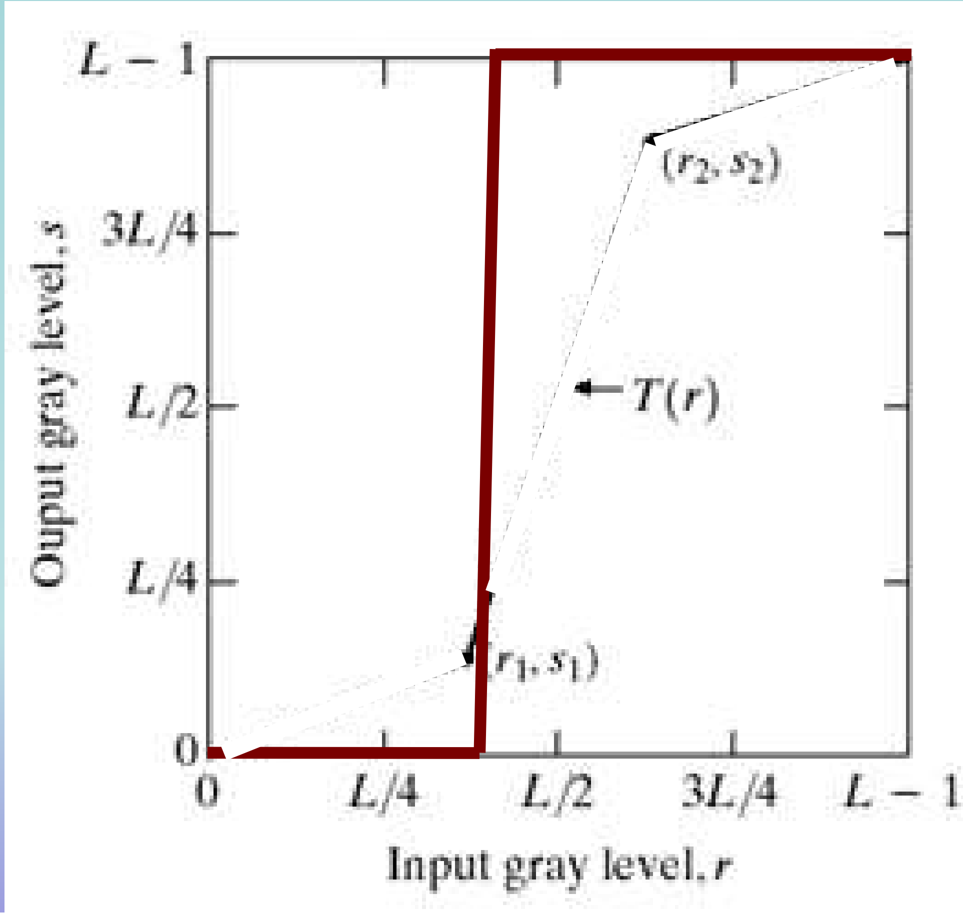


CSE-BUET

Piecewise-Linear Transformation

- The values r_1, r_2, s_1, s_2 change the shape of the transform

- $r_1 = r_2$, $s_1 = 0$, $s_2 = L\text{-}1$: thresholding



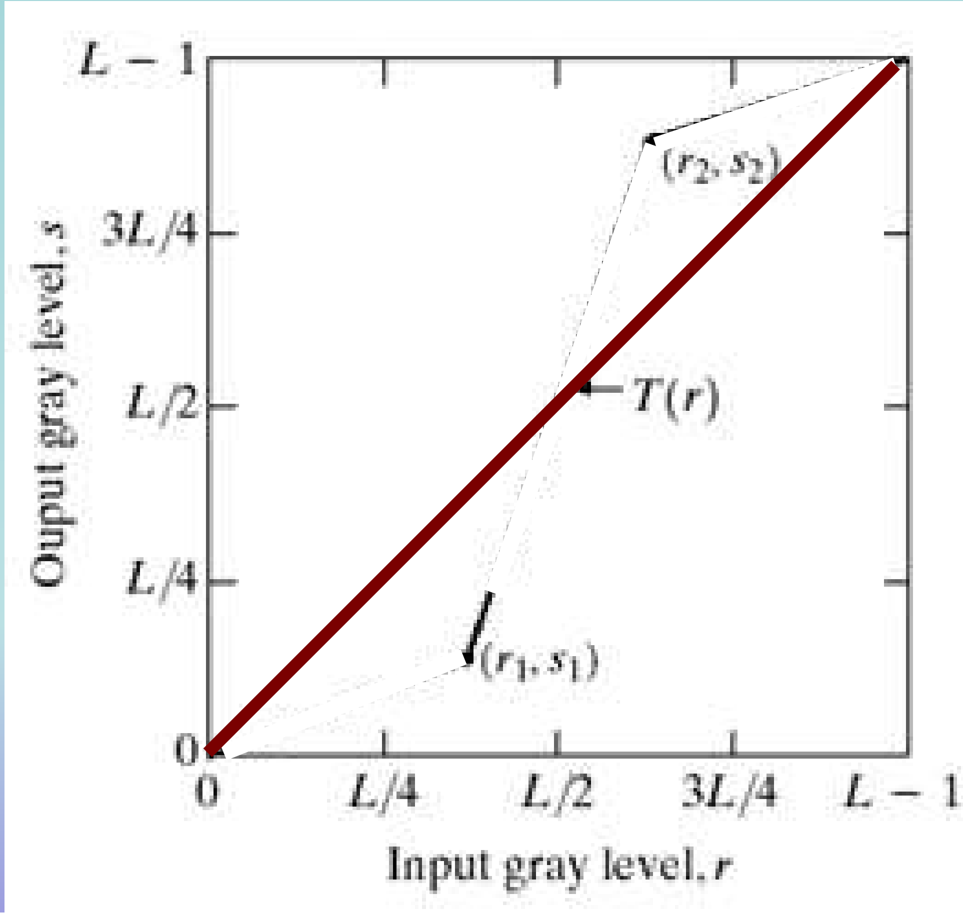
CSE-BUET

Piecewise-Linear Transformation

- The values r_1, r_2, s_1, s_2 change the shape of the transform

- $r_1 = r_2$, $s_1 = 0$, $s_2 = L-1$: thresholding

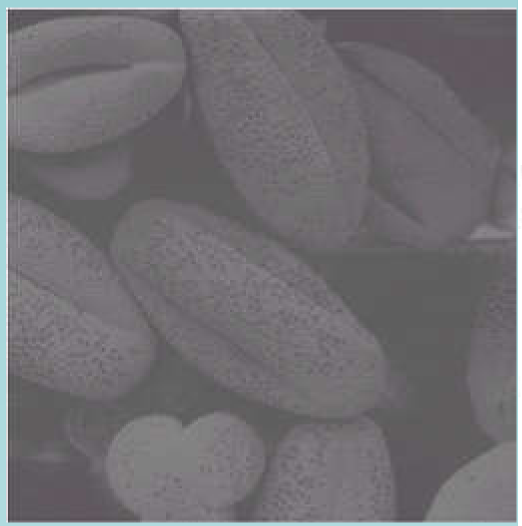
- $r_1 = s_1$, $r_2 = s_2$: linear transformation

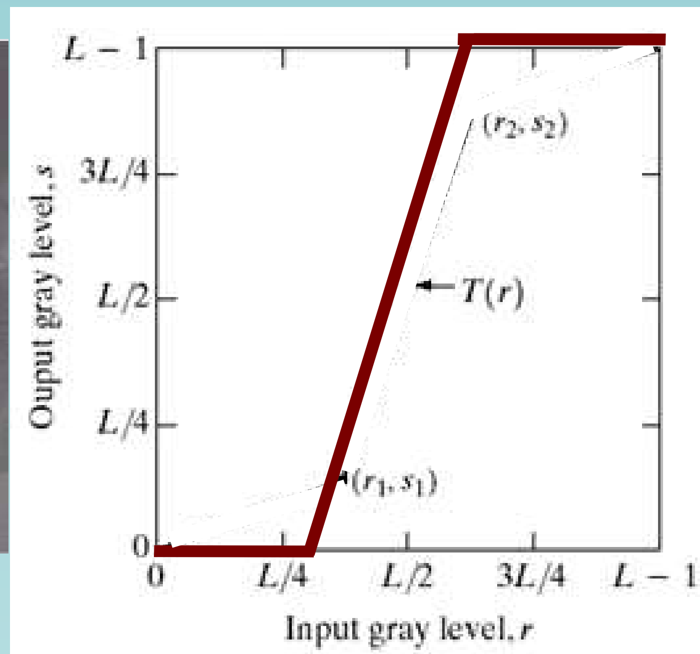


CSE-BUET

Piecewise-Linear Transformation

Piecewise-Linear Transformation

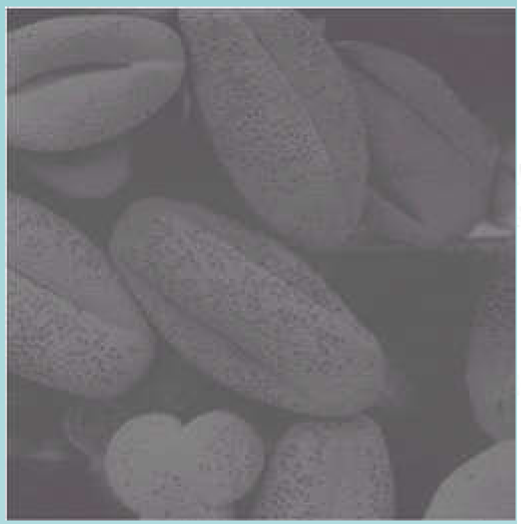


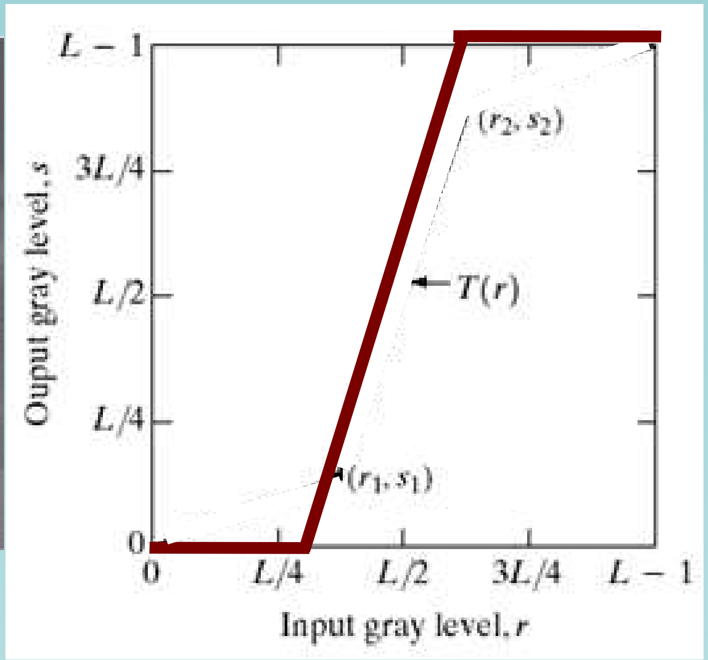


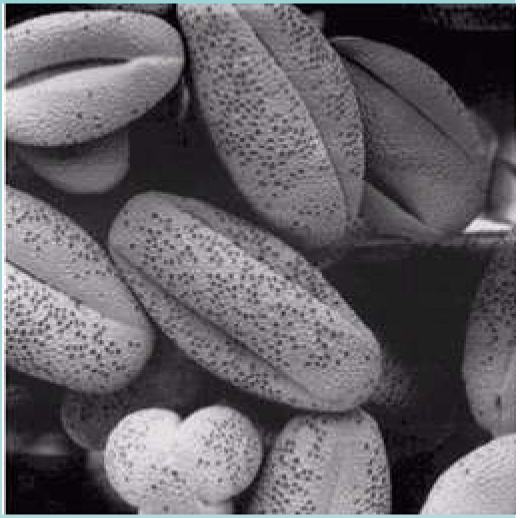
- r_1 = min gray value
- r_2 =max gray value
- s_1=0, s_2 =L-1

CSE-BUET

Piecewise-Linear Transformation



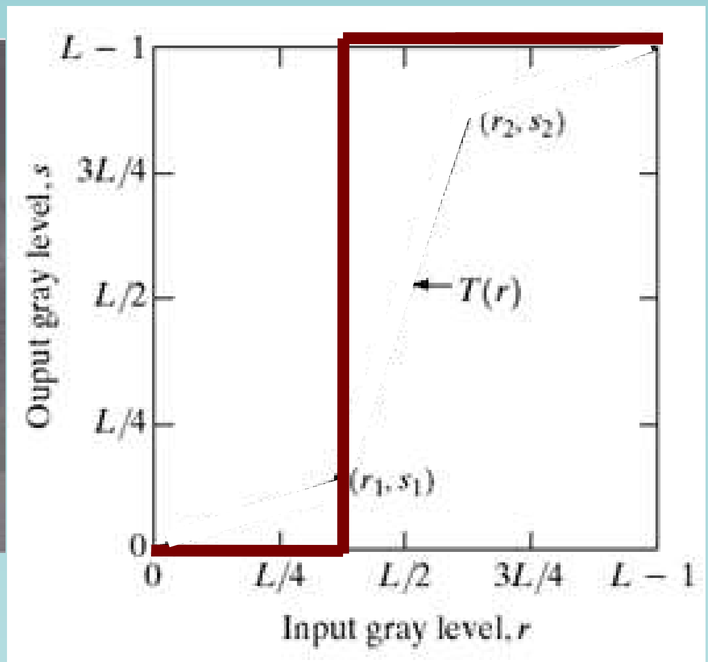




- r_1 = min gray value
- r_2 =max gray value
- s_1=0, s_2 =L-1

CSE-BUET

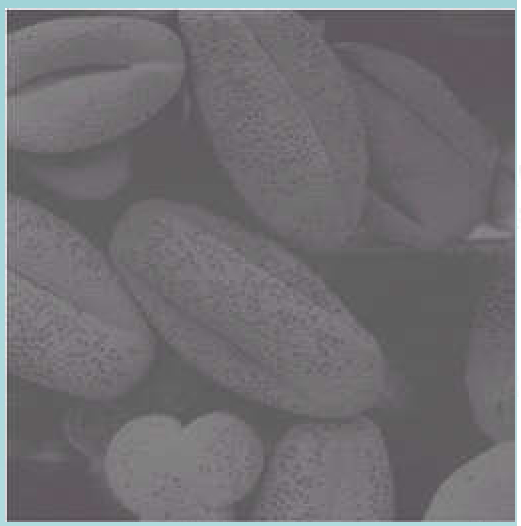
Piecewise-Linear Transformation

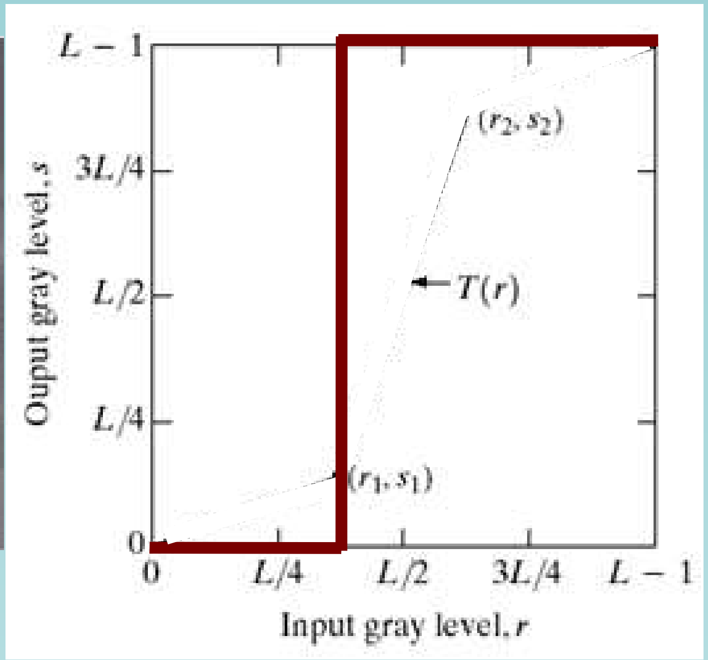


- $r_1= r_2$=m =avg. gray value
- s_1=0, s_2 =L-1

CSE-BUET

Piecewise-Linear Transformation

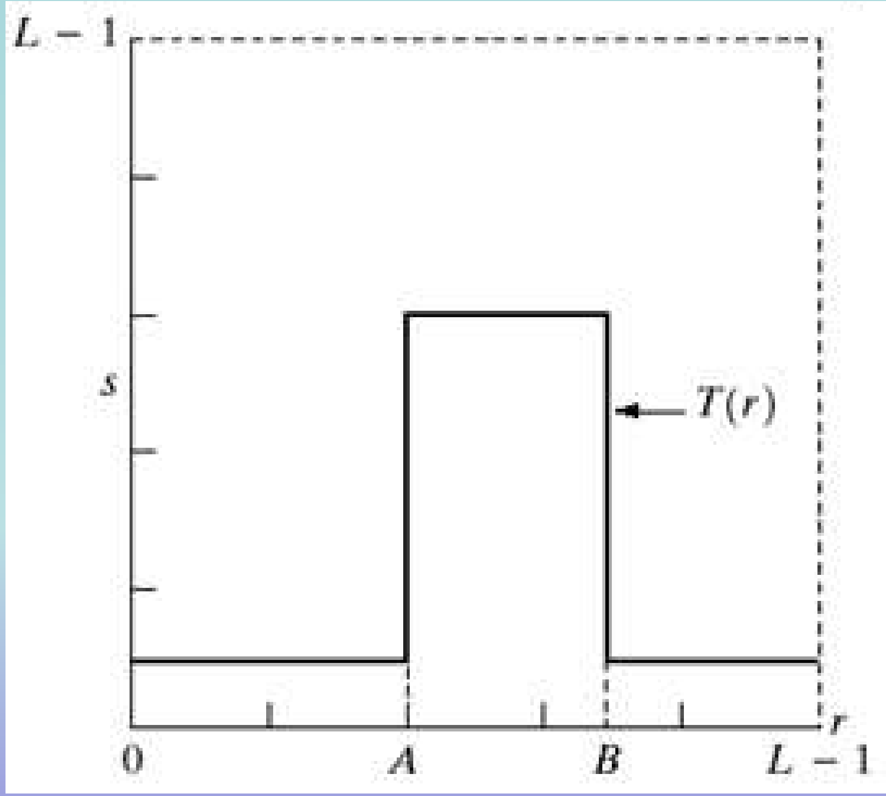




- $r_1 = r_2 = m$ = avg. gray value
- $s_1 = 0$, $s_2 = L-1$

Gray Level Slicing

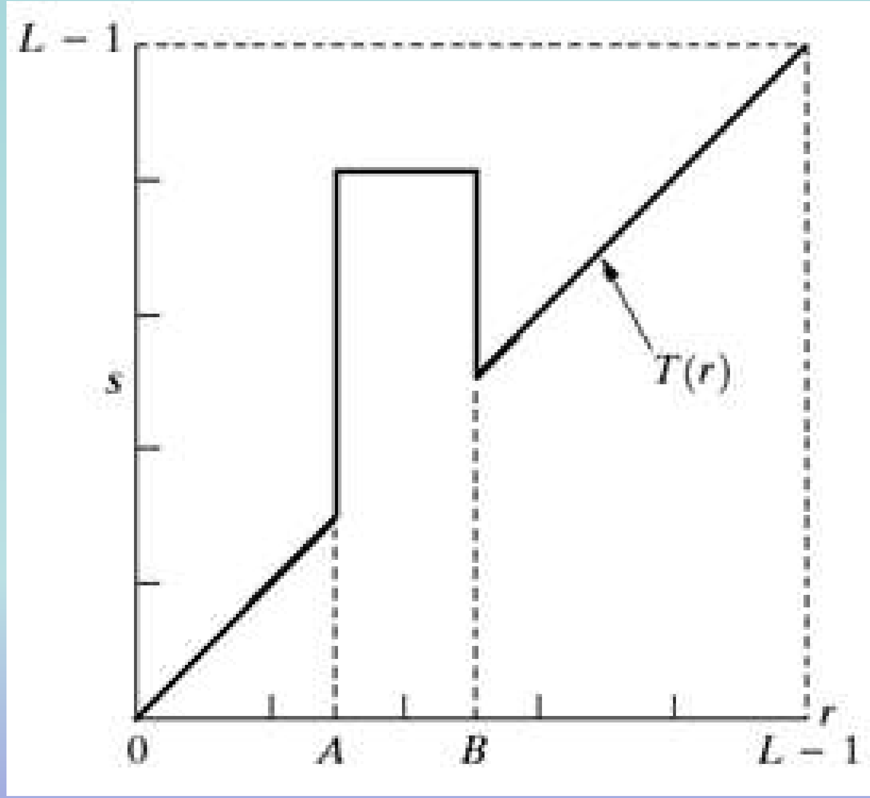
- Highlighting a specific range of gray levels
- Similar to band-pass filtering



CSE-BUET

Gray Level Slicing

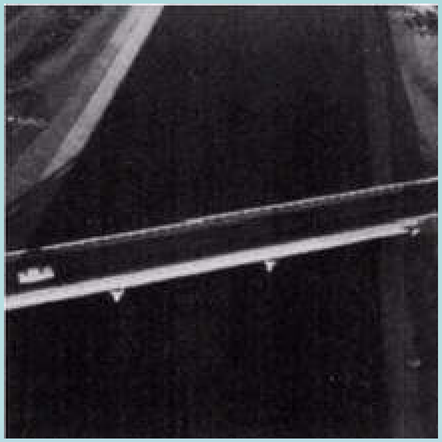
- Highlighting a specific range of gray levels
- Similar to band-pass filtering



CSE-BUET

Gray Level Slicing

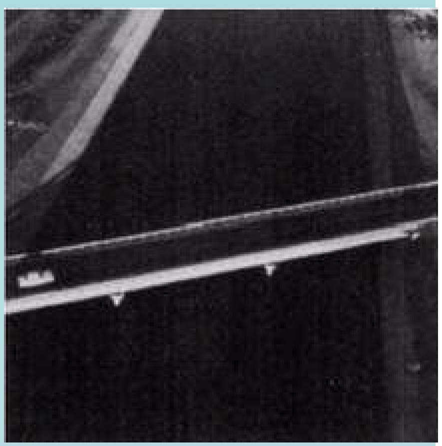
- Highlighting a specific range of gray levels
- Similar to band-pass filtering

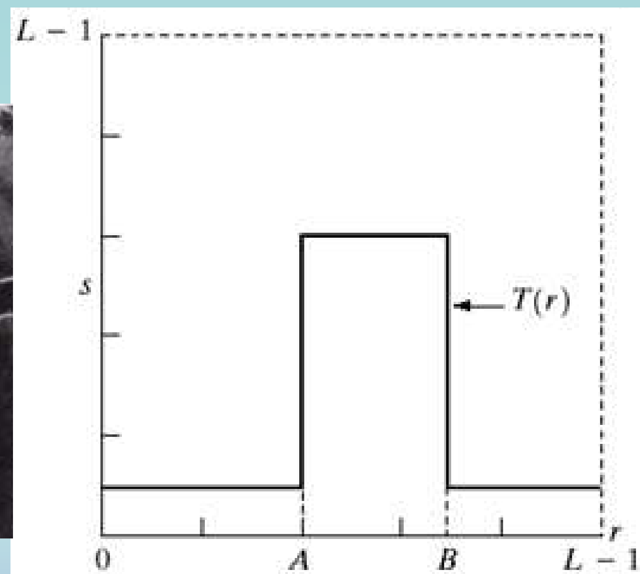


CSE-BUET

Gray Level Slicing: Example(1)

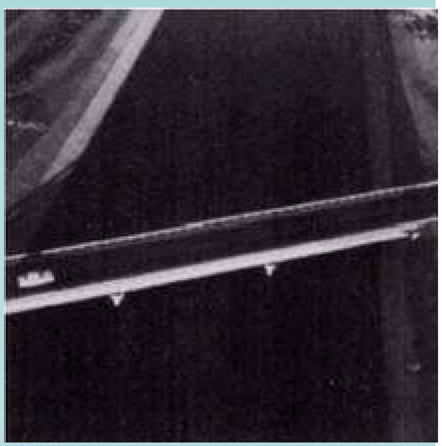
- Highlighting a specific range of gray levels
- Similar to band-pass filtering

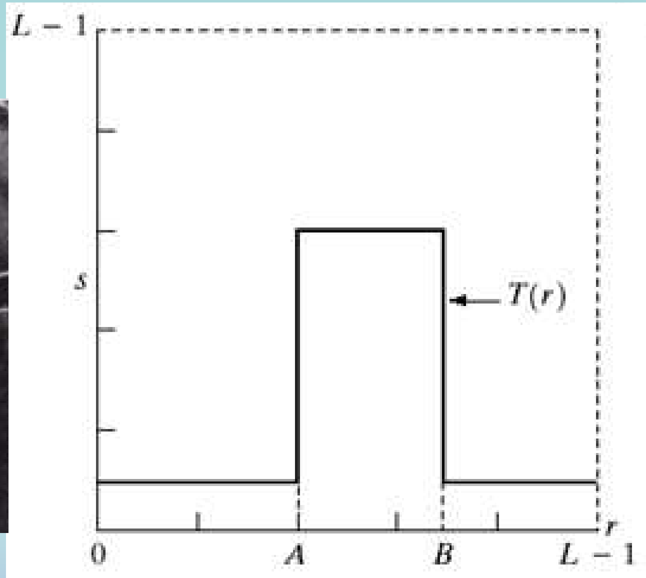


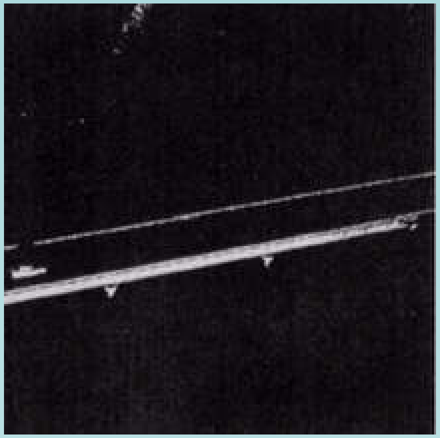


Gray Level Slicing: Example(1)

- Highlighting a specific range of gray levels
- Similar to band-pass filtering

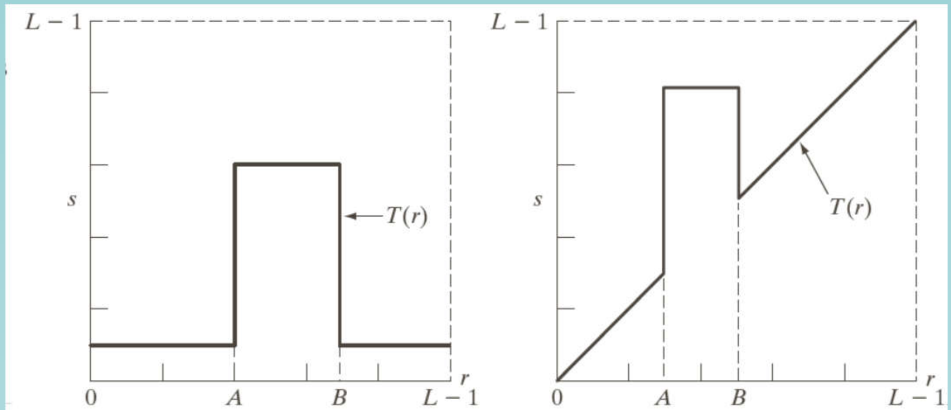


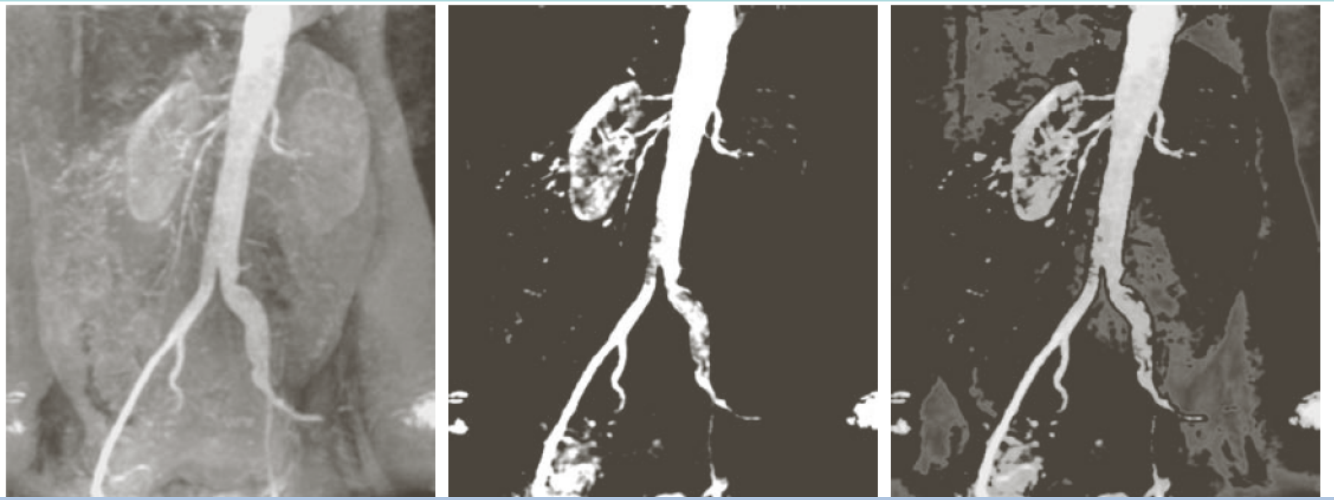




CSE-BUET

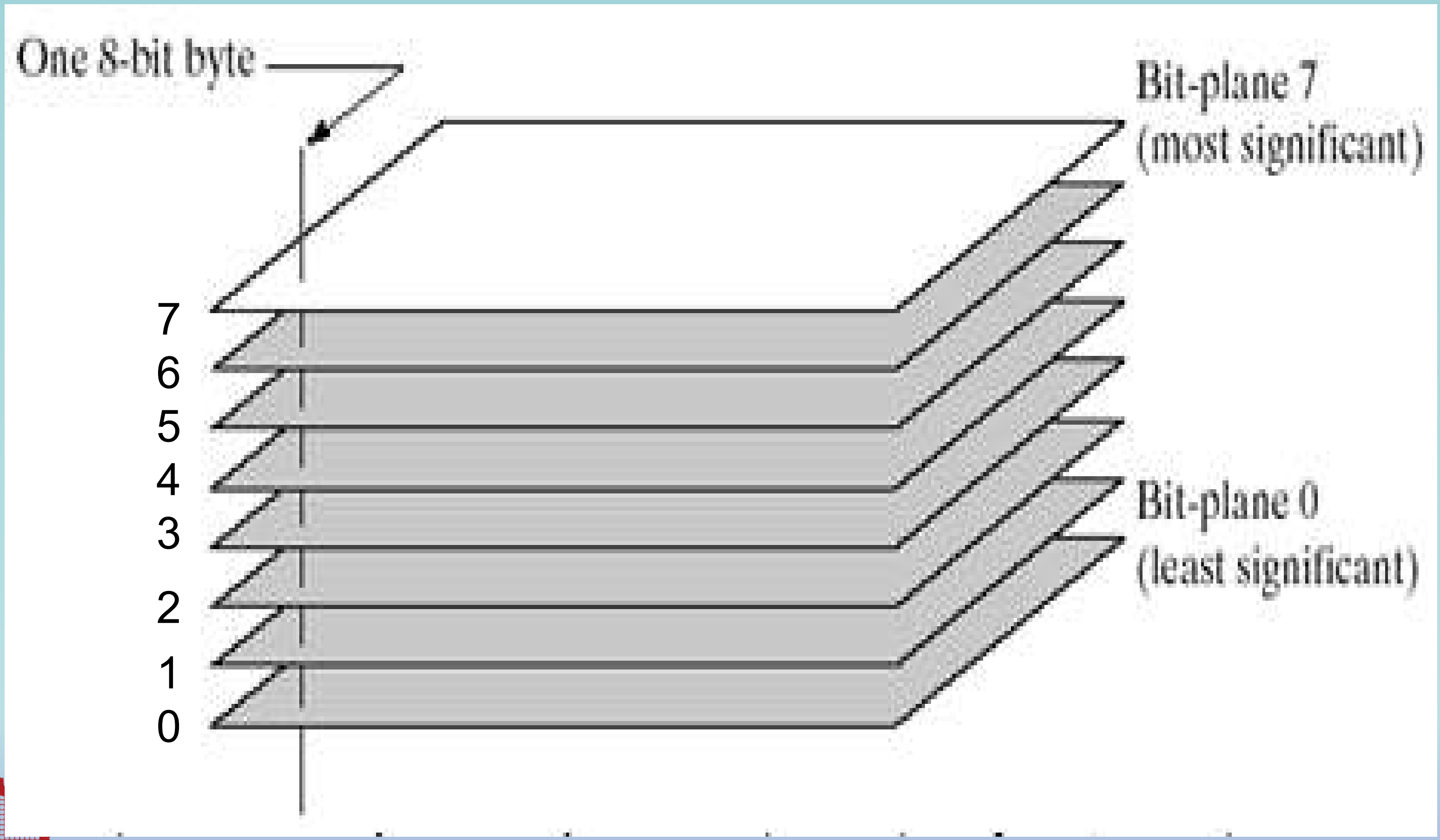
Gray Level Slicing: Example(2)





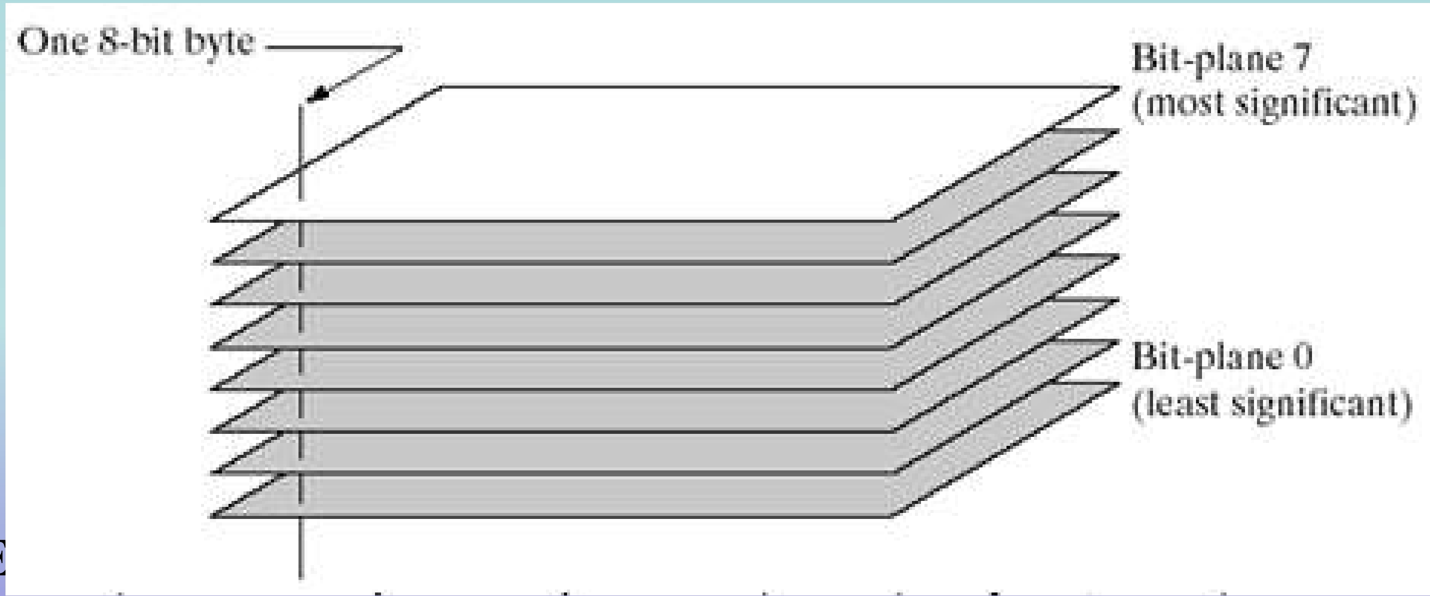
CSE-BUET

Bit-Plane Slicing

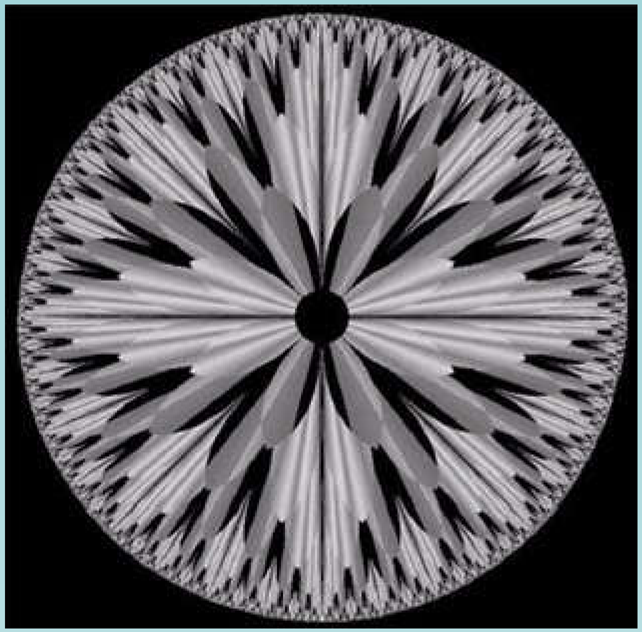


Bit-Plane Slicing

- Gets the contribution of each bit
- Determines the number of bits required to quantize a pixel
- Used in compression



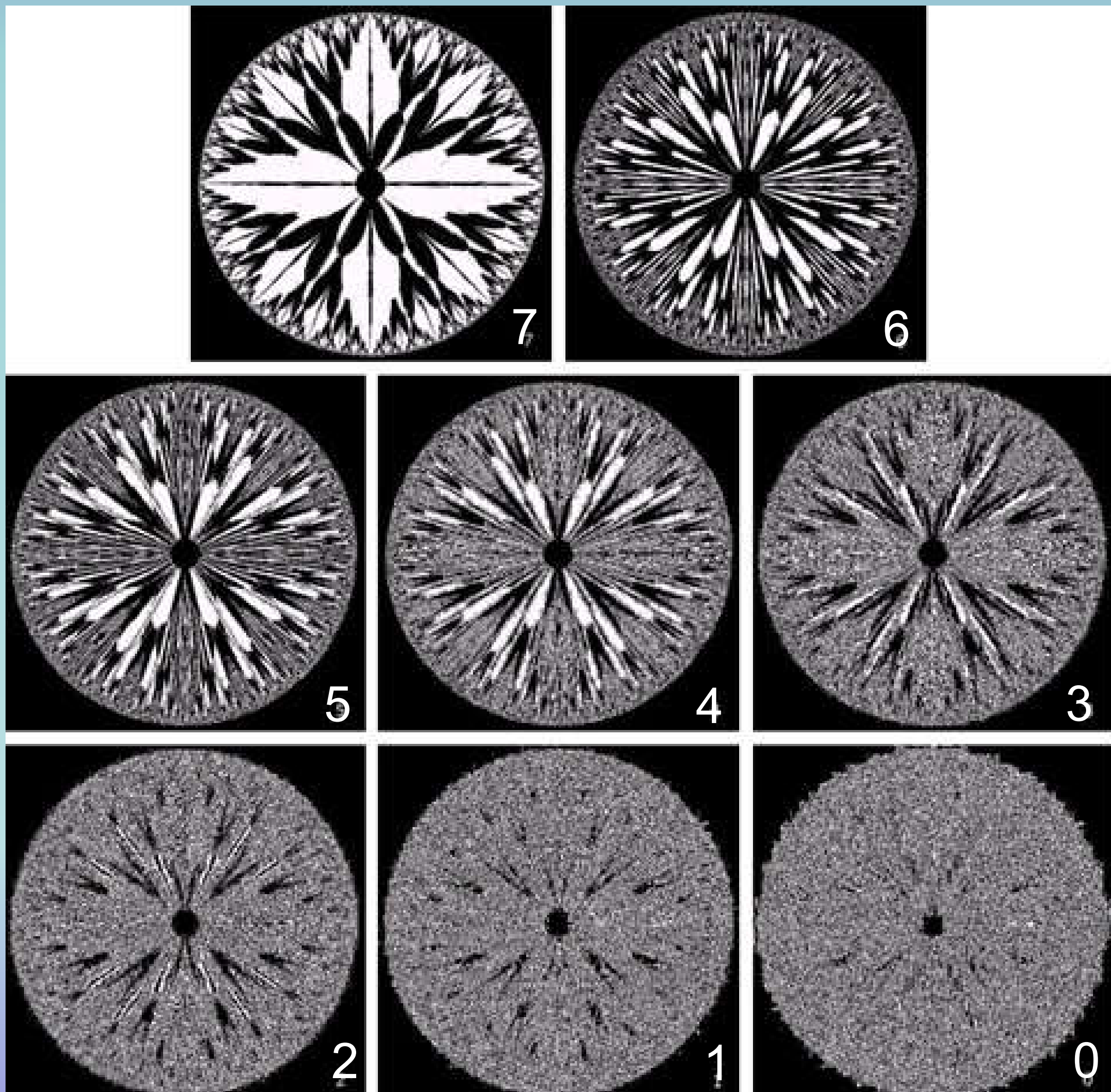
Bit-Plane Slicing: Example (1)



Example Image

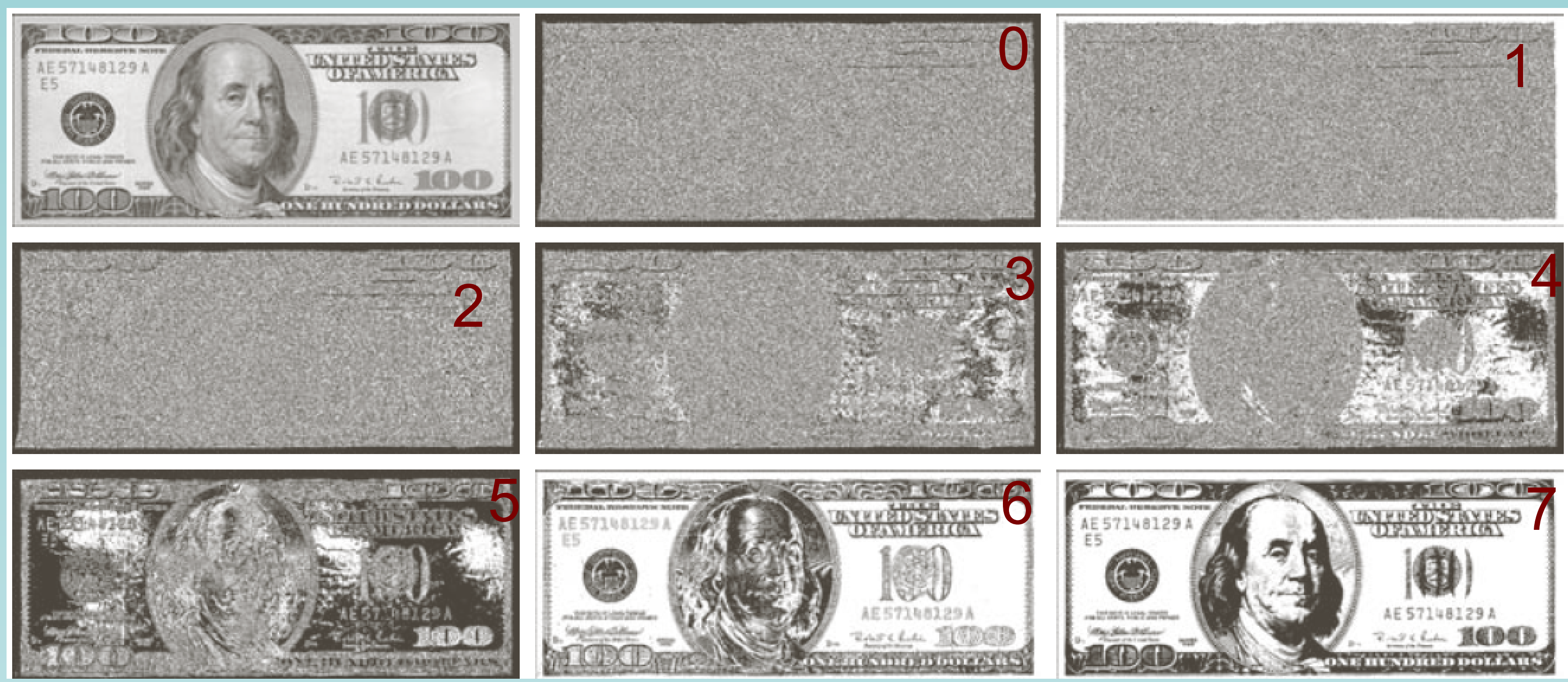
CSE-BUET

Bit-Plane Slicing: Example (1)



CSE-BUET

Bit-Plane Slicing: Example (2)



CSE-BUET