Image Enhancement & Spatial Filtering

Computer Vision

Aziz M. Qaroush

Birzeit University

Outline

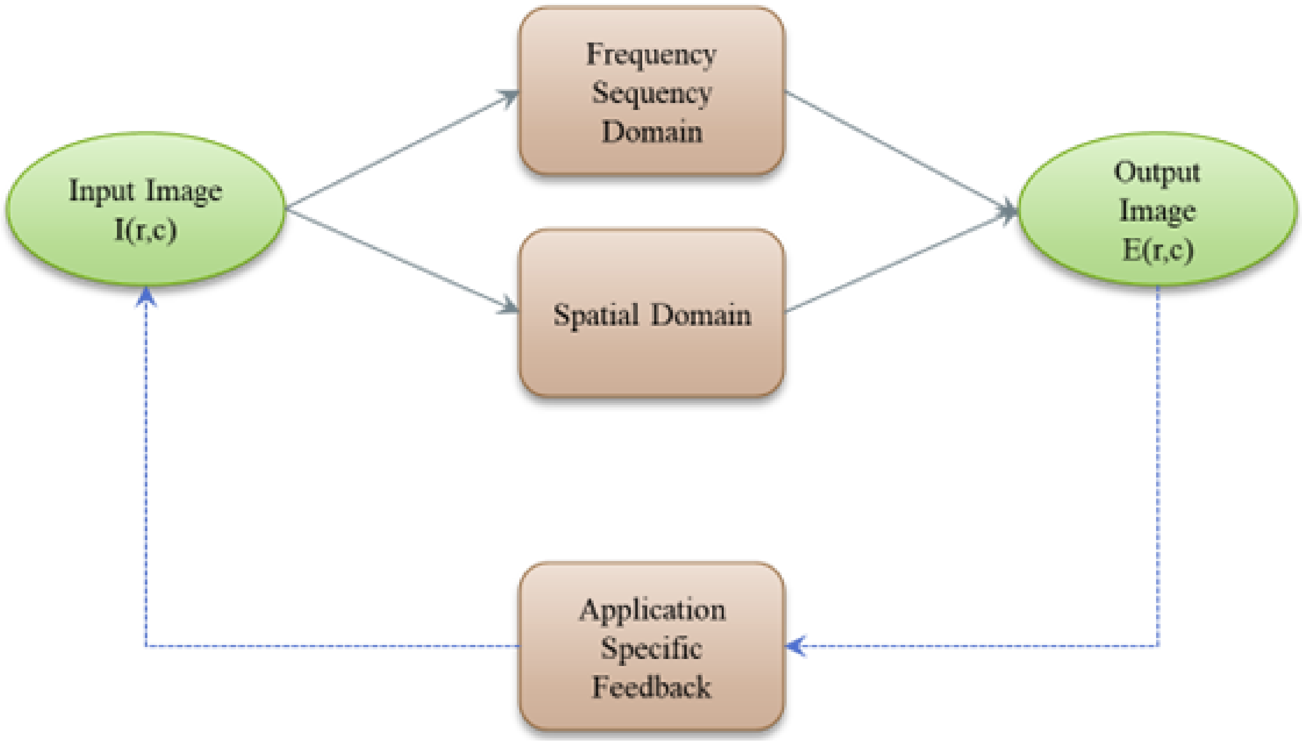
- **What is Image Enhancement?**
- Contrast Enhancement
  - Intensity Transformation Functions
  - Histogram Processing
- Spatial Filtering
  - Basic Concepts
  - Smoothing Filters
  - Sharping Filters
  - Nonlinear Filters
- Image Quality Assessment
  - Subjective image quality assessment
  - Objective image quality assessment

Image enhancement

- Process an image so that the result will be more suitable than the original image for a specific application.
- Enhancement can be done in
  - Spatial Domain (image plane):
    - Techniques are based on direct manipulation of pixels in an image
  - Frequency Domain:
    - Techniques are based on modifying the Fourier transform of an image
  - There are some enhancement techniques based on various combinations of methods from these two categories.
- Common reasons for enhancement include
  - Improving visual quality,
  - Improving recognition accuracy.

Image enhancement

- The suitableness is up to each application.
  - A method which is quite useful for enhancing an image may not necessarily be the best approach for enhancing another image



What is Image Enhancement? (cont…)

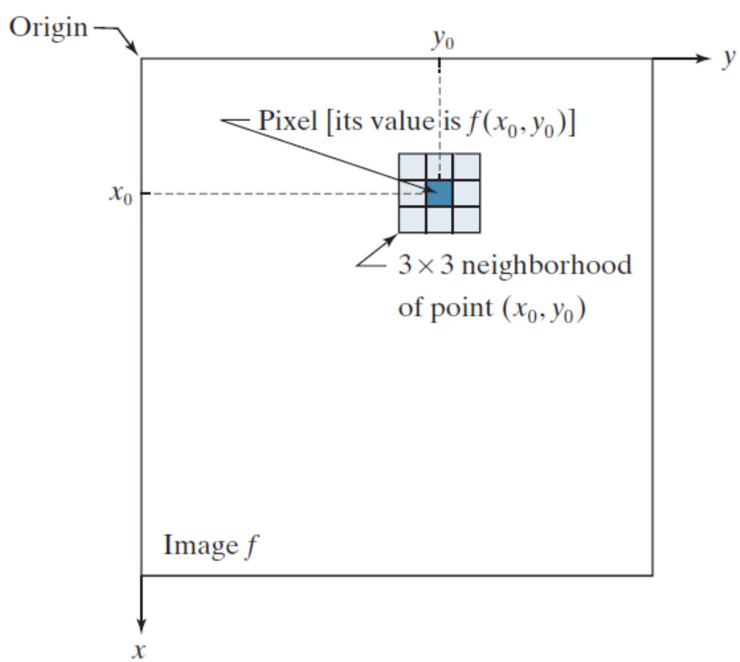
- Spatial domain of the image is the set of pixels composing the image
- Enhancement in the spatial domain involves direct operation on the pixel intensities
- This can be expressed mathematically as

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

  - f(x,y) is the input image
  - g(x,y) is the output image
  - T[ ] is an operator defined over some neighborhood of (x,y)
- **Important**
  - Keep in mind that g(x,y) may take any value from the set of available gray levels only. Thus, when mapping we should assign the mapped value to the closest level

What is Image Enhancement? (cont…)

- Defining the neighborhood around (x,y)
  - Use a square/rectangular subimage that is centered at (x,y)
- Operation
  - Move the center of the subimage from pixel to pixel and apply the operator T at each location (x,y) to compute the output g(x,y)



Point Processing

- The simplest form of the operator T is when the neighborhood size is 1x1 pixels.
  - Accordingly, g(x,y) is only dependent on the value of f at (x,y)
- In this case, T is called the gray-level or intensity transformation function that can be represented as

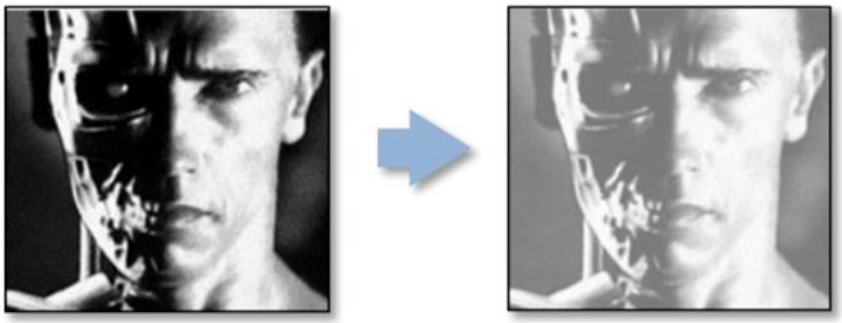
$$s = T(r)$$

  - s is a variable denoting g(x,y)
  - r is a variable denoting f(x,y)

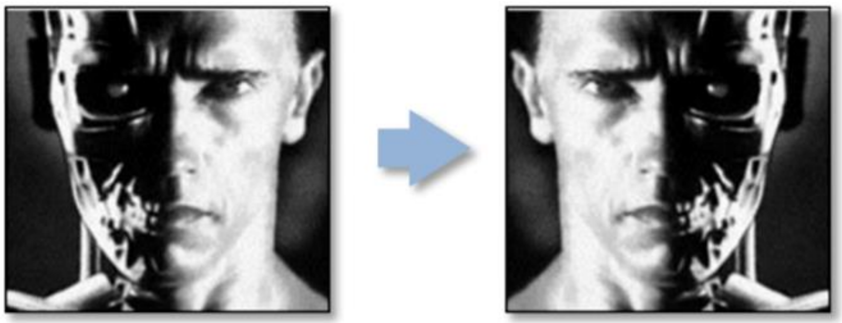
- This is kind of processing is referred as point processing

Point Processing

- Point operation deals with pixel intensity values individually
- Enhancement at any point depends only on the image value at that point.
- The transformed output pixel value does not depend on any of the neighboring pixel value of the input image
- Point Operation Examples:
  - Image Negative.
  - Contrast Stretching.
  - Thresholding.



$g(x,y) = f(x,y) + 20$



$g(x,y) = f(-x,y)$

Examples of point processing

original

x

darken

$x - 128$

lower contrast

$\frac{x}{2}$

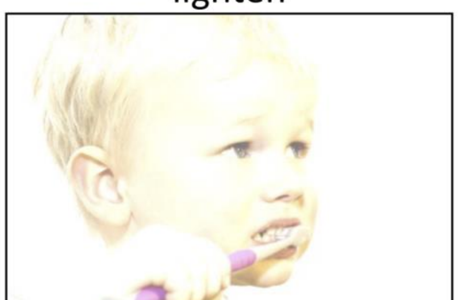
non-linear lower contrast

$\left(\frac{x}{255}\right)^{1/3} \times 255$

invert

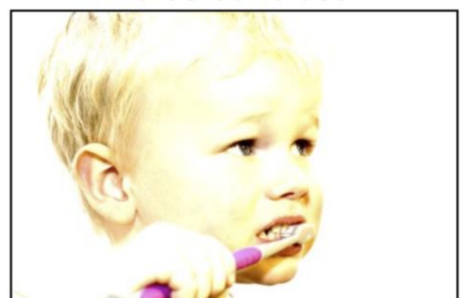
$255 - x$

lighten



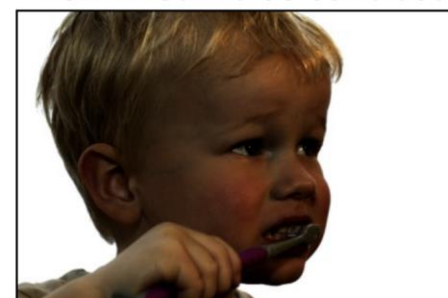
$x + 128$

raise contrast



$x \times 2$

non-linear raise contrast



$\left(\frac{x}{255}\right)^{2} \times 255$

Outline

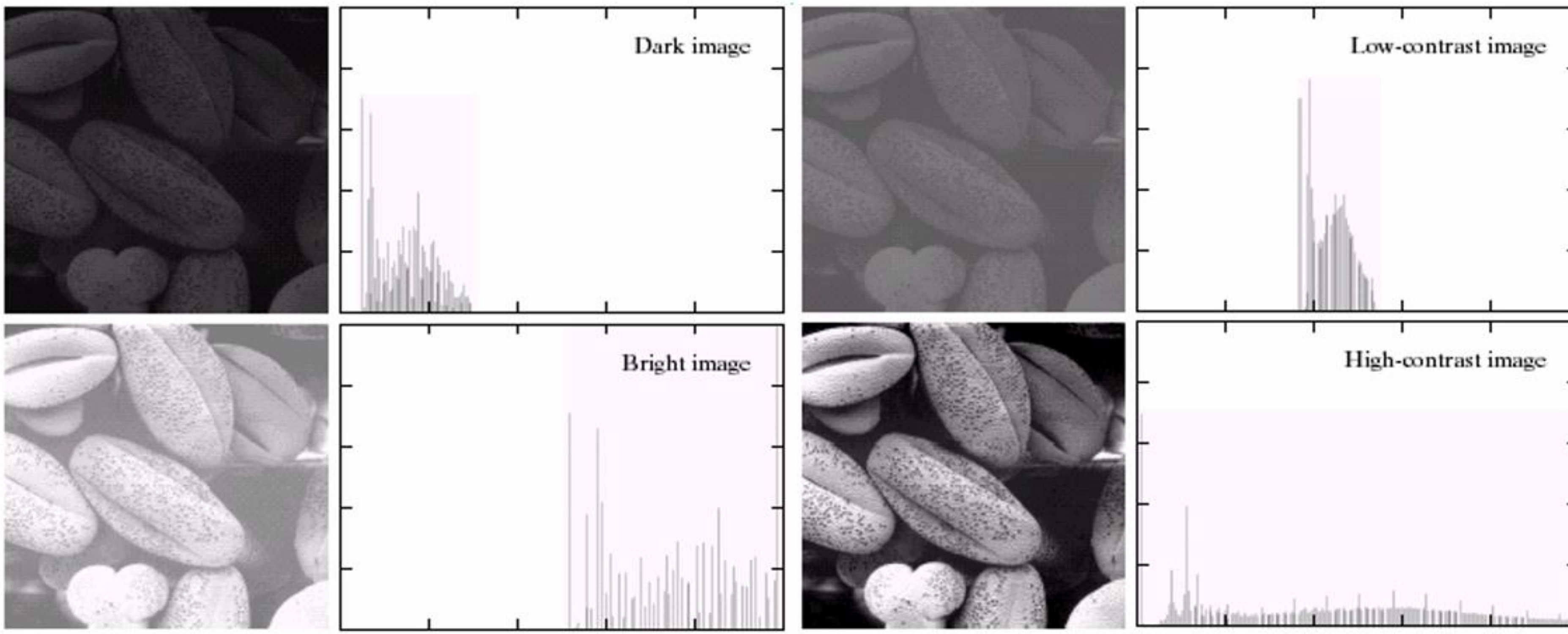
- What is Image Enhancement?
- Contrast Enhancement
  - **Intensity Transformation Functions**
  - Histogram Processing
- Spatial Filtering
  - Basic Concepts
  - Smoothing Filters
  - Sharping Filters
  - Nonlinear Filters
- Image Quality Assessment
  - Subjective image quality assessment
  - Objective image quality assessment

Image enhancement: Contrast Enhancement

- Contrast is an important factor in any subjective evaluation of image quality.
- Contrast is created by the difference in luminance reflected from two adjacent surfaces.
  - In other words, contrast is the difference in visual properties that makes an object distinguishable from other objects and the background.
- In visual perception, contrast is determined by the difference in the colour and brightness of the object with other objects.
- Contrast enhancements are typically performed as a contrast stretch followed by a tonal enhancement, although these could both be performed in one step.

Histogram Processing

- For enhancement, histograms can be used to infer the type of image quality: dark, bright, low or high contrast
- Intuitively, we expect that an image whose pixels tend to occupy the entire range of possible gray levels, tend to be distributed uniformly will have a high contrast and show a great deal of gray level detail.



Contrast Enhancement - Gray Level Transformations

- **Contrast Enhancement done by pixel value mapping called Gray Level Transformations**
- Mapping can be performed by mathematical substitution or lookup tables
- Gray-level transformation function

  S=T(r)

  where r is the gray level of f(x,y) and s is the gray level of g(x,y) at any point (x,y)
- Some common functions are
  - Linear (negative/identity)
  - Logarithmic (log/inverse log)
  - Power law (nth power/nth root)
  - Piecewise-Linear Transformations

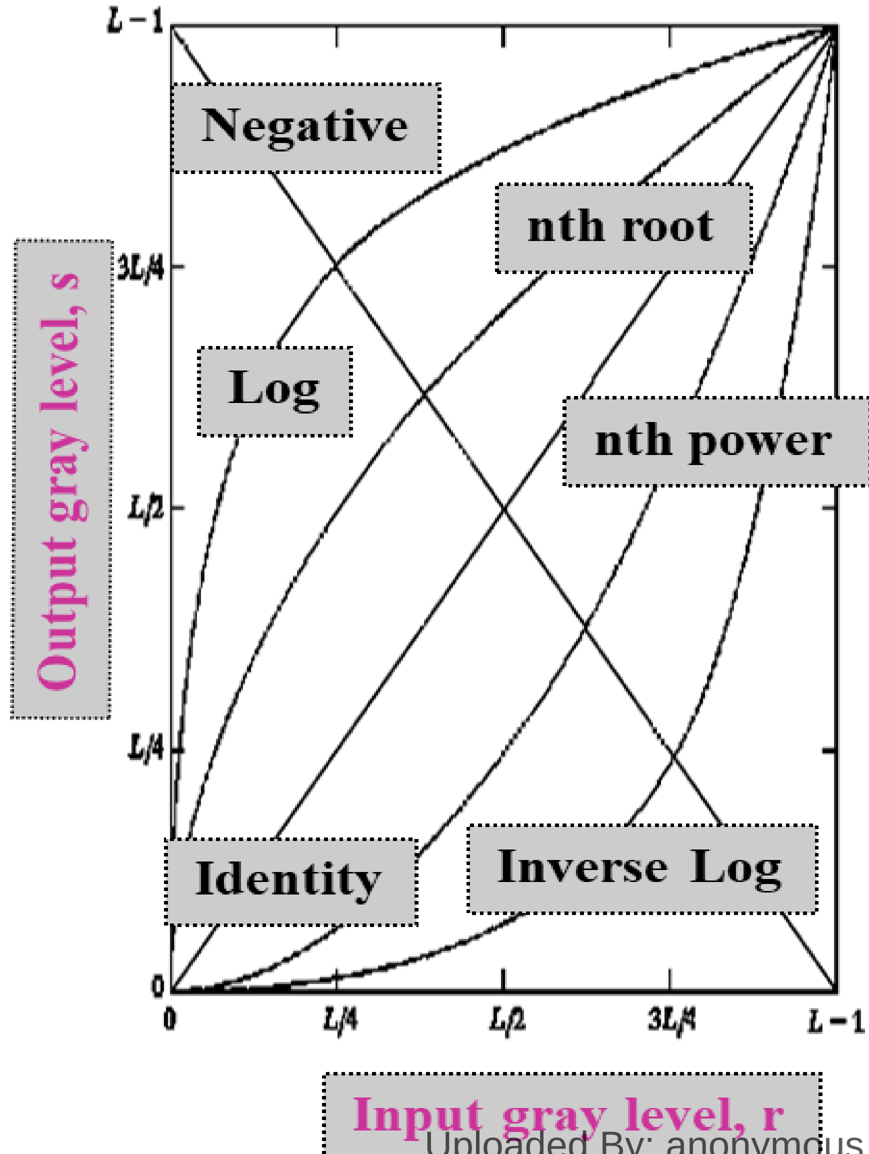


Image Negatives

- Reversing the intensity levels of an image.
- Image negative is produced by subtracting each pixel from the maximum intensity value.
- An image with gray level in the range [0, L-1] where $L = 2^n$; n = 1, 2… Negative transformation :

  $s = (L - 1) - r$

- This type of processing is used, for example, in enhancing white or gray detail embedded in dark regions of an image, especially when the black areas are dominant in size.

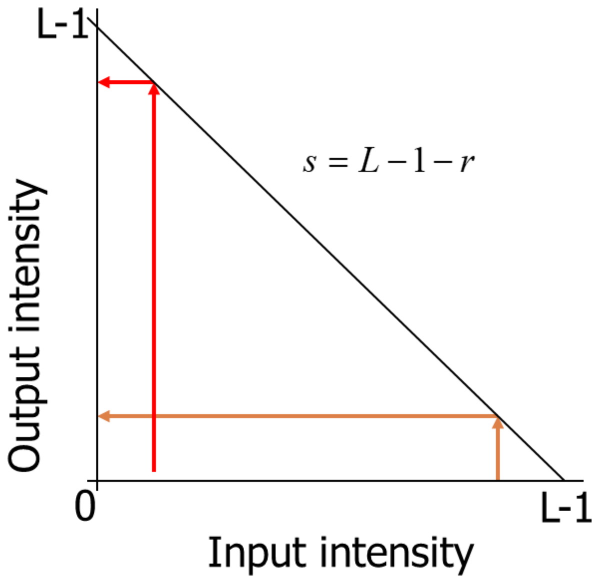
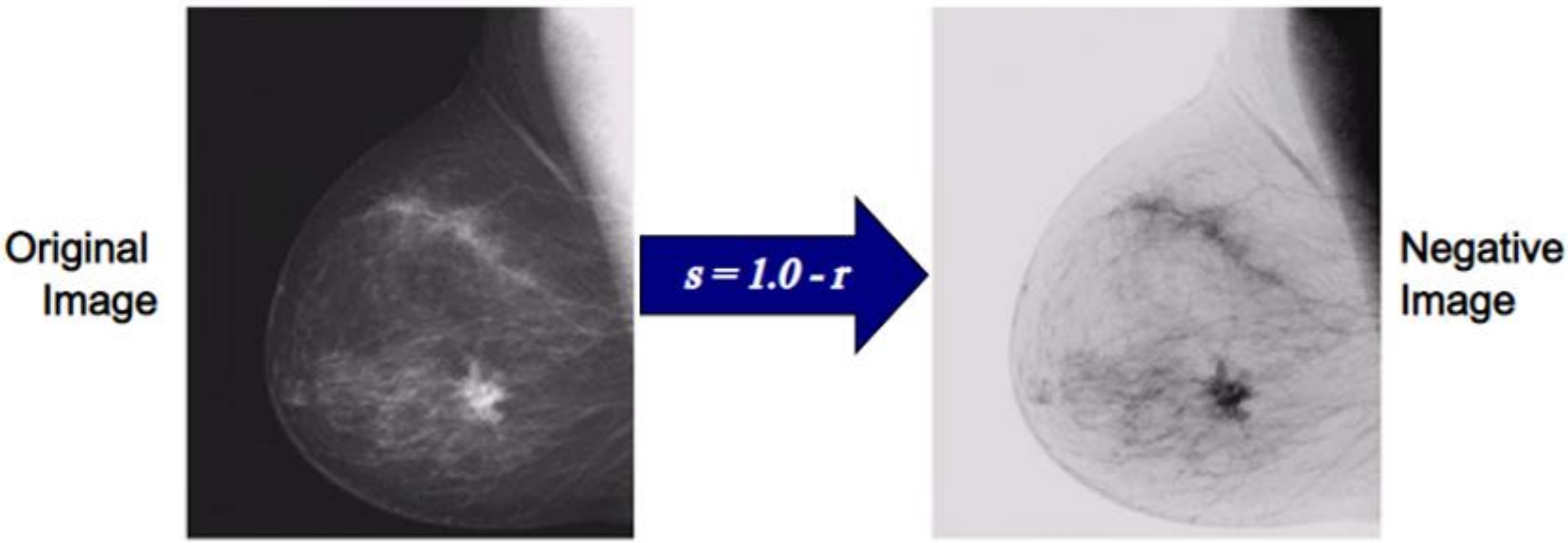
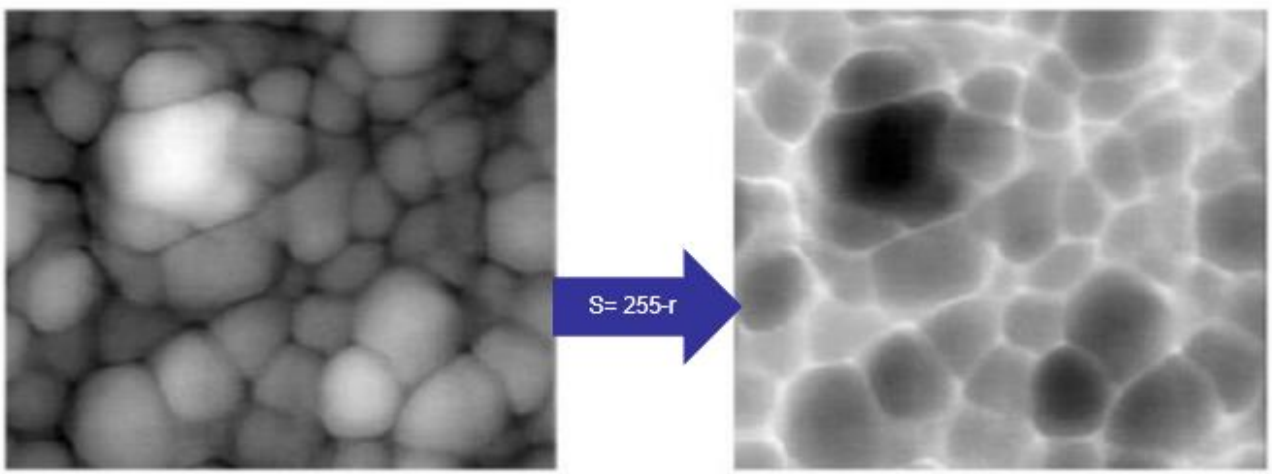


Image Negatives - Example





Log and inverse Log Transformations

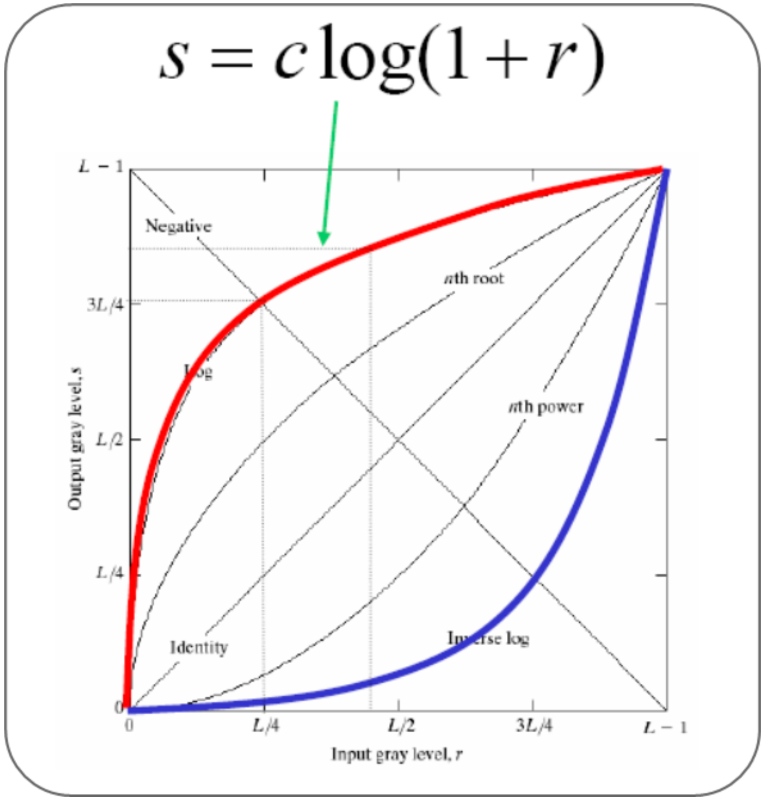
- The general form of the log transformation

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

  - b is the base
- Maps narrow range of low intensity levels to wider range and wide range of high intensity levels to narrower range
- **Usually used to expand the values of dark pixels and compress the higher level values**
- The general form of the inverse log

$$s = b^{cr} - 1$$

- Its operation is the opposite of the log transformation



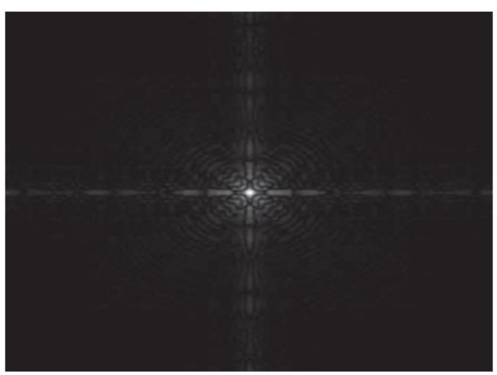
Log Transformation Example

- It is very important in mapping wide dynamic ranges into narrow ones
- Fourier spectrum values in the range [0,1.5x10^{6}] transformed to [0,255] using log transformation

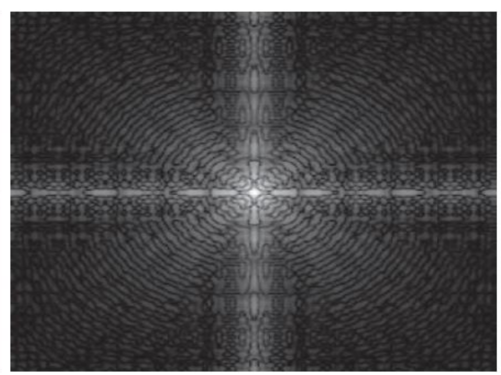
From the range 0 – 1.5×106 to the range 0 to 6.2

Original Image

Transformed Image

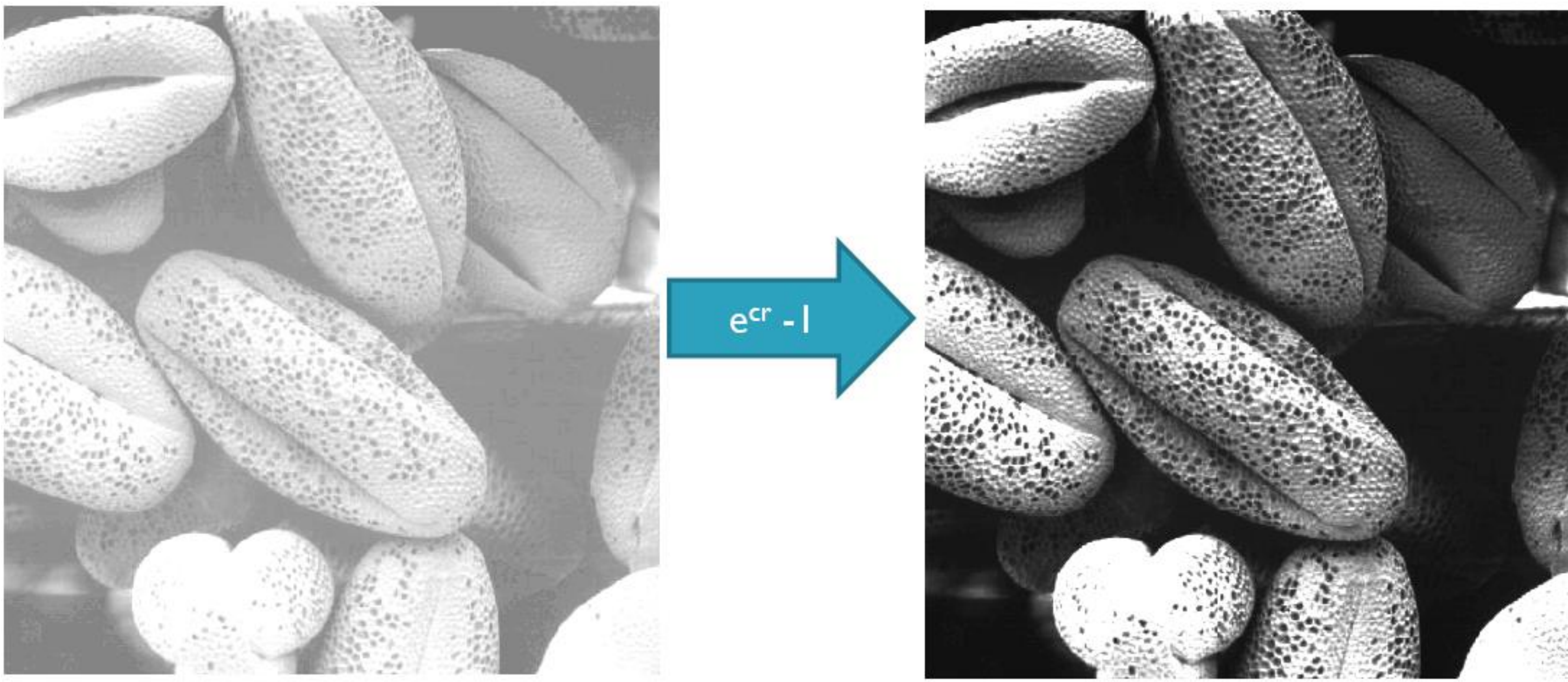


Fourier spectrum



Log Tr. Of Fourier spectrum

Inverse Log Transformation Example



Power-Law transformations

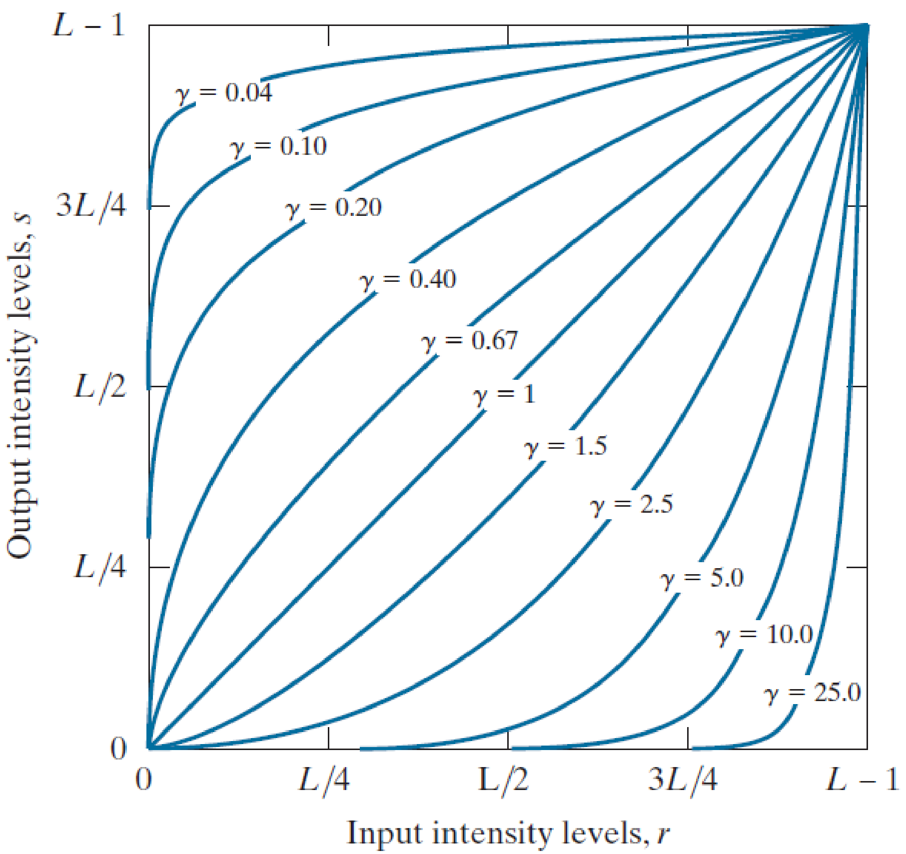
- Gamma correction function is used to correct image's luminance.
- Power-law transformations:

$$s=cr^{\gamma} \quad \text{or} \quad s=c(r+\varepsilon)^{\gamma}$$

  - γ : gamma, gamma correction
  - $\gamma<1$ maps a narrow range of dark input values into a wider range of output values [increase brightness]
  - $\gamma>1$ maps a narrow range of bright input values into a wider range of output values [increase darkness]
  - Power law is similar to log when gamma < 1 and similar to inverse log when gamma > 1
- Like the logarithmic transform, they are used to change the dynamic range of an image. However, in contrast to the logarithmic operator, they enhance high intensity pixel values.

Power-Law transformations

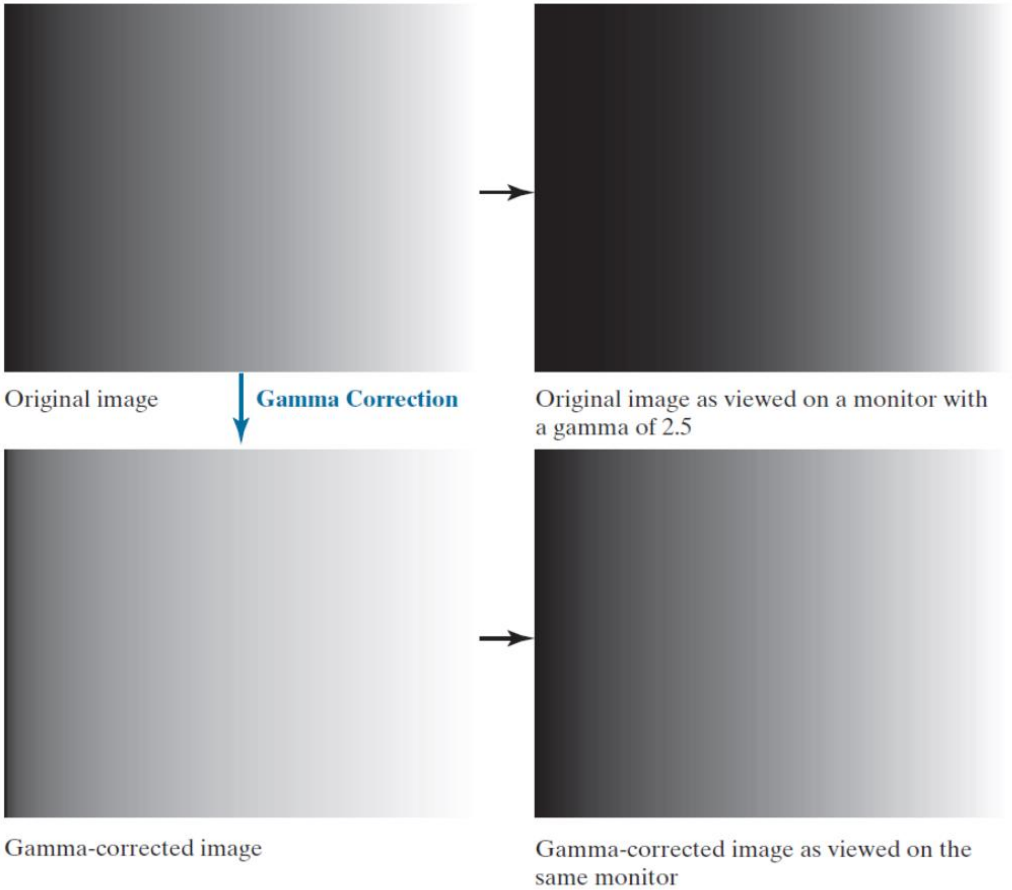
- Power-law curves with fractional values of γ map a narrow range of dark input values to a wider range of output values, and opposite for higher values of input levels



Power-Law transformations

Power-Law transformations

- Gamma-correction
  - Display devices have intensity-to-voltage response that is a power functions. Thus, images tend to be darker when displayed. Correction is needed using nth root before feeding the image to the monitor
  - Solution –display image after gamma correction to value that represents "average" of the types of monitors and computer systems to be used to display the image.



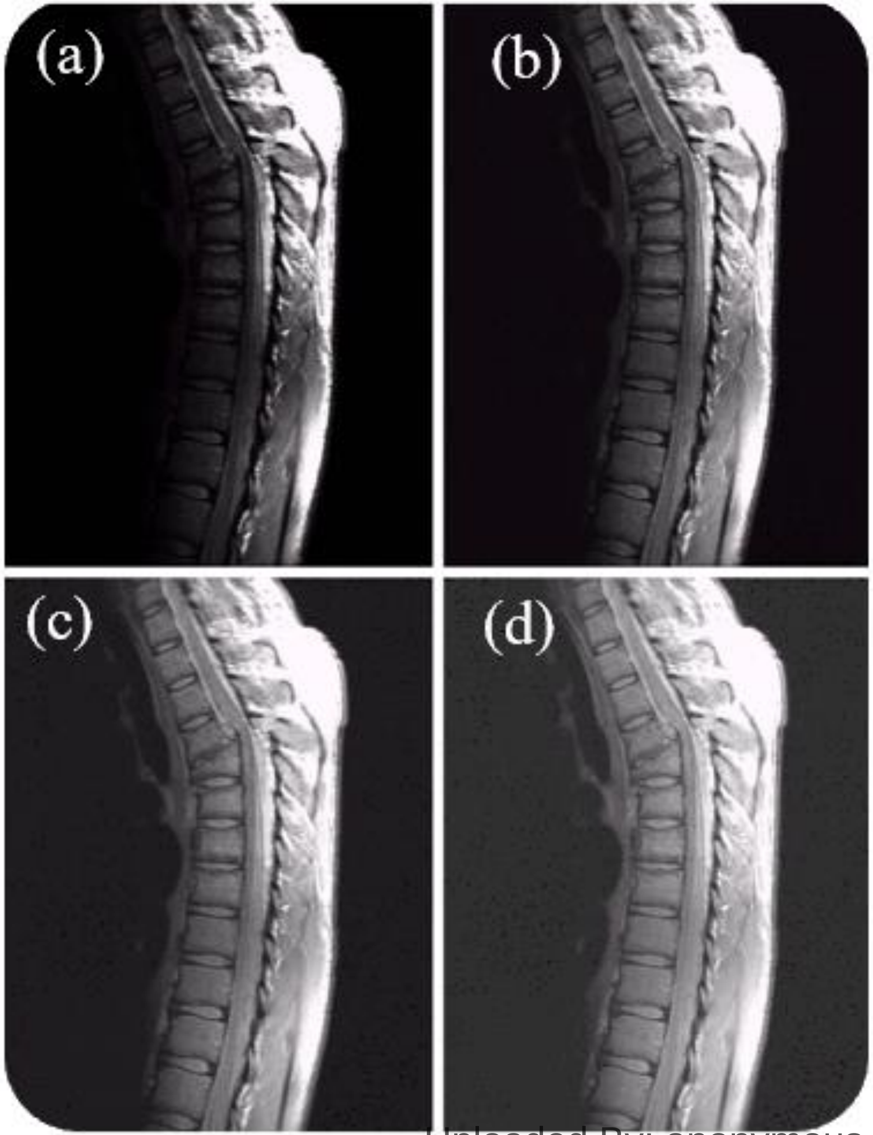
Power-Law transformations

a) a magnetic resonance image(MRI) of an upper thoracic human spine with a fracture dislocation and spinal cord impingement
- The picture is predominately dark
- An expansion of gray levels are desirable ⇨ needs $\gamma < 1$

b) result after power-law transformation with $\gamma = 0.6$, c=1

c) transformation with $\gamma = 0.4$ (best result)

d) transformation with $\gamma = 0.3$ (under acceptable level)

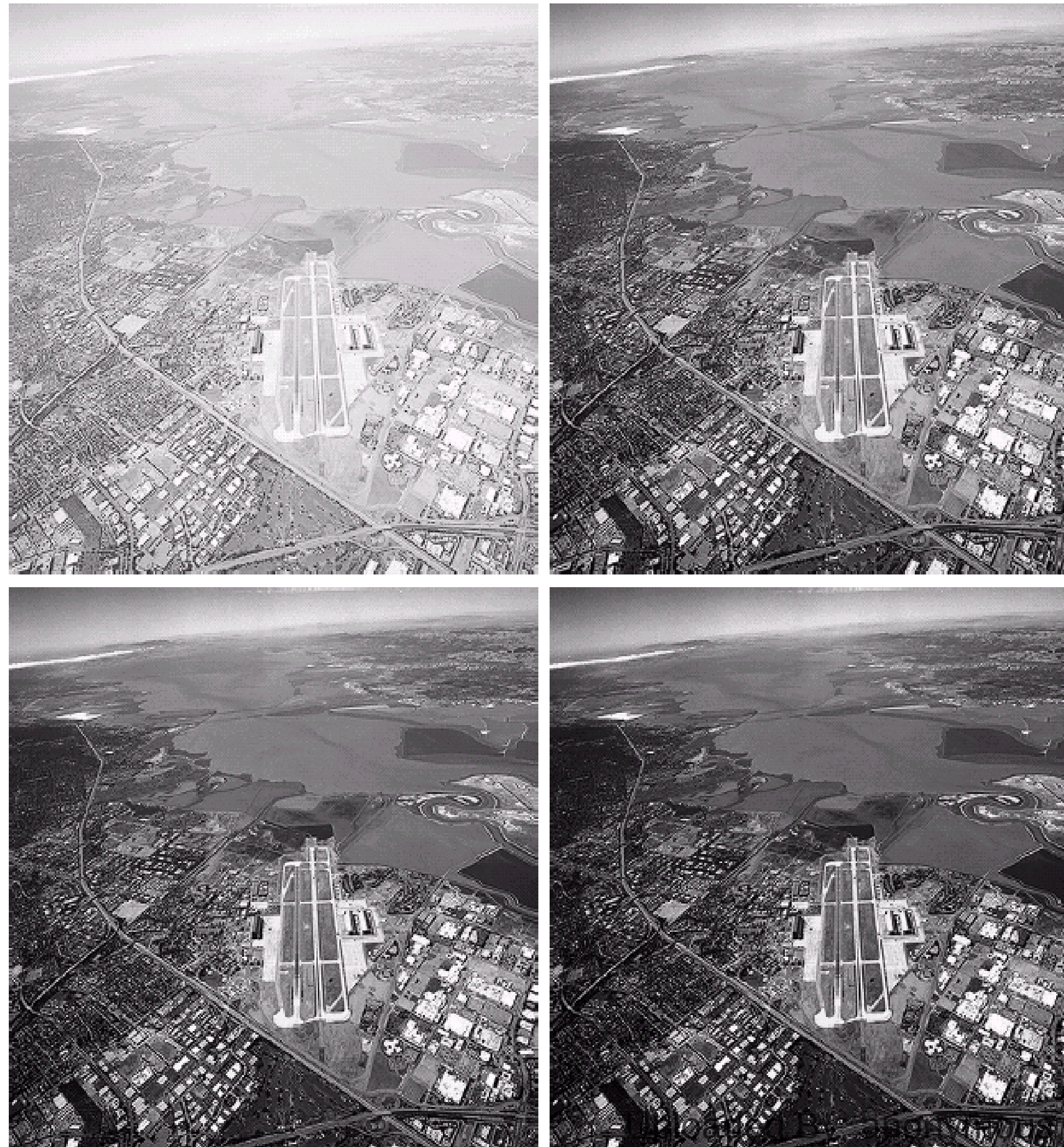


Power-Law transformations

- Gamma-correction application

a b
c d

FIGURE 3.9
(a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0,$ and 5.0, respectively. (Original image for this example courtesy of NASA.)

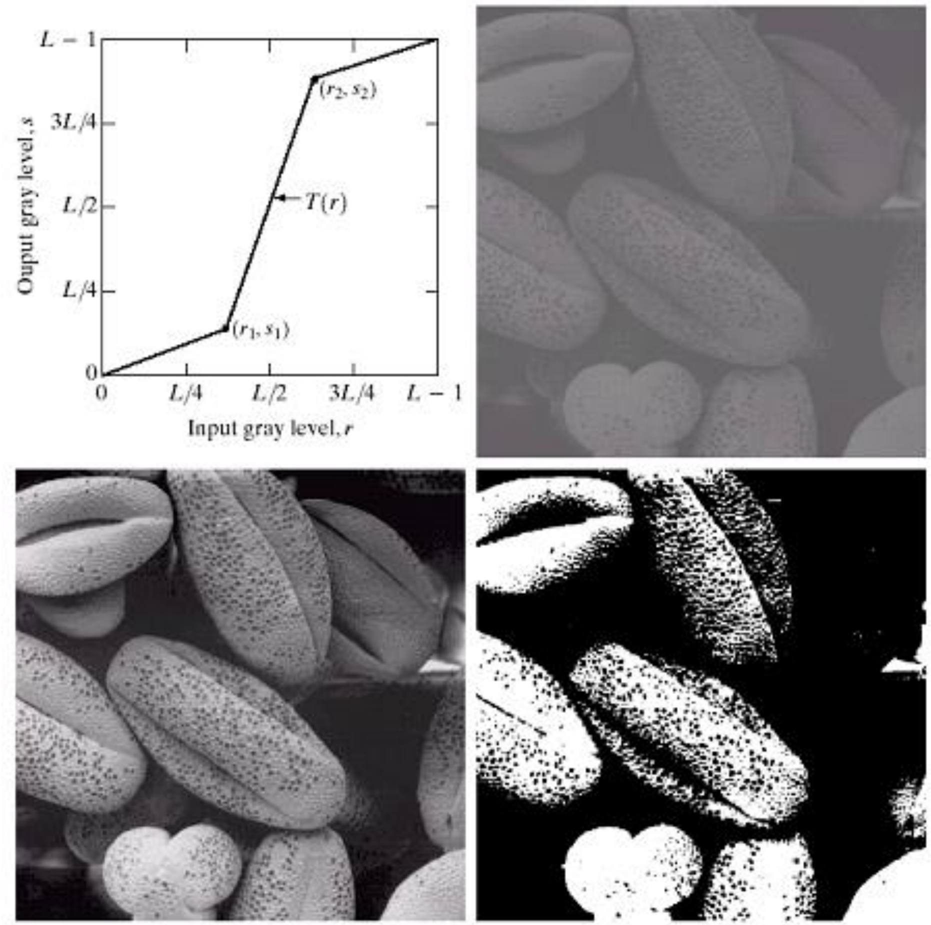


Piecewise-Linear Transformations

- Can represent arbitrarily complex functions to achieve different results.
- Piecewise-linear transformations offer more flexibility in terms of independent adjustments in different intensity ranges.
- Piecewise-linear transformations allow you to map specific input ranges to specific output ranges.
  - This makes them very versatile and useful for a variety of tasks, such as contrast stretching, thresholding, and gray-level slicing.
- However, piecewise-linear transformations can be complex to implement and can require a lot of trial and error to get the desired results.

Piecewise-Linear Transformations - Contrast Stretching

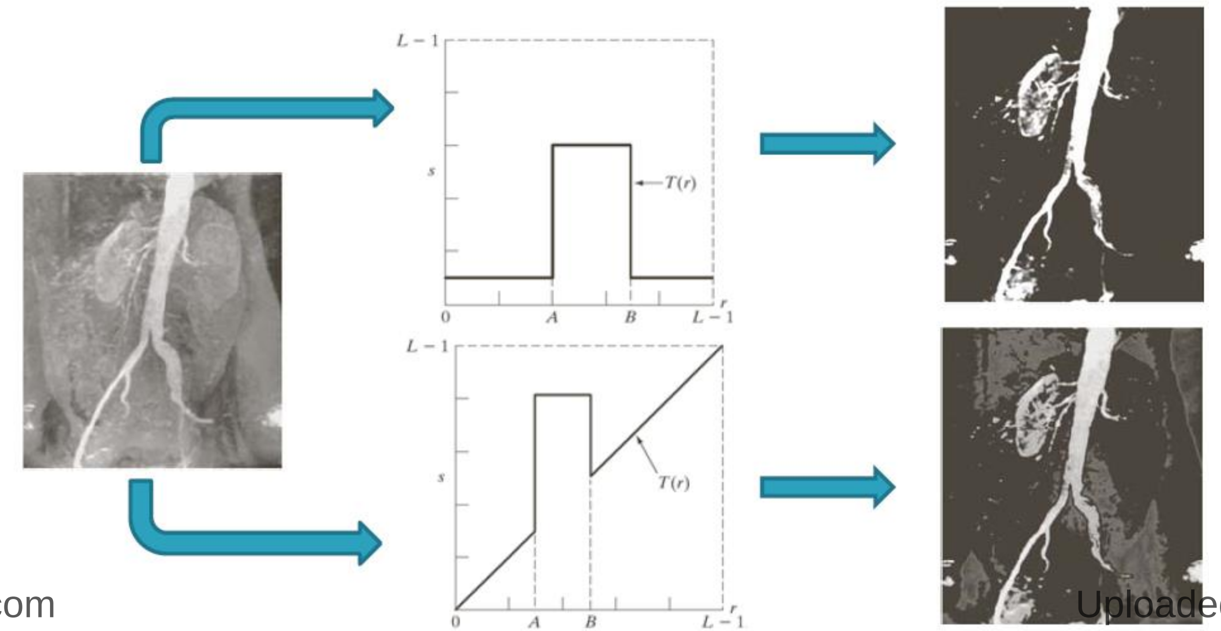
- Low-contrast images can result from:
  - Poor illumination,
  - Lack of dynamic range in the imaging sensor,
- *Contrast stretching* expands the range of intensity levels in an image so that it spans the ideal full intensity range of the recording medium or display device.
- $r1 \le r2$ and $s1 \le s2$ to preserve the order of gray levels
- The result depends on the values of r1, r2, s1, and s2



(a) Form of transformation function. (b) A low-contrast stretching. (c) Result of contrast stretching. (d) Result of thresholding

Piecewise-Linear Transformations - Gray-level Slicing

- Used to highlight specific range of gray levels
  - Enhancing features in satellite imagery, such as masses of water,
  - and enhancing flaws in X-ray images
- Two approaches
  - Transformation highlights range [A,B] of gray level and reduces all others to a constant level
  - Transformation highlights range [A,B] but preserves all other levels



Outline

- What is Image Enhancement?
- Contrast Enhancement
  - Intensity Transformation Functions
  - **Histogram Equalization**
- Spatial Filtering
  - Basic Concepts
  - Smoothing Filters
  - Sharping Filters
  - Nonlinear Filters
- Image Quality Assessment
  - Subjective image quality assessment
  - Objective image quality assessment

Histogram Processing

- For an image with gray levels in [0,L-1] and MxN pixels, the histogram is a discrete function given by

$$h(r_k) = n_k$$

  - Where
    - r_k : the kth gray level
    - n_k : the number of pixels in the image having gray level r_k
    - $h(r_k)$: histogram of a digital image with gray levels r_k
- It is a common practice to normalize the histogram function by the number of pixels in the image by

$$p(r_k) = n_k / n$$

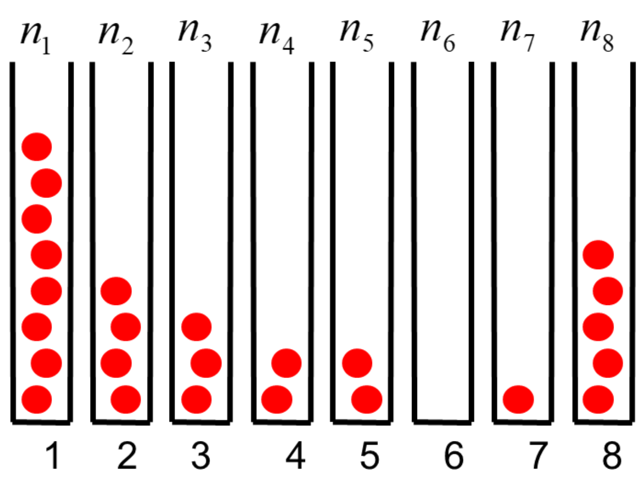
  - The normalized histogram can be used as an estimate of the probability density function of the image
  - The sum of the normalised histogram function over the range of all intensities is 1.
- Histograms are widely used in image processing: enhancement, compression, segmentation …

Histogram Processing: Example

1	8	4	3	4
1	1	1	7	8
8	8	3	3	1
2	2	1	5	2
1	1	8	5	2

$$h(r_k) = n_k$$

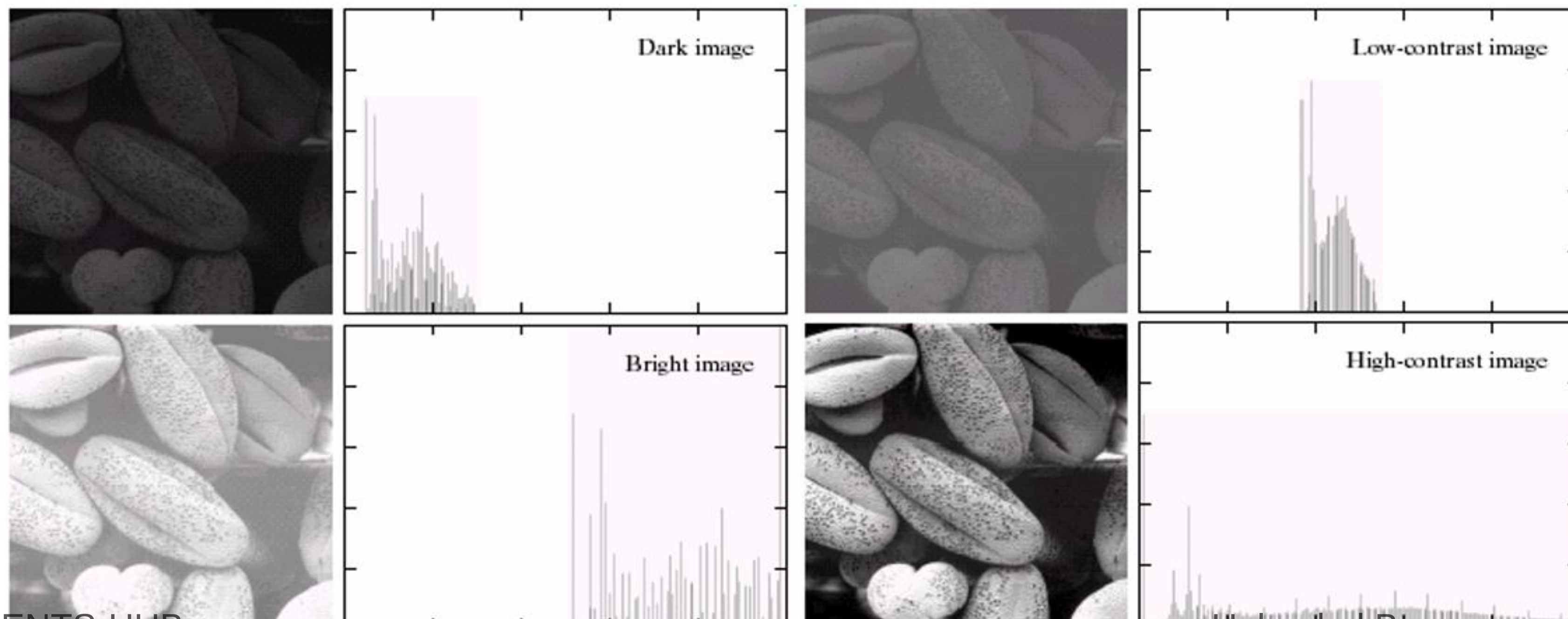
$h(r_1) = 8$
$h(r_2) = 4$
$h(r_3) = 3$
$h(r_4) = 2$
$h(r_5) = 2$
$h(r_6) = 0$
$h(r_7) = 1$
$h(r_8) = 5$



$p(r_1) = 8/25 = 0.32$
$p(r_2) = 4/25 = 0.16$
$p(r_3) = 3/25 = 0.12$
$p(r_4) = 3/25 = 0.08$
$p(r_5) = 2/25 = 0.08$
$p(r_6) = 0/25 = 0.00$
$p(r_7) = 1/25 = 0.04$
$p(r_8) = 5/25 = 0.20$

Histogram Processing

- For enhancement, histograms can be used to infer the type of image quality: dark, bright, low or high contrast
- Intuitively, we expect that an image whose pixels tend to occupy the entire range of possible gray levels, tend to be distributed uniformly will have a high contrast and show a great deal of gray level detail.
- It is possible to develop a transformation function that can achieve this effect using histograms.

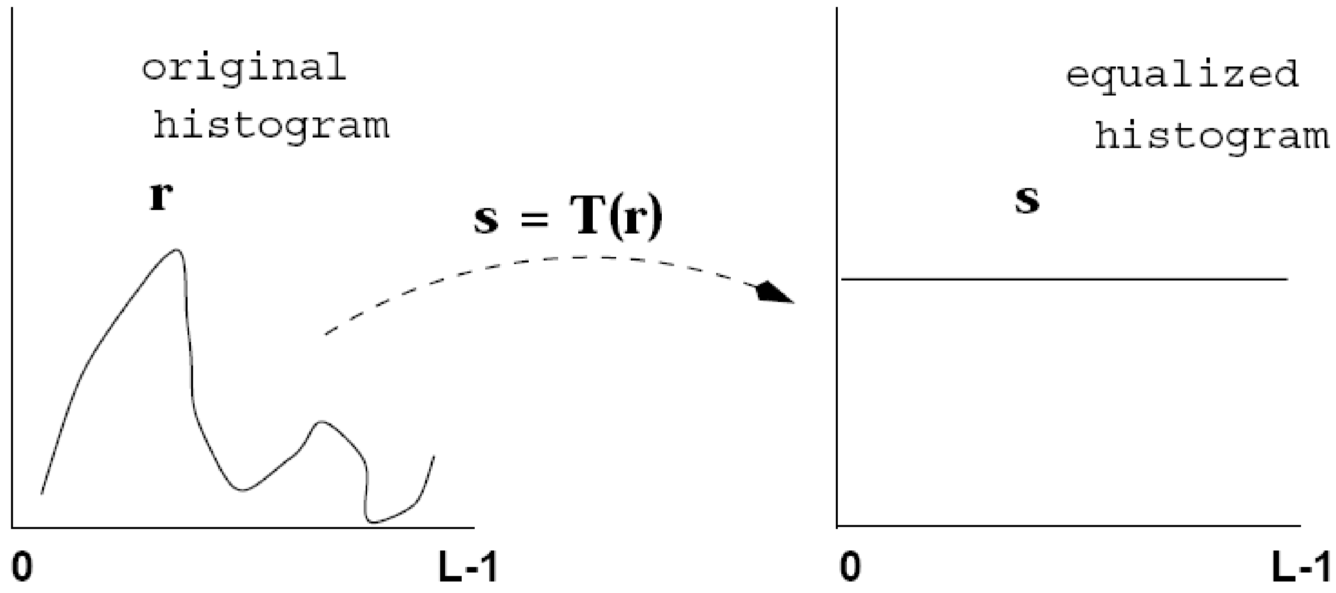


Histogram Equalization

- It is quite acceptable that high contrast images have flat histograms (uniform distribution)
- As the low-contrast image's histogram is narrow and centered toward the middle of the gray scale, if we distribute the histogram to a wider range the quality of the image will be improved.
- We can do it by adjusting the probability density function of the original histogram of the image so that the probability spread equally
- **Histogram equalization attempts to transform the original histogram into a flat one for the goal of better contrast**

Histogram Equalization

- The main idea is to redistribute the gray-level values uniformly.



Histogram Equalization

- Let r be a continuous variable that represents the intensity values in the range [0,L-1], then a valid transformation function for enhancement purposes s = T(r) should satisfy
  - T(r) is monotonically increasing in the interval $0 \le r \le L-1$
    - Preserves the increasing order from black to white in the output image thus it won't cause a negative image
  - T(r) is bounded by [0,L-1] for all values of r
    - Guarantees that the output gray levels will be in the same range as the input levels.
  - The inverse transformation function that maps s back to r

$$r = T^{-1}(s)$$

    - requires that T(r) to be strictly monotonically increasing

Histogram Equalization

- The probability of occurrence of gray level in an image is approximated by

$$\boldsymbol{p_r(r_k) = \frac{n_k}{n}} \qquad \boldsymbol{where\ k\ =\ 0,\ 1,\ ...,\ L\text{-}1}$$

- The intensity transformation function for histogram equalization is

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$

$$= (L-1)\sum_{j=0}^{k} \frac{n_j}{n} \qquad where\ k\ =\ 0,\ 1,\ ...,\ L\text{-}1$$

Thus, an output image is obtained by mapping each pixel with level $\boldsymbol{r_k}$ in the input image into a corresponding pixel with level $\boldsymbol{s_k}$ in the output image

Histogram equalization example

- Compute histogram function

0	0	1	0	2	0
1	0	7	7	7	0
0	7	0	0	7	0
1	0	0	7	2	0
0	0	7	1	0	1
1	0	7	7	7	0

frequencies

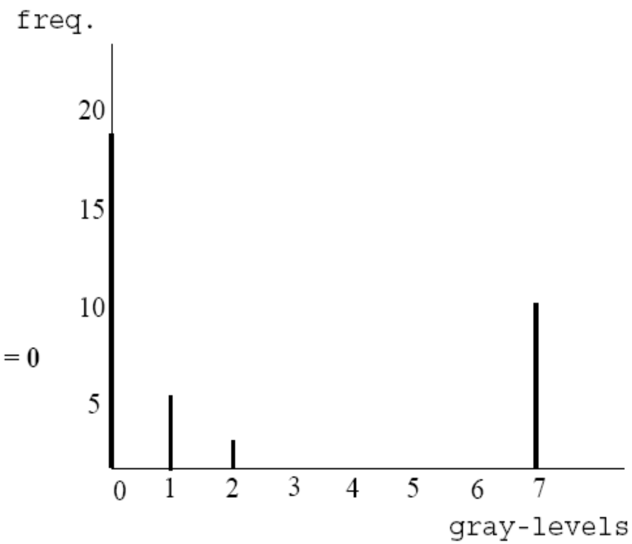
$f(0) = 18$

$f(1) = 6$

$f(2) = 2$

$f(3) = f(4) = f(5) = f(6) = 0$

$f(7) = 10$



- Divide frequencies by total number of pixels to represent as probabilities.

$$p_k = n_k / N$$

$$P(0) = \frac{f(0)}{36} = \frac{1}{2} \qquad P(1) = \frac{f(1)}{36} = \frac{1}{6}$$

$$P(2) = \frac{f(2)}{36} = \frac{1}{18} \qquad P(3) = P(4) = P(5) = P(6) = 0$$

$$P(7) = \frac{f(7)}{36} = \frac{5}{18}$$

Histogram equalization example

- Compute intensity transformation function

$$T(r_k) = \sum_{j=1}^{k} p(r_k)$$

$$P(0) = \frac{f(0)}{36} = \frac{1}{2} \qquad P(1) = \frac{f(1)}{36} = \frac{1}{6}$$

$$P(2) = \frac{f(2)}{36} = \frac{1}{18} \qquad P(3) = P(4) = P(5) = P(6) = 0$$

$$P(7) = \frac{f(7)}{36} = \frac{5}{18}$$

$$T(r_0) = \sum_{j=0}^{0} p_r(r_k) = p_r(r_0) = \frac{1}{2} = 0.5$$

$$T(r_1) = \sum_{j=0}^{1} p_r(r_k) = \frac{1}{2} + \frac{1}{6} = 0.5 + 0.17 = 0.66$$

$$T(r_2) = \sum_{j=0}^{2} p_r(r_k) = \frac{1}{2} + \frac{1}{6} + \frac{1}{18} = 0.5 + 0.17 + 0.06 = 0.72$$

$$T(r_3) = T(r_4) = T(r_5) = T(r_6) = 0.72$$

$$T(r_7) = \sum_{j=0}^{7} p_r(r_k) = 0.72 + \frac{5}{18} = 1.01$$

- Multiple intensity transformation function by intensity level – 1

$$S_0 = (L\text{-}1) * T(r_0) = 7*0.5 = \lfloor 3.5 \rfloor = 3$$

$$S_1 = (L\text{-}1) * T(r_1) = 7*0.66 = \lfloor 4.6 \rfloor = 4$$

$$S_2 = (L\text{-}1) * T(r_2) = 7*0.72 = \lfloor 5.04 \rfloor = 5$$

$$S_3 = S_4 = S_5 = S_6 = 7*0.72 = \lfloor 5.04 \rfloor = 5$$

$$S_7 = (L\text{-}1) * T(r_7) = 7*1.01 = \lfloor 7.01 \rfloor = 7$$

Histogram Equalization another Example

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

TABLE 3.1 Intensity distribution and histogram values for a 3-bit, 64×64 digital image.

s_k	Value	Appr.	Ps(s)
s0	1.33	1	
s1	3.08	3	790/4096
s2	4.55	5	
s3	5.67	6	1023/4096
s4	6.23	6	
s5	6.65	7	850/4096
s6	6.86	7	(656+329)/4096
s7	7.00	7	(245+122+81)/4096

$S_0 = T(r_0) = 7 \sum_{j=0}^{0} p_r(r_k) = 7 * p_r(r_0) = 7*0.19 = \lfloor 1.4 \rfloor = 1$

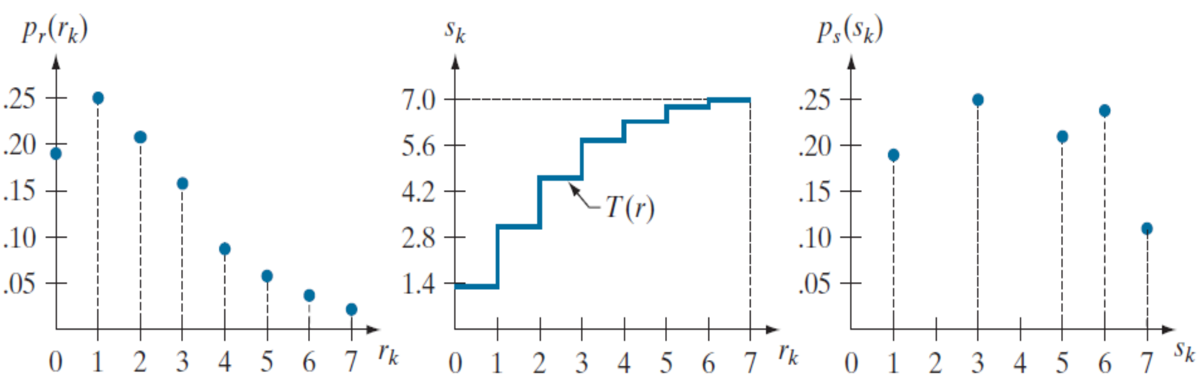
$S_1 = T(r_1) = 7 \sum_{j=0}^{1} p_r(r_k) = 7 * 0.44 = \lfloor 3.08 \rfloor = 3$

.....

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} \frac{n_j}{n}$$

a b c

FIGURE 3.19 Histogram equalization. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

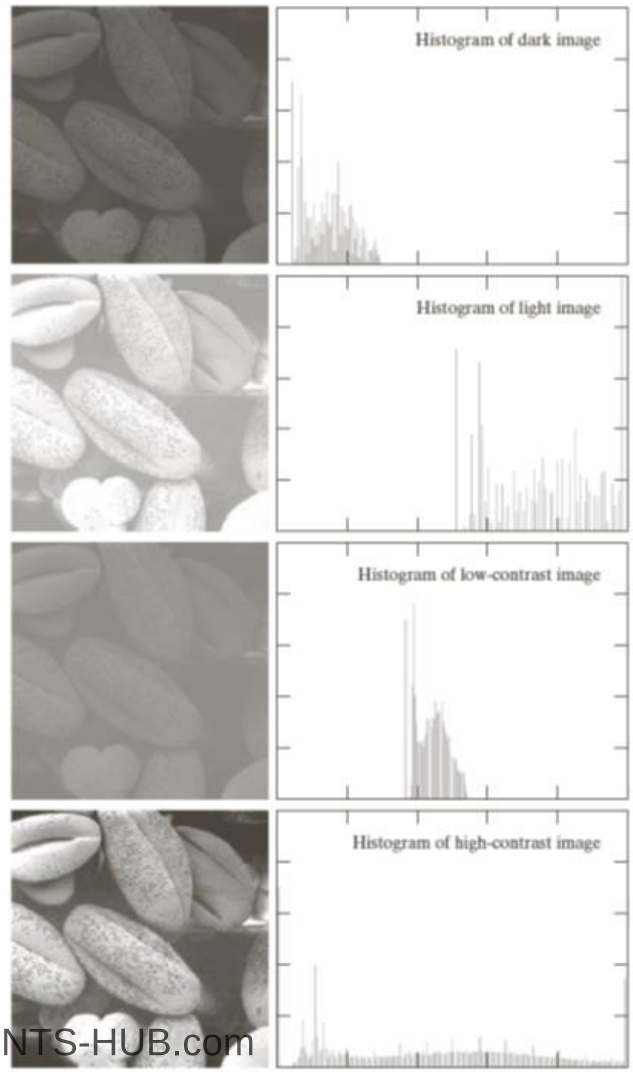


Histogram Equalization

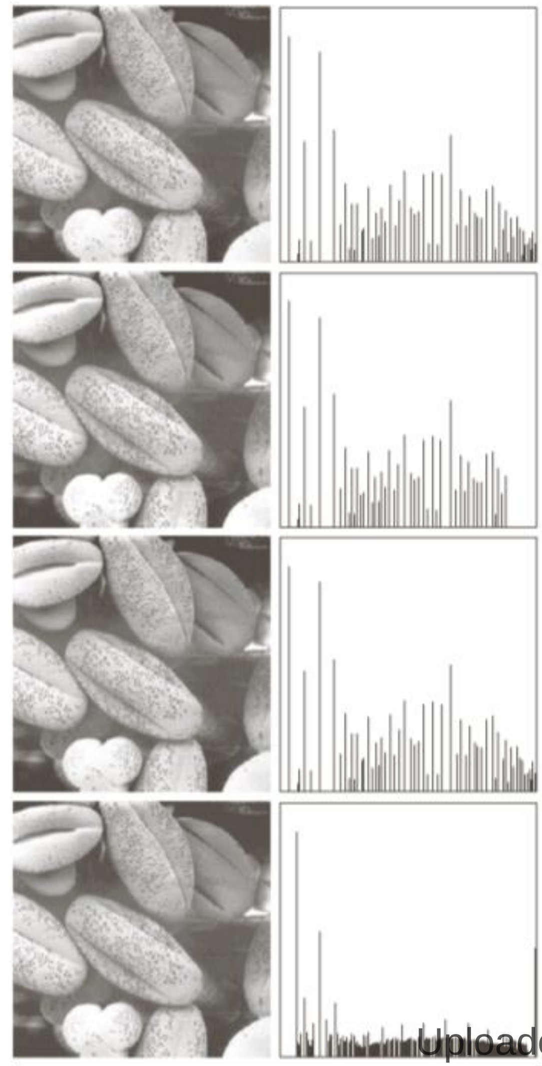
- It is clearly seen that
  - Histogram equalization distributes the gray level to reach the maximum gray level (white) because the cumulative distribution function equals 1 when $0 \leq r \leq L-1$
  - If the cumulative numbers of gray levels are slightly different, they will be mapped to little different or same gray levels as we may have to approximate the processed gray level of the output image to integer number
  - Thus the discrete transformation function can't guarantee the one to one mapping relationship

Histogram Equalization

Original images and histograms

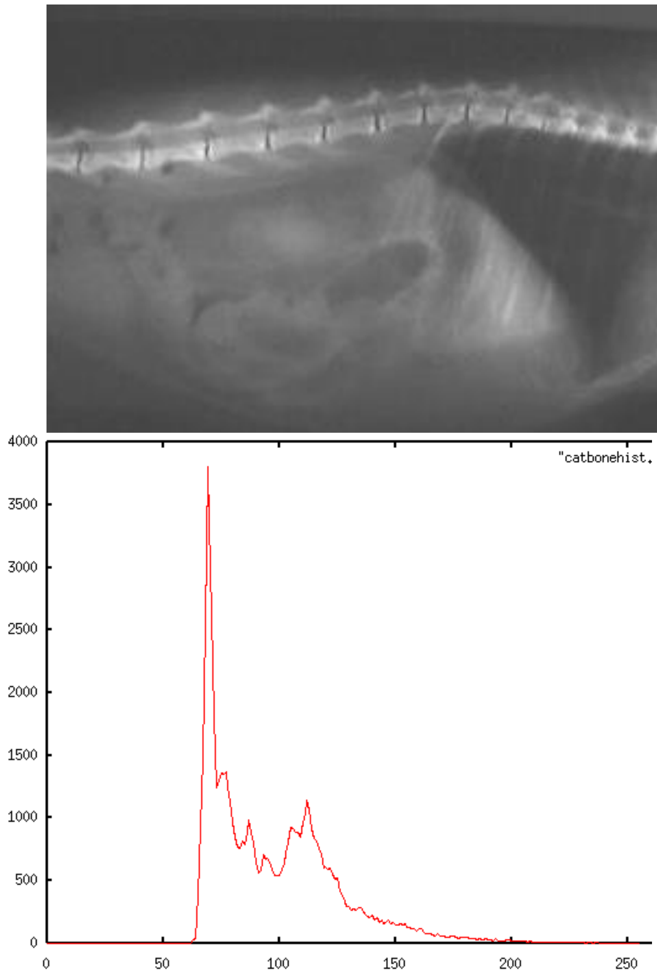


Equalized images and histograms

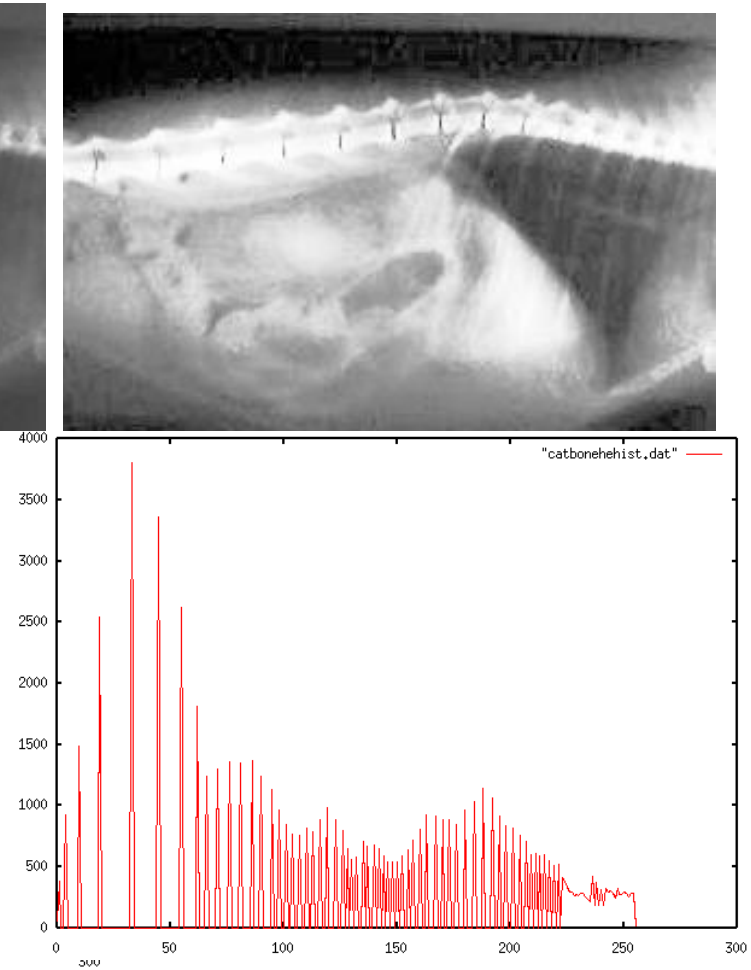


Histogram equalization Examples

Low contrast

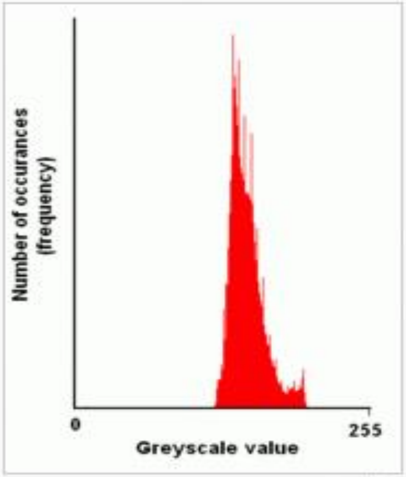


High contrast



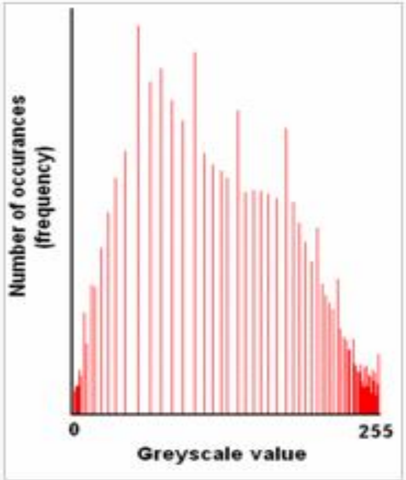
Histogram equalization Examples

An unequalized image



Corresponding histogram

Same image after histogram equalization



Corresponding histogram

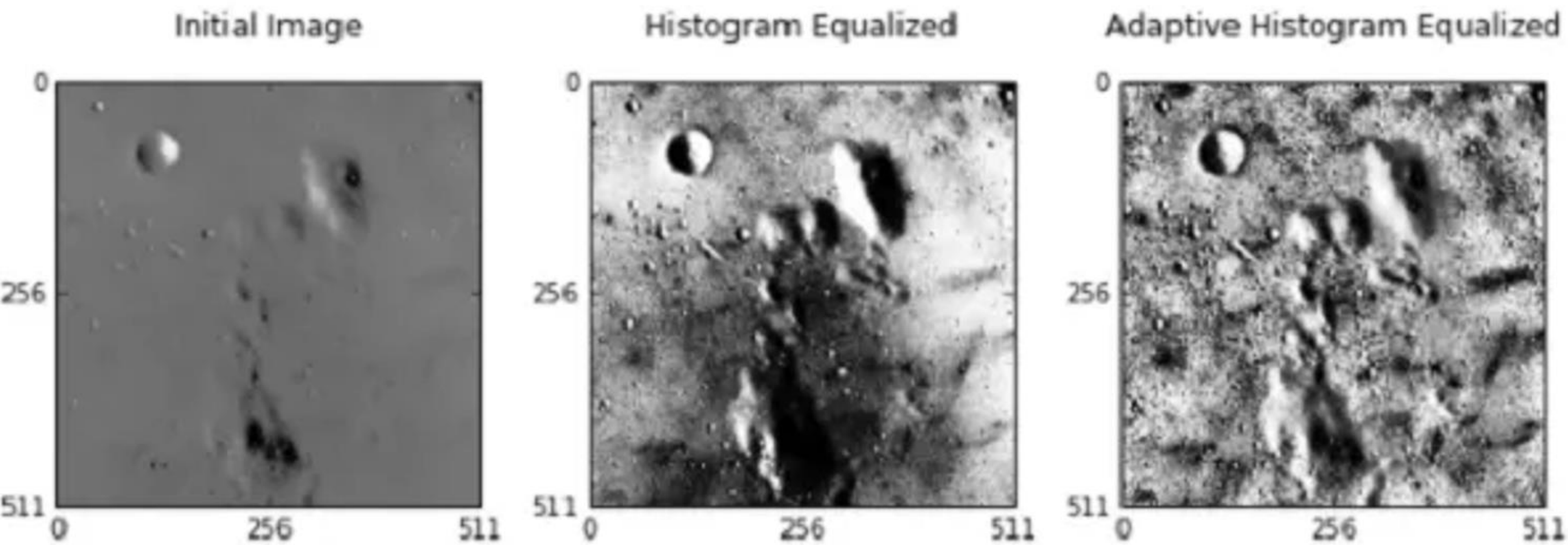
Histogram Equalization – Pros and Cons

- **Pros:**
  - **Simple Implementation**: The basic algorithm for histogram equalization is relatively simple to implement and computationally efficient, making it accessible for various applications.
  - **No Additional Information Required**: Histogram equalization does not require additional information about the image or scene. It operates solely based on the pixel intensities present in the image.
  - **No Parameter Tuning**: The basic histogram equalization method does not require tuning of parameters. It is a one-step process that can be applied directly to the image.
- **Cons:**
  - **Global Adjustment**: Basic histogram equalization is a global operation, meaning it considers the entire image. In some cases, this global approach may not be suitable, especially when local variations in intensity are essential.
  - **Lack of Control**: Histogram equalization does not provide fine-grained control over the enhancement process. In situations where specific adjustments are needed, more advanced methods or post-processing steps may be required.

Adaptive Histogram Equalization

- Adaptive Histogram Equalization is an extension of the standard histogram equalization technique that adapts the transformation locally within the image rather than globally.
- This is particularly useful in situations where different regions of an image have varying lighting conditions or contrast levels.
- To perform adaptive histogram equalization (AHE) on an image, follow these steps:
  1. Divide the image into a grid of tiles. The tile size is typically chosen to be between 16x16 and 64x64 pixels.
  2. For each tile:
     a. Calculate the histogram of the pixel values.
     b. Generate a cumulative distribution function (CDF) from the histogram.
     c. Map the pixel values in the tile to new values using the CDF.
     d. Choose the new values such that the histogram of the output tile is more uniform.
  3. Combine the processed tiles back into a single image.
  4. After equalization, to remove artifacts in tile borders, bilinear interpolation is applied

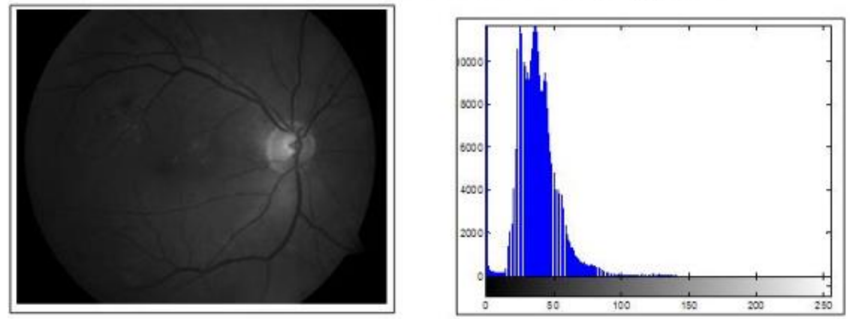
Adaptive Histogram Equalization



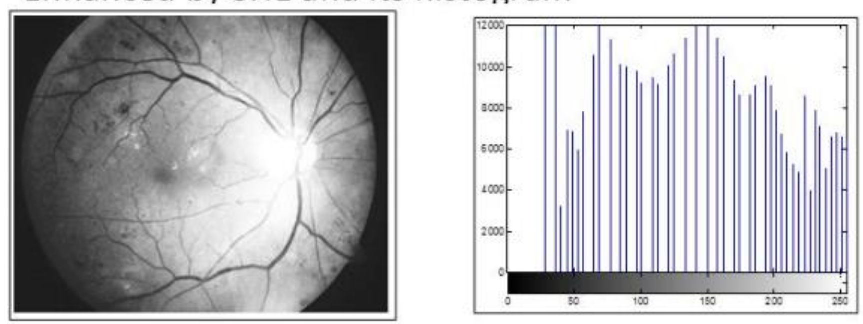
Adaptive Histogram Equalization

- In Adaptive Histogram Equalization, if noise is there, it will be amplified.
- To avoid this, **contrast limiting** is applied.
- If any histogram bin is above the specified contrast limit (by default 40 in OpenCV), those pixels are clipped and distributed uniformly to other bins before applying histogram equalization.

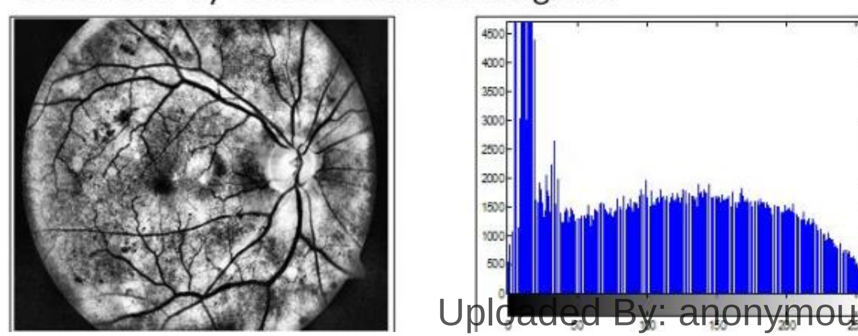
Original Green component and its histogram



Enhanced by SHE and its histogram



Enhanced by CLAHE and its histogram



Multiple Histogram Equalization

- Applying multiple histogram equalization operations to an image can have a number of effects, both positive and negative.
- **Positive side:** multiple histogram equalization can further enhance the contrast of an image and make it easier to see details. This can be useful for images with very low contrast or images that have been taken in poor lighting conditions.
- **Negative Side:** applying too much histogram equalization can also have a number of negative effects.
  - Applying histogram equalization excessively can make an image look unnatural. Fine details may be exaggerated, and the visual appearance may deviate significantly from the original scene.
  - Multiple histogram equalization can amplify noise in the image and make it more noticeable.
  - Multiple histogram equalization can lead to clipping, which is when the pixel values in the image are saturated to the maximum or minimum value. This can cause the image to lose detail and look washed out.

Histogram Specification

- Histogram equalization has a disadvantage which is that it can generate only one type of output image.
- With Histogram Specification, we can specify the shape of the histogram that we wish the output image to have.
- We can use the method used in deriving the transformation function of histogram equalization to find transformation function for the desired histogram, however
  - This requires the availability of $p_s(s)$ in mathematical form and the ability to express s in terms of r
  - It doesn't have to be a uniform histogram
  - Histogram specification is a trial-and-error process
  - There are no rules for specifying histograms, and one must resort to analysis on a case-by-case basis for any given enhancement task.

Contrast Enhancement: Summary

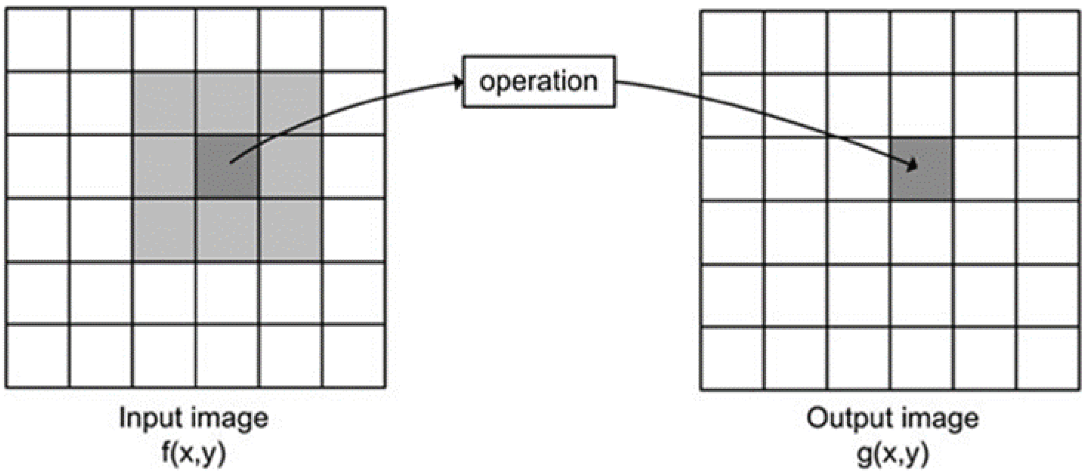
- **Log Transformation and Power-law Transformation**
  - Use when the image has a wide range of intensity values, and you want to enhance details in the darker regions.
  - Common in medical imaging for visualizing features in X-ray and MRI images.
- **Contrast Stretching**
  - Use when the image has poor contrast and the pixel values are not covering the full dynamic range.
  - Suitable for enhancing visibility of details in poorly-contrasted images.
- **Gray-level Slicing**
  - Use when you want to highlight specific intensity ranges, such as highlighting objects with specific intensity values.
  - Effective in emphasizing certain features while suppressing others.
- **Histogram Equalization**
  - Use when you want a global enhancement of the image contrast.
  - Suitable for improving the visibility of details in images with uneven intensity distributions.

Outline

- What is Image Enhancement?
- Contrast Enhancement
  - Intensity Transformation Functions
  - Histogram Processing
- **Spatial Filtering**
  - **Basic Concepts**
  - Correlation and Convolution
  - Smoothing Filters
  - Sharping Filters
  - Nonlinear Filters
- Image Quality Assessment
  - Subjective image quality assessment
  - Objective image quality assessment

Spatial domain filtering

- Capabilities of point operations are limited
- Filters: combine pixel's value + values of neighbors



- Combining multiple pixels needed for certain operations:
  - Enhance an image, e.g., denoise, Blurring, Sharpening.
  - Extract information, e.g., texture, edges.
  - Detect patterns, e.g., template matching.

Fundamentals of Spatial Filtering

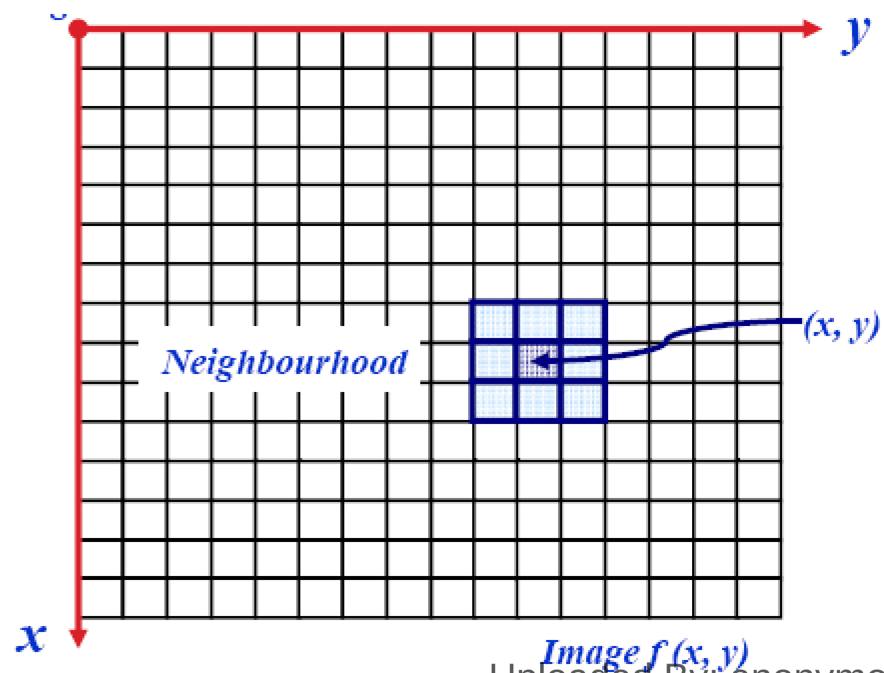
- Filtering is borrowed from the frequency domain processing and refers to the process of passing or rejecting certain frequency components
  - Highpass, lowpass, band-reject , and bandpass filters
- Filtering is achieved in the frequency domain by designing the proper filter
- Filtering can done in the spatial domain also by using filter masks (kernels, templates, or windows)
  - Unlike frequency domain filters, spatial filters can be nonlinear !

Spatial Filtering Mechanics

- A spatial filter is characterized by
  - A rectangular neighborhood of size mxn (usually m and n are odd)
  - A predefined operation that is specified by the mask values at each position.
- Spatial filtering Operation
  - The filter mask is centered at each pixel in the image and the output pixel value is computed based on the operation specified by the mask

w(-1,-1)	w(-1,0)	w(-1,1)
w(0,-1)	w(0,0)	w(0,1)
w(1,-1)	w(1,0)	w(1,1)

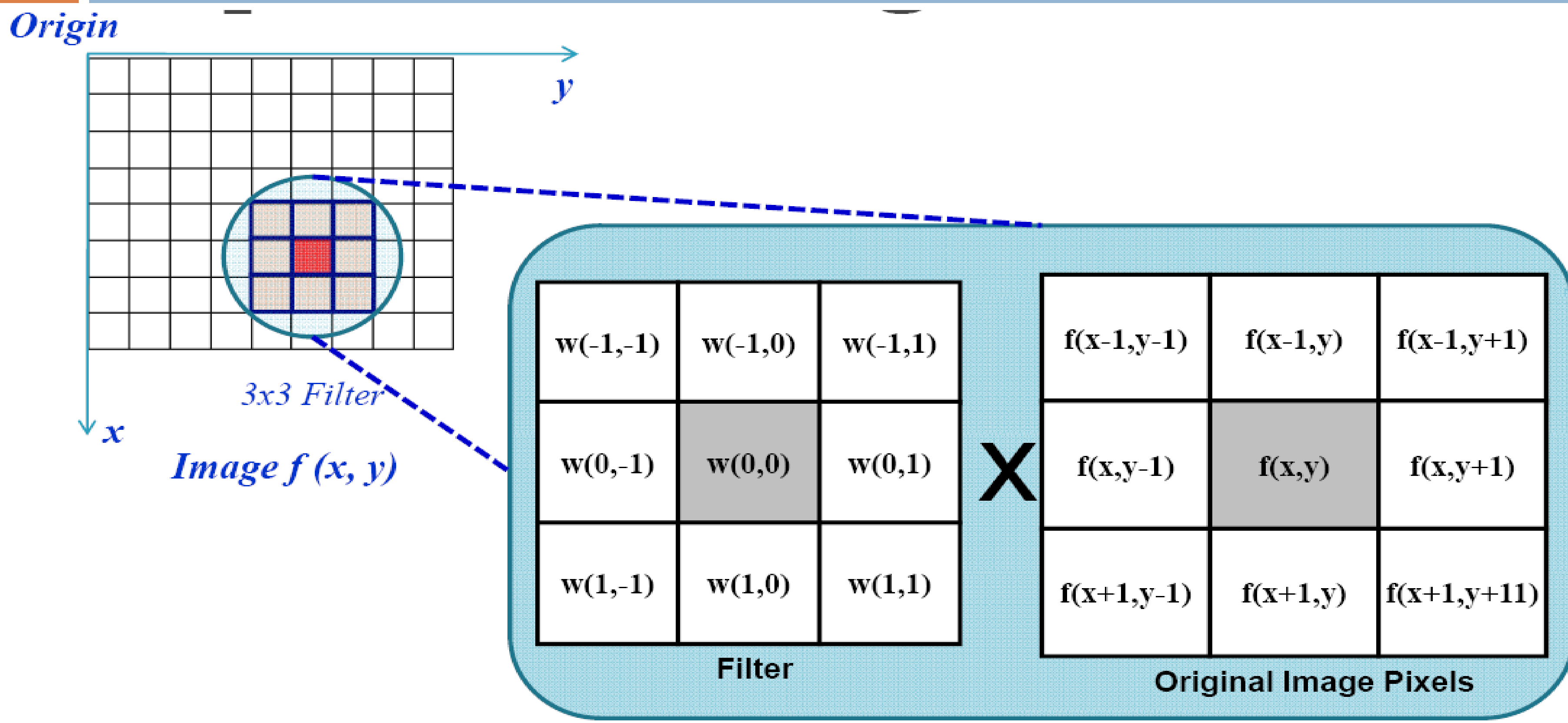
3x3 filter mask example



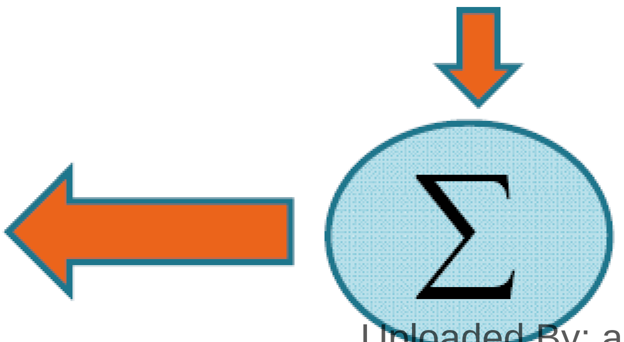
Spatial Filtering Mechanics

- **Spatial Filtering Process:** The spatial filtering on the whole image is given by:
  1. Move the mask over the image at each location.
  2. Compute sum of products between the mask coefficients and pixels inside subimage under the mask.
  3. Store the results at the corresponding pixels of the output image.
  4. Move the mask to the next location and go to step 2 until all pixel locations have been used.

Spatial Filtering Mechanics



$$g(x,y)=\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)f(x+s,y+t)$$



Spatial Filtering Mechanics - Example

Image

$$f_i = \sum_{k=1}^{9} w_k I_k(i)$$

$=(-1x10) + (-1x11) + (-1x8) + (-1x40) + (8x35)$

$+(-1x42) + (-1x38) + (-1x36) + (-1x46) = 14$

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

=

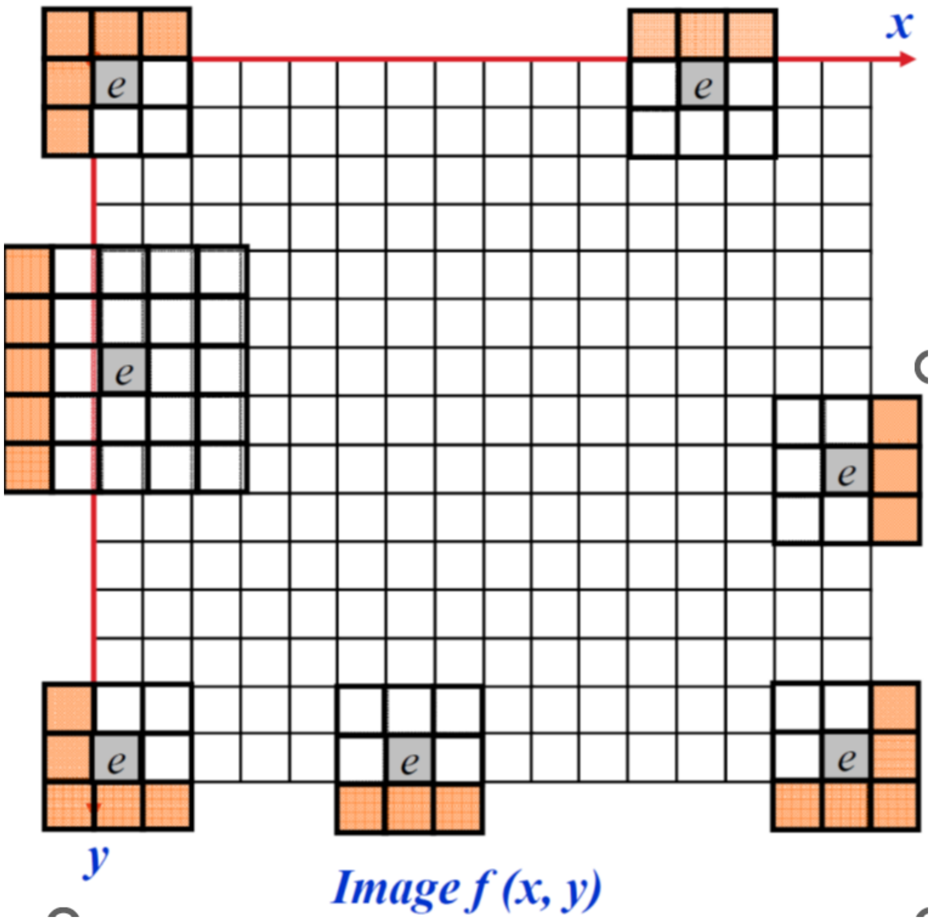
–1	–1	–1
–1	8	–1
–1	–1	–1

12	11	12	13	13	9
10	8	10	11	8	13
32	36	40	35	42	40
40	37	38	36	46	41
41	36	89	39	42	39
42	37	39	43	45	38

The mechanics of image filtering with an N*N=3*3 kernel filter

Treatment of Pixels at Edges

- There are a few approaches to deal with missing edge pixels:
  - Pad the image
    - Typically with either all white or all black pixels
  - Replicate border pixels
  - Truncate the image
  - Allow pixels *wrap around* the image
    - Can cause some strange image artefacts



Average filtering example with border padding

Input

Average = round(2+5+0+3+0+0+0+0+0)/9 = 1

0	0	0	0	0	0	0	0
0	1	4	0	1	3	1	0
0	2	2	4	2	2	3	0
0	1	0	1	0	1	0	0
0	1	2	1	0	2	2	0
0	2	5	3	1	2	5	0
0	1	1	4	2	3	0	0
0	0	0	0	0	0	0	0

Output

1	1	1	1	1	1
1	2	2	2	1	1
1	2	1	1	1	1
1	2	1	1	1	1
1	2	2	2	2	2
1	2	2	2	1	1

Average filtering example with border replication

Input

Average = round(1+4+0+1+4+0+2+2+4)/9 = 2

1	1	4	0	1	3	1	1
1	1	4	0	1	3	1	1
2	2	2	4	2	2	3	3
1	1	0	1	0	1	0	0
1	1	2	1	0	2	2	2
2	2	5	3	1	2	5	5
1	1	1	4	2	3	0	0
1	1	1	4	2	3	0	0

Output

2	2	2	2	2	2
2	2	2	2	1	2
1	2	1	1	1	2
2	2	1	1	1	2
2	2	2	2	2	2
2	2	3	3	2	2

Average filtering example with border truncation

Average = round(1+4+0+2+2+4+1+0+1)/9 = 2

Input

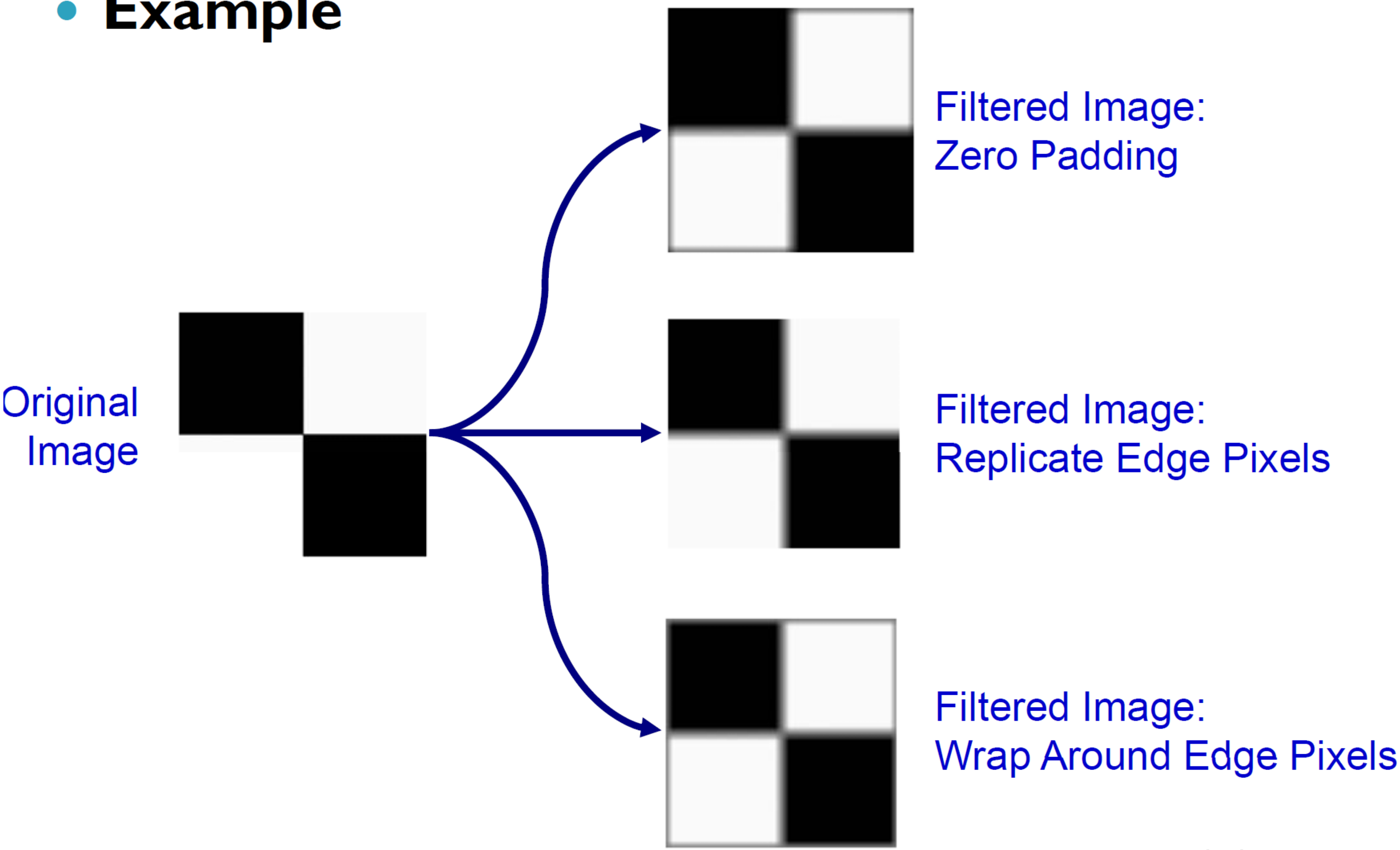
1	4	0	1	3	1
2	2	4	2	2	3
1	0	1	0	1	0
1	2	1	0	2	2
2	5	3	1	2	5
1	1	4	2	3	0

Output

1	4	0	1	3	1
2	2	2	2	1	3
1	2	1	1	1	0
1	2	1	1	1	2
2	2	2	2	2	5
1	1	4	2	3	0

Treatment of Pixels at Edges

- **Example**



Key properties of linear filters

- Image filters - linear or nonlinear.
  - Linear filters are also know as convolution filters
  - Nonlinear operations such as median filter.
- Linearity: $\text{filter}(f_1 + f_2) = \text{filter}(f_1) + \text{filter}(f_2)$
- Shift invariance: same behavior regardless of pixel location.
  - The value of the output depends on the pattern in the image neighborhood, not the position of the neighborhood

    $\text{filter}(\text{shift}(f)) = \text{shift}(\text{filter}(f))$
  - Any linear, shift-invariant operator can be represented as a convolution
- Commutative: $a * b = b * a$

Key properties of linear filters

- Associative: $a * (b * c) = (a * b) * c$
  - Often apply several filters one after another: $(((a * b_1) * b_2) * b_3)$
  - This is equivalent to applying one filter: $a * (b_1 * b_2 * b_3)$
- Distributes over addition: $a * (b + c) = (a * b) + (a * c)$
- Scalars factor out: $ka * b = a * kb = k (a * b)$
- Identity: unit impulse $e = [0, 0, 1, 0, 0]$,

  $a * e = a$
- Differentiation: $\frac{\partial}{\partial x}(f * g) = \frac{\partial f}{\partial x} * g$

Spatial Correlation and Convolution

- Spatial correlation and convolution are two mathematical operations that are commonly used in image processing
- **Spatial Correlation** involves computing the dot product of the values in a kernel (filter) with the corresponding values in the input image. The kernel slides or moves across the image, and at each position, the dot product is computed.
  - Spatial correlation is typically used to find the similarity between two signals or images.

$$g(x,y)=\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)f(x+s,y+t)$$

- **Spatial Convolution** is similar to spatial correlation but involves flipping the kernel horizontally and vertically before applying it to the image.
  - Convolution is typically used to filter signals or images.

$$g(x,y)=\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)f(x-s,y-t)$$

Spatial Correlation and Convolution

- Some examples of how spatial correlation and convolution are used in practice
- **Spatial correlation:**
  - **Image matching**: Spatial correlation can be used to find the best match between a template image and a larger image. This is useful for applications such as object tracking and image registration.
  - **Motion tracking**: Spatial correlation can be used to track the movement of objects in a video sequence. This is useful for applications such as video surveillance and human-computer interaction.
- **Spatial Convolution:**
  - **Image denoising**: Convolution can be used to remove noise from images. This is useful for improving the quality of images that have been taken in low light conditions.
  - **Image sharpening**: Convolution can be used to sharpen the edges of images. This is useful for improving the clarity of images that have been blurred.
  - **Edge detection**: Convolution can be used to detect edges in images. This is useful for applications such as image segmentation and object recognition.
  - **Feature extraction**: Convolution can be used to extract features from images. This is useful for applications such as image classification and object tracking.

Spatial Correlation and Convolution

- In convolution, the mask is flipped vertically and horizontally
- Zero padding is done in both directions by m-1 and n-1

f

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

Zero padding by m-1 and n-1 →

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

w

1	2	3
4	5	6
7	8	9

Mask rotation →

9	8	7
6	5	4
3	2	1

Spatial Correlation and Convolution

1	2	3	0	0	0	0	0	0
4	5	6	0	0	0	0	0	0
7	8	9	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Initial position in correlation

9	8	7	0	0	0	0	0	0
6	5	4	0	0	0	0	0	0
3	2	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Initial position in convolution

Spatial Correlation and Convolution

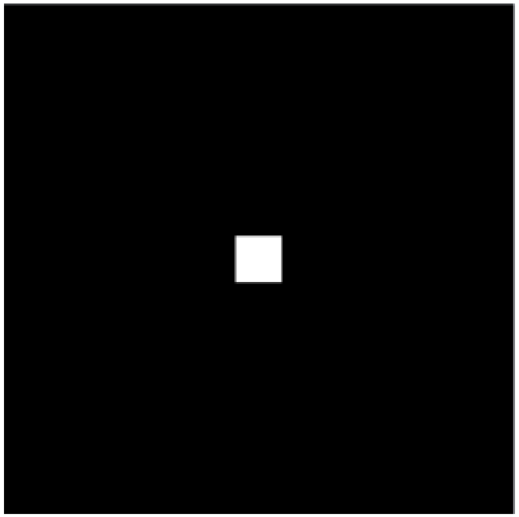
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	9	8	7	0	0	0
0	0	0	6	5	4	0	0	0
0	0	0	3	2	1	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

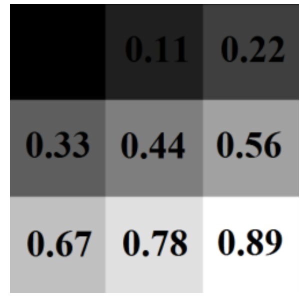
Full correlation result

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	1	2	3	0	0	0
0	0	0	4	5	6	0	0	0
0	0	0	7	8	9	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

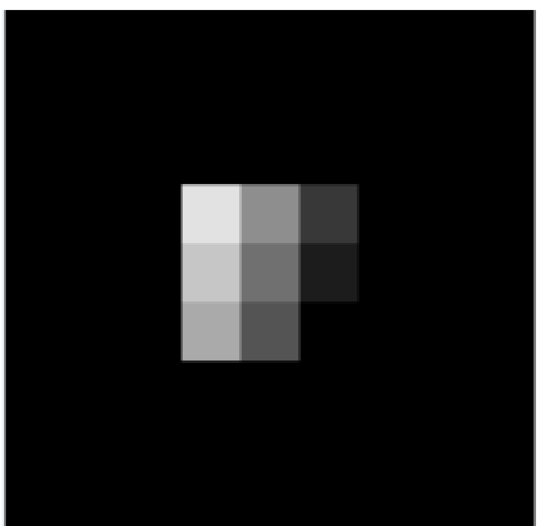
Full convolution result

Correlation vs Convolution

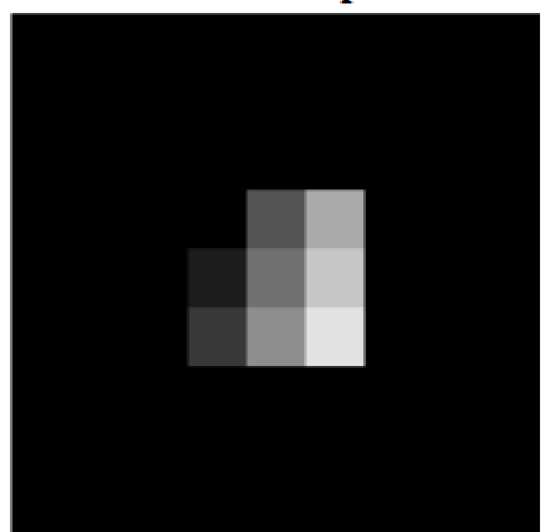




Cross-Correlation Output



Convolution Output



Outline

- What is Image Enhancement?
- Contrast Enhancement
  - Intensity Transformation Functions
  - Histogram Processing
- Spatial Filtering
  - Basic Concepts
  - **Smoothing Filters**
  - Sharping Filters
  - Nonlinear Filters
- Image Quality Assessment
  - Subjective image quality assessment
  - Objective image quality assessment

Smoothing Spatial Filters

- A smoothing (averaging, blurring) filter replaces each pixel with the average value of all pixels under the mask
- Used for blurring and for noise reduction
  - Blurring is used in preprocessing steps, such as
    - Removal of small details from an image prior to object extraction
    - Bridging of small gaps in lines or curves
  - Noise reduction can be accomplished by blurring (noise as it is characterized with sharp transitions
- Replacing the value of every pixel in an image by the average of the gray levels in the neighborhood will reduce the "sharp" transitions in gray levels.
  - sharp transitions
    - random noise in the image
    - edges of objects in the image
- Thus, smoothing can reduce noises (desirable) and blur edges (undesirable)

Smoothing by Box Filter Kernels

- The simplest lowpass filter kernel is the *box kernel*, whose coefficients have the same value (typically 1).
- An $m \times n$ box filter is an $m \times n$ array of 1's, with a normalizing constant in front, whose value is 1 divided by the sum of the values of the coefficients (i.e., *1/mn* when all the coefficients are 1's).

$$g(x,y) = \frac{\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)\, f(x+s, y+t)}{\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)}$$

- This normalization, which we apply to all lowpass kernels, has two purposes.
  - First, the average value of an area of constant intensity would equal that intensity in the filtered image, as it should.
  - Second, normalizing the kernel in this way prevents introducing a *bias* during filtering; that is, the sum of the pixels in the original and filtered images will be the same.

Smoothing by Box Filter: Example

$h[\cdot\,,\cdot\,]\;\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$g[.,.]$

$$g[m,n]=\sum_{k,l} h[k,l]\; f[m+k,n+l]$$

Smoothing by Box Filter: Example

$h[\cdot,\cdot] \quad \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$g[.,.]$

	0	10							

$$g[m,n] = \sum_{k,l} h[k,l]\, f[m+k, n+l]$$

Smoothing by Box Filter: Example

$h[\cdot\,,\cdot\,]\ \frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$g[.,.]$

	0	10	20						

$$g[m,n] = \sum_{k,l} h[k,l]\; f[m+k, n+l]$$

Smoothing by Box Filter: Example

$h[\cdot\,,\cdot\,]\ \frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$g[.,.]$

	0	10	20	30					

$$g[m,n] = \sum_{k,l} h[k,l]\ f[m+k, n+l]$$

Smoothing by Box Filter: Example

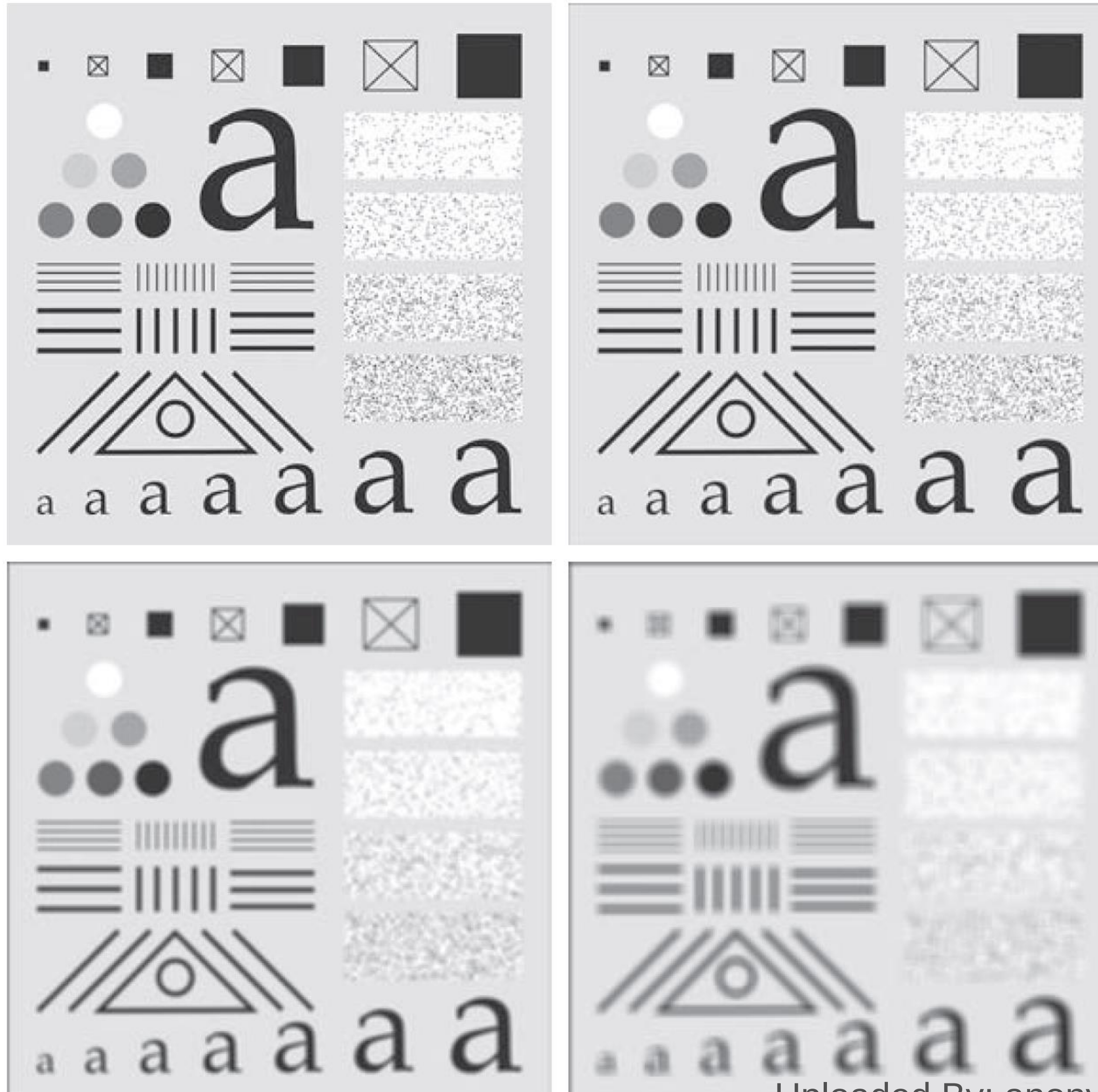
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

Example

a b
c d

FIGURE 3.33
(a) Test pattern of size 1024×1024 pixels.
(b)-(d) Results of lowpass filtering with box kernels of sizes 3×3, 11×11, and 21×21, respectively.



Notice how detail begins to disappear

Box Filters - Summary

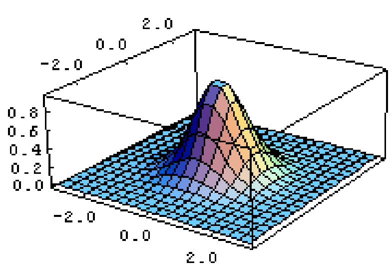
- Because of their simplicity, box filters are suitable for quick experimentation and they often yield smoothing results that are visually acceptable.
- They are useful also when it is desired to reduce the effect of smoothing on edges.
- However, box filters have limitations that make them poor choices in many applications.
  - Box filters favor blurring along perpendicular directions.
  - In applications involving images with a high level of detail, or with strong geometrical components, the directionality of box filters often produces undesirable results.

Smoothing using Gaussian Filter

- A Gaussian kernel gives less weight to pixels further from the center of the window
- Filtering with a m x m mask
  - The weights are computed according to a Gaussian function:

$$h(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{\sigma^2}}$$

  - Where σ is the standard deviation of the distribution. The distribution is assumed to have a mean of 0.

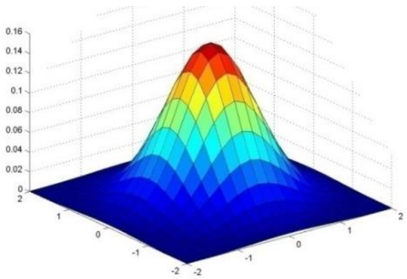


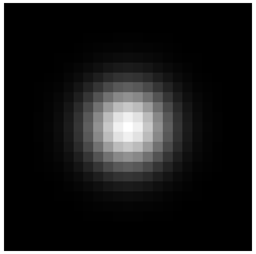
$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$H[u,v]$

Gaussian Filtering

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

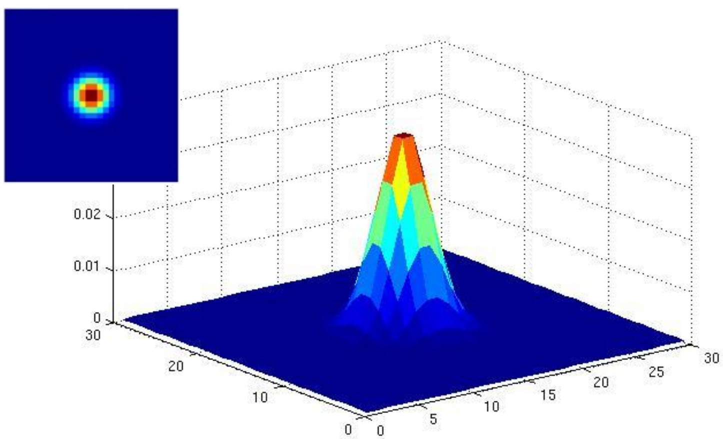




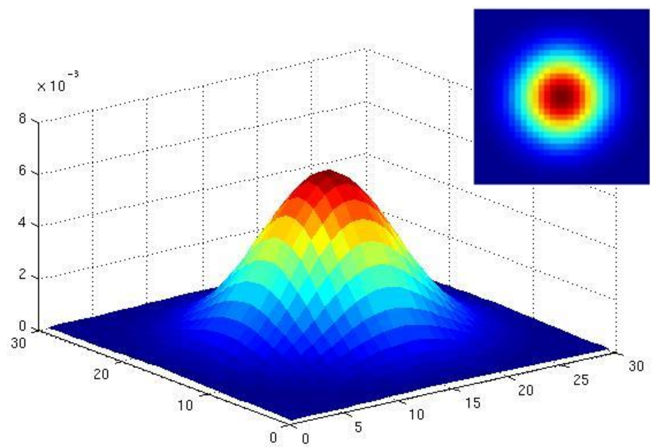
0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

5 x 5, $\sigma = 1$

- Parameter σ is the "scale" / "width" / "spread" of the Gaussian kernel, and controls the amount of smoothing.



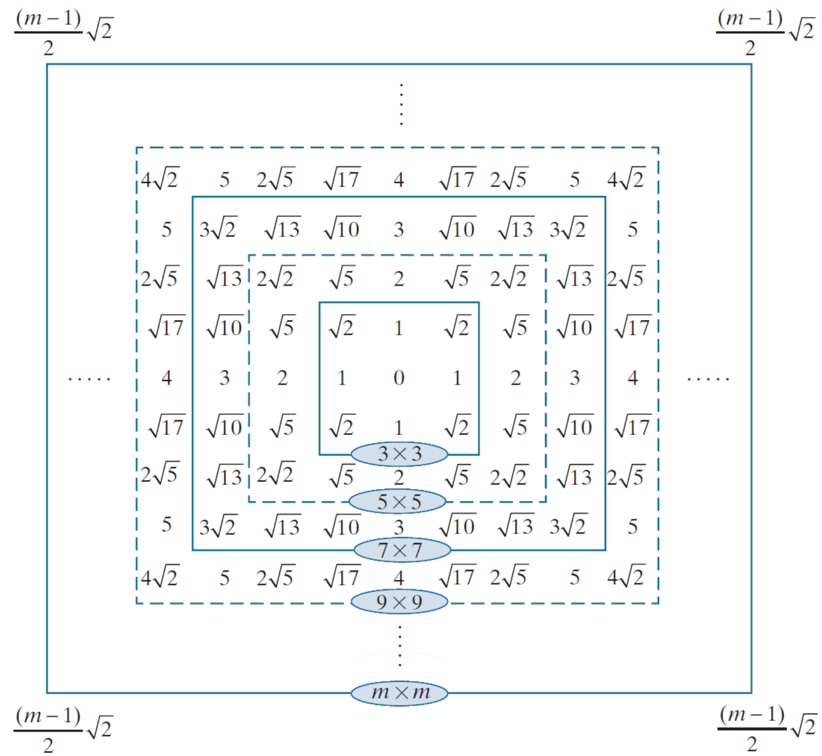
$\sigma = 2$ with 30 x 30 kernel



$\sigma = 5$ with 30 x 30 kernel

Gaussian kernels

- The kernel was obtained by sampling the Gaussian.
- Small Gaussian kernels cannot capture the characteristic Gaussian bell shape, and thus behave more like box kernels.
- Gaussian function at a distance larger than 3**σ** from the mean are small enough that they can be ignored.
- This property tells us that there is nothing to be gained by using a Gaussian kernel larger than **6σ** * **6σ** for image processing.
- Because typically we work with kernels of odd dimensions, we would use the smallest *odd* integer that satisfies this condition

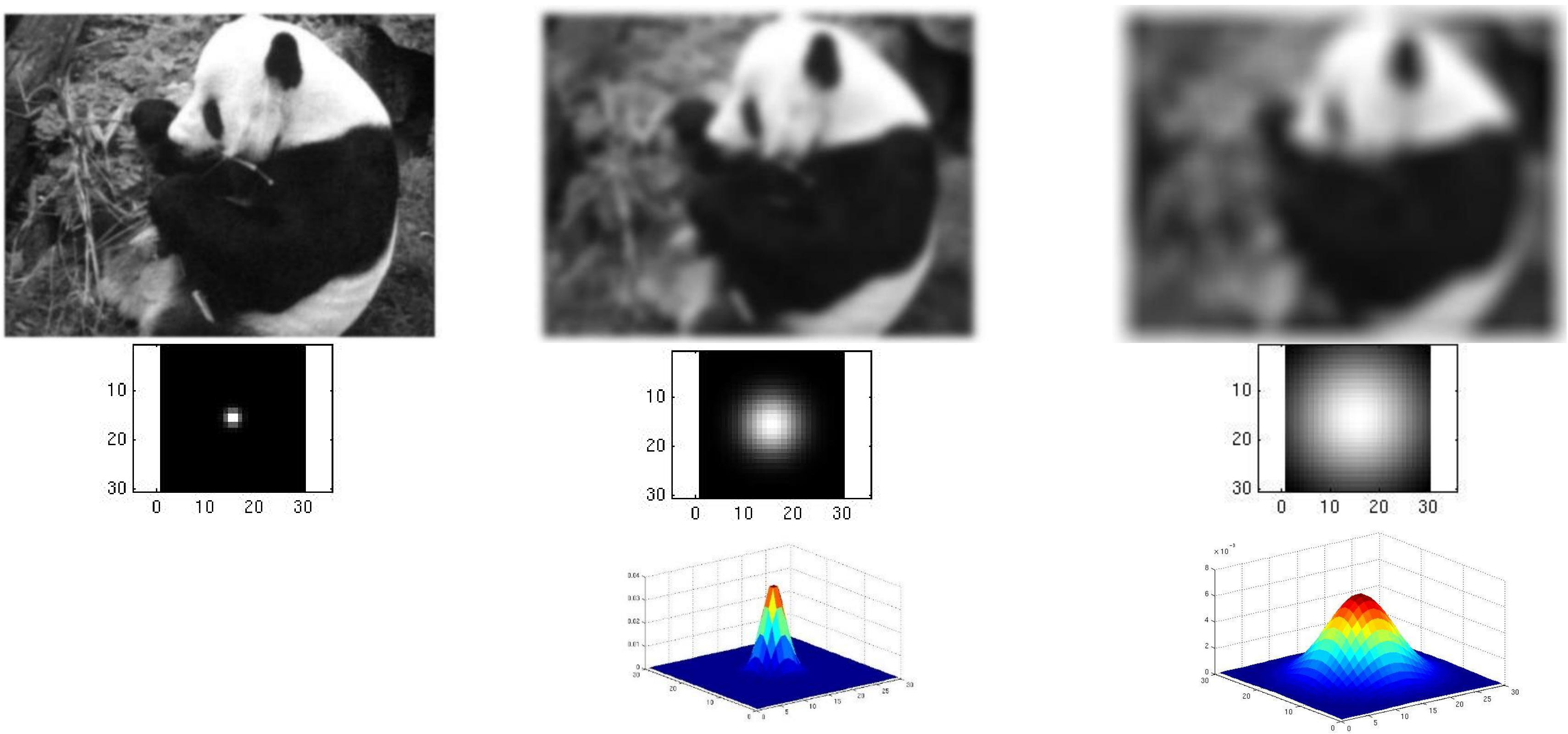


Properties of Gaussian Filtering

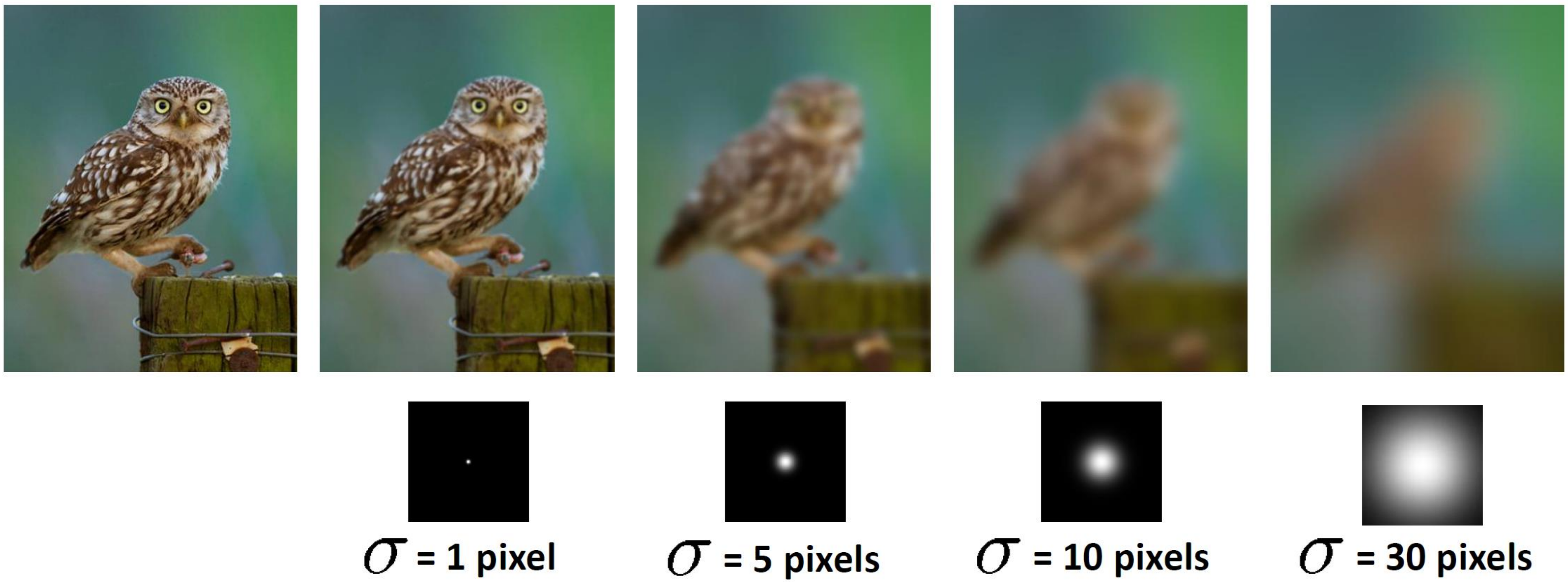
- Gaussian smoothing is very effective for removing Gaussian noise
- The weights give higher significance to pixels near the edge (reduces edge blurring)
- They are linear low pass filters
- Computationally efficient (large filters are implemented using small 1D filters)
- Rotationally symmetric (perform the same in all directions)
- The degree of smoothing is controlled by σ (larger σ for more intensive smoothing)

Smoothing with a Gaussian: Example

- Parameter σ is the "scale" / "width" / "spread" of the Gaussian kernel, and controls the amount of smoothing.



Gaussian Smoothing Filter - Effect of σ

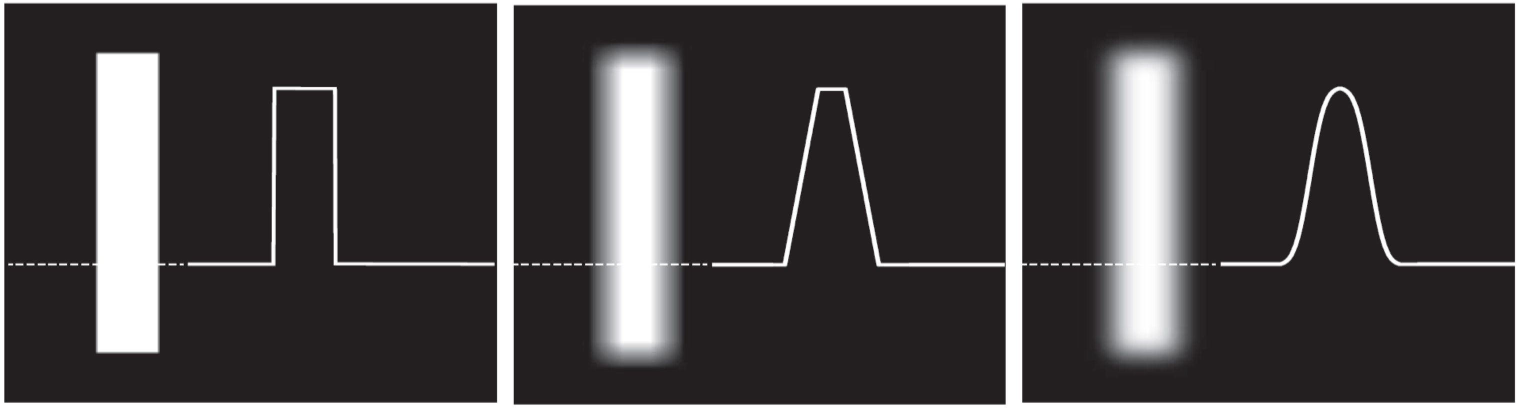


Gaussian vs Box Filter

Aspect	Box Filter	Gaussian Filter
Filtering Operation	Averages pixel values in a local neighborhood.	Applies a weighted average using a Gaussian function.
Smoothing/Blurring	Simple and uniform blur.	Controlled and smoother blur.
Edge Preservation	Preserves edges to some extent.	Better at preserving edges due to weighted averaging.
Computational Complexity	Computationally simpler.	More complex calculations due to Gaussian function.
Flexibility	Limited flexibility (fixed kernel size).	More flexible due to adjustable standard deviation.
Artifact Generation	Can introduce artifacts, e.g., ring artifacts.	Generally produces fewer artifacts.
Applications	Common for basic blurring and noise reduction.	Widely used for smoothing, blurring, and advanced filtering.

Gaussian vs Box Filter

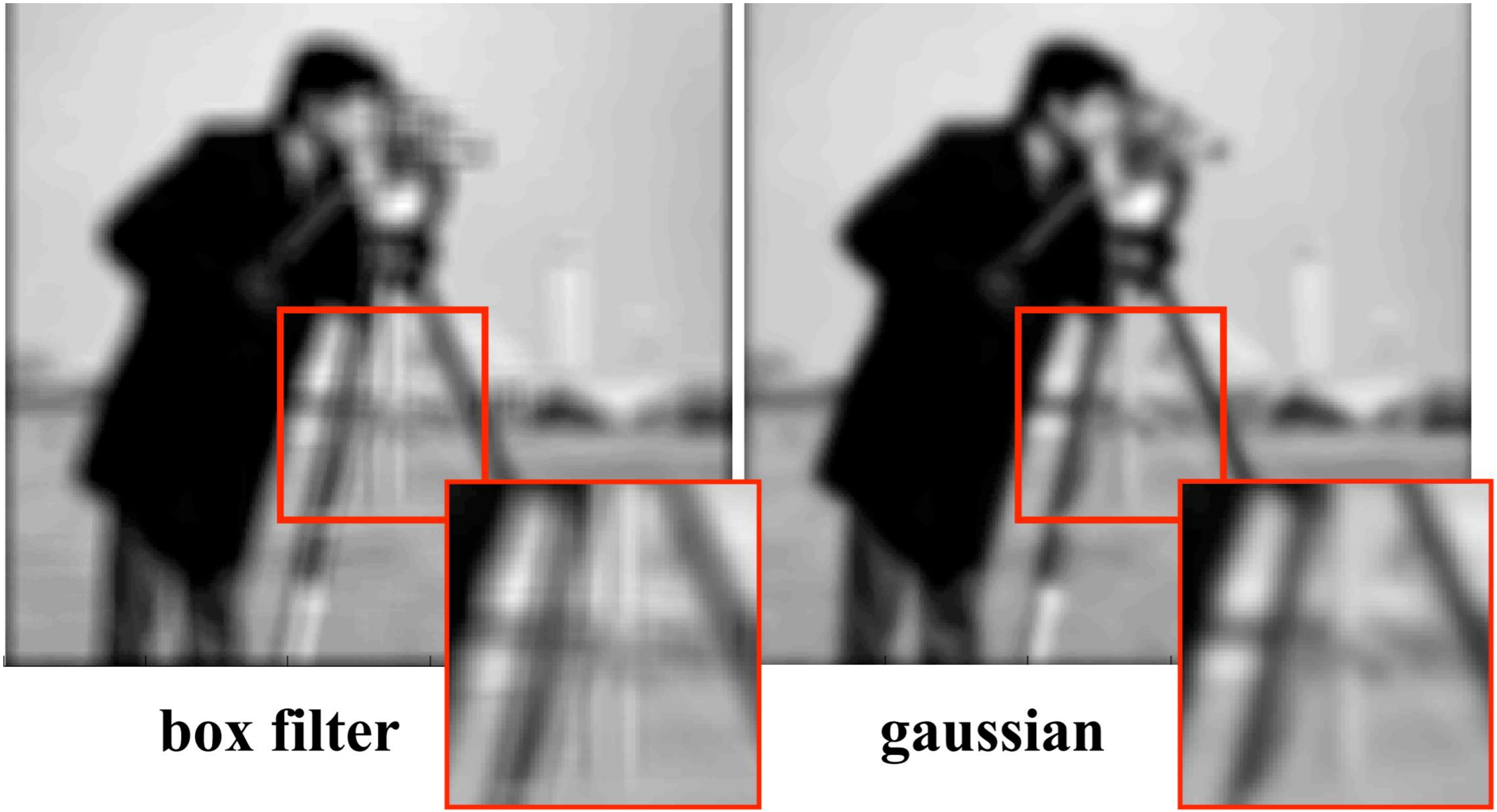
- As the intensity profiles show, the box filter produced linear smoothing, with the transition from black to white (i.e., at an edge) having the shape of a ramp. The important features here are hard transitions at the onset and end of the ramp. We would use this type of filter when less smoothing of edges is desired.
- Conversely, the Gaussian filter yielded significantly smoother results around the edge transitions. We would use this type of filter when generally uniform smoothing is desired.



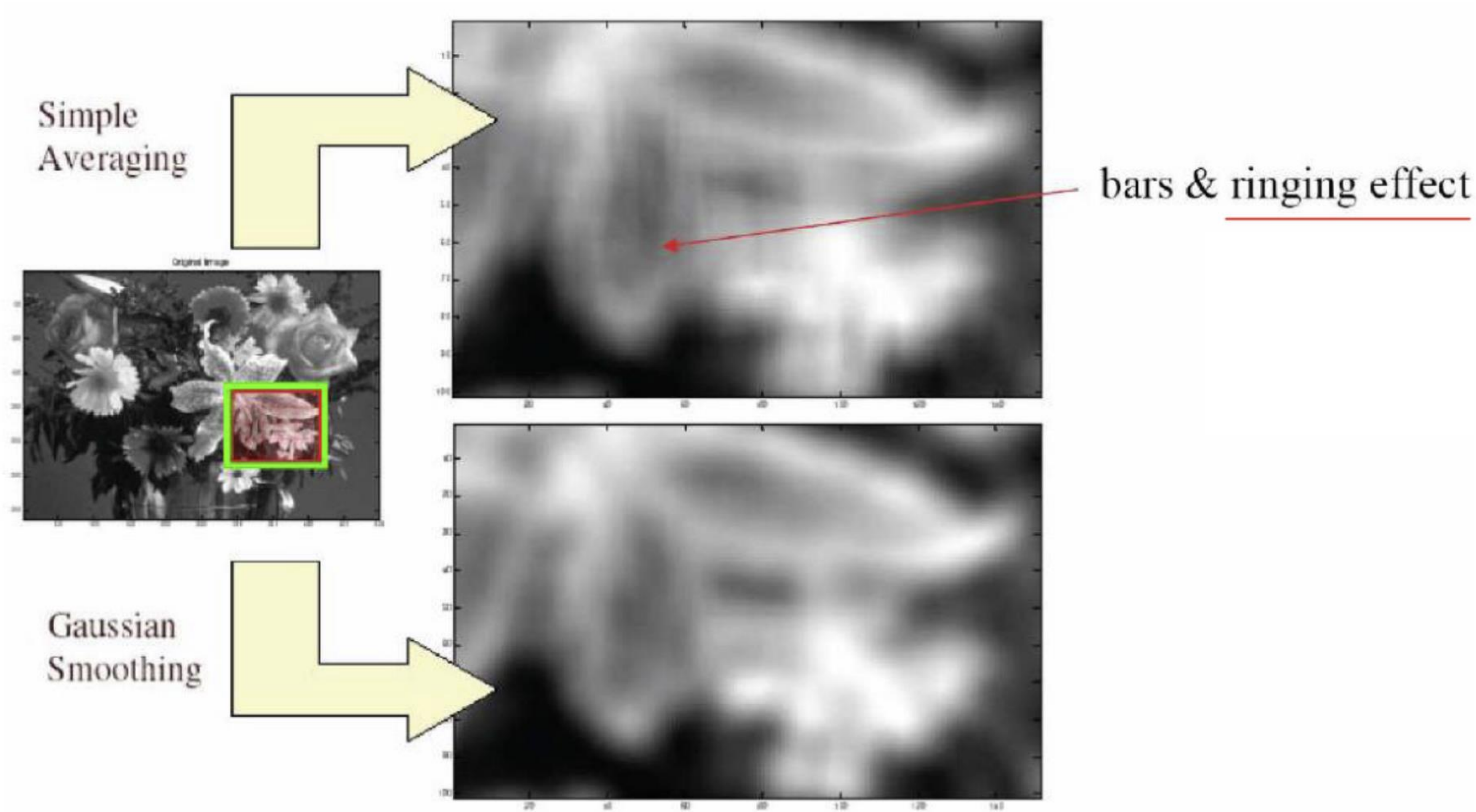
a b c

FIGURE 3.38 (a) Image of a white rectangle on a black background, and a horizontal intensity profile along the scan line shown dotted. (b) Result of smoothing this image with a box kernel of size 71×71, and corresponding intensity profile. (c) Result of smoothing the image using a Gaussian kernel of size 151×151, with $K = 1$ and $\sigma = 25$. Note the smoothness of the profile in (c) compared to (b). The image and rectangle are of sizes 1024×1024 and 768×128 pixels, respectively.

Gaussian vs. Box Filter

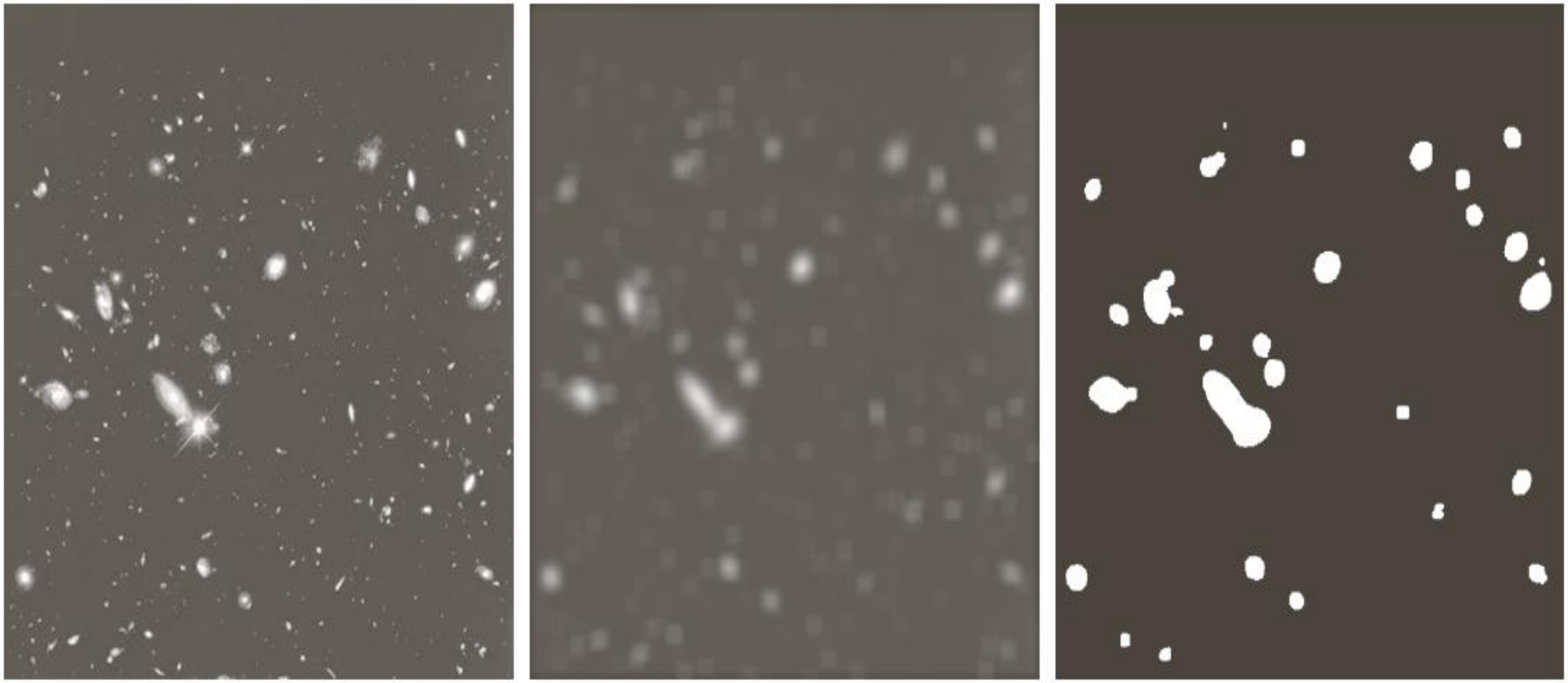


Gaussian vs Box Filter



Smoothing Spatial Filters - Applications

- Smoothing highlights gross details. Could be useful in providing better segmentation results

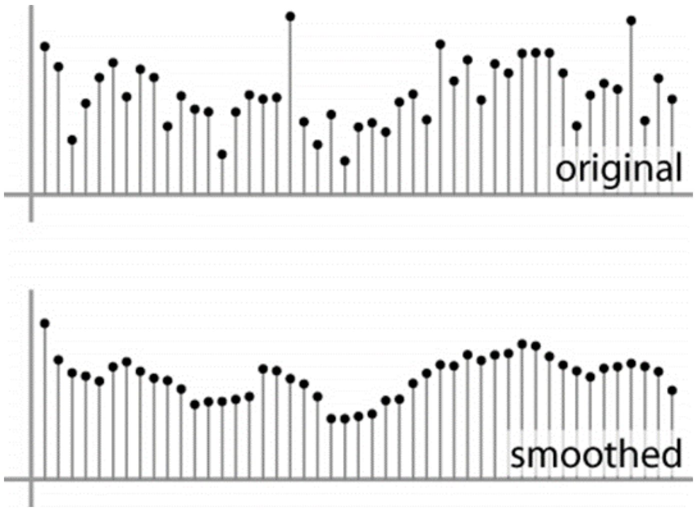


a b c

FIGURE 3.34 (a) Image of size 528 × 485 pixels from the Hubble Space Telescope. (b) Image filtered with a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

Smoothing Reduces Noise

- Why Averaging Reduces Noise
  - Intuitive explanation: variance of noise in the average is smaller than variance of the pixel noise (assuming zero-mean Gaussian noise).



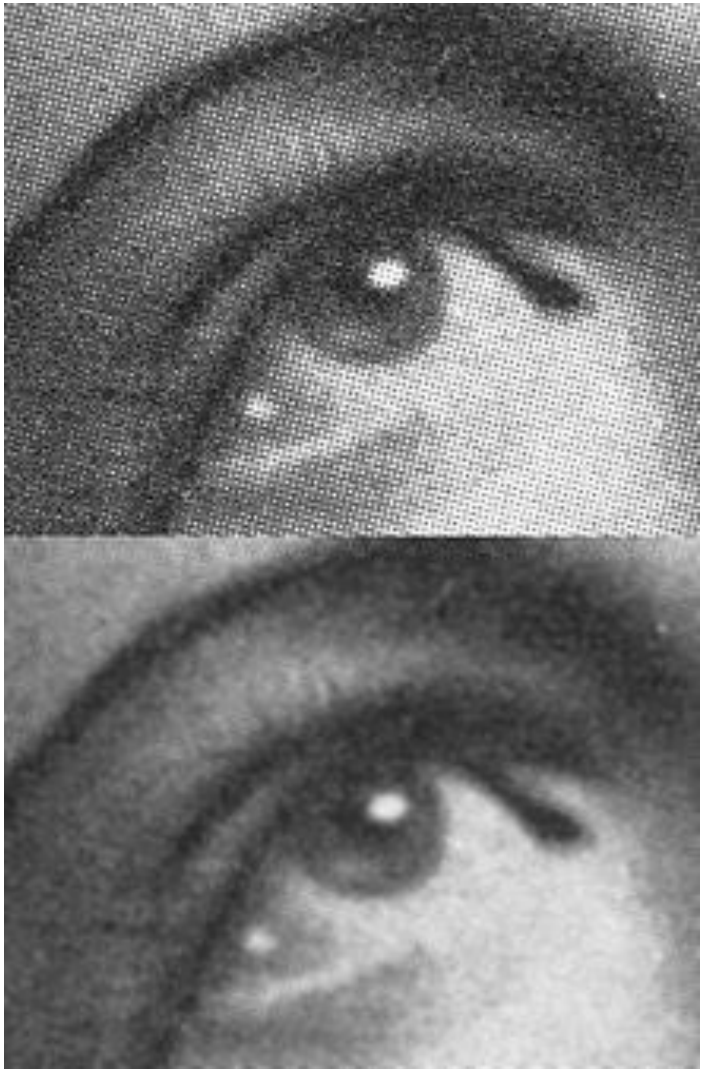


Image Noise

- Noise in image , is any degradation in an image signal , caused by external disturbance while an image is being sent from one place to another place via satellite , wireless and network cable .
- Noise sources
  - Light Variations
  - Camera Electronics
  - Surface Reflectance
  - Lens
- Let *f(i,j)* be the true pixel values and *n(i,j)* be the noise added to the pixel *(i,j)*
  - *Additive noise:* $\hat{f}(i,j) = f(i,j) + n(i,j)$
  - *Multiplicative noise:* $\hat{f}(i,j) = f(i,j) * n(i,j)$
- Averaging filters is useful for noise reduction.

Noise removing using average filters

- On the left is an image containing a significant amount of salt and pepper noise. On the right is the same image after processing with an average filter.

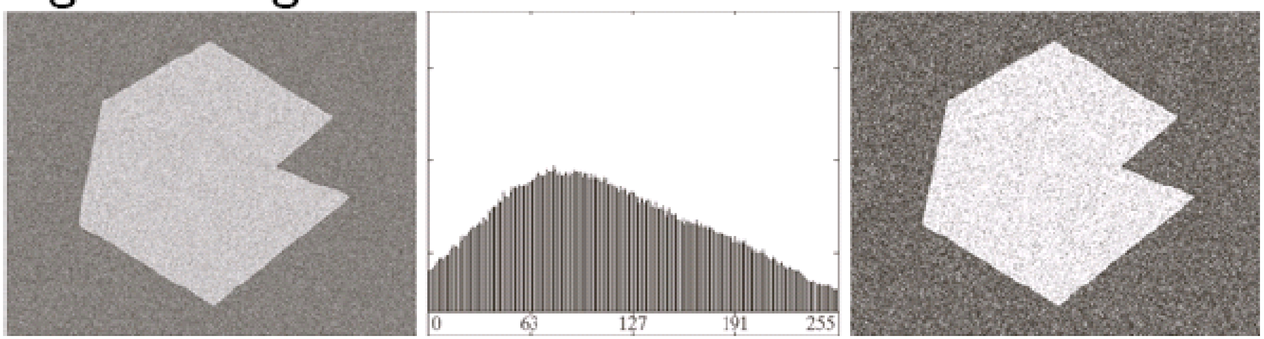
Noise Elimination

After Averaging

After Gaussian Smoothing

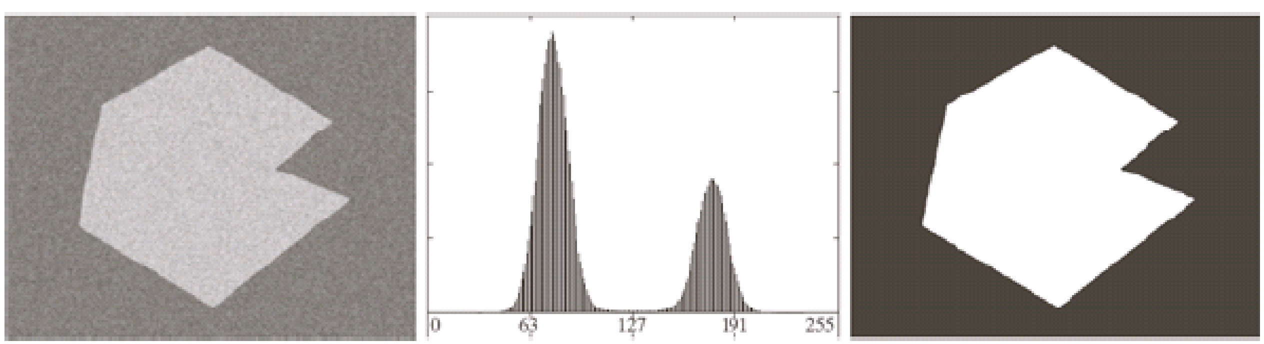
Enhancing Global Thresholding by Smoothing

- It was noted earlier that the presence of noise highly affect the result of thresholding
- One approach to improve thresholding result is to smooth the original image



Segmentation Fails

Noisy image, its histogram, and Otsu's segmentation result



Noisy image after smoothing by 5x5mask, its histogram, and Otsu's segmentation result

Notes on Smoothing Spatial Filters

- The weighted average filter gives more weight to pixels near the center
- The basic strategy behind weighting the center point the highest and then reducing the value of the coefficients as a function of increasing distance from the origin is simply an attempt to reduce blurring in the smoothing process.
- Averaging attenuates noise (reduces the variance), leading to a more "accurate" estimate.
- However, the more accurate estimate is of the mean of a local pixel neighborhood! This might not be what you want.
- Balancing act: smooth enough to "clean up" the noise, but not so much as to remove important image gradients.

Common Problems

- Mean Filters: blurs image, removes simple noise, no details are preserved
- Gaussian Filters: blurs image, preserves details only for small σ.
- Can we find a filter that not only smooths regions but preserves edges?
  - **Adaptive Filters**
  - **Bilateral Filter**
  - **Winner Filters**
  - **Anisotropic Diffusion Filtering**

Smoothing Color Images

1. Per-color-plane method: for RGB, CMY color models smooth each color plane using moving averaging and the combine back to RGB

$$\bar{\mathbf{c}}(x,y) = \frac{1}{K}\sum_{(x,y)\in S_{xy}} \mathbf{c}(x,y) = \begin{bmatrix} \frac{1}{K}\sum_{(x,y)\in S_{xy}} R(x,y) \\ \frac{1}{K}\sum_{(x,y)\in S_{xy}} G(x,y) \\ \frac{1}{K}\sum_{(x,y)\in S_{xy}} B(x,y) \end{bmatrix}$$

2. Smooth only Intensity component of a HSI image while leaving H and S unmodified.

Color Image Smoothing Example (cont.)

Color image

Red

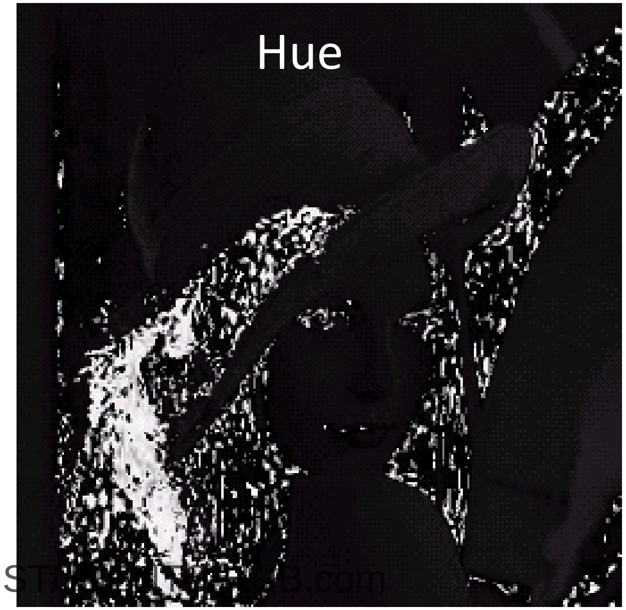
Green

Blue

Color Image Smoothing Example (cont.)

Color image

HSI Components



Color Image Smoothing Example (cont.)

Smooth all RGB components

Smooth only I component of HSI

(faster)

Color Image Smoothing Example (cont.)

Difference between smoothed results from 2 methods in the previous slide.

Outline

- What is Image Enhancement?
- Contrast Enhancement
  - Intensity Transformation Functions
  - Histogram Processing
- Spatial Filtering
  - Basic Concepts
  - Smoothing Filters
  - **Sharping Filters**
  - Nonlinear Filters
- Image Quality Assessment
  - Subjective image quality assessment
  - Objective image quality assessment

Sharpening Spatial Filters

- In addition to noise removal, the other two main uses of image filtering are for (a) feature extraction and (b) feature enhancement.
- The principle objective of sharpening is to highlight transitions in intensity which usually correspond to edges in images; thus sharpening is the opposite of smoothing
- If we examine the smoothing operation we can think of it as integration
- Thus to perform sharpening in the spatial domain, it is intuitive to use differentiation
- The strength of the response of a derivative operator is proportional to the degree of discontinuity of the image at the point at which the operator is applied.
- Thus, image differentiation
  - Enhances edges and other discontinuities (noise)
  - Deemphasizes area with slowly varying gray-level values.

Sharpening Spatial Filters

- Derivatives can be approximated as differences
- 1st derivative at x

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

- 2nd derivative at x

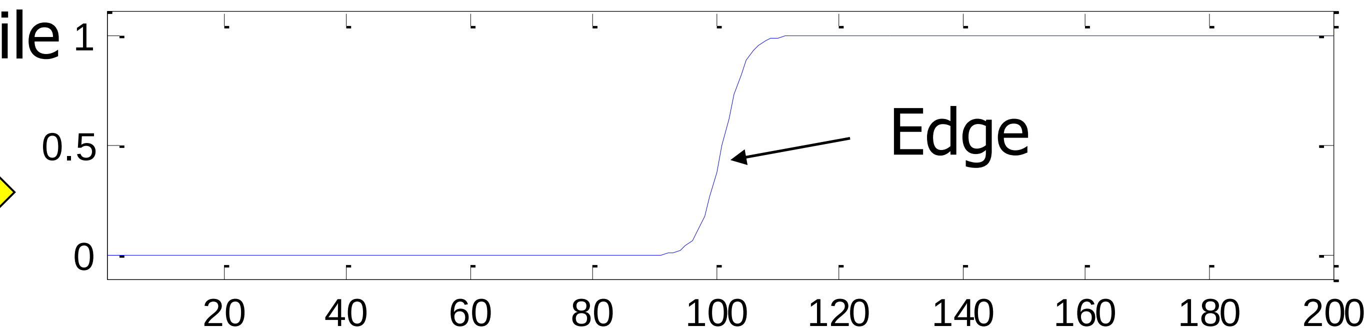
$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

Sharpening Spatial Filters

Investigation of derivatives behavior

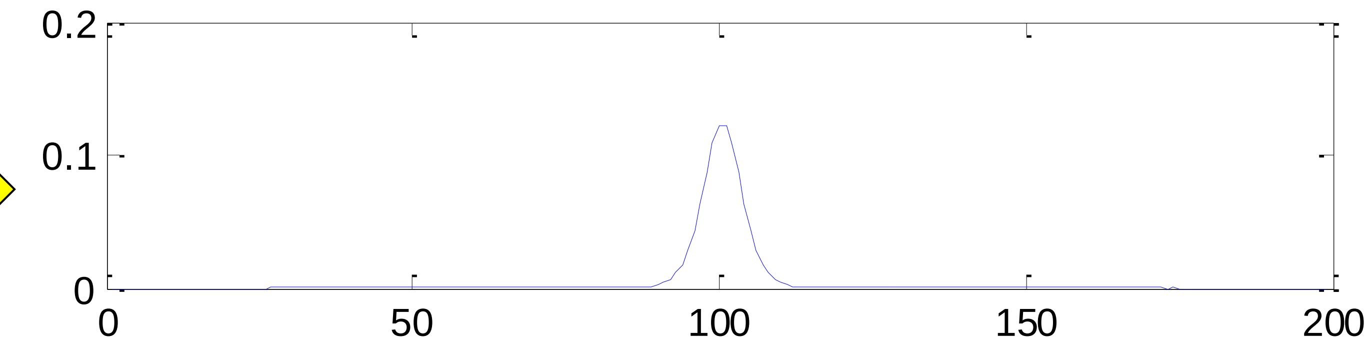
Intensity profile

$p(x)$



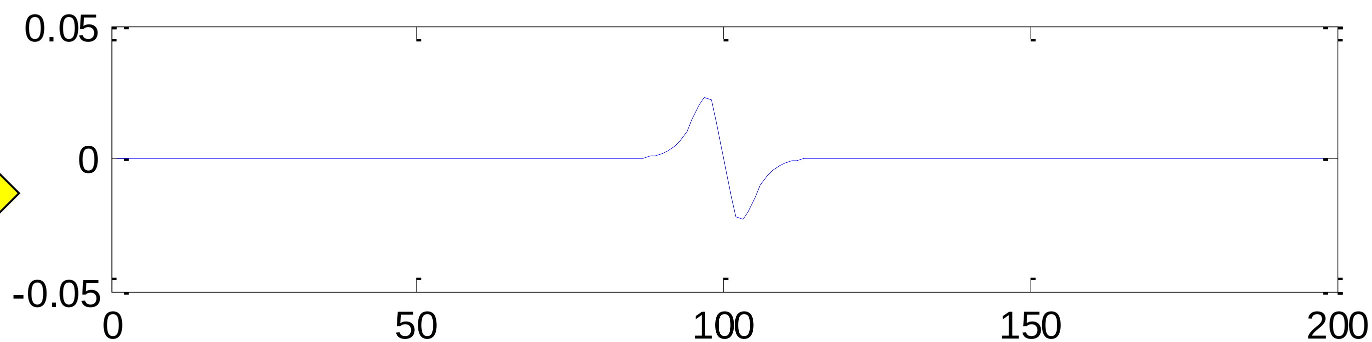
1st derivative

$\frac{dp}{dx}$



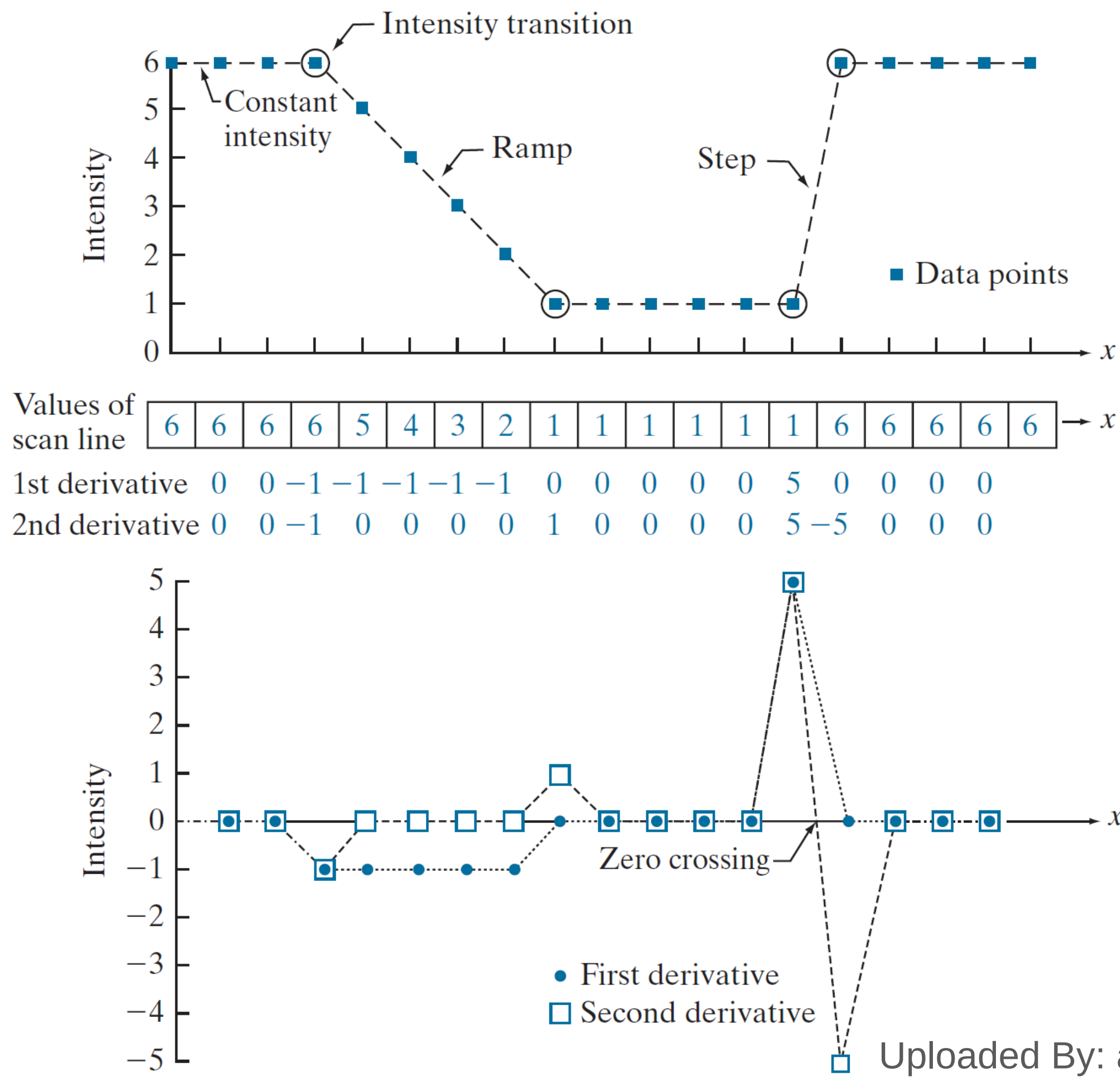
2nd derivative

$\frac{d^2p}{dx^2}$



Sharpening Spatial Filters

- Investigation of derivatives behavior



a
b
c

FIGURE 3.44
(a) A section of a horizontal scan line from an image, showing ramp and step edges, as well as constant segments.
(b)Values of the scan line and its derivatives.
(c) Plot of the derivatives, showing a zero crossing. In (a) and (c) points were joined by dashed lines as a visual aid.

Sharpening Spatial Filters

- We are concerned about the behavior of 1st and 2nd derivatives in the following areas
  - Constant intensity
  - Onset and end of discontinuities (ramps and steps)
  - Intensity ramps
- Properties of 1st derivative
  - Zero in areas of constant intensity
  - Nonzero at the onset of a step and intensity ramp
  - Nonzero along intensity ramp
- Properties 2nd derivative
  - Zero in areas of constant intensity
  - Nonzero at the onset and end of a step and intensity ramp
  - Zero along intensity ramp

Sharpening Spatial Filters

- **Notes**
  - Examining the 1st and 2nd derivatives plots shows that all of their properties are satisfied
  - 1st derivative produce thicker edges than 2nd derivatives
  - 2nd derivative produce double edge separated by a zero crossing
  - 2nd derivative is commonly used in sharpening since:
    - 2nd derivative is more sensitive to intensity variations than a first order operator.
    - It has stronger response to fine details, such as thin lines and isolated points
    - 2nd derivative or the Laplacian has a response that is zero for constant regions or slowly varying intensity regions. It tends to suppress low-frequency components, which are often associated with smooth areas in an image. This characteristic helps in isolating and enhancing the details and edges.

Sharpening Using 2nd Derivative

- When we consider an image function of two variables, f(x,y), at which time we will dealing with partial derivatives along the two spatial axes.
- The second derivative (Laplacian) in 2-D is defined as

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

- If we define

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

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

- Then

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

Sharpening Using 2nd Derivative

- The Laplacian can be implemented as a filter mask

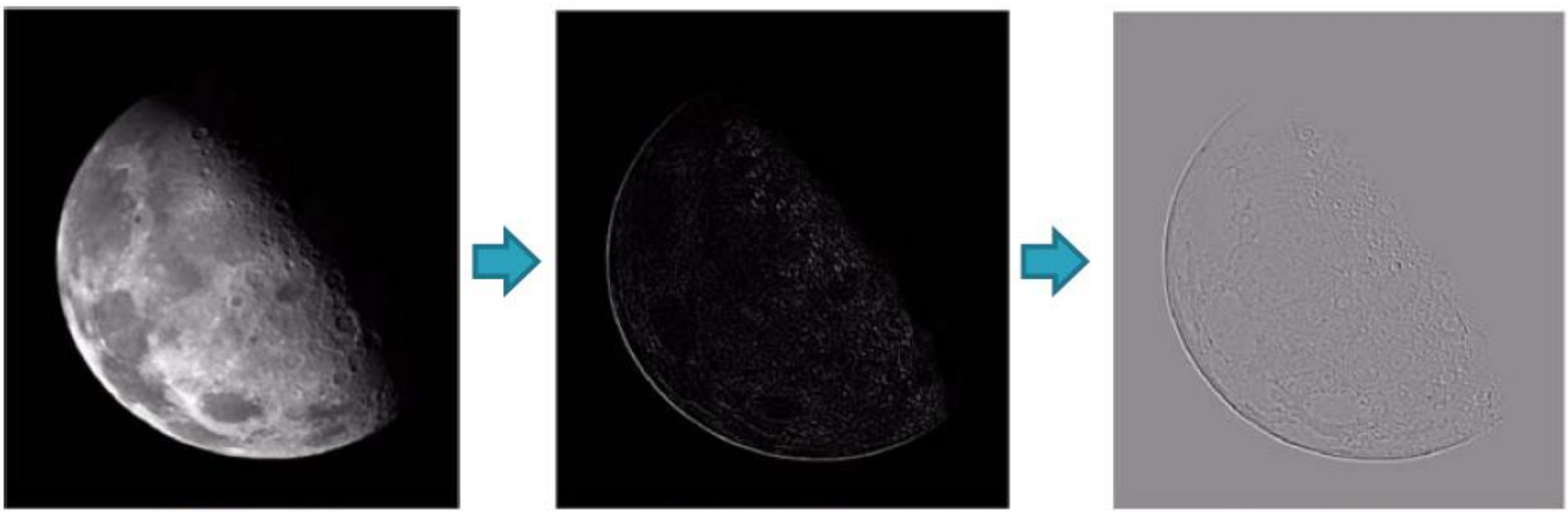
0	1	0
1	-4	1
0	1	0

- Or

1	1	1
1	-8	1
1	1	1

Sharpening Using 2nd Derivative

- Computing the Laplacian doesn’t produce a sharpened image. However, grayish edge lines and discontinuities superimposed on a dark background
  - It is common practice to scale the Laplacian image to [0,255] for better display



Original Image

Laplacian Filtered Image

Laplacian Filtered Image Scaled for Display

Sharpening Using 2nd Derivative

- Because the Laplacian is a derivative operator, it highlights sharp intensity transitions in an image and de-emphasizes regions of slowly varying intensities. This will tend to produce images that have grayish edge lines and other discontinuities, all superimposed on a dark, featureless background.
- Thus, the basic way in which we use the Laplacian for image sharpening is:

$$g(x, y) = f(x, y) + c\left[\nabla^2 f(x, y)\right]$$

where $f(x, y)$ and $g(x, y)$ are the input and sharpened images, respectively. We let $c = -1$ if the Laplacian kernels center is negative and 1 if the Laplacian kernels center is positive.

Sharpening Using 2nd Derivative

- The two steps required to achieve sharpening can be combined into a single filtering operation

$$
\begin{aligned}
g(x,y) = f(x,y) - [\, & f(x+1,y) + f(x-1,y) \\
& + f(x,y+1) + f(x,y-1) + 4f(x,y)] \\
= 5f(x,y) - [\, & f(x+1,y) + f(x-1,y) \\
& + f(x,y+1) + f(x,y-1)]
\end{aligned}
$$

0	-1	0
-1	5	-1
0	-1	0

Sharpening Using 2nd Derivative

- Example

a	b
c	d

FIGURE 3.46 (a) Blurred image of the North Pole of the moon. (b) Laplacian image obtained using the kernel in Fig. 3.45(a). (c) Image sharpened using Eq. (3-54) with $c = -1$. (d) Image sharpened using the same procedure, but with the kernel in Fig. 3.45(b). (Original image courtesy of NASA.)

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

Unsharp Masking and High-Boost Filtering

- The technique known as *unsharp masking* is a method of common use in graphics for making the images sharper.
- It consists of:
  1. defocusing the original image;
  2. obtaining the mask as the difference between the original image and its defocused copy;
  3. adding the mask to the original image.
- The process can be formalized as:

$$g = f + k \cdot (f - f * h)$$

  where f is the original image, h is the smoothing filter and k is a constant for tuning the mask contribution.
  - If $k > 1$, the process is called *highboost* filtering.

Unsharp Masking and High-Boost Filtering

0	−1	0
−1	$A+4$	−1
0	−1	0

−1	−1	−1
−1	$A+8$	−1
−1	−1	−1

- If $A \geq 1$, High-Boost
- if $A = 1$, it becomes “standard” Laplacian sharpening

Unsharp Masking and High-Boost Filtering

FIGURE 3.55

(a) Unretouched "soft-tone" digital image of size 469×600 pixels. (b) Image blurred using a 31×31 Gaussian lowpass filter with $\sigma = 5$ (c) Mask. (d) Result of unsharp masking using Eq. (3-65) with $k = 1$. (e) and (f) Results of highboost filtering with $k = 2$ and $k = 3$, respectively.

Color Image Sharpening

We can do in the same manner as color image smoothing:

1. Per-color-plane method for RGB,CMY images
2. Sharpening only I component of a HSI image

Sharpening all RGB components

Sharpening only I component of HSI

Color Image Sharpening Example (cont.)

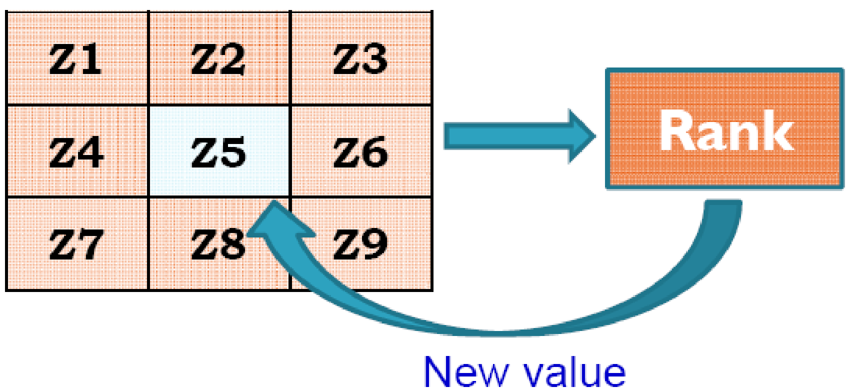
Difference between sharpened results from 2 methods in the previous slide.

Outline

- What is Image Enhancement?
- Contrast Enhancement
  - Intensity Transformation Functions
  - Histogram Processing
- Spatial Filtering
  - Basic Concepts
  - Smoothing Filters
  - Sharping Filters
  - **Nonlinear Filters**
- Image Quality Assessment
  - Subjective image quality assessment
  - Objective image quality assessment

Nonlinear (Order-statistic) Spatial Filters

- Order-statistic filters are nonlinear filters whose response is based on ordering the pixels under the mask and then replacing the centre pixel with the value determined by the ranking result



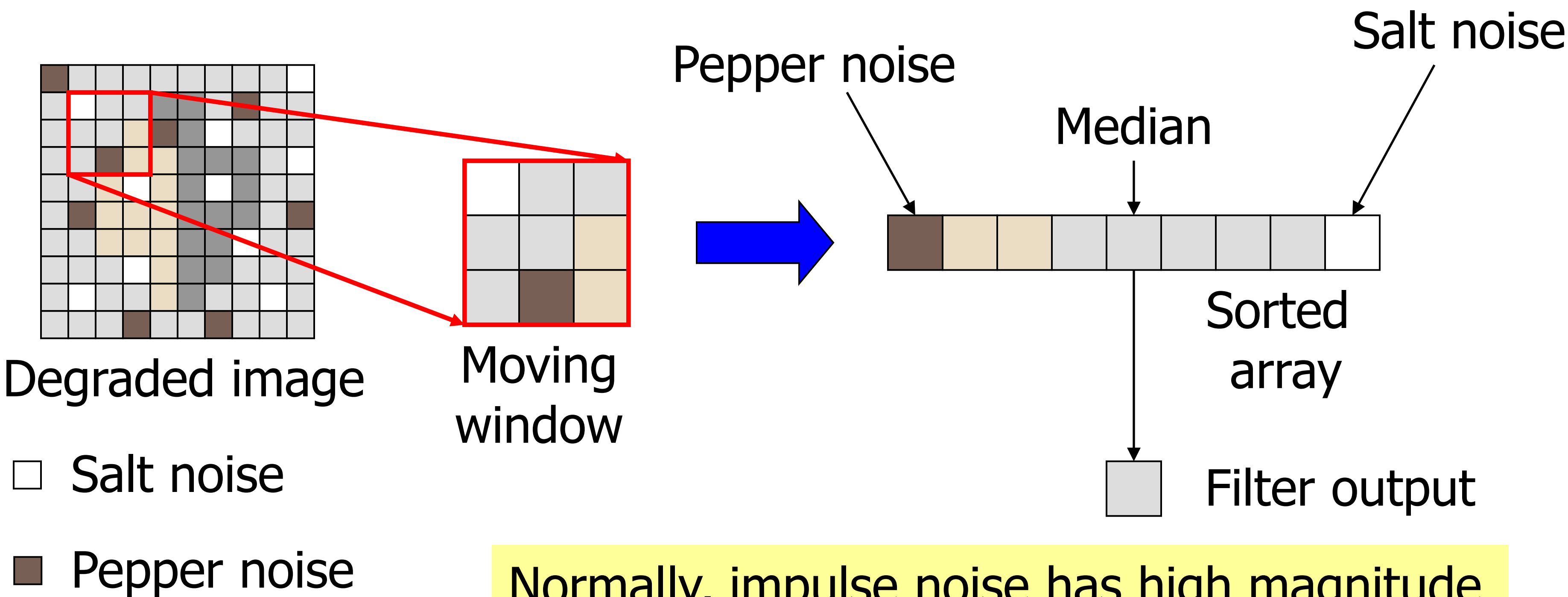
- Example
  - Median filter : R = median{z_k |k = 1,2,…,n x n}
  - Max filter : R = max{z_k |k = 1,2,…,n x n}
  - Min filter : R = min{z_k |k = 1,2,…,n x n}
    - note: n x n is the size of the mask

Median Filters

- Replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel
  - The original value of the pixel is included in the computation of the median)
- Quite popular because for certain types of random noise (impulse noise ⇨ salt and pepper noise) , they provide excellent noise-reduction capabilities, with considering less blurring than linear smoothing filters of similar size.
- Forces the points with distinct gray levels to be more like their neighbors.
- Isolated clusters of pixels that are light or dark with respect to their neighbors, and whose area is less than $n^2/2$ (one-half the filter area), are eliminated by an n x n median filter.
- Eliminated = forced to have the value equal the median intensity of the neighbors.
- Larger clusters are affected considerably less

Median Filters

A median filter is good for removing impulse, isolated noise

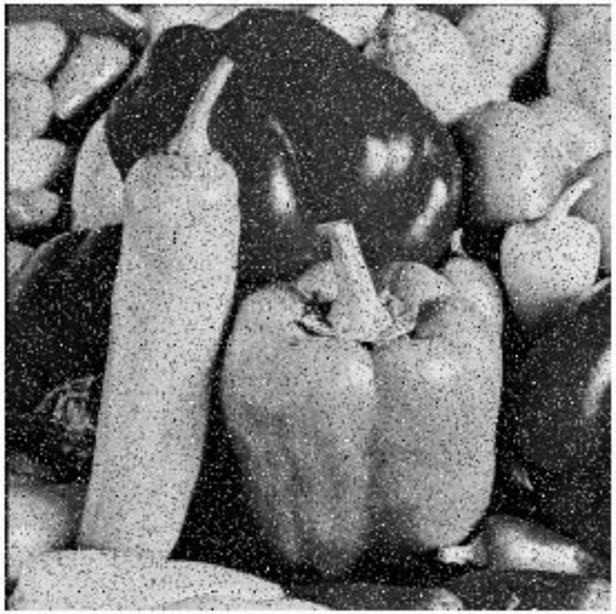


Normally, impulse noise has high magnitude and is isolated. When we sort pixels in the moving window, noise pixels are usually at the ends of the array.

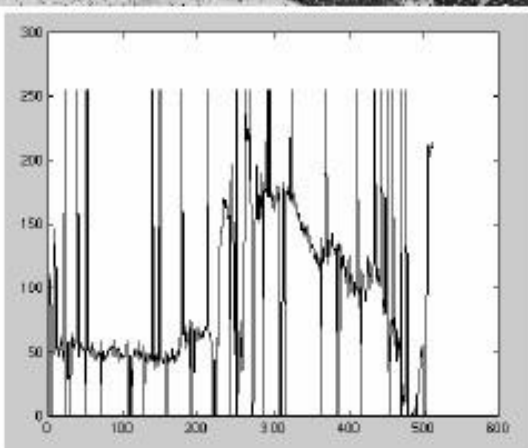
Therefore, it's rare that the noise pixel will be a median value.

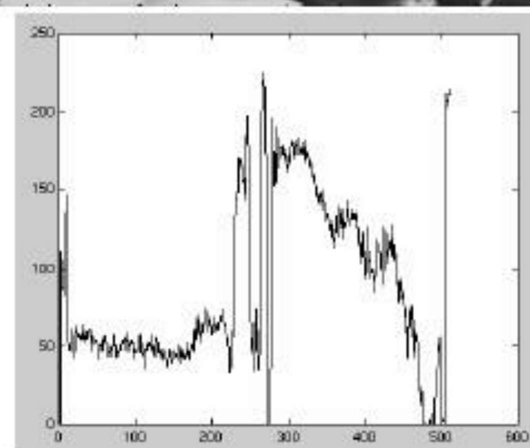
Median filters: Example

Salt and pepper noise



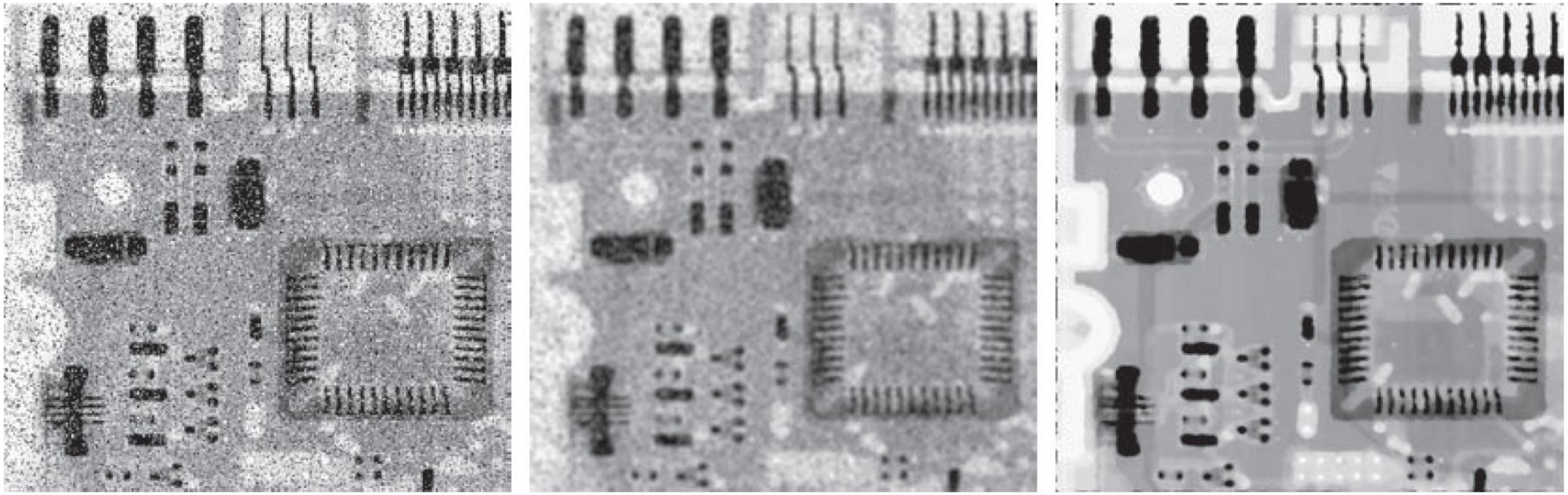
Median filtered





Plots of a row of the image

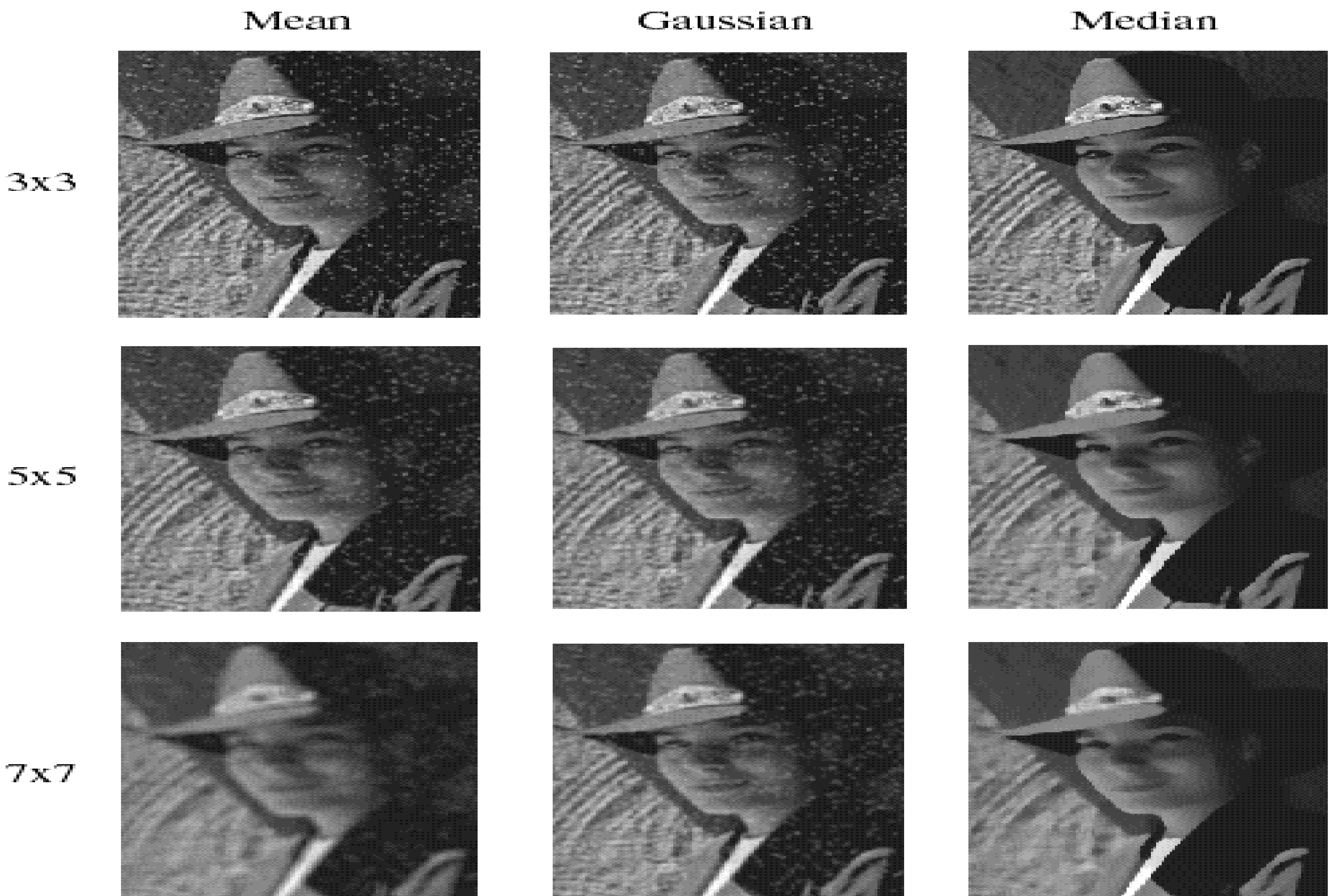
Median Filters - Example



a b c

FIGURE 3.43 (a) X-ray image of a circuit board, corrupted by salt-and-pepper noise. (b) Noise reduction using a 19×19 Gaussian lowpass filter kernel with $\sigma = 3$. (c) Noise reduction using a 7×7 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Averaging Filter Vs. Median Filter Example



Outline

- What is Image Enhancement?
- Contrast Enhancement
  - Intensity Transformation Functions
  - Histogram Processing
- Spatial Filtering
  - Basic Concepts
  - Smoothing Filters
  - Sharping Filters
  - Nonlinear Filters
- **Image Quality Assessment**
  - Subjective image quality assessment
  - Objective image quality assessment

Image Quality Assessment

- Through different processing stages, e.g., acquisition, compression, and transmission, images are subjected to different types of distortions which degrade the quality of them.
- The goal of image quality assessment (IQA) methods is to automatically evaluate the quality of images in agreement with human quality judgments.
- The application scope of IQA includes image acquisition, segmentation, printing and display systems, image fusion, and biomedical imaging.
- IQA methods can be categorized into subjective and objective methods.

Subjective Image Quality Assessment

- The most reliable method for assessing the quality of images is through subjective testing, since human observers are the ultimate users in most of the multimedia applications.
- In subjective testing a group of people are asked to give their opinion about the quality of each image.
- In order to perform a subjective image quality testing, several international standards are proposed which provide reliable results.

Subjective Image Quality Assessment

- Subjective quality assessment methods provide accurate and reliable measurements of the quality of visual signals. However, these methods suffer from different drawbacks that limits their applications:
  - They are time consuming and expensive. This is due to the fact that subjective results are obtained through experiments with many observers.
  - They cannot be incorporated into real-time applications such as image compression, and transmission systems.
  - Their results depend heavily on the subjects' physical conditions and emotional state. Moreover, other factors such as display device and lighting condition affect the results of such experiments.

Objective Image Quality Assessment

- The goal of objective IQA is to design mathematical models that are able to predict the quality of an image accurately and also automatically.
- An ideal objective IQA method should be able to mimic the quality predictions of an average human observer based on the availability of a distortion-free, perfect quality reference image
- Objective IQA methods can be classified into three categories.
  - Full-reference image quality assessment (FR-IQA) where the reference image is fully available.
    - Mean squared error (MSE)
    - peak-signal-to-noise ratio (PSNR).
  - Reduced-reference image quality assessment (RR-IQA) where only partial information about the reference image is available.
  - No-reference image quality assessment (NRIQA) where neither the reference image nor its features are available for quality evaluation

Acknowledgement

- The material in these slides are based on:
  - Digital Image Processing: Rafael C. Gonzalez, and Richard
  - Forsythe and Ponce: Computer Vision: A Modern Approach
  - Rick Szeliski's book: Computer Vision: Algorithms and Applications
  - cs131@ Stanford University
  - cs131n@ Stanford University
  - CS198-126@ University of California, Berkely
  - CAP5415@ University of Central Florida
  - CSW182 @ University of California, Berkely
  - 11-785@ Carnegie Mellon University
  - CSCI1430@ Brown University
  - Computer Vision@ Bonn University
  - ICS 505@ KFUPM
  - Digital Image Processing@ University of Jordan