# CoE4TN4
# Image Processing

## Chapter 3: Intensity Transformation and Spatial Filtering

McMaster University

# Image Enhancement

- Enhancement techniques: to process an image so that the result is more suitable than the original image for a specific application.
- Specific: techniques are very much problem oriented
  - A technique that is useful for X-ray images might not be the best for pictures transmitted from a space probe.
- Enhancement approaches:
  1. Spatial domain
  2. Frequency domain

McMaster University

# Basics

- Spatial domain: collection of pixels forming an image
- Spatial domain techniques are techniques that operate directly on pixels
- Frequency domain techniques are based on modifying the Fourier transform of an image

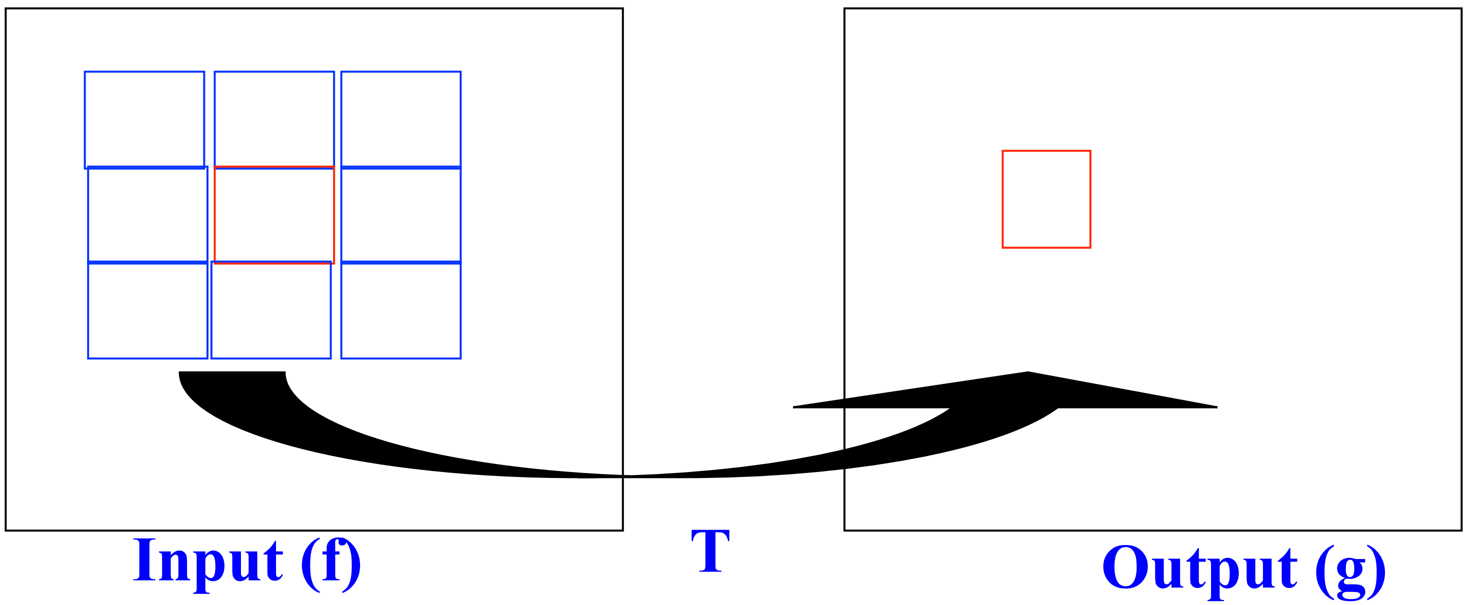
McMaster University

# Spatial domain: background

- Spatial domain processing: procedures that operate directly on the pixels of the input image to generate the pixel values of processed (output) image.
- g(x,y)=T[f(x,y)]
  - f(x,y): input image
  - g(x,y): processed image
  - T: an operator defined over some neighborhood of (x,y)

# Spatial domain: background

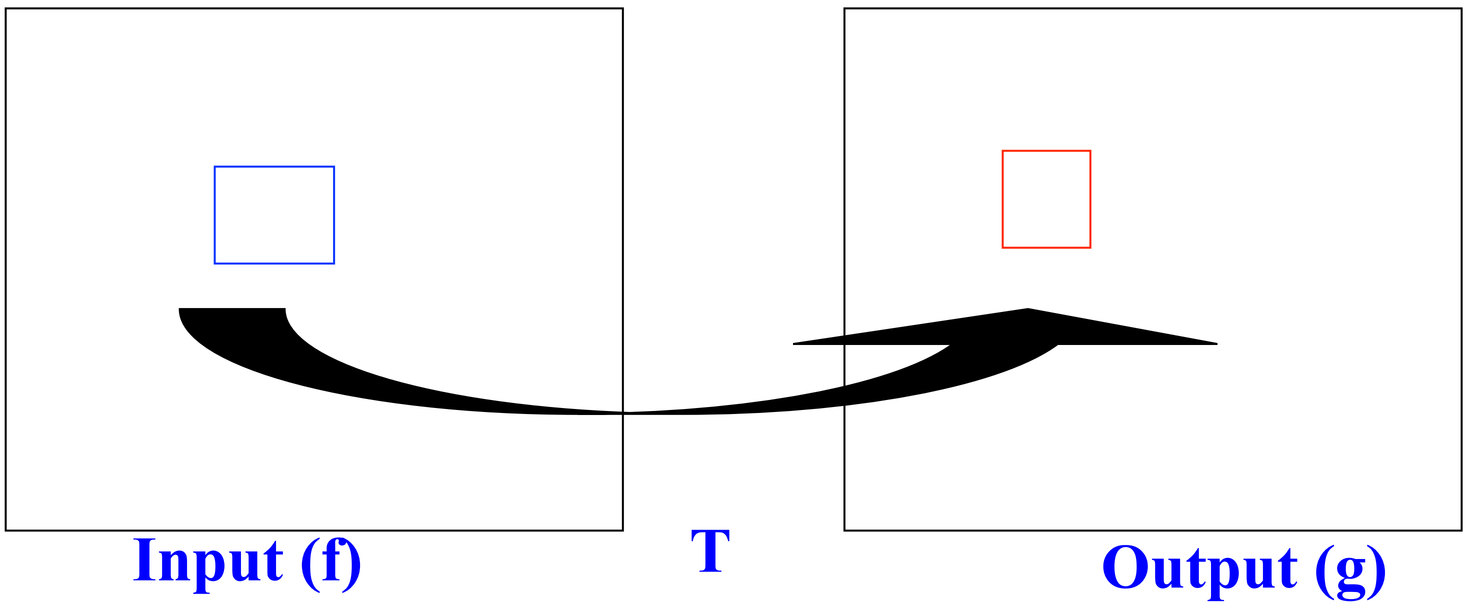
- Neighborhood around (x,y): usually a square or rectangular subimage area centered at (x,y).
- Center of subimage is moved pixel by pixel. At each location (x,y) the operator T is applied to find the value of g(x,y).




McMaster University

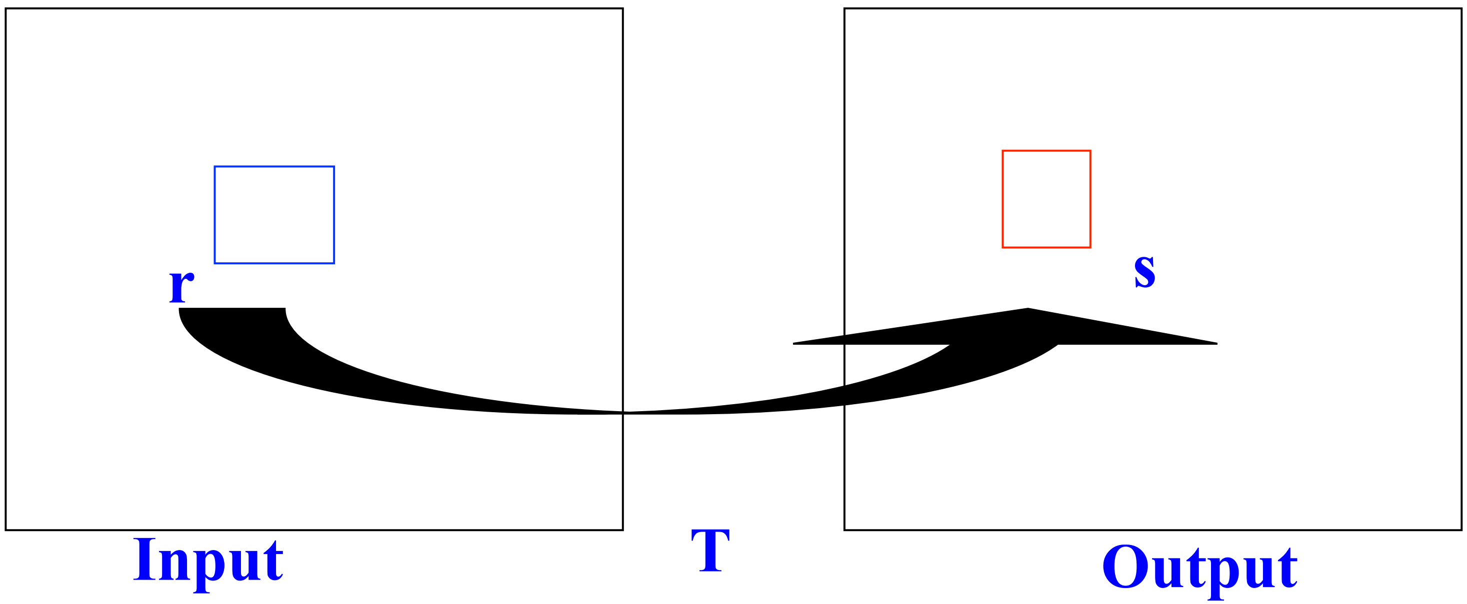
# Spatial domain: background

- Simplest form of T (the operator): the neighborhood is 1x1.
- g(x,y) only depends of value of f at (x,y).
- T: a gray-level transformation (mapping)
- This type of processing is called point processing



# Spatial domain: Point Processing

- s=T(r)
- r: gray-level at (x,y) in original image f(x,y)
- s: gray-level at (x,y) in processed image g(x,y)
- T is called gray-level transformation or mapping

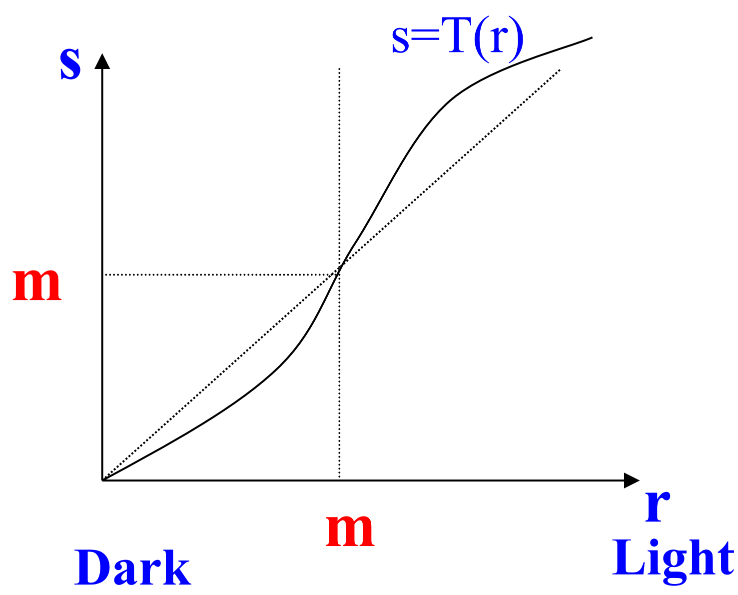



McMaster University

# Spatial domain: Point Processing

- The relation s=T(r) can be shown as a curve
- Example: effect of the transform shown below is that an image with higher contrast than the original image
- How: the gray levels below m are darkened and the levels above m are brightened .

**Contrast stretching**




McMaster University

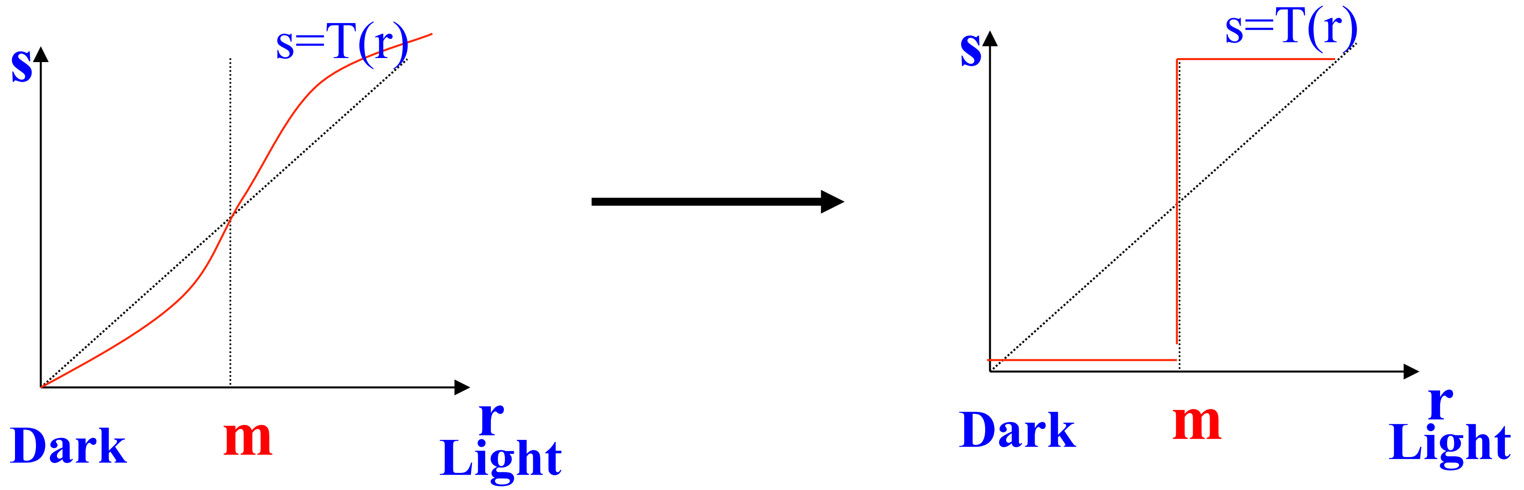
# Spatial domain: Point Processing

## Contrast stretching

McMaster University

# Spatial domain: Point Processing

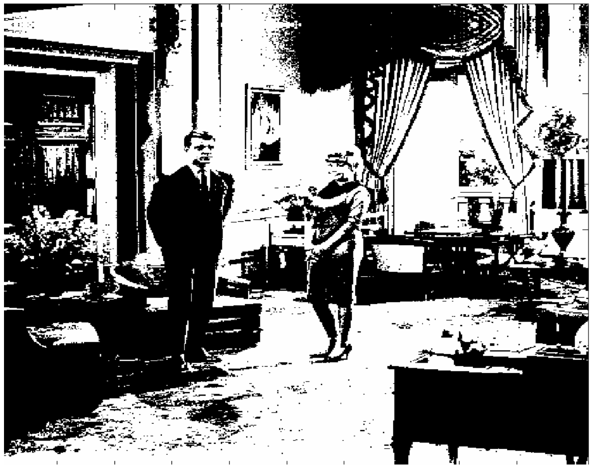
- Limiting case: produces a binary image (two level) from the input image




**Thresholding**

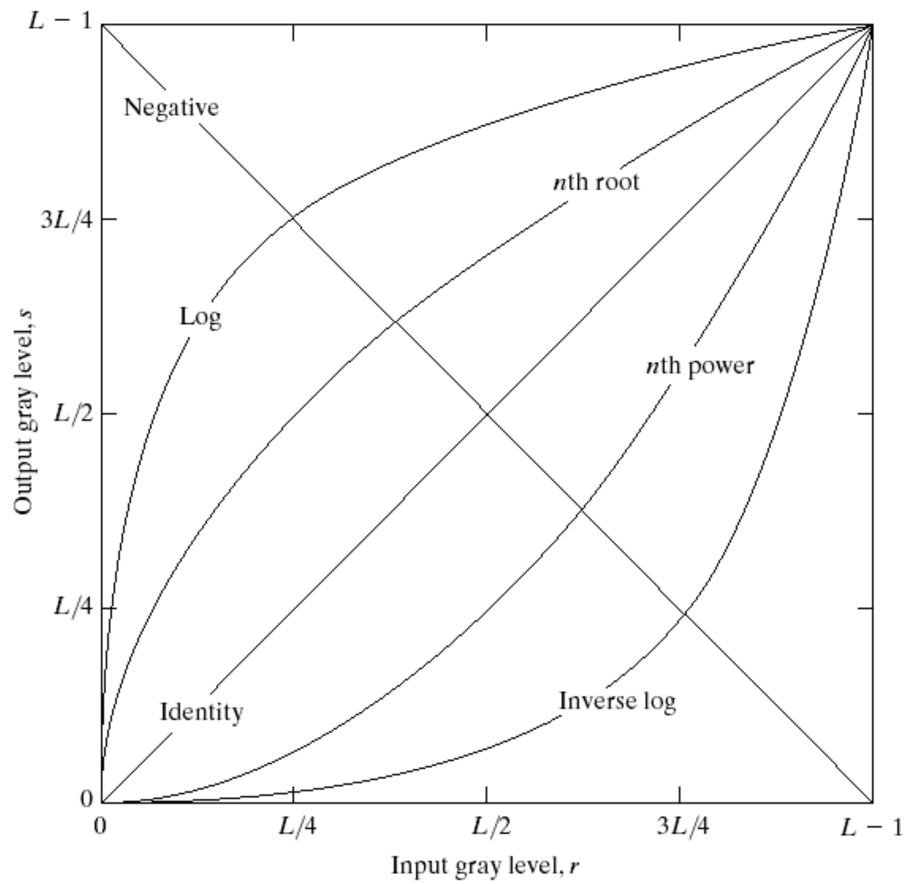
# Spatial domain: Point Processing

**Contrast stretching**



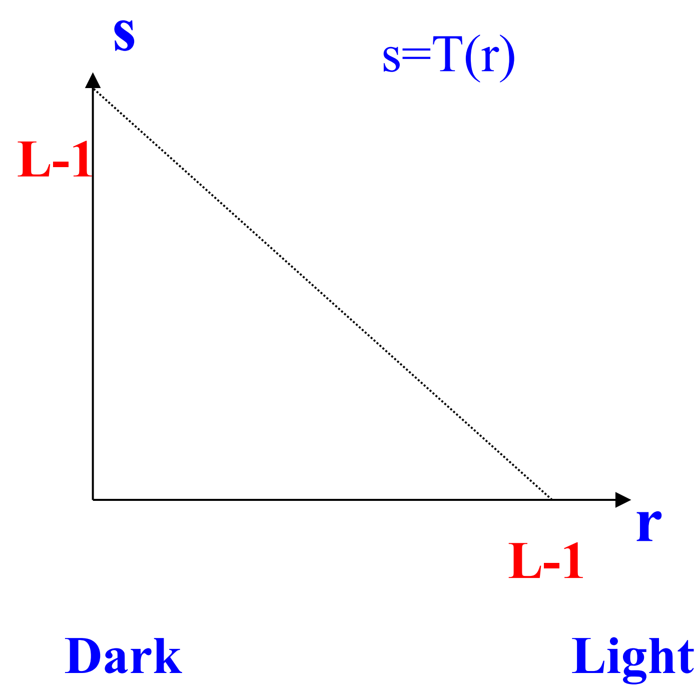
# Gray-level transforms

**FIGURE 3.3** Some basic gray-level transformation functions used for image enhancement.



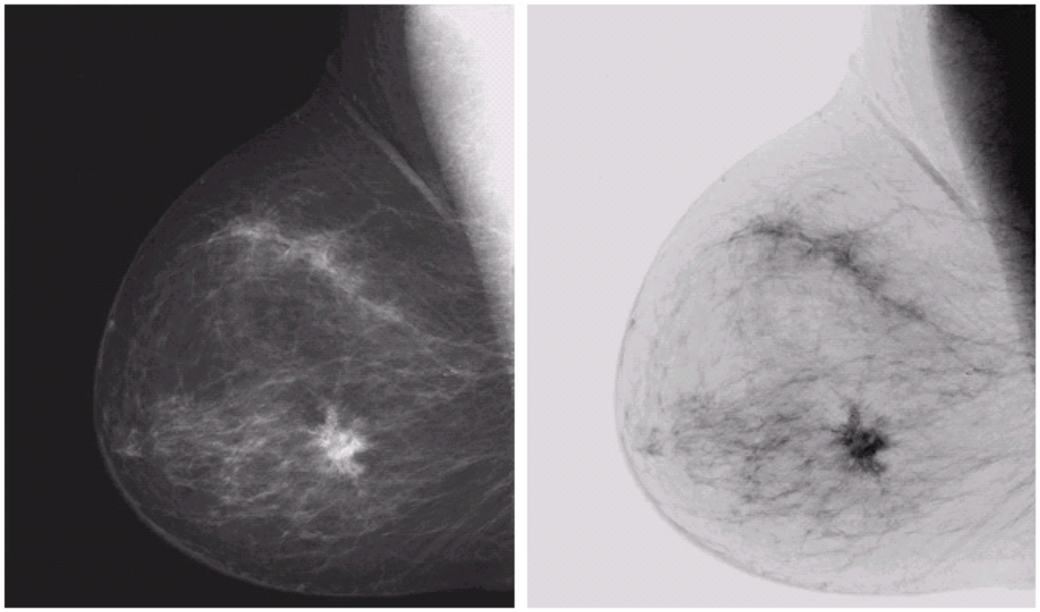
# Image Negative

- Suited for enhancing white detail embedded in dark regions
- Has applications in medical imaging




McMaster University

# Image Negative



a b

**FIGURE 3.4** (a) Original digital mammogram. (b) Negative image obtained using the negative transformation in Eq. (3.2-1). (Courtesy of G.E. Medical Systems.)

# Log Transformation

$$s = c\log(1 + r)$$

- Log transformation: maps a narrow range of low gray-level values in the input image into a wider range of output levels.
- The opposite is true for higher values of input levels
- Expand the values of dark pixels in an image while compressing the higher-level values

# Log Transformation

- Log transformation has the important property of compressing the dynamic range of images with large variations in pixel values
- Compression of dynamic range: Sometimes the dynamic range exceeds capability of the display device. An effective way to compress the dynamic range of pixel values is

$$s = c\log(1 + r)$$

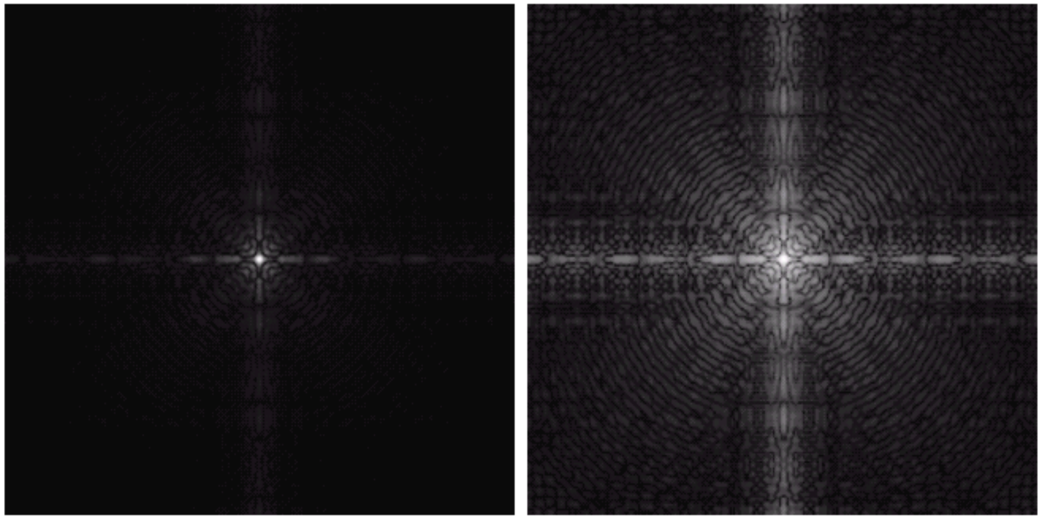
- Example: range=[0, $2.5\text{x}10^6$] ⟹ [0, 6.4]

  choose c=255/6.4

McMaster University

a b

**FIGURE 3.5** (a) Fourier spectrum. (b) Result of applying the log transformation given in Eq. (3.2-2) with $c = 1$.



McMaster University

# Power-law transformation

$$s = cr^{\gamma}$$

- If $\gamma < 1$: transformation maps a narrow range of dark input values into a wider range of output values
- If $\gamma > 1$:opposite of the above effect
- Many devices used for image capture, printing and display respond according to a power low.
- The process used to correct this power-low response phenomena is called gamma correction.
- Exp: CRT devices, intensity to voltage relation is a power function with $\gamma$=1.8 to 2.5
  - The output of CRT is a darker image
- To correct we pre-process the image with $s = cr^{1/\gamma}$

McMaster University

# Power-law transformation

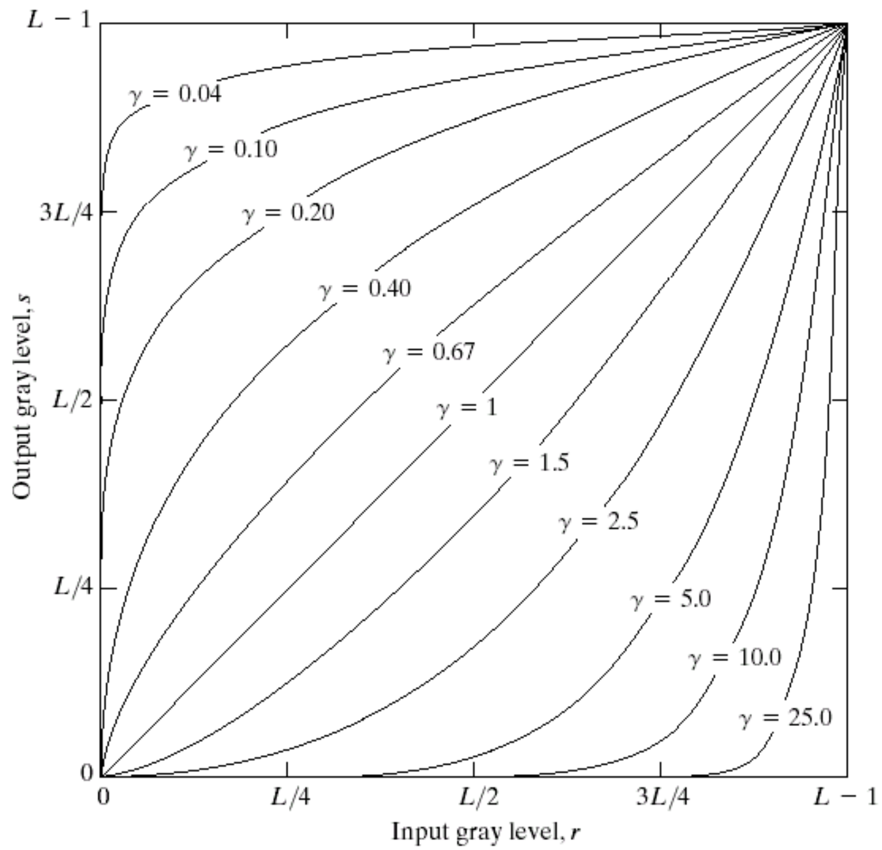



**FIGURE 3.6** Plots of the equation $s = cr^{\gamma}$ for various values of $\gamma$ ($c = 1$ in all cases).

# Contrast stretching

- Low contrast images occur often due to poor or non-uniform lighting conditions or due to non-linearity or small dynamic range of the imaging sensor.
- The transformation looks like:

$$s = \begin{cases} \alpha r, & 0 \le r \le r_1 \\ \beta(r - r_1) + s_1, & r_1 \le r \le r_2 \\ \gamma(r - r_2) + s_2, & r_2 \le r \le L-1 \end{cases}$$

- The locations of points $(r_1, s_1)$ and $(r_2, s_2)$ control the shape of the transformation function.

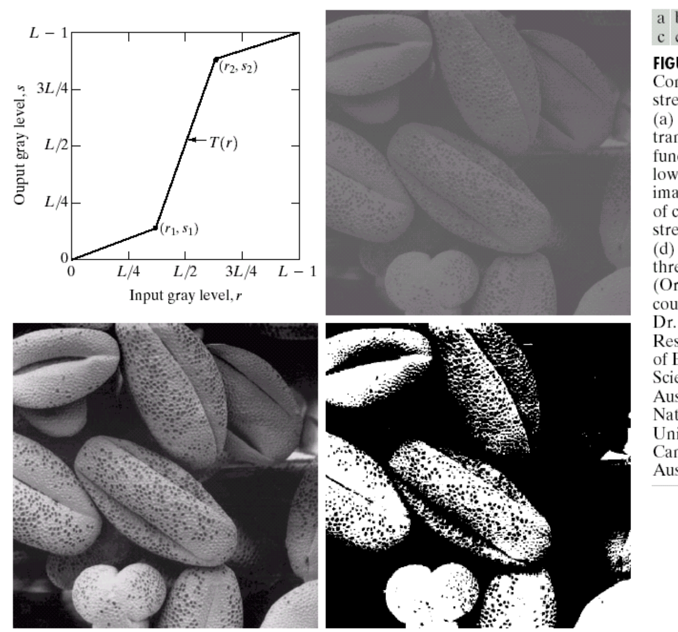



**FIGURE 3.10** Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

# Contrast stretching

- Special cases:
  - if $r_1 = s_1$ & $r_2 = s_2$, transformation is linear (no change)
  - if $r_1 = r_2$, $s_1 = 0$ & $s_2 = L\text{-}1$, thresholding transformation

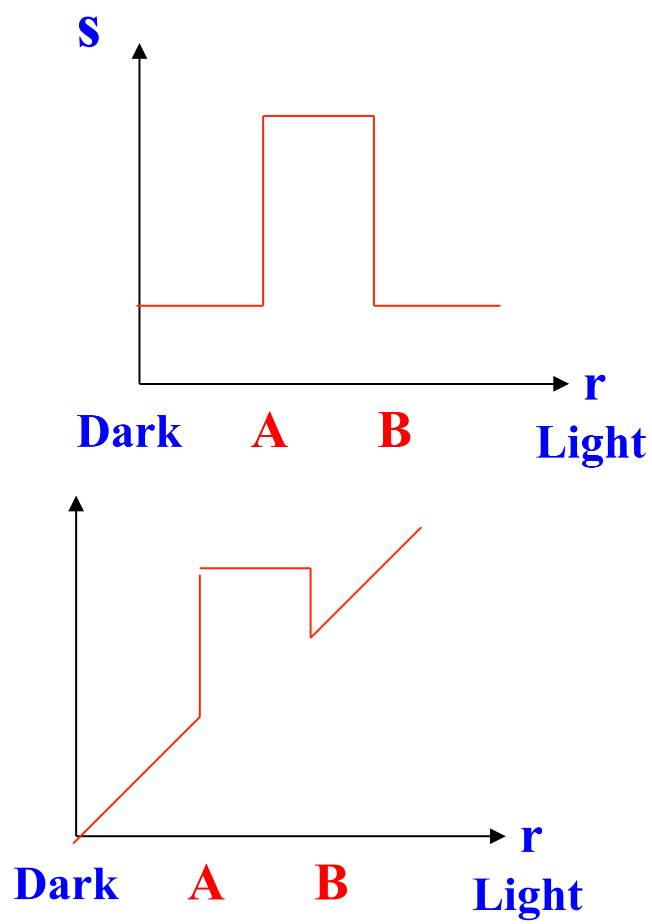
McMaster University

# Intensity-level Slicing

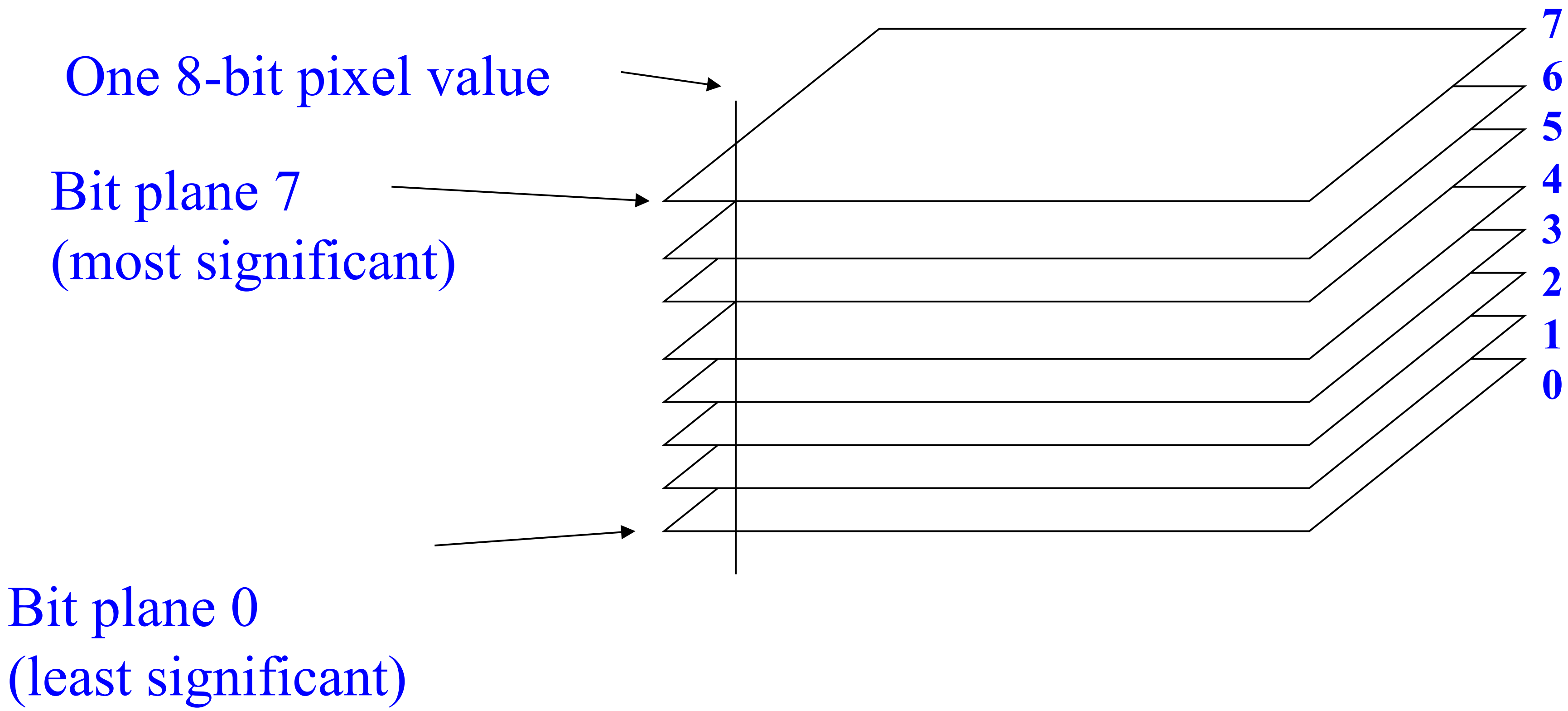
- Highlights a specific range of gray-levels in an image

2 basic methods:

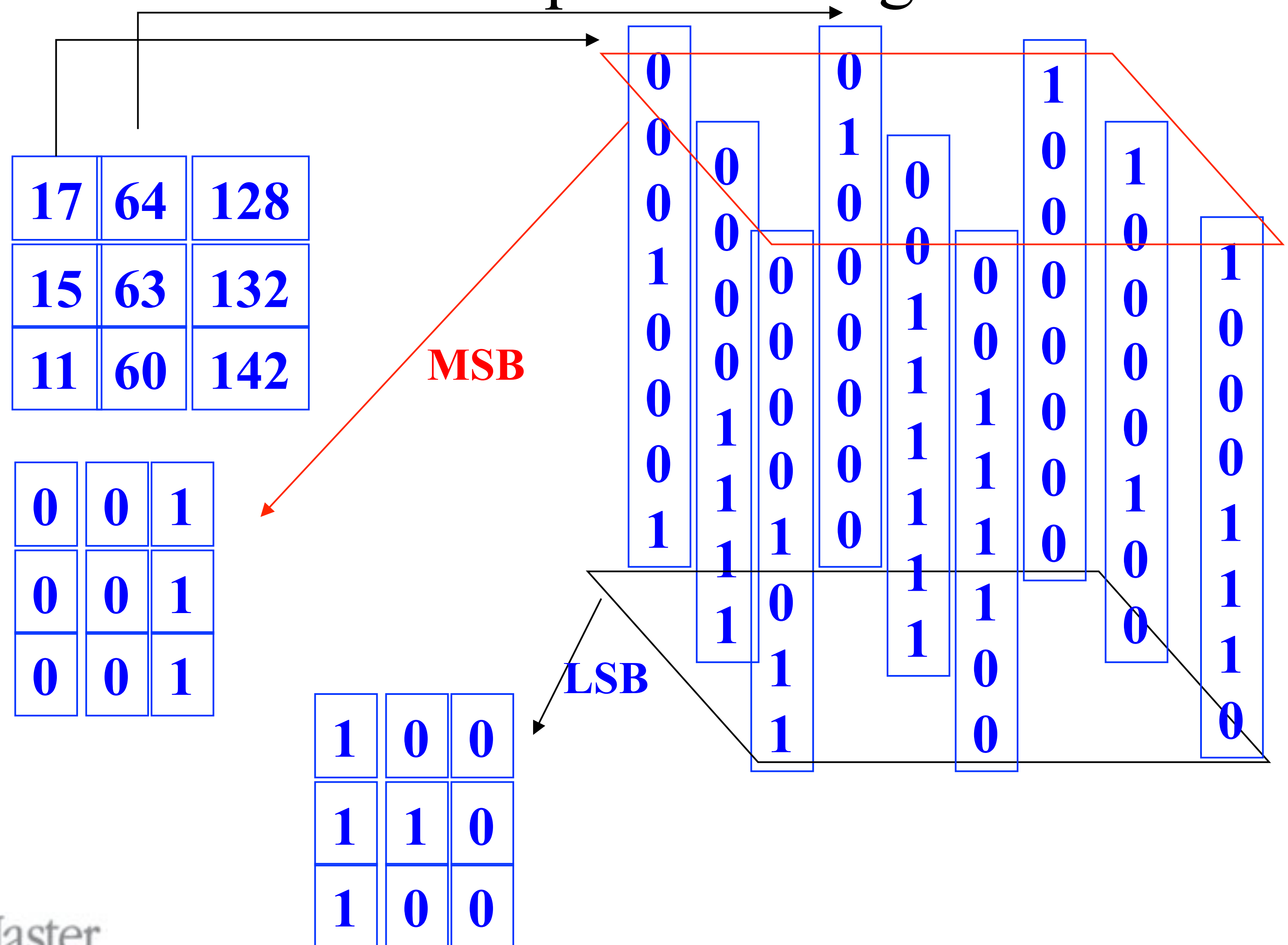
1. Display a high value for all gray levels in the range of interest and a low value for all other
2. Brighten the desired range of gray levels but preserve the gray level tonalities



# Bit plane slicing



# Bit plane slicing



# Bit plane slicing

a b c
d e f
g h i

**FIGURE 3.14** (a) An 8-bit gray-scale image of size 500 × 1192 pixels. (b) through (i) Bit planes 1 through 8, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image.

# Bit plane slicing

- Higher order bit planes of an image carry a significant amount of visually relevant details
- Lower order planes contribute more to fine (often imperceptible) details

McMaster University

# Histogram Processing

- Histogram of a digital image is a discrete function that is formed by counting the number of pixels in the image that have a certain gray level.
- Often the histogram is normalized by dividing by the total number of pixels in the image
- In an image with gray levels in [0,L-1] normalized histogram is given by $p(r_k)= n_k/n$ where:
  - $r_k$ is the k th gray level, k=0, 1, 2, …, L-1
  - $n_k$ number of pixels in the image with gray level $r_k$
  - n total number of pixels in the image
- Loosely speaking, $p(r_k)$ gives an estimate of the probability of occurrence of gray level $r_k$.

McMaster University

# Histogram Processing

- Problem: an image with gray levels between 0 and 7 is given below. Find the histogram of the image

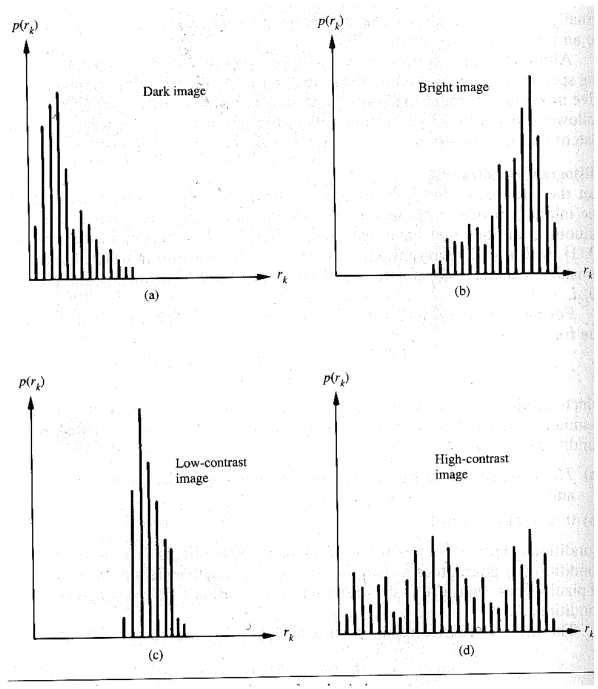
| | | | |
|---|---|---|---|
| 1 | 6 | 2 | 2 |
| 1 | 3 | 3 | 3 |
| 4 | 6 | 4 | 0 |
| 1 | 6 | 4 | 7 |

$$p_r(r_k) = \frac{n_k}{n}$$

McMaster University

# Histogram equalization

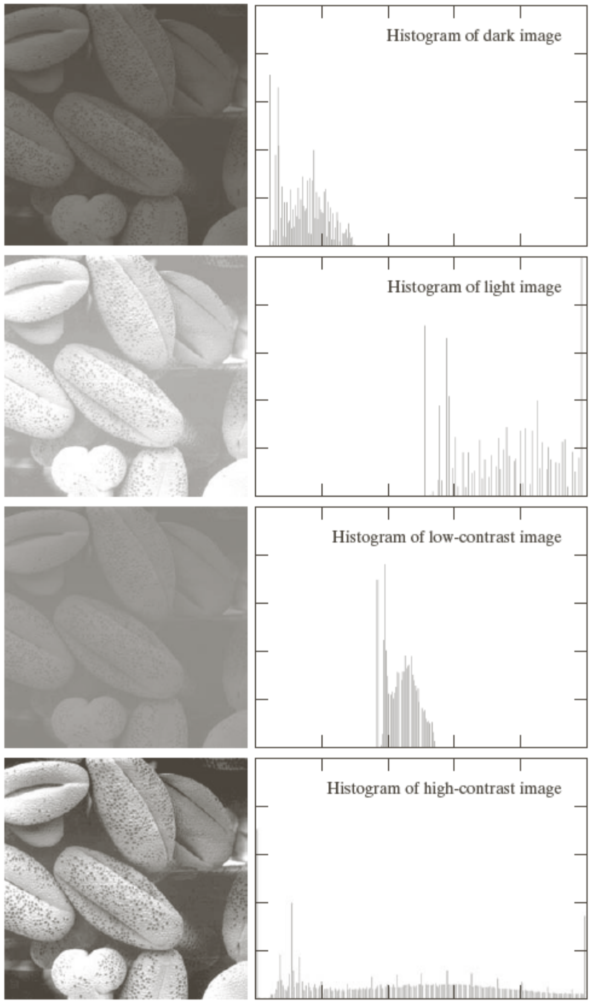
- Histogram provides a global description of the appearance of an image:
  - In a dark image, histogram is centered in the dark side of gray scale
  - In a bright image, the histogram is biased toward the high side of gray levels
- An image whose pixels occupy the entire range of possible gray levels and is uniformly distributed will appear as high-contrast.




McMaster University

# Histogram equalization

- Goal: find a transform s=T(r) such that the transformed image has a flat (equalized) histogram

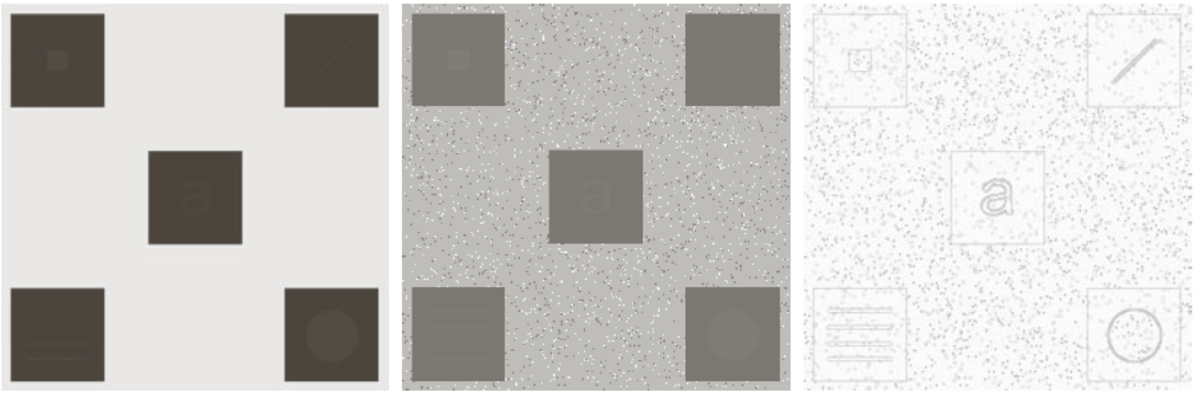



McMaster University

# Local histogram processing

- Global histogram processing: pixels are modified by a transformation function based on the gray-level content of an entire image
- Sometimes we want to enhance the details over a small area
- Solution: transformation should be based on gray-level distribution in the neighborhood of every pixel
- Local histogram processing:
  - At each location the histogram of the points in the neighborhood is computed and a histogram equalization or histogram specification transformation function is obtained
  - The gray level of the pixel centered in the neighborhood is mapped
  - The center of the neighborhood is moved the next pixel and the procedure repeated

# Local histogram processing



a b c

**FIGURE 3.26** (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization applied to (a), using a neighborhood of size $3 \times 3$.

McMaster University

# Local Enhancement

$$m = \sum_{i=0}^{L-1} r_i p(r_i)$$

$$\sigma^2 = \sum_{i=0}^{L-1} (r_i - m)^2 p(r_i)$$

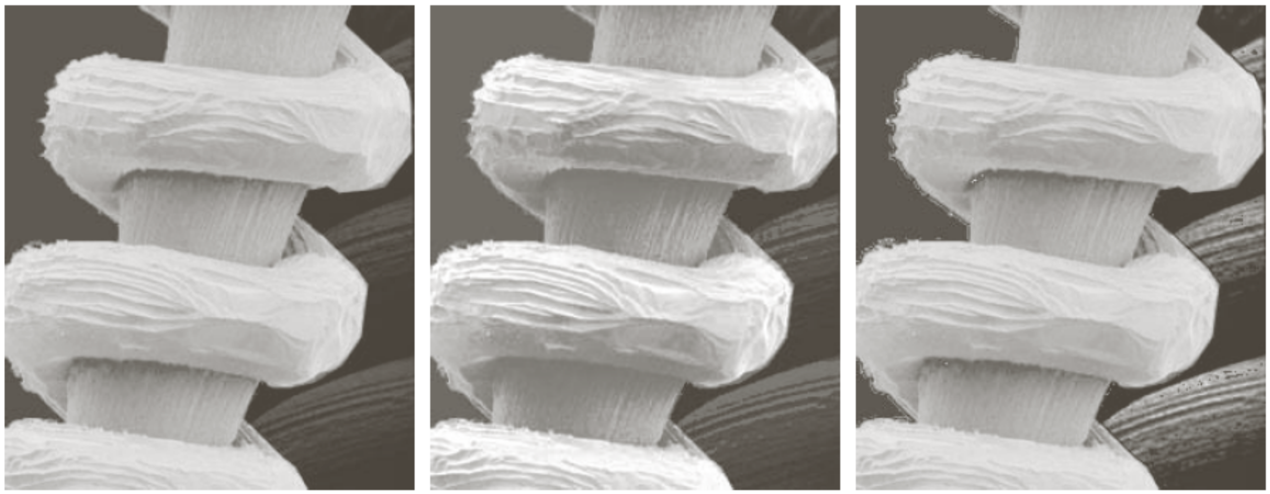
- Mean of gray levels in an image: a measure of darkness, brightness of the image
- Variance of gray levels in an image: a measure of average contrast
- Local mean and variance are used as the basis for making changes that depend on image characteristics in a predefined region about each pixel

$$m_{S_{xy}} = \sum_{(s,t)\in Sxy} r_{s,t} p(r_{s,t})$$

$$\sigma^2{}_{Sxy} = \sum_{(s,t)\in Sxy} (r_{s,t} - m_{Sxy})^2 p(r_{s,t})$$

McMaster University

# Local Enhancement

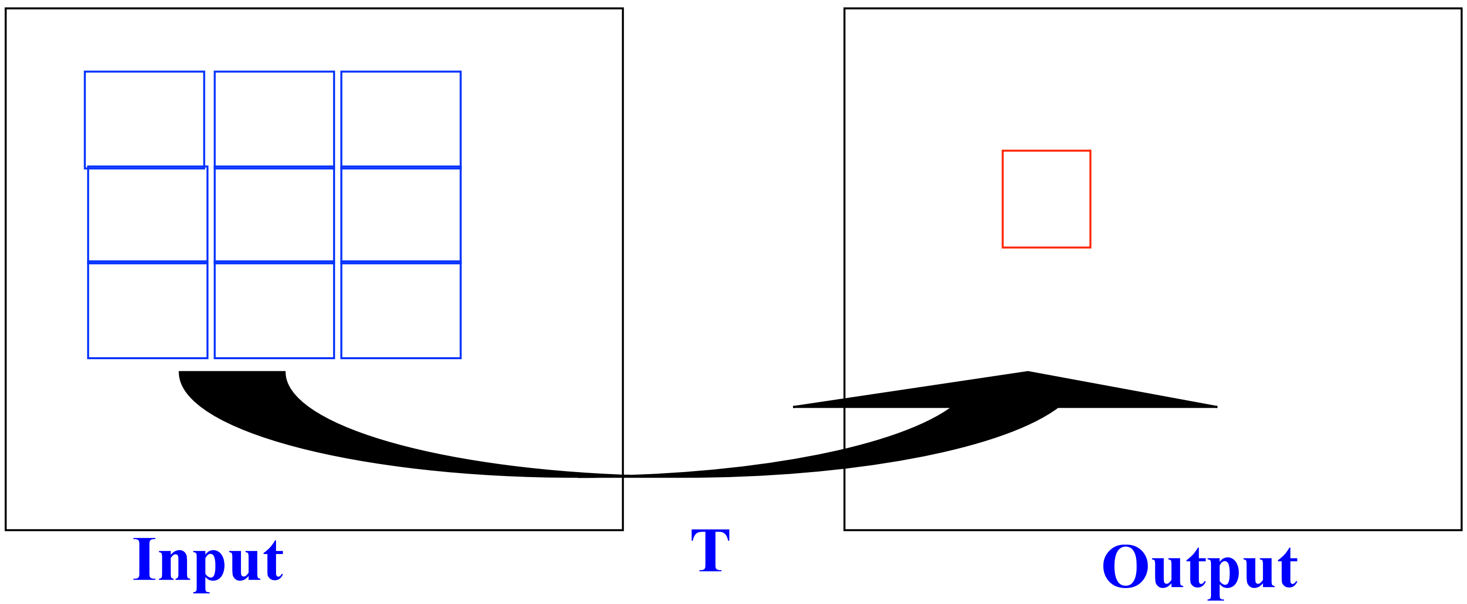


a b c

**FIGURE 3.27** (a) SEM image of a tungsten filament magnified approximately 130×. (b) Result of global histogram equalization. (c) Image enhanced using local histogram statistics. (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

# Spatial domain filtering

- Larger neighborhood around (x,y): usually a square or rectangular subimage area centered at (x,y).
- The center of the subimage is moved pixel by pixel. At each location (x,y) the operator T is applied to find the value of g(x,y).




McMaster University

# Spatial domain filtering

Based on the operator T:

- Spatial filtering
  - Linear filters
    - Average filtering
    - Weighted average filtering
    - High-boost filters
    - Derivative filters
  - Non-linear filters
    - Median filters
    - Order-statistics filters

Smoothing filters ← Average filtering, Weighted average filtering, Median filters

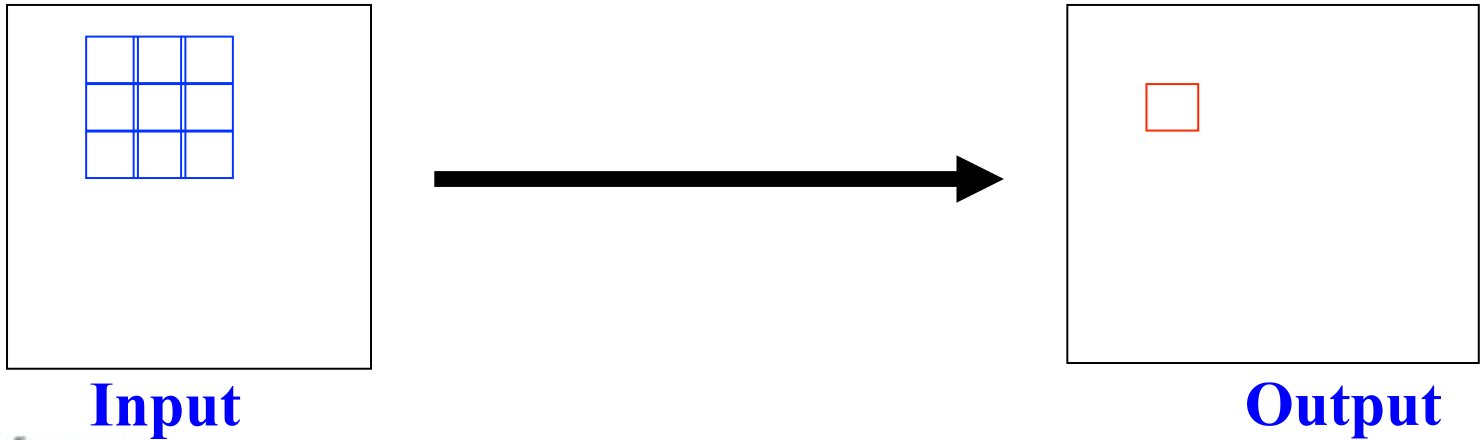
Sharpening filters ← High-boost filters, Derivative filters

Linear filters can be implemented by masks but not non-linear filters

McMaster University

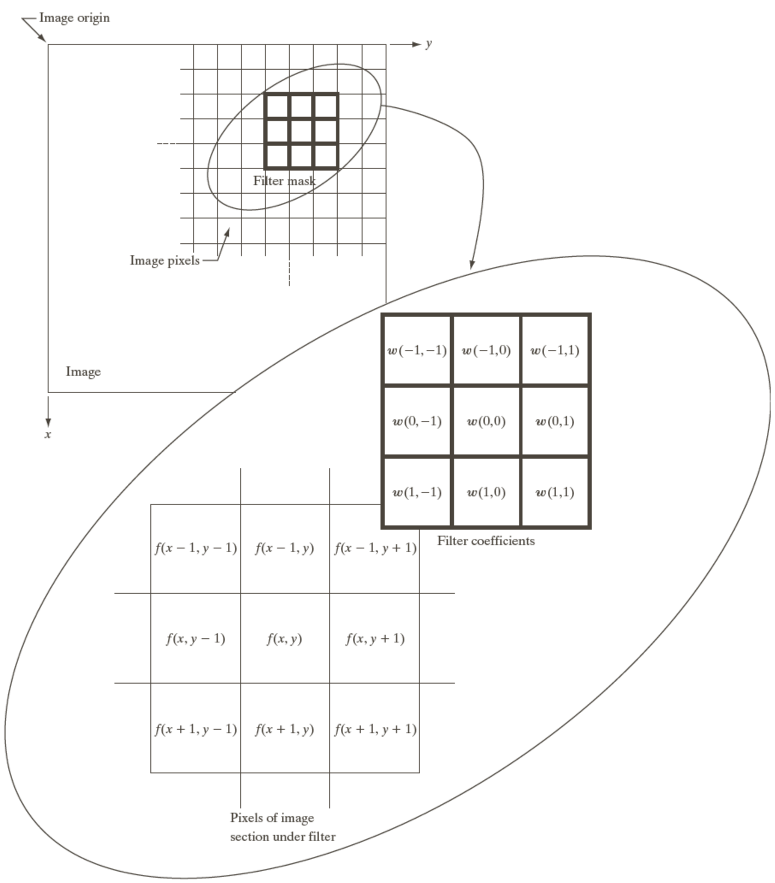
# Spatial domain filtering

- Linear filtering: Result of filtering is a linear combination of the gray-levels in the neighborhood of (x,y)
- Exp: g(x,y)=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)
- One approach to find the processed image in this case is to use a mask (window or filter)
- Mask: a small 2-D array. The values of the elements of the mask are the w's




McMaster University

# Spatial domain filtering




McMaster University

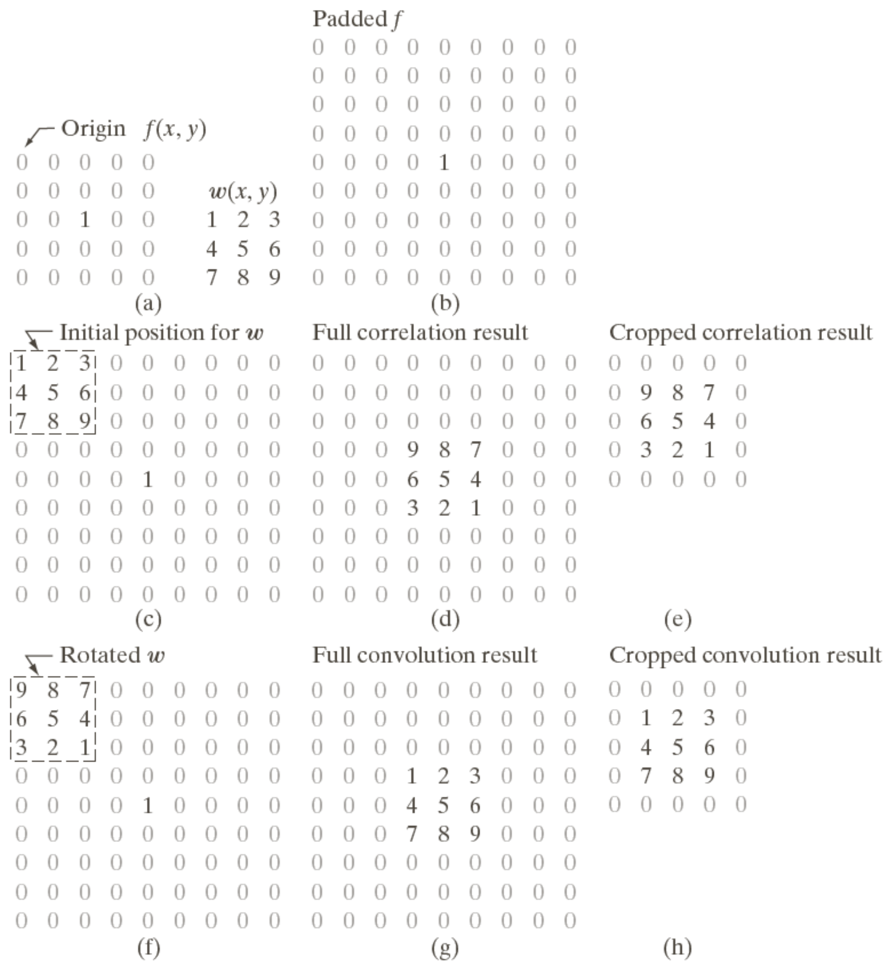
# Correlation and Convolution

```
                 Correlation                                  Convolution

     Origin   f              w                    Origin   f          w rotated 180°
(a)  0 0 0 1 0 0 0 0     1 2 3 2 8                0 0 0 1 0 0 0 0     8 2 3 2 1        (i)

(b)          0 0 0 1 0 0 0 0                              0 0 0 1 0 0 0 0              (j)
     1 2 3 2 8                                    8 2 3 2 1
             Starting position alignment

              Zero padding
(c)  0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0              0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0  (k)
     1 2 3 2 8                                    8 2 3 2 1

(d)  0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0              0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0  (l)
       1 2 3 2 8                                    8 2 3 2 1
       Position after one shift

(e)  0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0              0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0  (m)
             1 2 3 2 8                                    8 2 3 2 1
             Position after four shifts

(f)  0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0              0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0  (n)
                           1 2 3 2 8                                    8 2 3 2 1
              Final position

             Full correlation result                     Full convolution result
(g)      0 0 0 8 2 3 2 1 0 0 0 0                      0 0 0 1 2 3 2 8 0 0 0 0          (o)

             Cropped correlation result                  Cropped convolution result
(h)          0 8 2 3 2 1 0 0                              0 1 2 3 2 8 0 0              (p)
```

**FIGURE 3.29** Illustration of 1-D correlation and convolution of a filter with a discrete unit impulse. Note that correlation and convolution are functions of *displacement*.

McMaster University

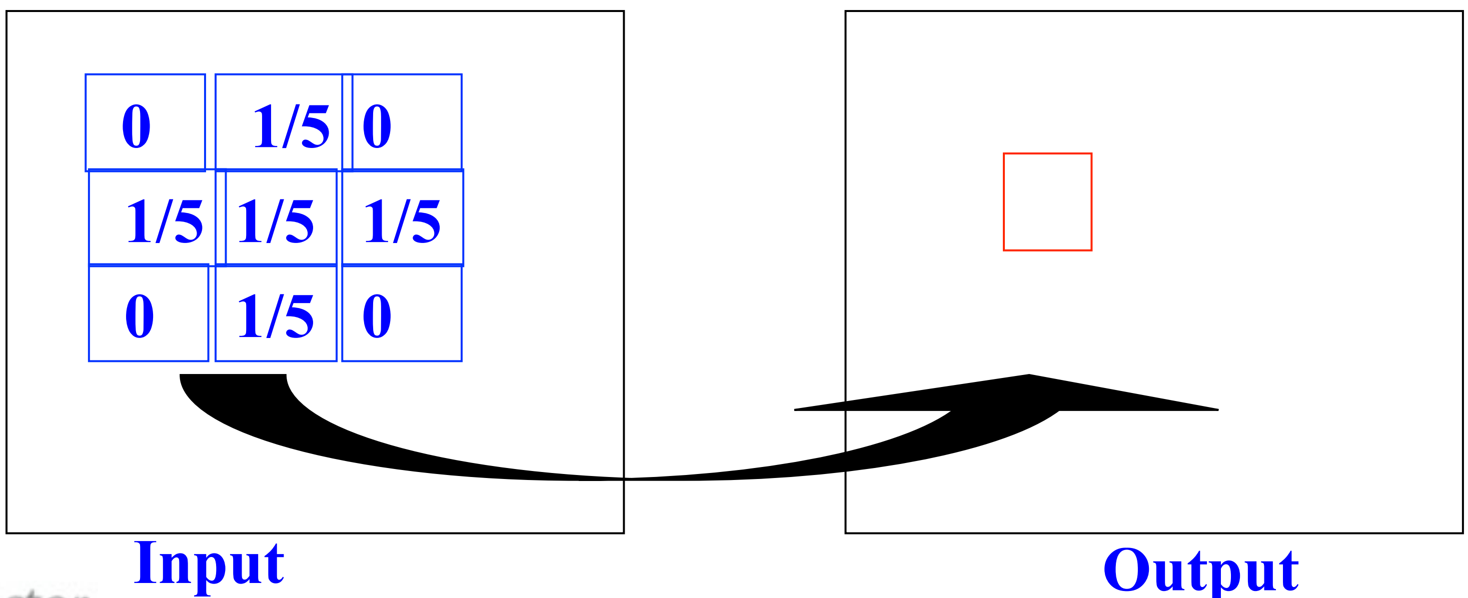
# Correlation and Convolution




McMaster University

# Spatial domain filtering

- Example: each pixel in the processed image is the average of the gray levels of pixels to the right, left, top, bottom and itself.
- This spatial domain processing can be implemented by the following mask:




McMaster University

# Smoothing Filters

- Smoothing filters: used for blurring and noise reduction.
- Blurring: used in preprocessing steps such as removing small details from an image before object extraction, bridging small gaps in lines and curves
- Smoothing:
  - Averaging (weighted averaging)
  - Median filtering

# Smoothing Filters

- Averaging: By replacing the value of every pixel in an image by the average of the gray levels in the neighborhood, we get an image with reduced sharp transitions.
- Because random noise typically consists of sharp transitions in gray-levels, averaging can be used in noise reduction
- Edges in an image also have sharp transitions
- Average filtering has the side effect of blurring edges

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

# Smoothing Filters

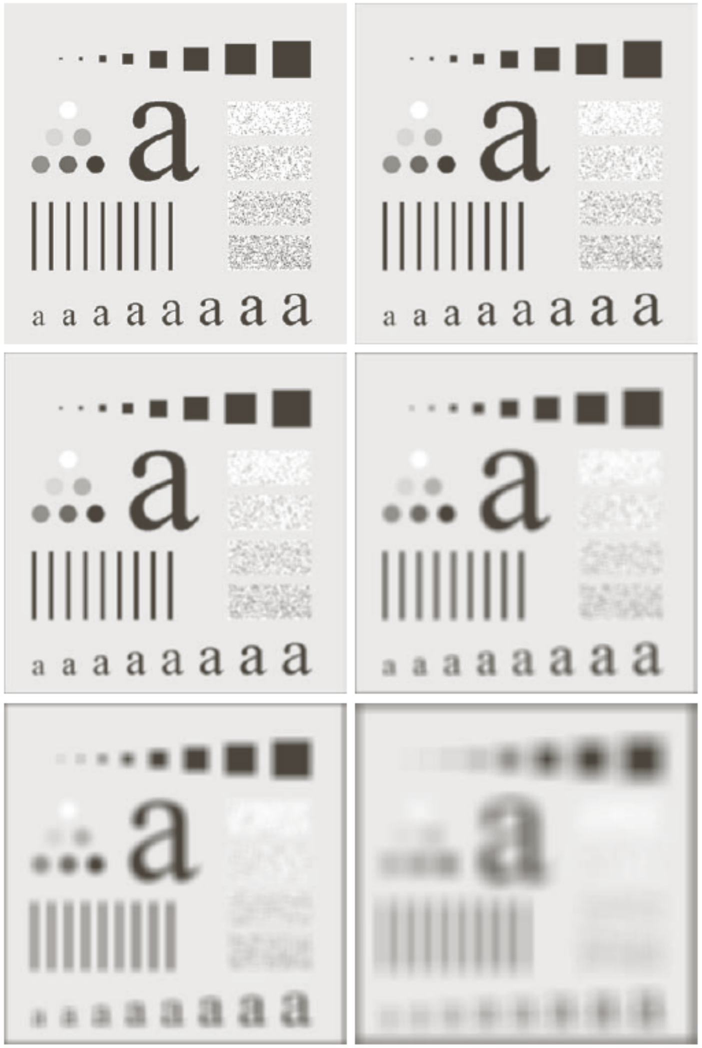


**FIGURE 3.33** (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $m = 3, 5, 9, 15$, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. 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 intensity 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.

a b
c d
e f

McMaster University

# Median Filtering

- Median filter belongs to a group of filters called order-statistic filters
- These filters are non-linear
- The output of the filter is obtained by ordering the values of pixels in the neighborhood and performing some operation on the ranked data (e.g., min, max, median)

McMaster University

# Median Filtering

- Median filters are particularly effective in the presence of impulse noise (salt and pepper noise)
- Unlike average filtering, median filtering does not blur edges and other sharp details.
- Example: Consider the example of filtering the sequence below using a 3-pt median filter:

  16 14 15 12 2 13 15 52 51 50 49
- The output of the median filter is:

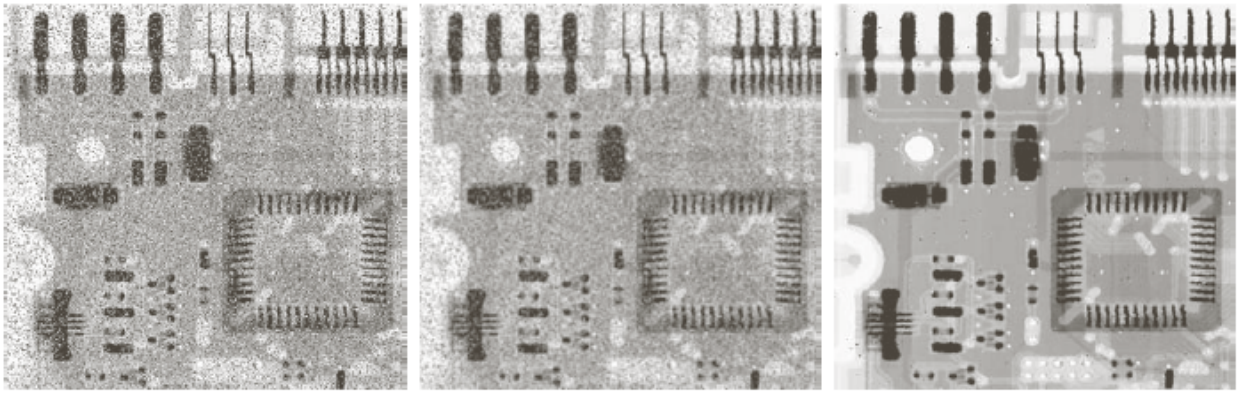
  15 14 12 12 13 15 51 51 50
- Note that the impulse noise is removed while the edge is preserved.

McMaster University

# Median filtering

- Principal function of median filtering is to force points with distinct intensities to be more like their neighbors, eliminating intensity spikes that appear isolated in the neighborhood
- Advantages:
  - Removes impulsive noise
  - Preserves edges
- Disadvantages:
  - performance poor when # of noise pixels in the window is greater than 1/2 # in the window
  - performs poorly with Gaussian noise

# Median Filtering



a b c

**FIGURE 3.35** (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a $3 \times 3$ averaging mask. (c) Noise reduction with a $3 \times 3$ median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

# Sharpening filters

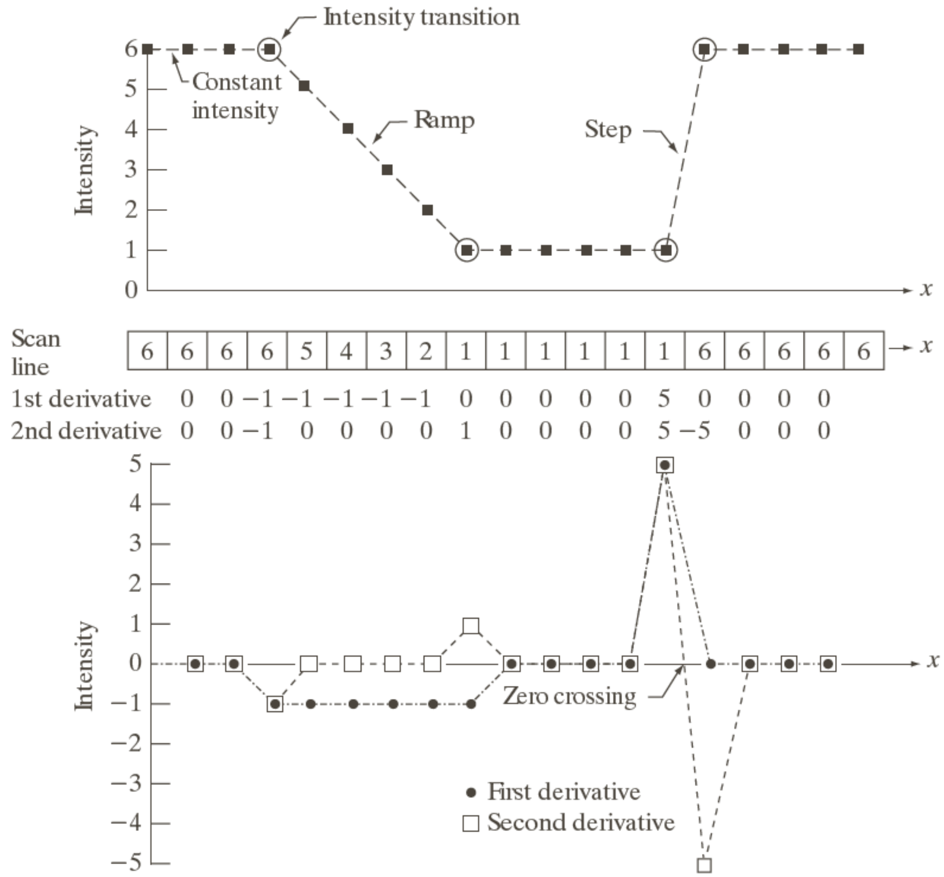
- Objective: highlight fine detail in an image or to enhance detail that has been blurred
- Sharpening can be achieved by spatial differentiation
- Since images are digital we should define digital differentiation operators.
- First and second order derivatives are commonly used for sharpening.
- We consider 1-D case first and then expand the results to images.

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

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

McMaster University

# Sharpening filters



# Sharpening filters

- Comparing first and second order derivatives:

1. First-order derivatives generally produce thicker edges in an image
2. Second order derivatives have a stronger response to fine details such as thin lines and isolated points
3. Second order derivates produce a double response at step changes in gray level

- For image enhancement (sharpening) second order derivative has more applications because of the ability to enhance fine details

# Laplacian

- How to obtain 2-D second order derivative for image enhancement and find a mask corresponding to it?
- We would like our filter to be isotropic: response of the filter is independent of the direction of the discontinuity in the image
- Simplest isotropic second order derivative is the Laplacian:

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

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

# Laplacian

(a)

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | −4 | 1 |
| 0 | 1 | 0 |

(b)

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | −8 | 1 |
| 1 | 1 | 1 |

(c)

| | | |
|---|---|---|
| 0 | −1 | 0 |
| −1 | 4 | −1 |
| 0 | −1 | 0 |

(d)

| | | |
|---|---|---|
| −1 | −1 | −1 |
| −1 | 8 | −1 |
| −1 | −1 | −1 |

a b
c d

**FIGURE 3.37**
(a) Filter mask used to implement Eq. (3.6-6). (b) Mask used to implement an extension of this equation that includes the diagonal terms. (c) and (d) Two other implementations of the Laplacian found frequently in practice.

McMaster University

# Laplacian

- Laplacian is a derivative operator: its use highlights discontinuities in an image and de-emphasizes regions with slowly varying gray levels.
- All the background are removed
- Background can be recovered simply by adding original and Laplacian images.

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) & \text{center coefficient negative} \\ f(x,y) + \nabla^2 f(x,y) & \text{center coefficient positive} \end{cases}$$

McMaster University

# Laplacian

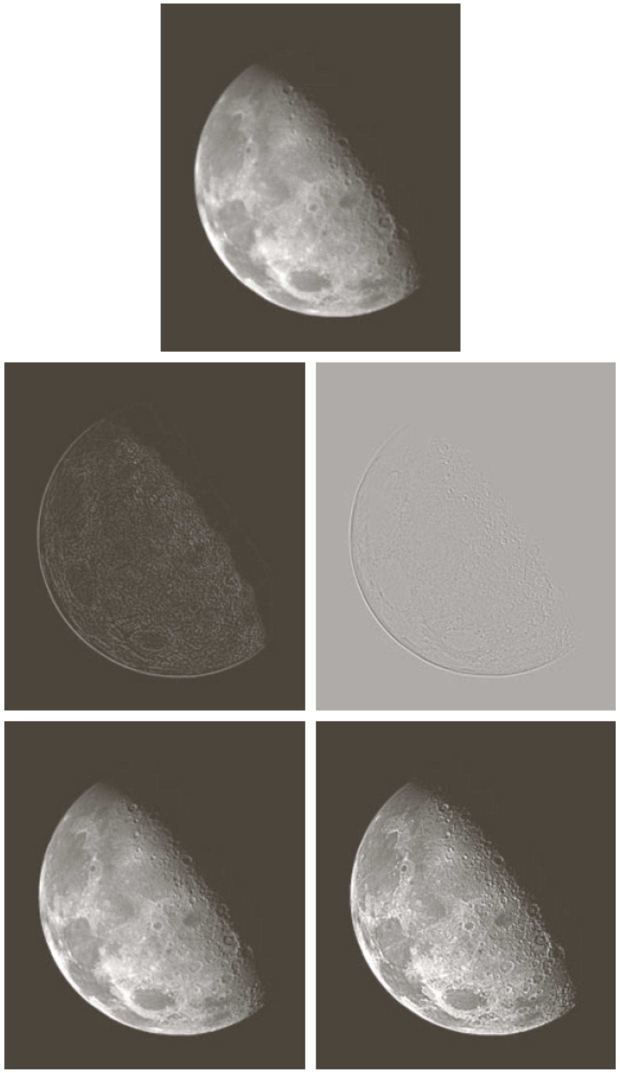


FIGURE 3.38
(a) Blurred image of the North Pole of the moon. (b) Laplacian without scaling. (c) Laplacian with scaling. (d) Image sharpened using the mask in Fig. 3.37(a). (e) Result of using the mask in Fig. 3.37(b). (Original image courtesy of NASA.)

McMaster University

# Laplacian

- Instead of computing the Laplacian filtered image and then subtracting it from the original image we can combine the two operations.

$$G(x,y) = f(x,y) - [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)] + 4f(x,y)$$

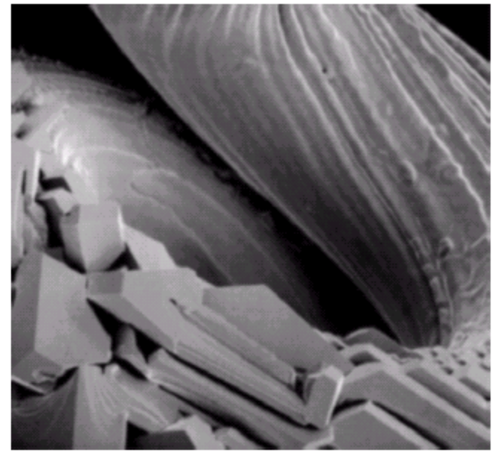
$$= 5f(x,y) - [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)]$$

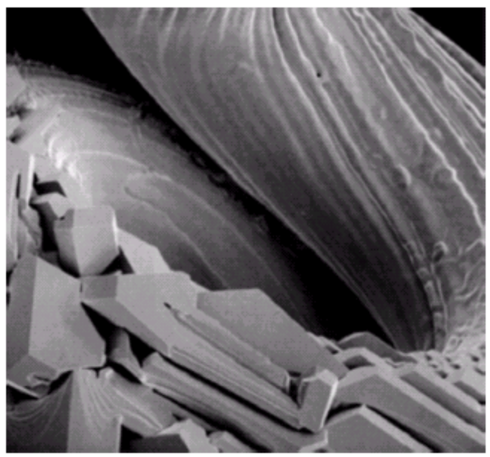
McMaster University

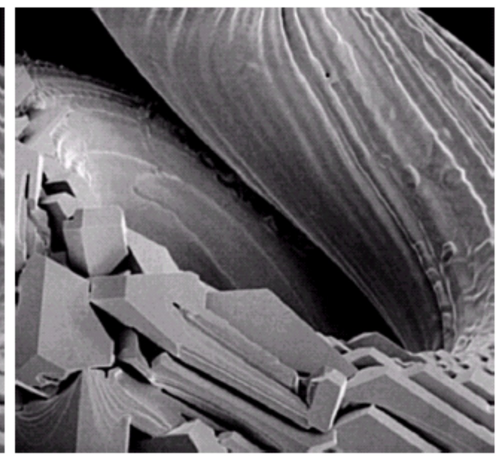
# Laplacian

| | | |
|---|---|---|
| 0 | −1 | 0 |
| −1 | 5 | −1 |
| 0 | −1 | 0 |

| | | |
|---|---|---|
| −1 | −1 | −1 |
| −1 | 9 | −1 |
| −1 | −1 | −1 |







a b c
d e

**FIGURE 3.41** (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.)

McMaster University

# Unsharp masking & High-boost filtering

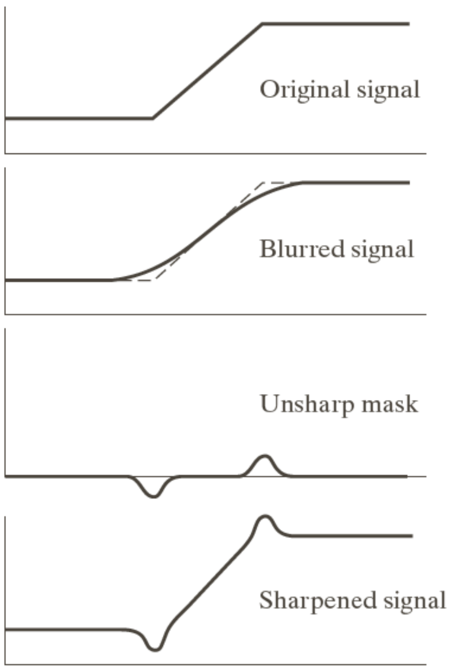
- A method for sharpening an image is to subtract a blurred version of the image from the image itself to obtain a mask.
- Add the mask back to the original image
- This method is called unsharp masking

$$g_{mask}(x,y) = f(x,y) - \bar{f}(x,y)$$
$$g(x,y) = f(x,y) + k.g_{mask}(x,y)$$

- When k>1, the process is called high-boost filtering.

McMaster University

# Unsharp masking & High-boost filtering



# Unsharp masking & High-boost filtering


McMaster University

# Derivative filters

- Image is a 2-D signal: when we are talking about derivative we should specify the direction.
- First order derivates are implemented using magnitude of the gradient
- Gradient:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

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

McMaster University

# Derivative filters

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

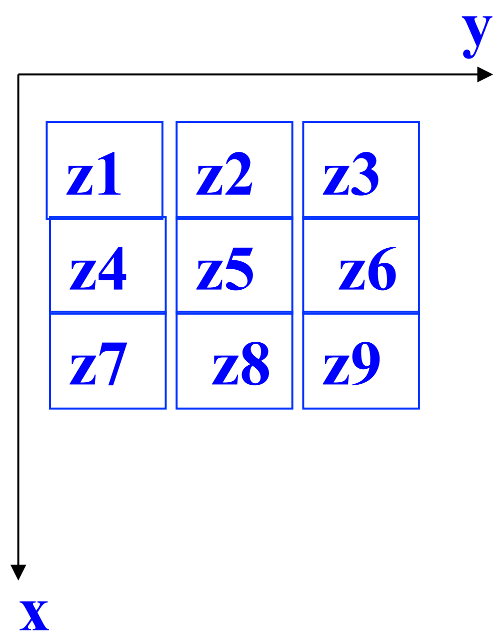
$$|\nabla f| \approx |\nabla_x f| + |\nabla_y f|$$

$$|\nabla_x f| \approx |z_8 - z_5|$$

$$|\nabla_y f| \approx |z_6 - z_5|$$

| | |
|---|---|
| -1 | 0 |
| 1 | 0 |

| | |
|---|---|
| -1 | 1 |
| 0 | 0 |

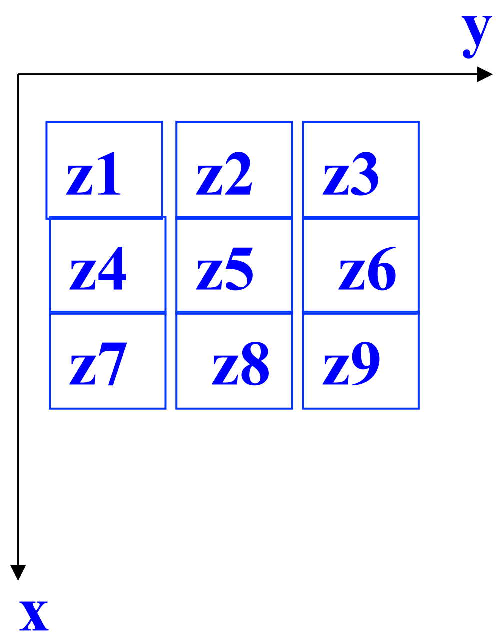


# Derivative filters

$$|\nabla f| \approx |\nabla_x f| + |\nabla_y f|$$

$$|\nabla_x f| \approx |z_7 + z_8 + z_9 - (z_1 + z_2 + z_3)|$$

$$|\nabla_y f| \approx |z_3 + z_6 + z_9 - (z_1 + z_4 + z_7)|$$




| | | |
|---|---|---|
| -1 | -1 | -1 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |

| | | |
|---|---|---|
| -1 | 0 | 1 |
| -1 | 0 | 1 |
| -1 | 0 | 1 |

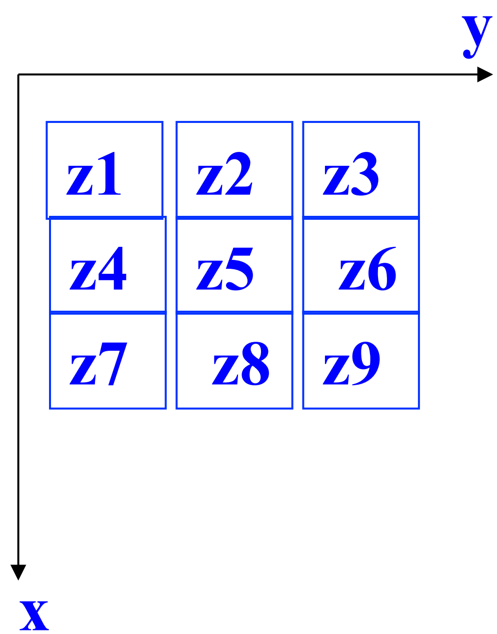
McMaster University

# Derivative filters

$$|\nabla f| \approx |\nabla_x f| + |\nabla_y f|$$

$$|\nabla_x f| \approx |z_7 + 2z_8 + z_9 - (z_1 + 2z_2 + z_3)|$$

$$|\nabla_y f| \approx |z_3 + 2z_6 + z_9 - (z_1 + 2z_4 + z_7)|$$




| | | |
|---|---|---|
| -1 | -2 | -1 |
| 0 | 0 | 0 |
| 1 | 2 | 1 |

| | | |
|---|---|---|
| -1 | 0 | 1 |
| -2 | 0 | 2 |
| -1 | 0 | 1 |

# Derivative filters

| | | |
|---|---|---|
| $z_1$ | $z_2$ | $z_3$ |
| $z_4$ | $z_5$ | $z_6$ |
| $z_7$ | $z_8$ | $z_9$ |

| | |
|---|---|
| −1 | 0 |
| 0 | 1 |

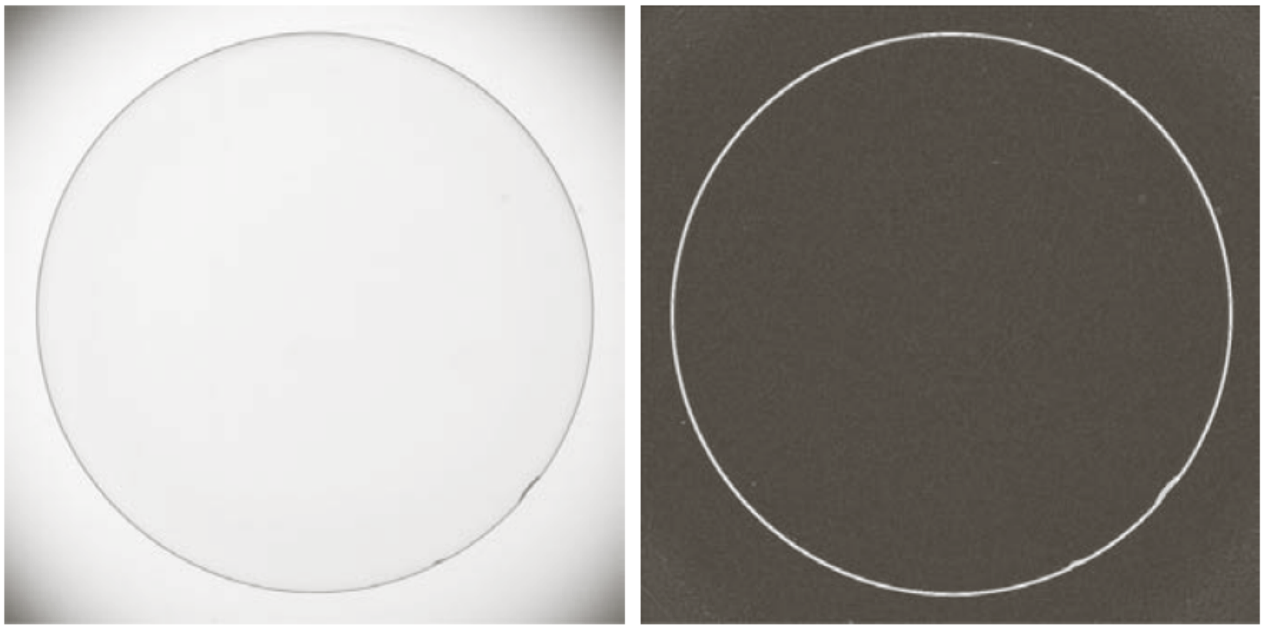
| | |
|---|---|
| 0 | −1 |
| 1 | 0 |

| | | |
|---|---|---|
| −1 | −2 | −1 |
| 0 | 0 | 0 |
| 1 | 2 | 1 |

| | | |
|---|---|---|
| −1 | 0 | 1 |
| −2 | 0 | 2 |
| −1 | 0 | 1 |

McMaster University

# Derivative filters

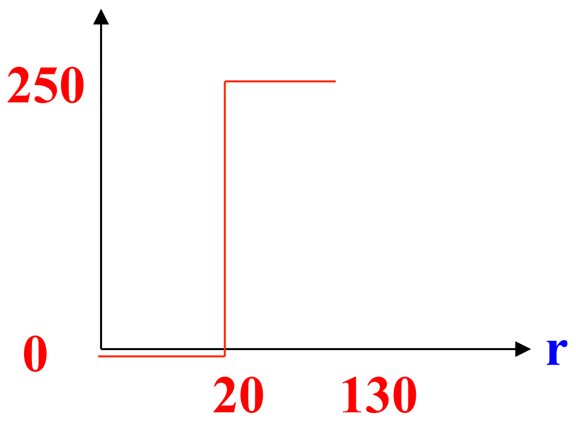


# Problem

- A 4x4 image is given as follow.

1) The image is transformed using the point transform shown. Find the pixel values of the output image.

| | | | |
|---|---|---|---|
| **17** | **64** | **128** | **128** |
| **15** | **63** | **132** | **133** |
| **11** | **60** | **142** | **140** |
| **11** | **60** | **142** | **138** |

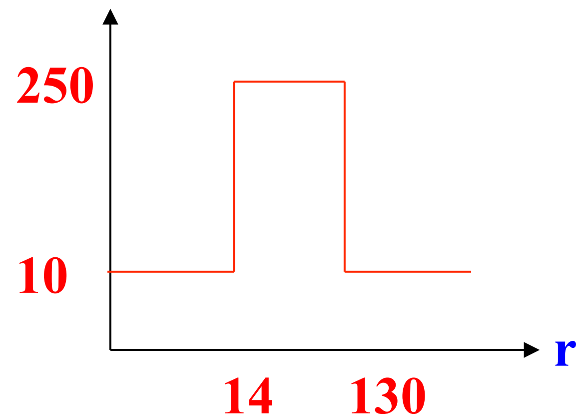


# Problem

- A 4x4 image is given as follow.

1) The image is transformed using the point transform shown. Find the pixel values of the output image.
2) What is the 7-th bit plane of this image

| | | | |
|---|---|---|---|
| **17** | **64** | **128** | **128** |
| **15** | **63** | **132** | **133** |
| **11** | **60** | **142** | **140** |
| **11** | **60** | **142** | **138** |




McMaster University

# Problem

- A 4x4 image is given as follow.

1) Suppose that we want to process this image by replacing each pixel by the difference between the pixels to the top and bottom. Give a 3x1 mask that performs this.
2) Apply the mask to the second row of the image

| | | | |
|---|---|---|---|
| **17** | **64** | **128** | **128** |
| **15** | **63** | **132** | **133** |
| **11** | **60** | **142** | **140** |
| **11** | **60** | **142** | **138** |

McMaster University