Image Processing
Enhancement

Image Enhancement

- **Enhancement:** to process an image so that the certain features can be derived/ improve quality.
- Features:
  - Remove Noise
  - Improve contrast
  - Highlight edges
- **Enhancement approaches:**
  - Spatial domain (image plane) techniques are techniques that operate directly on pixels.
  - Frequency domain techniques are based on modifying the Fourier transform of an image.

Image Enhancement

Simple transformation: $g(x,y)=T\,[f(x,y)]$

- $f(x,y)$: input image, $g(x,y)$: processed image
- T: an operator or transformation function

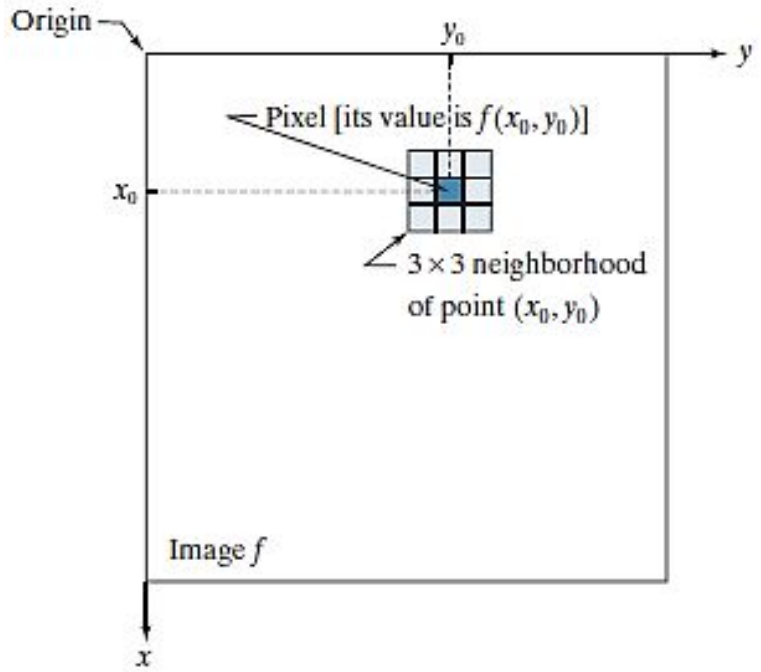
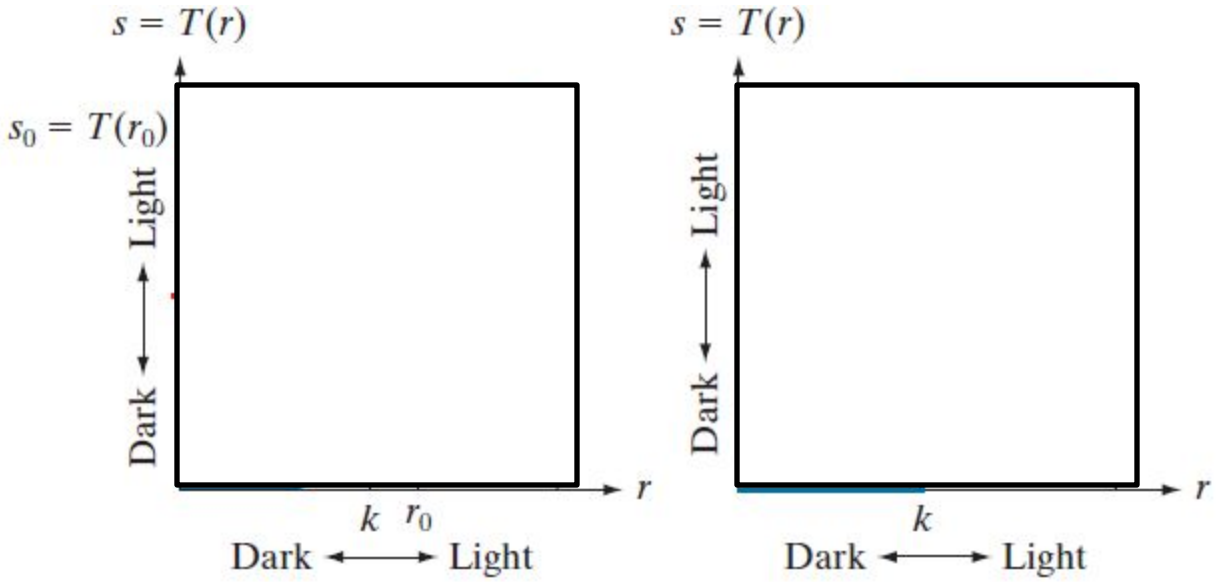


Image Enhancement

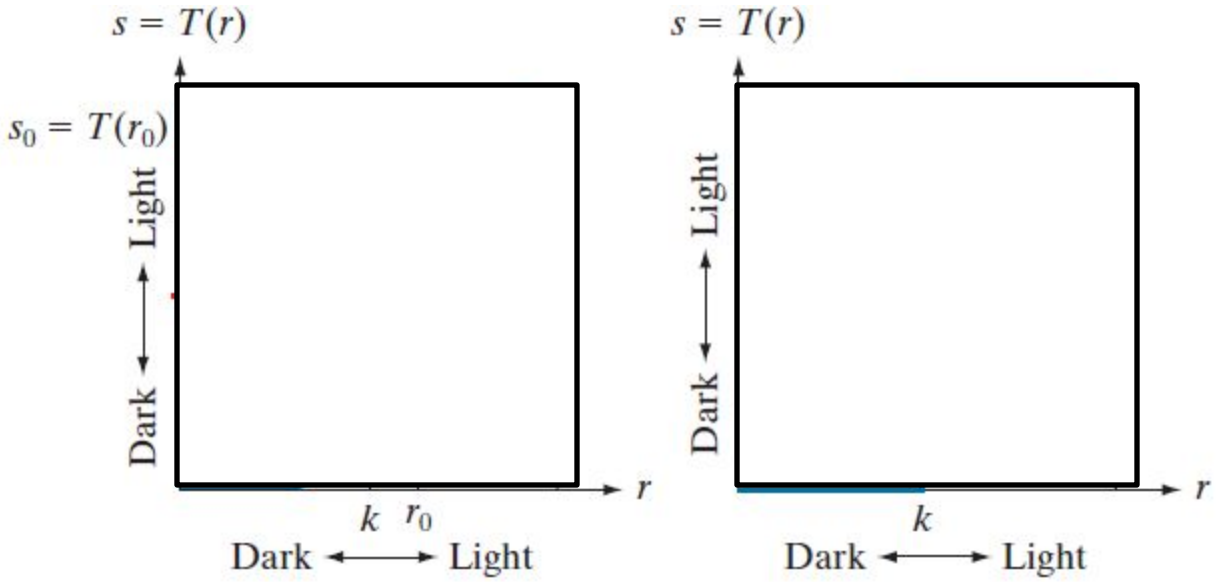
- *In case of point processing: Neighborhood is of size 1×1*
  Transform can written as : $s=T(r)$
  - r: gray-level at (x,y) in original image $f(x,y)$
  - s: gray-level at (x,y) in processed image $g(x,y)$
  - T is called gray-level transformation or mapping



Intensity transformation functions. (a) Contrast stretching (b)Thresholding

Image Enhancement

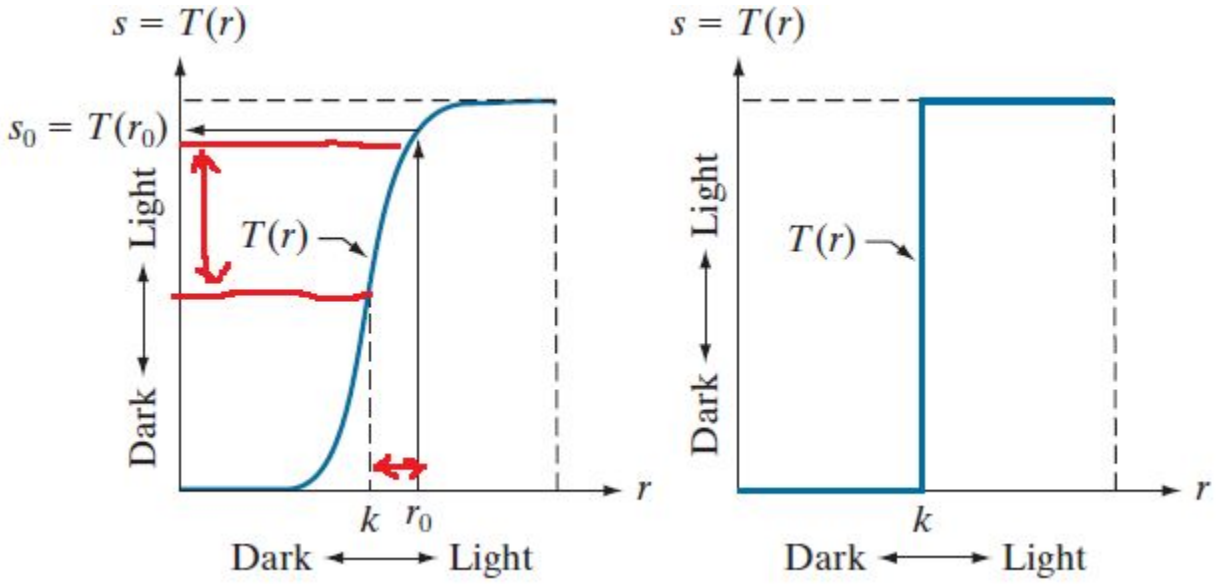
- *In case of point processing: Neighborhood is of size 1×1*
  Transform can written as : $s=T(r)$
  - r: gray-level at (x,y) in original image $f(x,y)$
  - s: gray-level at (x,y) in processed image $g(x,y)$
  - T is called gray-level transformation or mapping



Intensity transformation functions. (a) Contrast stretching (b)Thresholding

Image Enhancement

- *In case of point processing: Neighborhood is of size 1×1*
  Transform can written as : $s=T(r)$
  - 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



Intensity transformation functions. (a) Contrast stretching (b)Thresholding

Image Enhancement

Contrast Stretching

Original

Enhanced

Image Enhancement

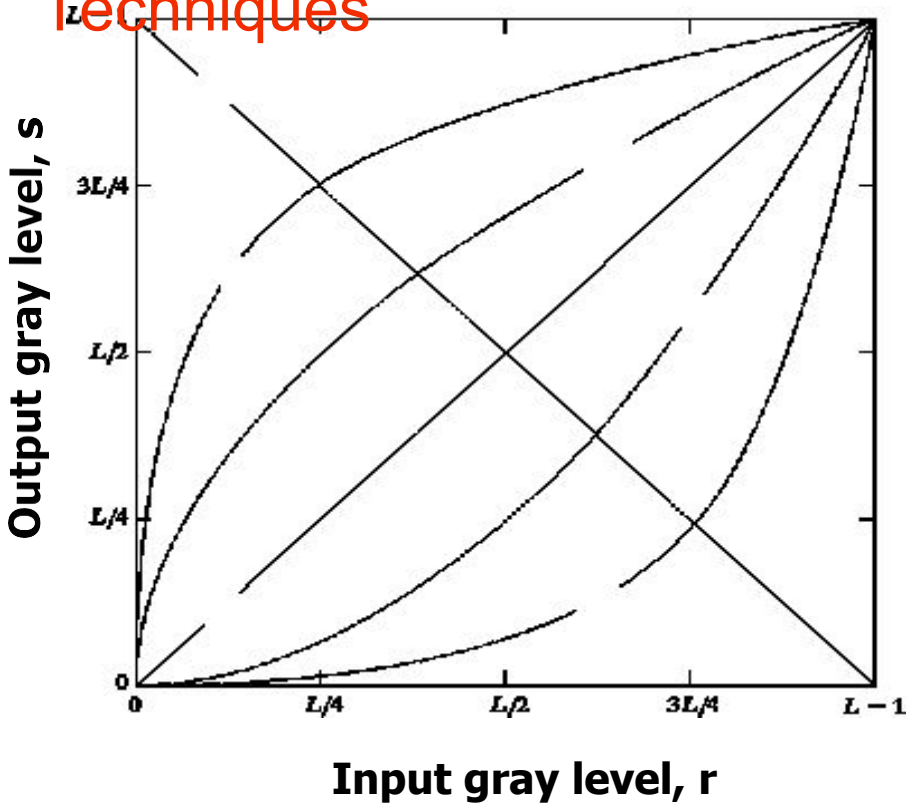
Contrast Stretching: Thresholding

Original

Enhanced

Basic Gray-level Intensity Transformation Functions

Simplest of all image enhancement Techniques



- Linear function
  - Negative and identity transformations
- Logarithm function
  - Log and inverse-log transformation
- Power-law function
  - n^{th} power and n^{th} root transformations

Image Negative

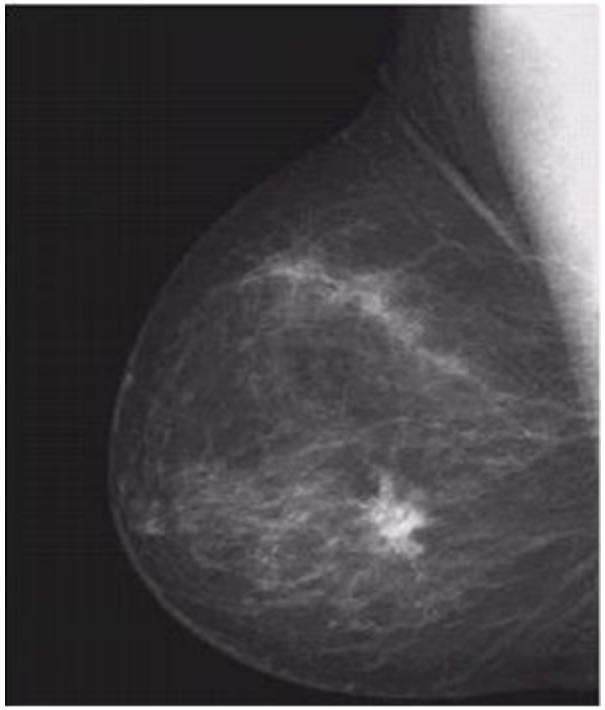
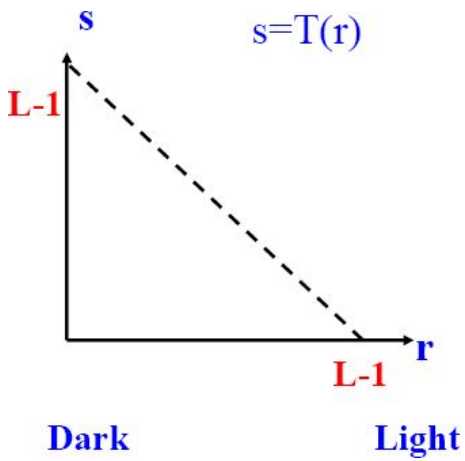


Image Negative

Image Negative

- Reversing the intensity levels of an image
  Given intensity levels in the range $[0, L-1]$
  Transformation: $\boldsymbol{s = (L-1) - r}$
- Produces the equivalent of a photographic negative.
- *Finds applications in medical imaging.*

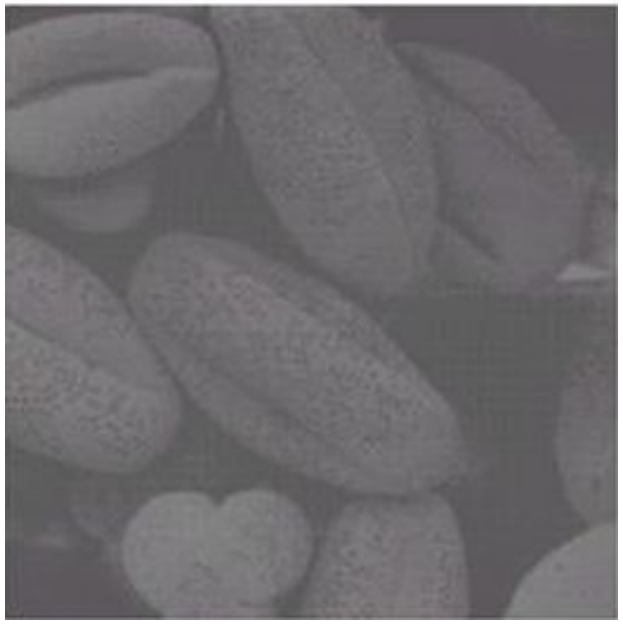


Original Digital mammogram

Negative image obtained using negative transformation

Contrast Stretching

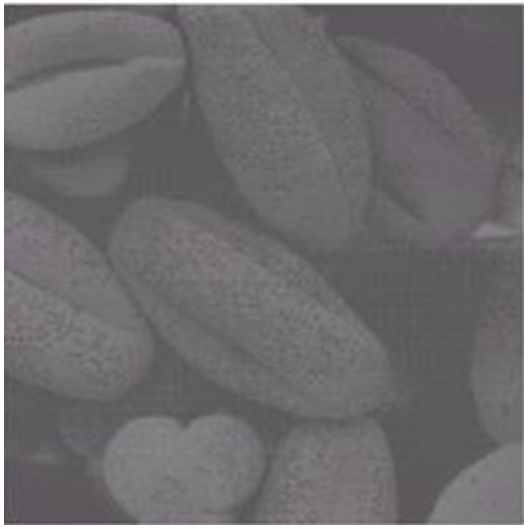
Low contrast image



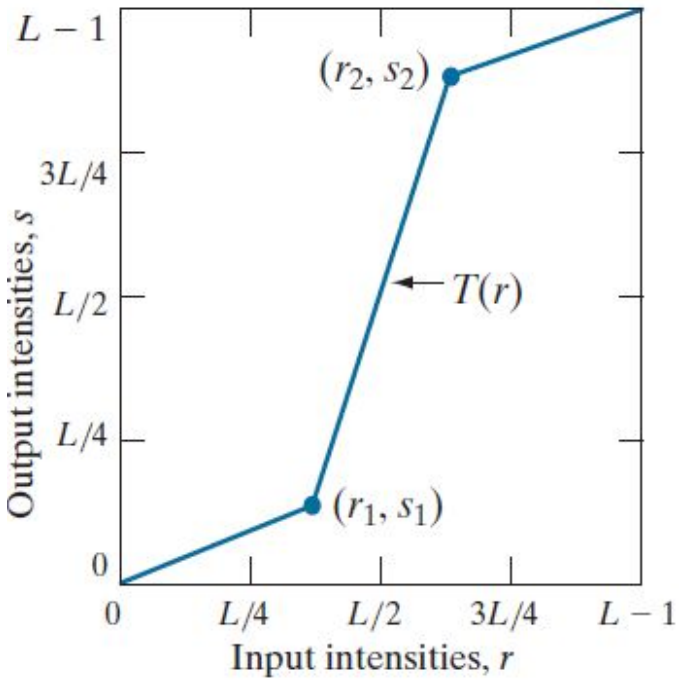
Contrast Stretching

- Piece-wise linear transformation
- To improve the range of the image intensity values

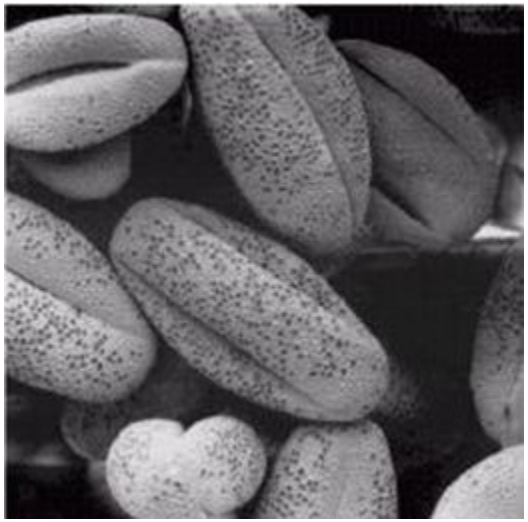
Low contrast image



Transformation function



Enhanced Image

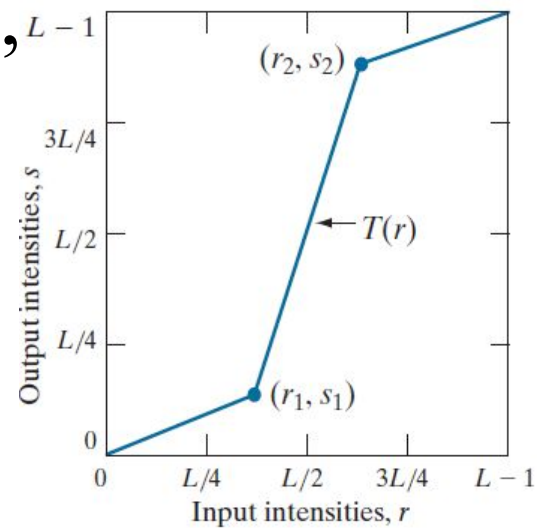


- The location of points (r_1,s_1) and (r_2,s_2) control the shape of the transformation function

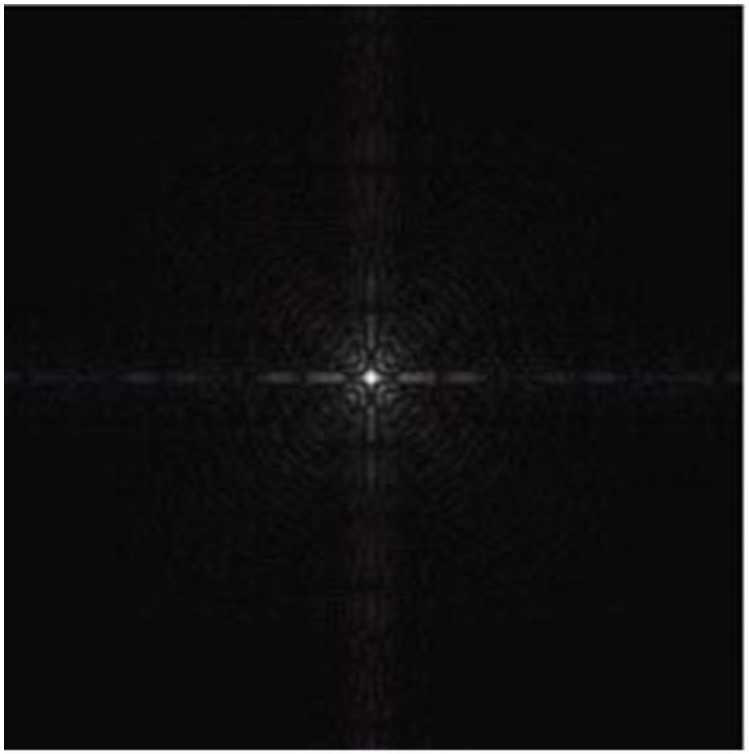
Contrast Stretching

- $r_1 = s_1$ and $r_2 = s_2$ ⇒ *linear transformation* function produces no changes in intensity levels
- $r_1 = r_2$ and $s_1 = 0$ and $s_2 = L - 1$ ⇒ *thresholding function*
- Intermediate values of (r_1, s_1) and (r_2, s_2) produce various degrees of spread in the intensity levels of output image, thus affecting its contrast

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



Logarithmic Transforms



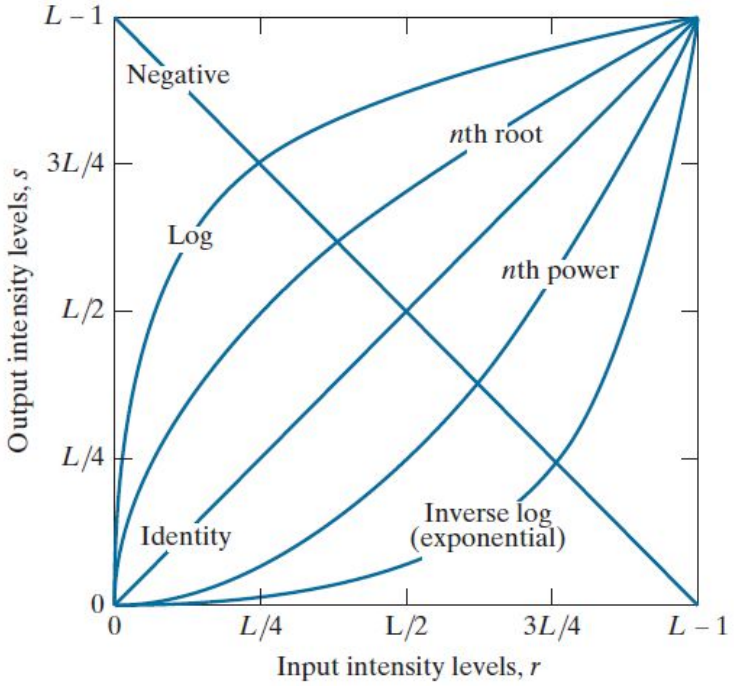
Logarithmic Transforms

- Maps a narrow range of low gray-level input image into a wider range of output levels.
- The opposite is true of higher values of input levels

Logarithmic Transforms

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

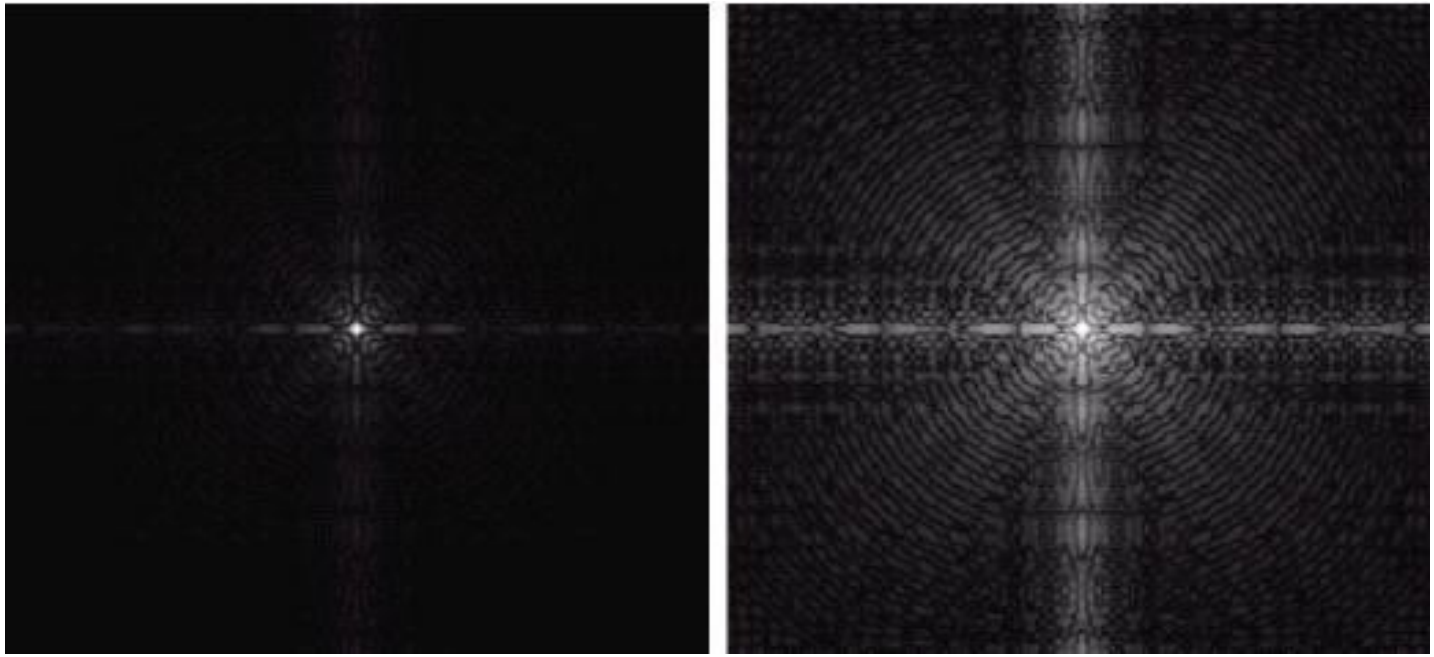
- Maps a narrow range of low gray-level input image into a wider range of output levels.
- The opposite is true of higher values of input levels



- Used to expand the values of dark pixels in an image while compressing the higher-level values.
- Opposite is true for inverse log transformation

Logarithmic Transforms

- Log function compresses the dynamic range of images with large variations in pixel values
- Classical example: Fourier Spectrum



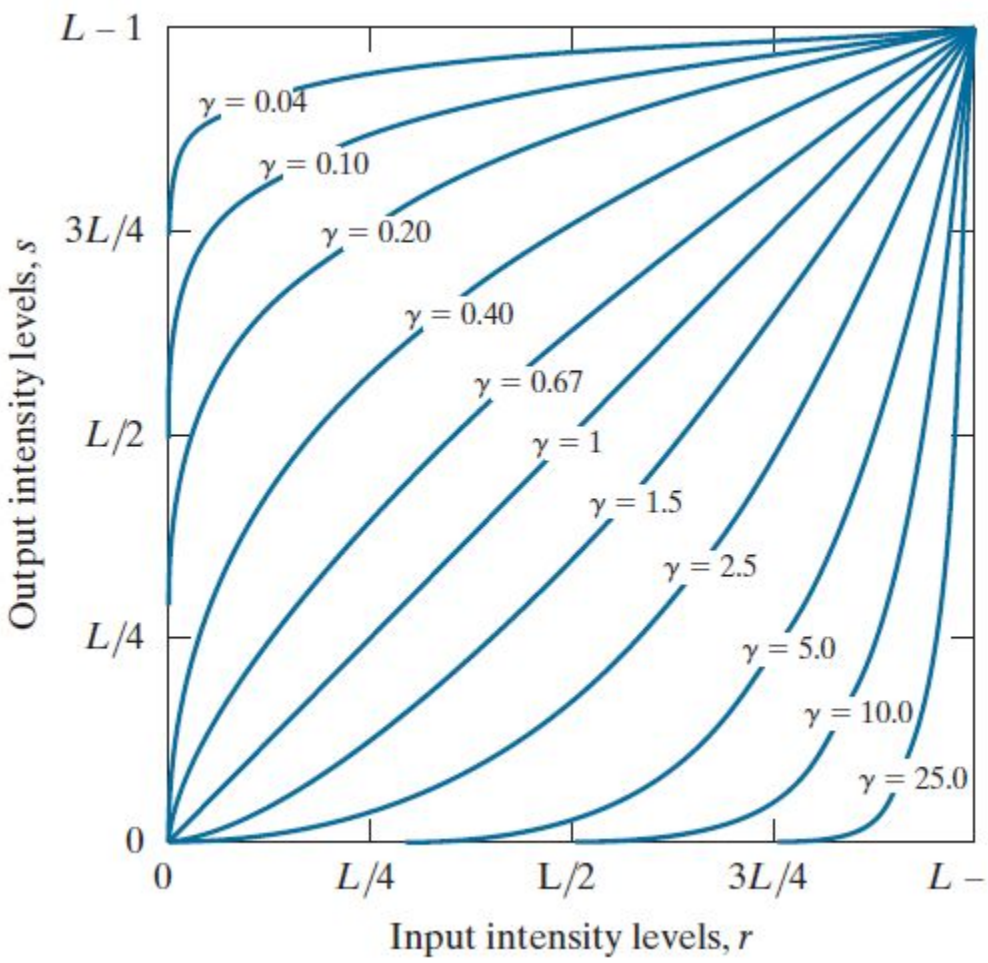
Fourier Spectrum
with values in the range = 0 to 10^6

Applying log transformation
with c=1 and range = 0 to 6.2

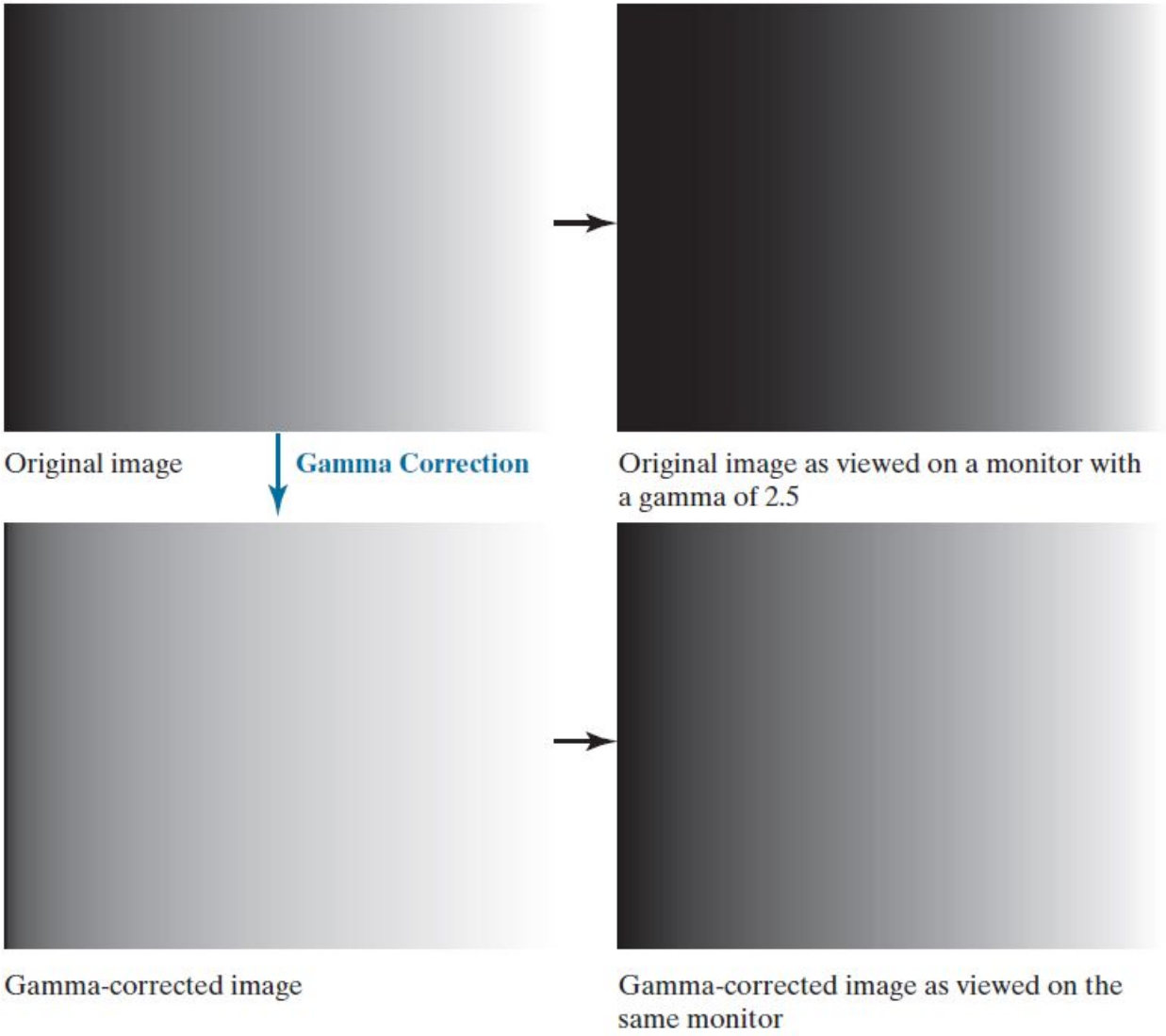
Power law Transform

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

- Power-law curves with fractional values of γ map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher
- $c = \gamma = 1$ ⇒ Identity function



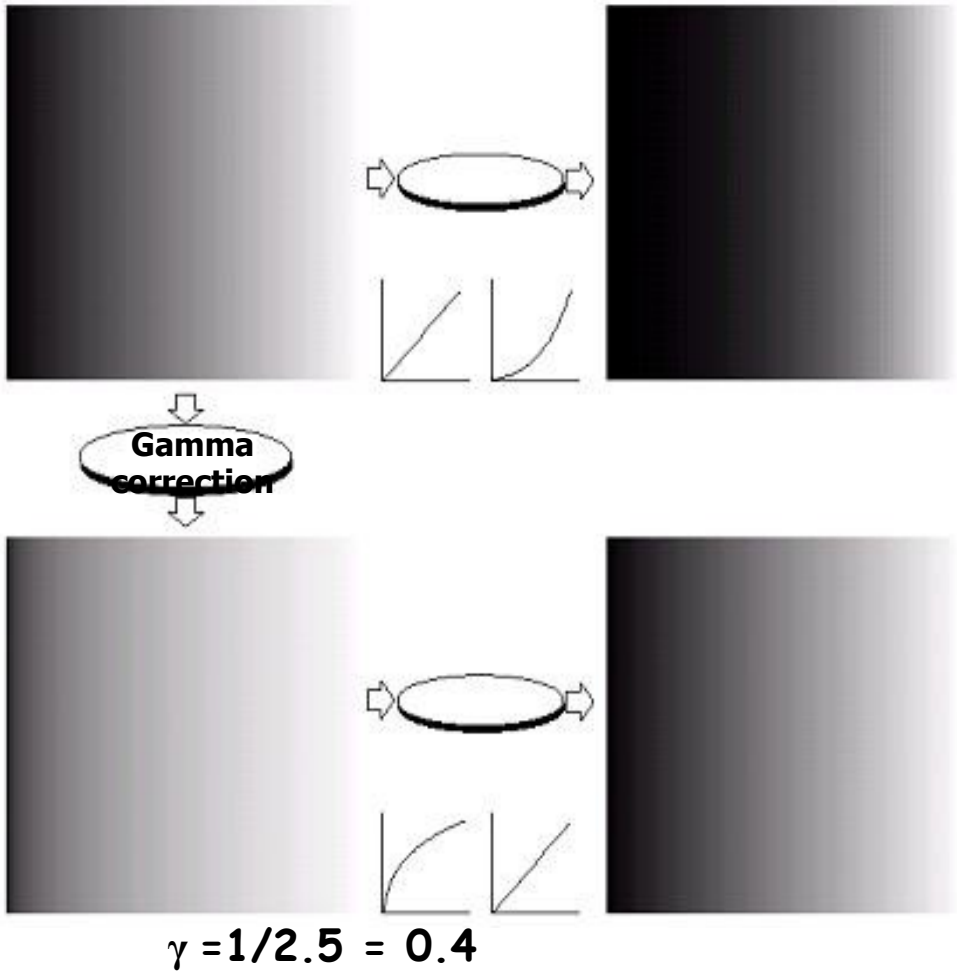
Gamma correction



a b
c d

FIGURE 3.7
(a) Intensity ramp image. (b) Image as viewed on a simulated monitor with a gamma of 2.5. (c) Gamma-corrected image. (d) Corrected image as viewed on the same monitor. Compare (d) and (a).

Gamma correction

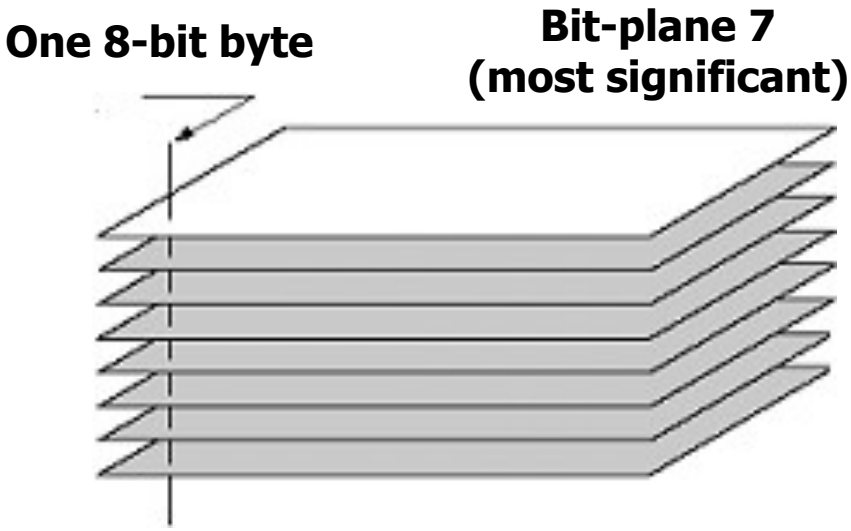


- A simple gray-scale linear wedge input to a monitor
- The output of the monitor appears darker than the input
- Gamma correction is done by performing the transformation $s = r^{1/2.5}$
- Gamma corrected input image produces an output that is close in appearance to the original image

- Similar analysis other imaging devices (scanner & printer)
- Only difference device dependent value of gamma

Bit Plane Slicing

- Highlighting the contribution made to total image appearance by specific bits
- Suppose each pixel is represented by 8 bits
- Higher-order bits contain the majority of the visually significant data
- Useful for analyzing the relative importance played by each bit of the image



- This type of decomposition is useful for image compression

Bit-plane Slicing: Another Example

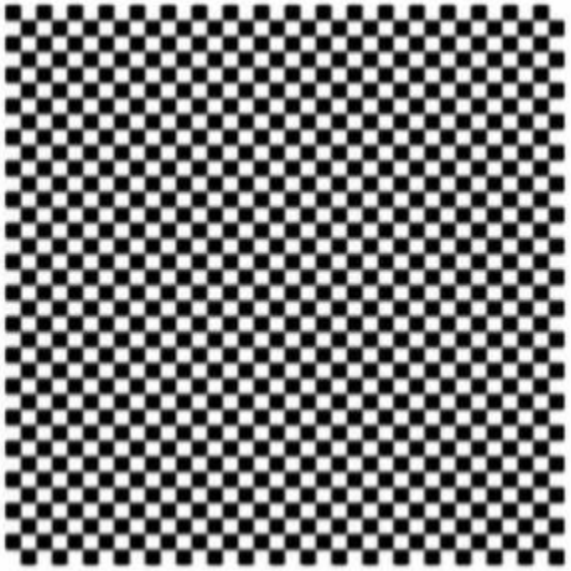
a b c
d e f
g h i

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.

Spatial Filtering

Spatial Frequencies

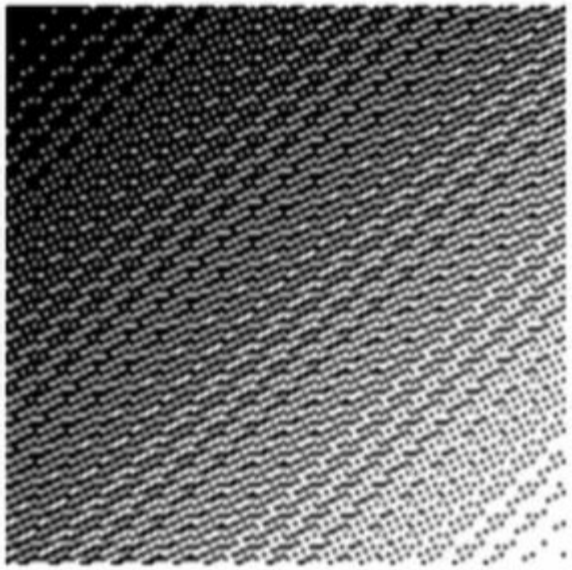
Radical change in intensity



High frequency Image

Changes in tone is abrupt over small areas

Slowly varying change in intensity

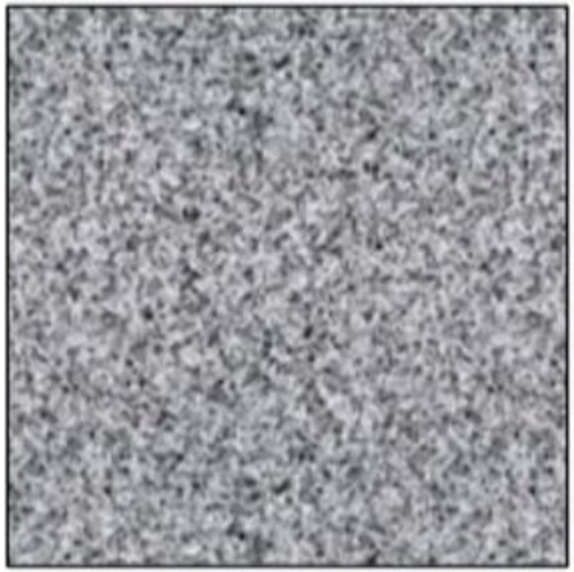


Low frequency Image

Little variation in tone over several pixels

Spatial Frequencies

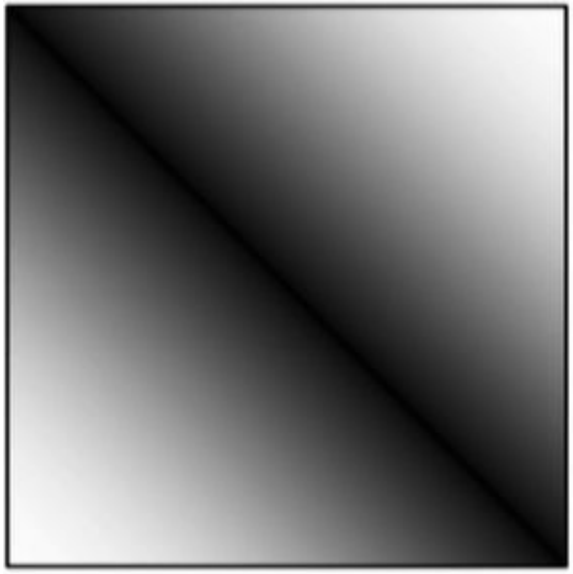
Radical change in intensity



High frequency Image

Changes in tone is abrupt over small areas

Slowly varying change in intensity



Low frequency Image

Little variation in tone over several pixels

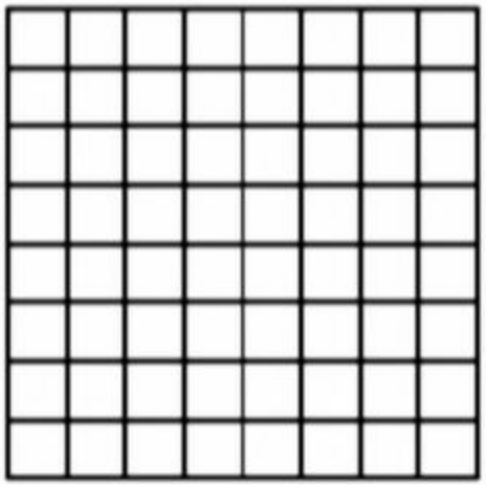
Spatial Frequency Definitions

- Number of changes in intensity value per unit distance for any particular part of the image.

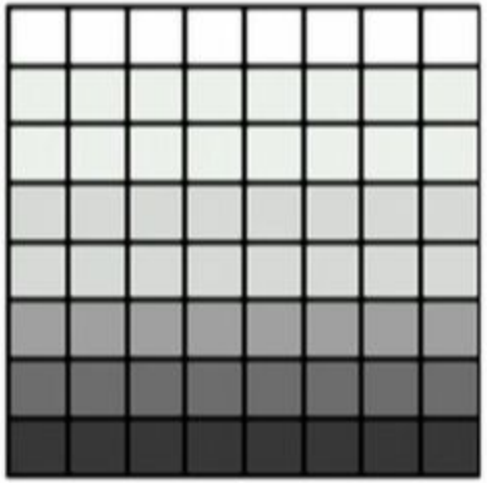
- Image is composed of a
  - Low frequency details- basic details, few changes over a given area
  - High Frequency details – fine details, dramatic changes over a given area

Spatial Frequency Examples

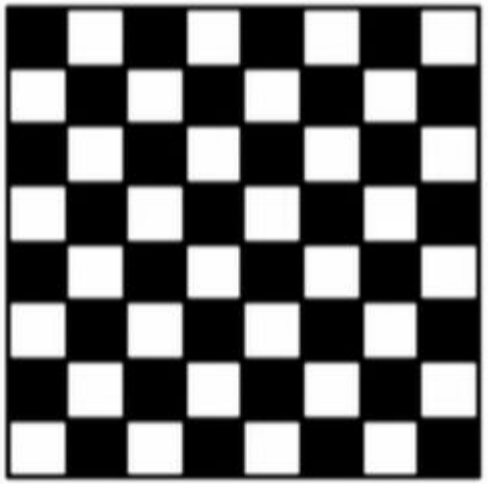
- Zero spatial frequency - flat image, all pixels same value.
- Low spatial frequency - smoothly varying grayscale image.
- High spatial frequency- an image consisting of check board fashion



Zero spatial frequency

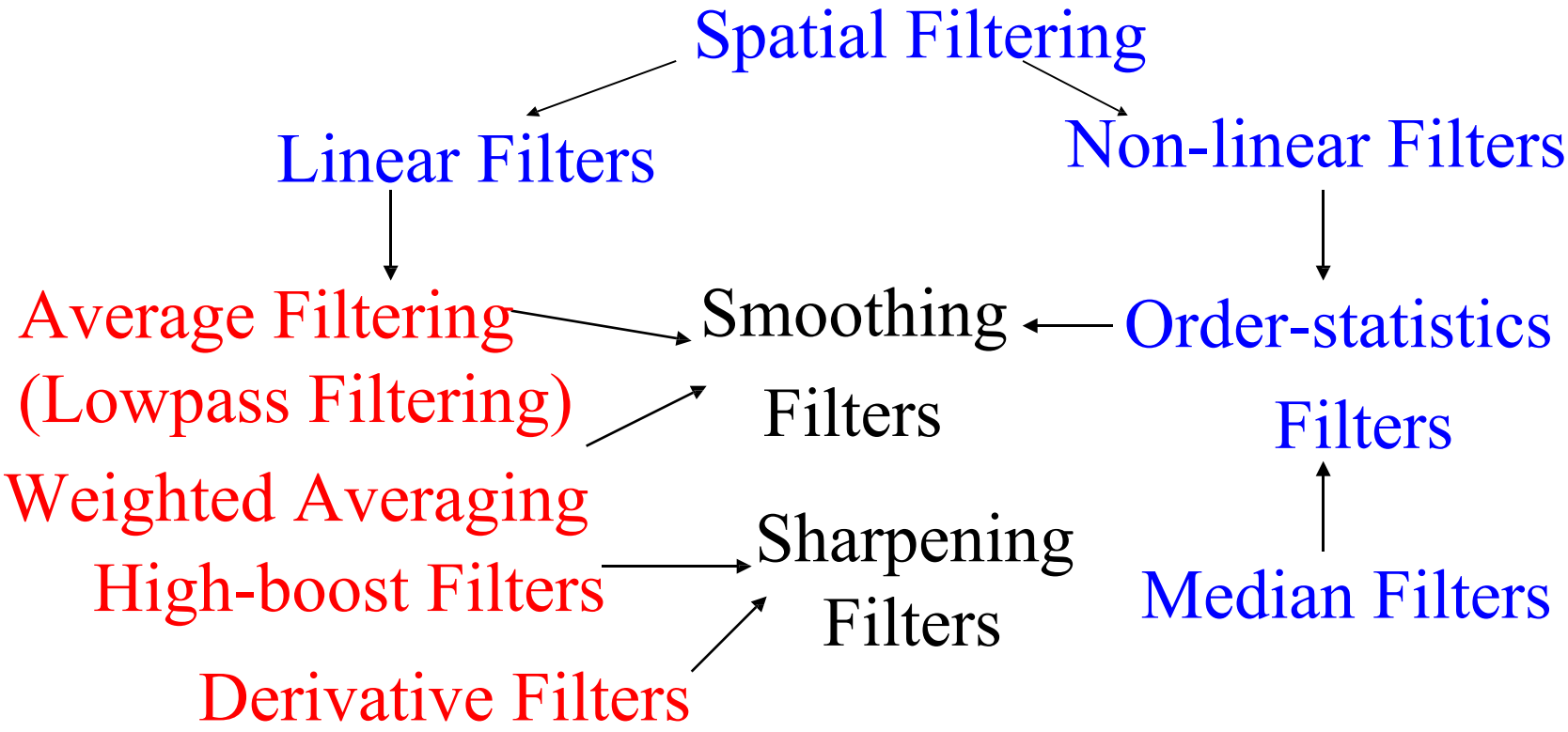


Low spatial frequency



High spatial frequency-

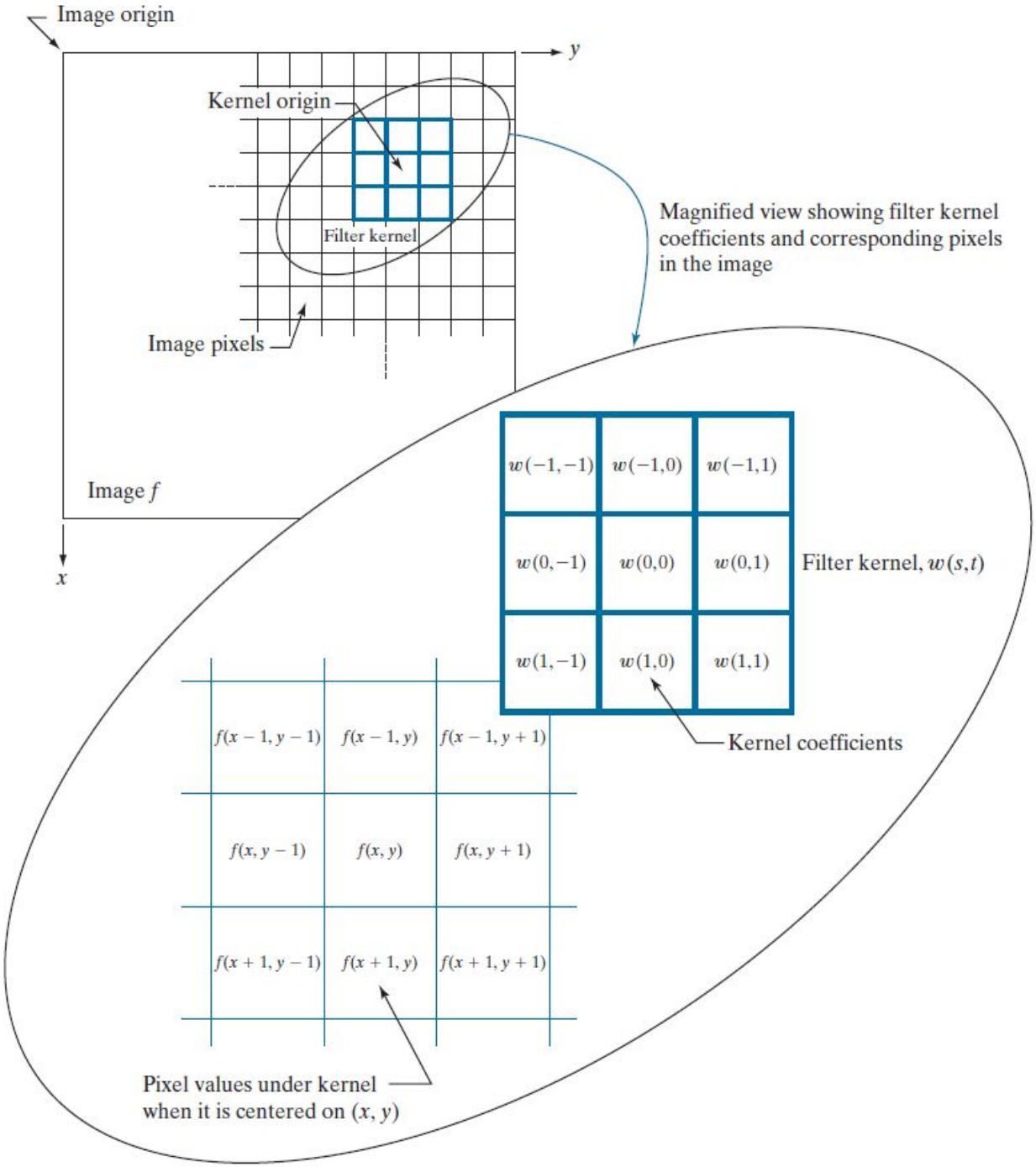
Spatial Filtering



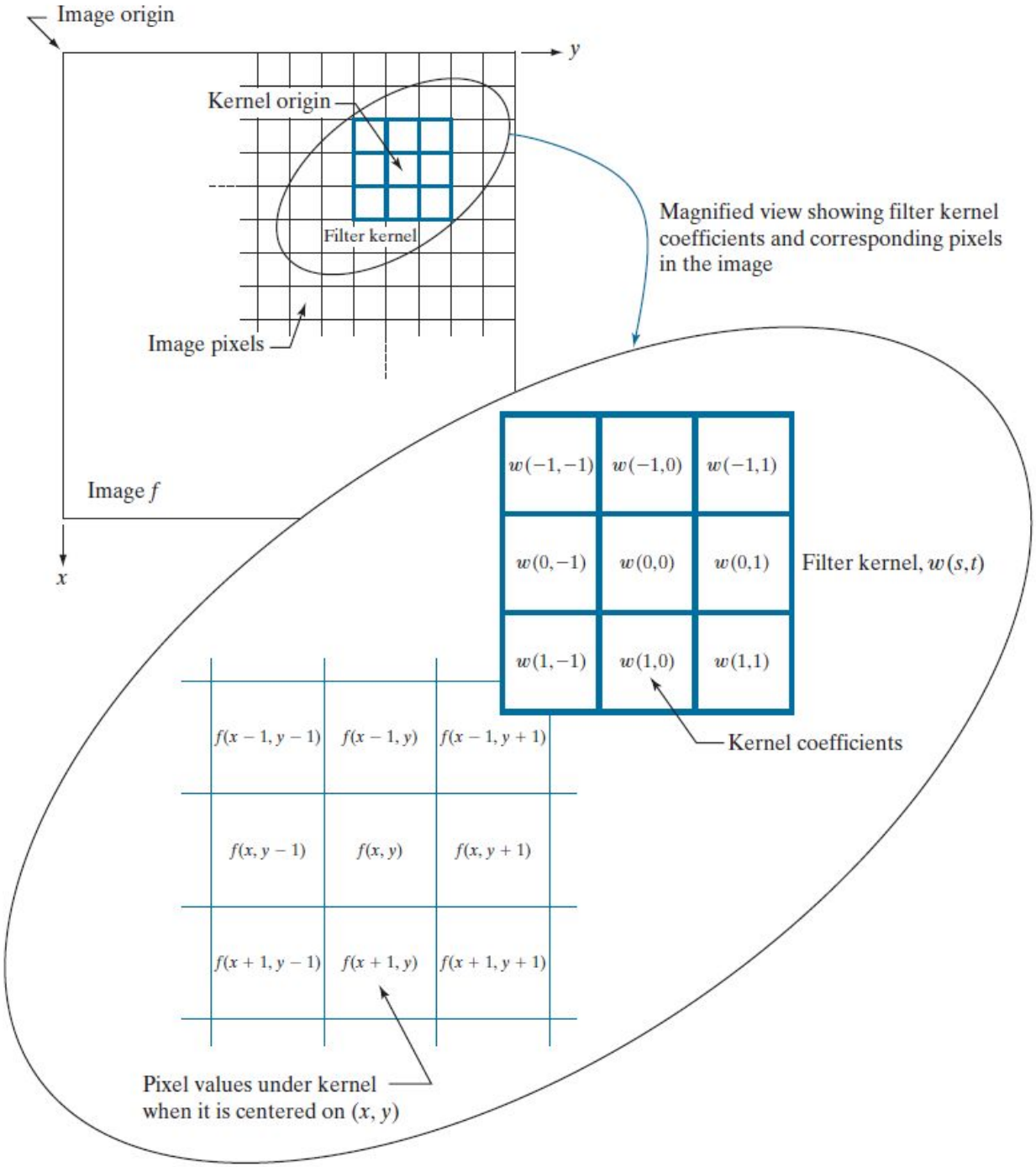
Smoothing filters: used for blurring and for noise reduction

Sharpening filters: used to highlight fine detail in an image or to enhance detail that has been blurred, either in error or as a natural effect of a particular method of image acquisition

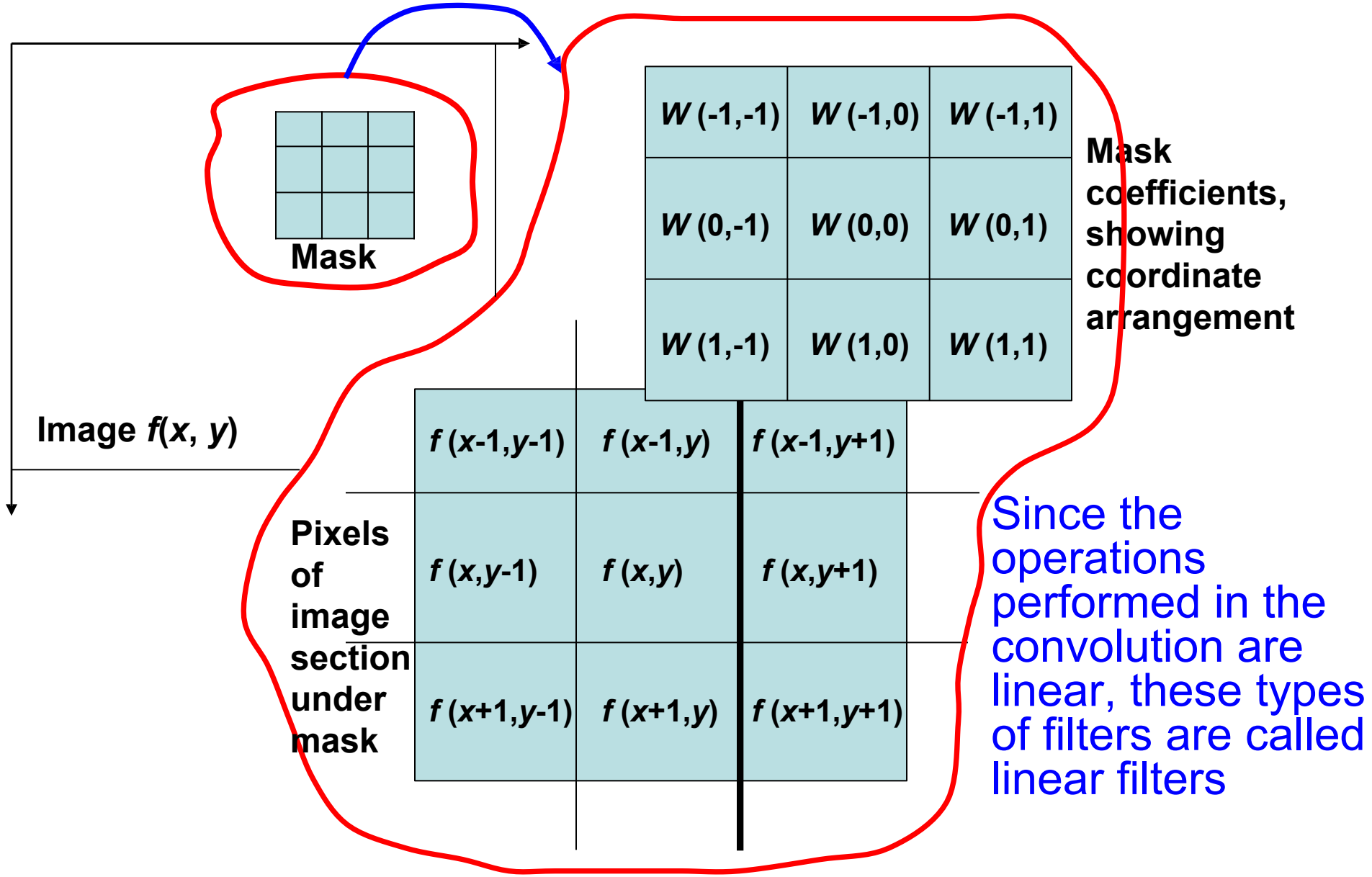
Linear Spatial Filtering

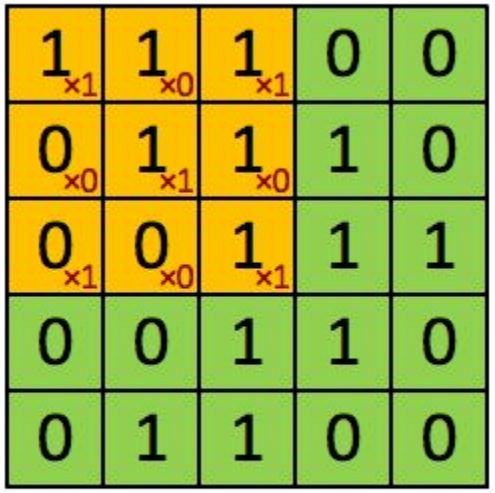


Linear Spatial Filtering

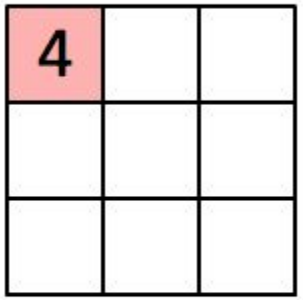


Linear Spatial Filtering

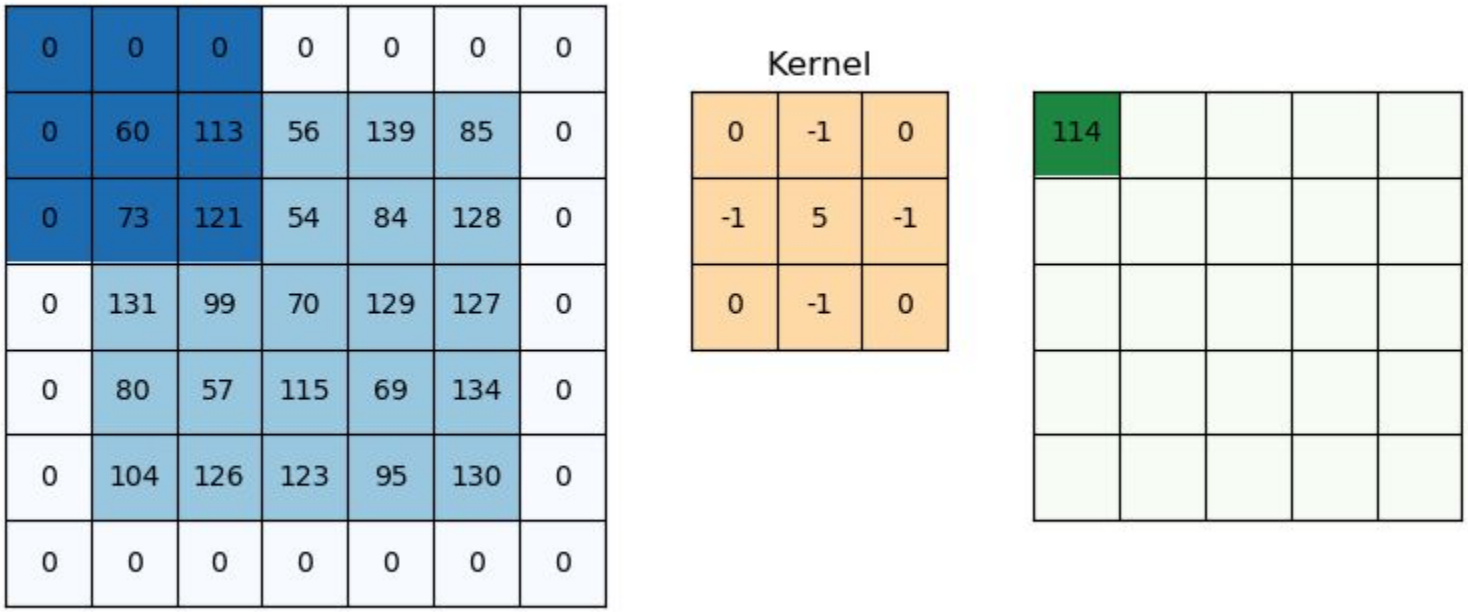




Image



Convolved Feature


0 0 0 0 0 0 0
0 60 113 56 139 85 0
0 73 121 54 84 128 0
0 131 99 70 129 127 0
0 80 57 115 69 134 0
0 104 126 123 95 130 0
0 0 0 0 0 0 0
Kernel
0 -1 0
-1 5 -1
0 -1 0
114

Linear Spatial Filtering

- The process consists simply of moving the filter mask from point to point in an image
- At each point (x, y), the response of the filter at that point is calculated using a predefined relationship
- The response is given by a sum of products of the filter coefficients and the corresponding image pixels in the area spanned by the filter mask
- The coefficient w (0,0) coincides with image value f (x, y), indicating that the mask is centered at (x, y) when the computation of the sum of products takes place

Linear Spatial Filtering

- For a 3x3 mask shown in figure, the response, R, of linear filtering with the filter mask at a point (x, y) in the image

$$R = w(-1,-1)\,f(x-1,y-1) + w(-1,0)\,f(x-1,y) + \ldots\ldots$$
$$+ w(0,0)\,f(x,y) + \ldots + w(1,0)\,f(x+1,y)$$
$$+ w(1,1)\,f(x+1,y+1)$$

$w(s,t)$: $(2a+1) \times (2b+1)$ mask

Where, „a“ & „b“ are nonnegative integers

- For a mask of size mxn, we assume that $m = 2a+1$ and $n = 2b+1$
- Focus: mask of odd sizes, smallest meaningful size: 3x3

Linear Spatial Filtering

- In general, Linear filtering of an image *f* of size *M*x*N* with a filter mask of size mxn is given by:

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

where $a = (m-1)/2$ and $b = (n-1)/2$

- To generate a complete filtered image this equation must be applied for $x = 0, 1, 2, \ldots, M-1$ and $y = 0, 1, 2, \ldots, N-1$
  - In this way, we are assured that the mask processes all pixels in the image
- Nonlinear spatial filters also operate on neighborhoods, and the mechanics of sliding a mask past an image are the same. They do not explicitly use coefficients in the sum-of-products

Linear Spatial Filtering

- Border Effects- Special Consideration
  - For a small value of *m* and *n*, it is not that much problem; otherwise we may loose substantial amount of data
  - Common procedures:
    - Border strips are added and set to zero
    - Rows and columns are considered to wrap around
    - Limit the excursion of the center of the filter mask to a distance no less than (*n*-1)/2 pixels from the border of the original image
      - The resulting filtered image will be smaller than the original, but all the pixels in the filtered image will have been processed with the full mask

Spatial Correlation and Convolution

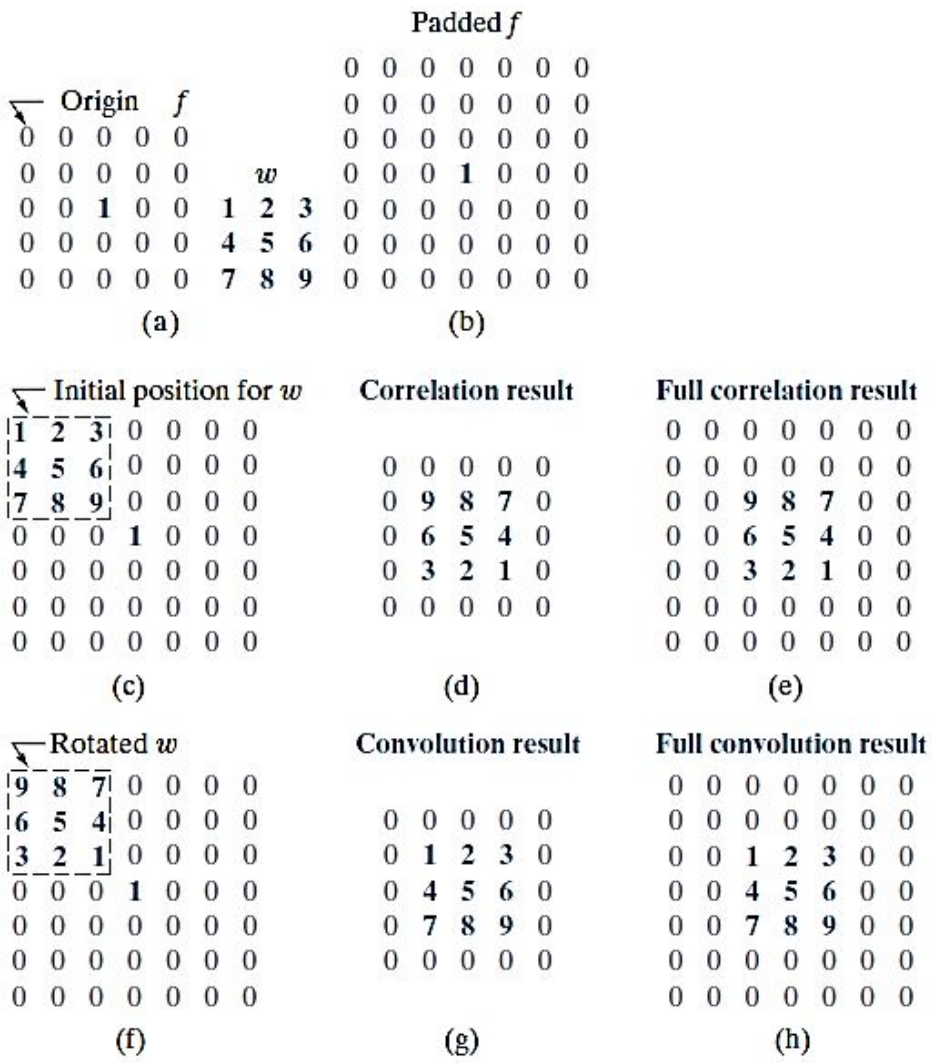


FIGURE 3.30 Correlation (middle row) and convolution (last row) of a 2-D filter with a 2-D discrete, unit impulse. The 0s are shown in gray to simplify visual analysis.

- See that convolution of a function with an impulse copies the function at the location of the impulse
- If the filter mask is symmetric, correlation and convolution yield the same result

Spatial Correlation and Convolution

- Correlation

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

- Convolution

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

where $a = (m-1)/2$ and $b = (n-1)/2$

Minus sign denotes 180^0 rotation of f, which has been done for notational simplicity and to follow convention

Spatial Correlation and Convolution

- Using correlation or convolution to perform spatial filtering is a matter of preference
  - Either can perform the function of the other by a simple rotation of the filter
  - Important is the filter mask used in a given filtering task be specified in a way that corresponds to the intended operation
- The linear spatial filtering discussed here are based on convolution
- Convolution filter, Convolution mask, Convolution kernel spatial filter (not necessarily used for true convolution)
- Convolving a mask with an image is often used to denote the sliding, sum-of-products process

Generating Spatial Filter Mask

- Example:

Consider the image: $\begin{bmatrix} 0 & 1 & 0 \\ & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$

1

Suppose the second row of the image is to be highlighted, what is the mask that is needed?

Generating Spatial Filter Mask

- One interesting observation:
  - Overall effect can be predicted on the general pattern, if one uses convolution mask
- If the sum of coefficients of the mask is one, average brightness of the image will be retained
- If the sum of coefficients of the mask is zero, average brightness of the image will be lost and will return a dark image
- If coefficients are alternating positive and negative, the mask is a filter that will sharpen an image
- If coefficients are all positive, it is a filter that will

Smoothing Spatial Filters

- Used for blurring and for noise reduction
- Blurring is used in preprocessing steps, such as
  - Removal of small details from an image prior to large object extraction, and
  - Bridging of small gaps in lines or curves
- Noise reduction can be accomplished by blurring with a linear filter and also by nonlinear filtering
- The output/response of a smoothing, linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask
- Also called „Averaging Filters“ or

Smoothing Linear Filters

- 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 intensity transitions
  - random noise in the image
  - edges of objects in the image
- Thus, smoothing can reduce noises (desirable) and blur edges (undesirable)
- Other uses
  - smoothing of false contours resulting from using an insufficient number of intensity levels
  - reduction of irrelevant detail in an image (here, "irrelevant" means pixel regions that are small wrt the size of filter

General Form : Smoothing Mask

- Filter of size m x n (m and n odd)

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

summation of all coefficient of the mask (computed once)

The complete filtered image is obtained by applying the operation for $x = 0, 1, 2, \ldots, M - 1$ and $y = 0, 1, 2, \ldots, N - 1$

Weighted Average Filter

- 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.

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

Weighted Average Filter

3x3 Smoothing Linear Filters

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

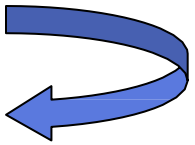
1	1	1
1	1	1
1	1	1

box filter

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

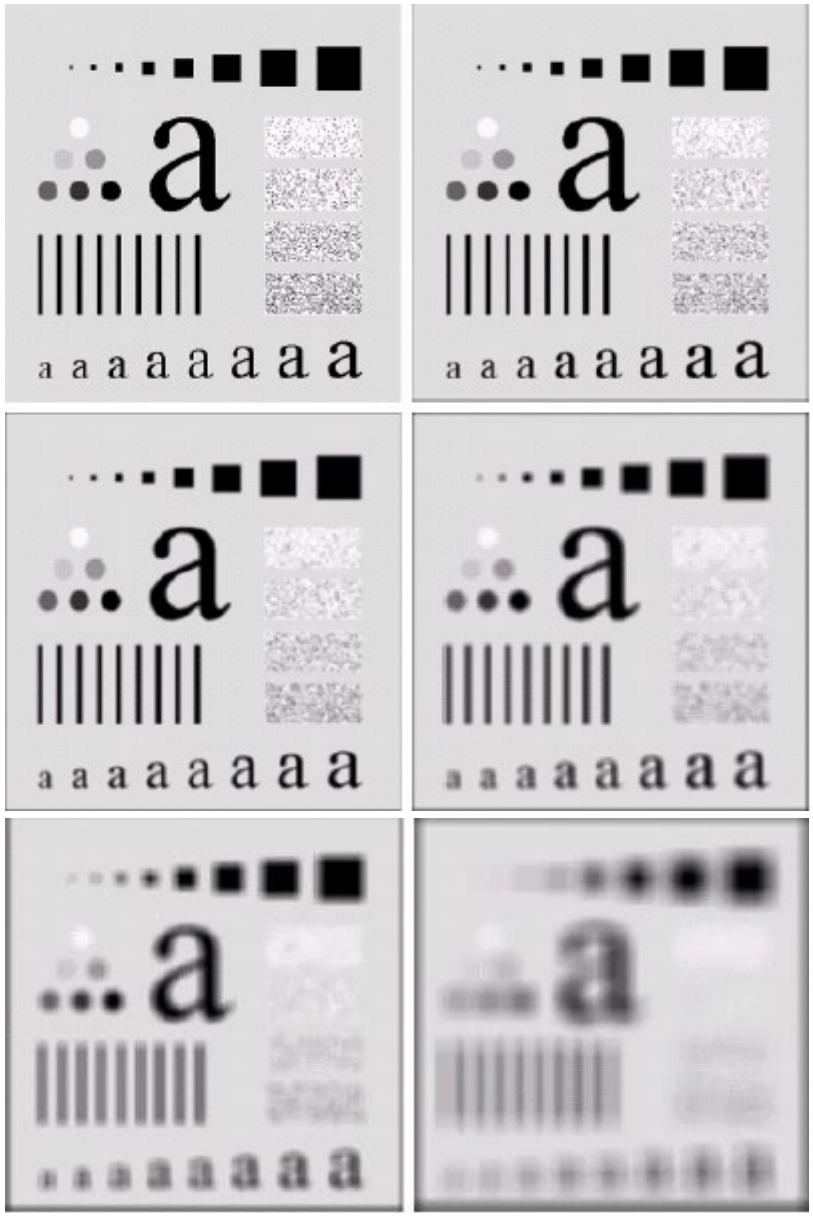
1	2	1
2	4	2
1	2	1

weighted average



the center is the most important and other pixels are inversely weighted as a function of their distance from the center of the mask

Example 1



a	b
c	d
e	f

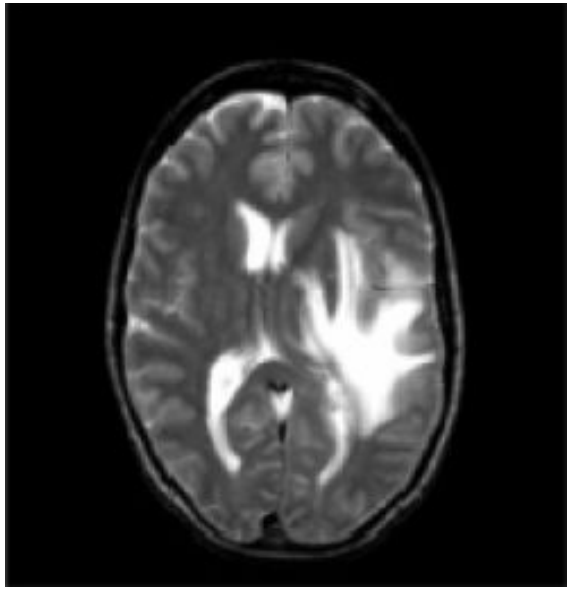
(a) Original image of size 500x500 pixel

(b)-(f) Results of smoothing with square averaging filter masks of sizes *n* = 3, 5, 9, 15, and 35 respectively

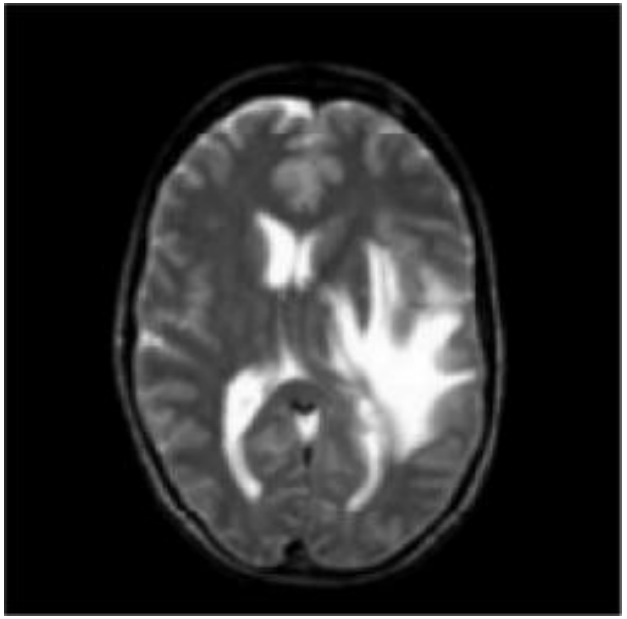
Note:

- details that are of approx. the same size as the filter mask are affected considerably more
- the size of the mask establishes the relative size of the objects that will be blended with the background
- big mask is used to eliminate small objects from an image

Example 2



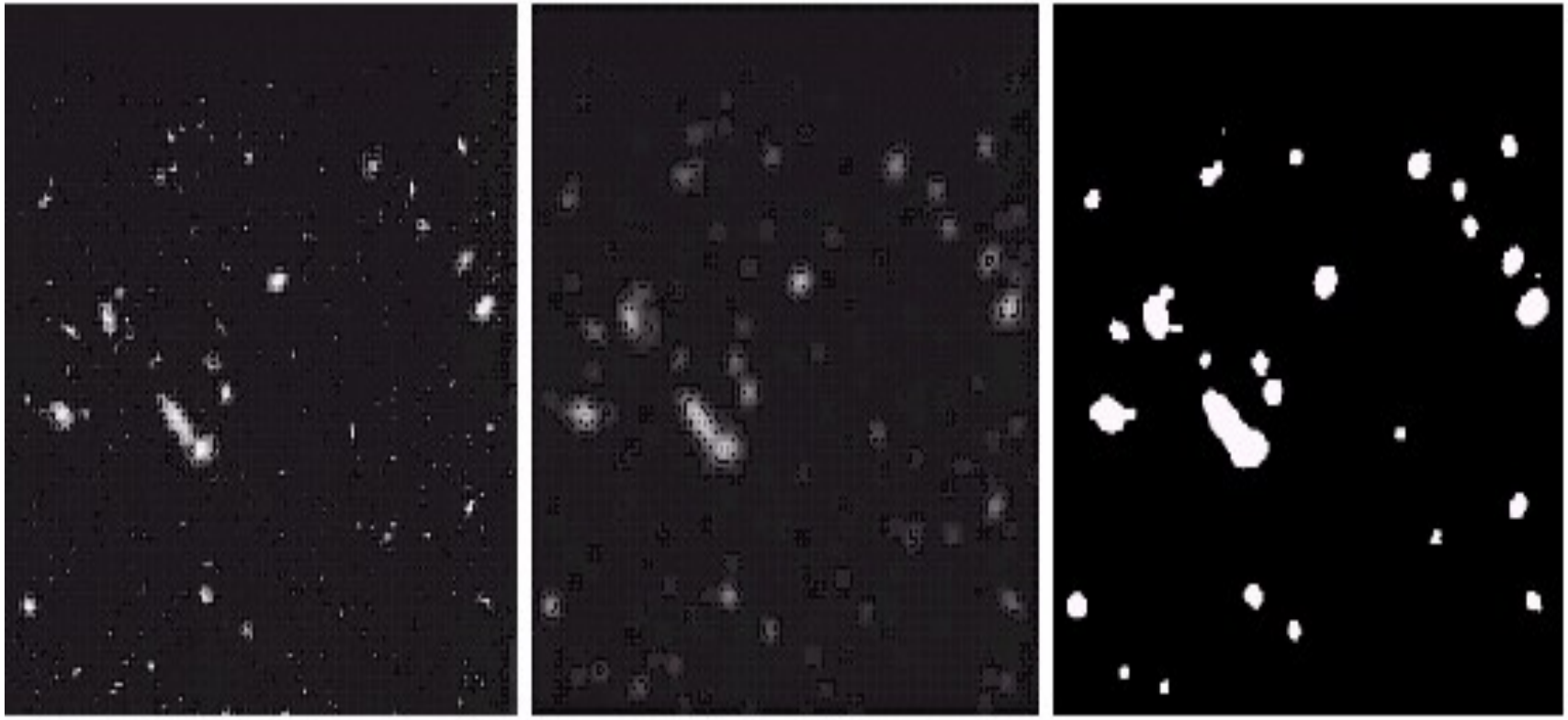
1/16	2/16	1/16
2/16	4/16	2/16
1/16	2/16	1/16



Before smoothing

After smoothing

Example 3



original image

result after smoothing with 15x15 averaging mask

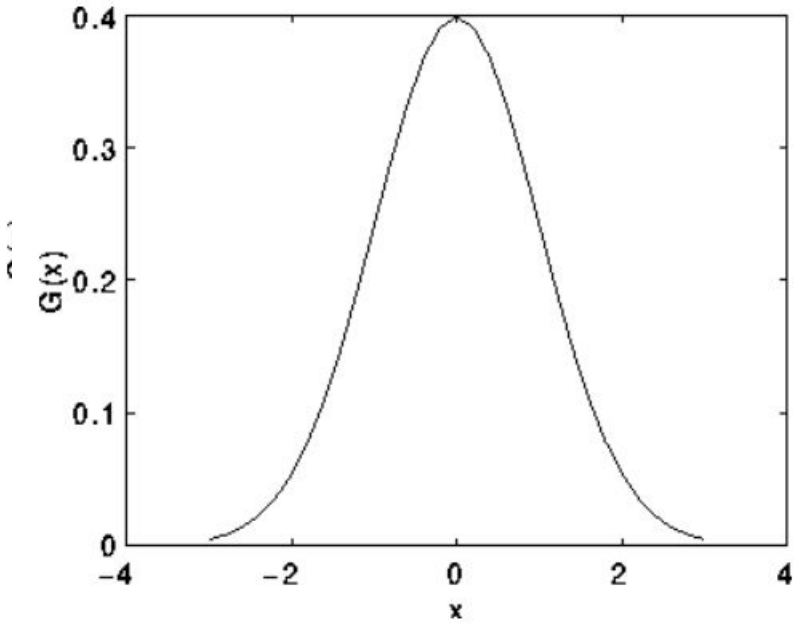
result of thresholding

we can see that the result after smoothing and thresholding, the remains are the largest and brightest objects in the

Gaussian Filters

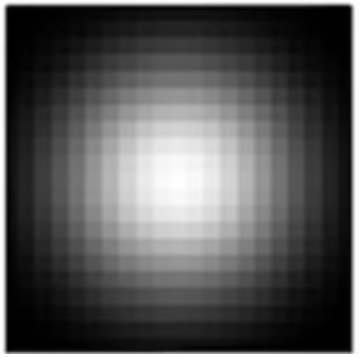
The Gaussian distribution in 1-D has the form:

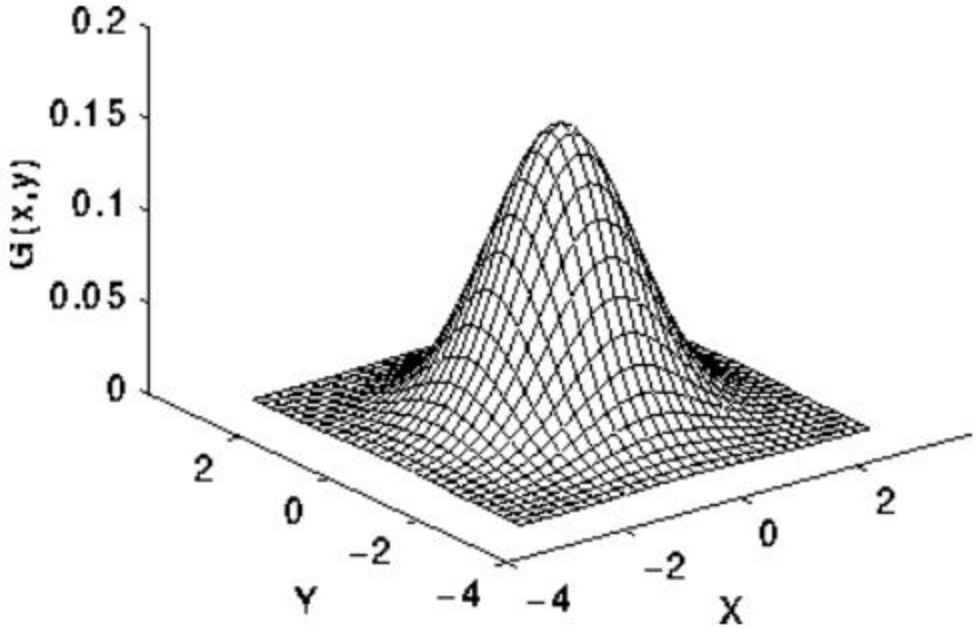
$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$



Gaussian Filters

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



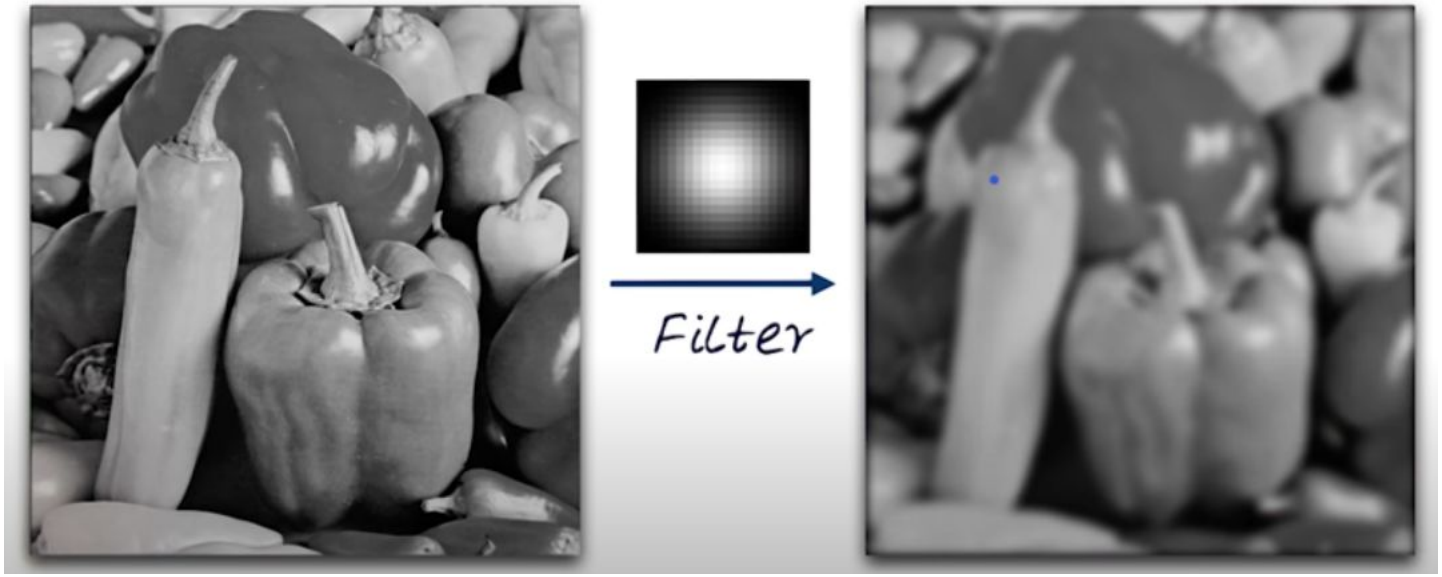


$$\frac{1}{273}$$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Figure 3 Discrete approximation to Gaussian function with σ=1.0

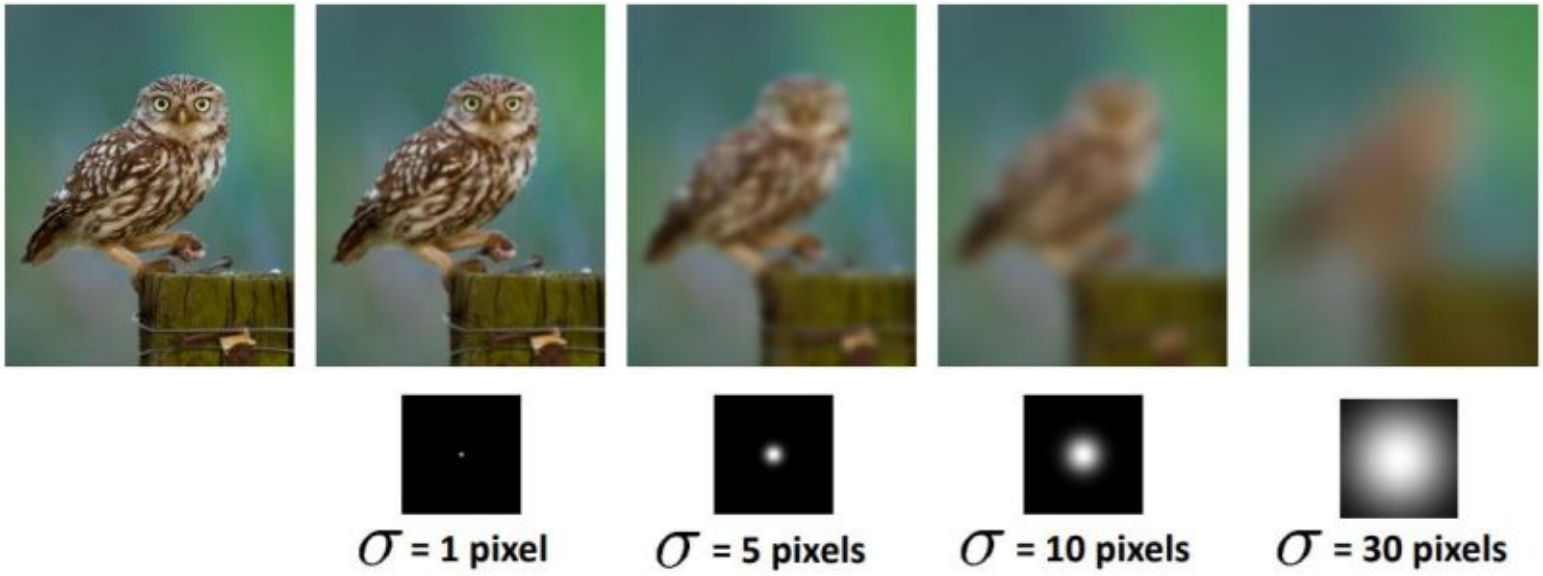
Gaussian Filters



Gaussian Filters

Gaussian Filters

- Effect of varying σ , as we increase it blurring effect will increase.



Order-Statistics Filters

- Nonlinear spatial filters
- The response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and

  then replacing the value of the center pixel with the value determined by the ranking results
- Example (n x n is the size of the mask)
  - median filter : $R = \text{median}\{z_k | k = 1, 2, \ldots, n \text{ x } n\}$
  - max filter : $R = \max\{z_k | k = 1, 2, \ldots, n \text{ x } n\}$
  - min filter : $R = \min\{z_k | k = 1, 2, \ldots, n \text{ x } n\}$
- Most popular: median filter

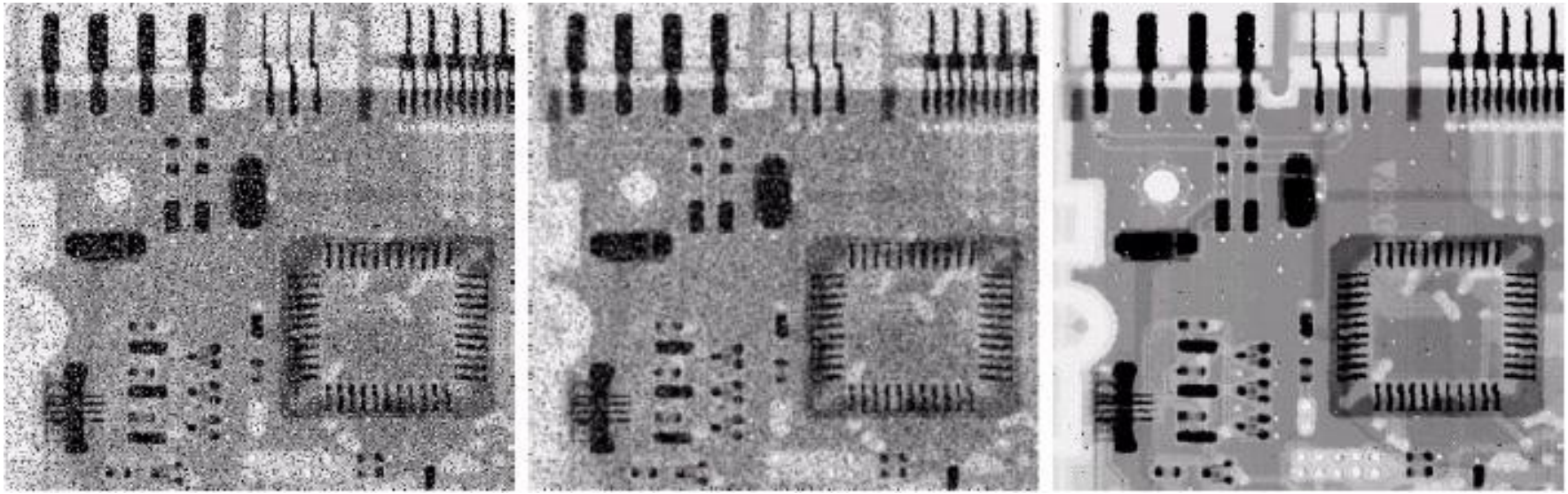
Median Filters

- It replaces the value of the centre pixel by the Median/Min/Max value in the neighborhood pixels
  - Original value of the pixel is included in the computation
  - Median: 50^{th} percentile of a ranked set of nos.
    - For ex: in a 3x3 neighborhood the median is the 5th largest value (5x5 neighborhood -> 13th largest value)
  - 100^{th} percentile: max filter; 0^{th} percentile: min filter
- 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

Median Filters

- 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 Filtering: Example 1



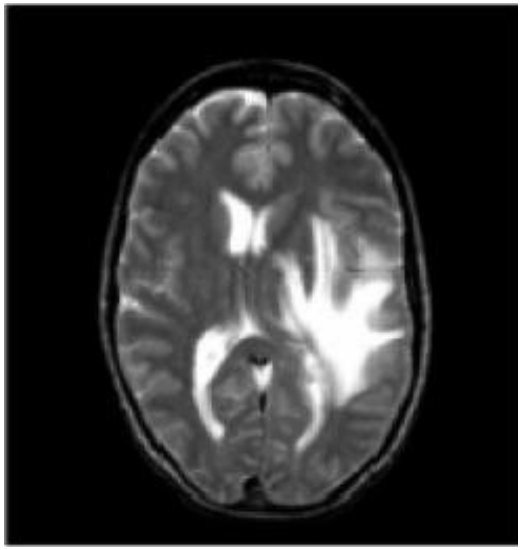
(a) (c) (b)

(a) X-ray image of circuit board corrupted by salt-and-pepper noise

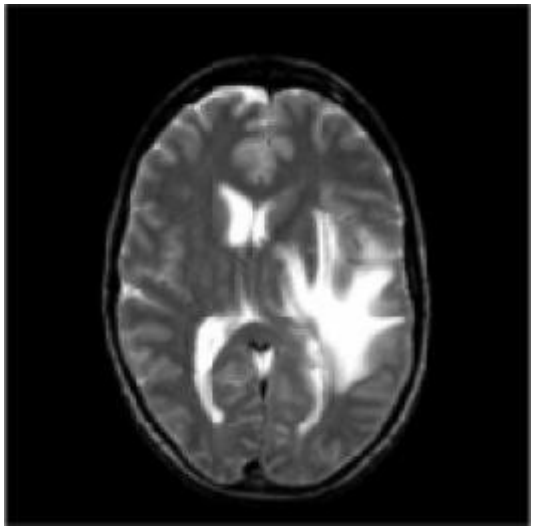
(b) Noise reduction with a 3x3 averaging mask
-less noise reduction but more blurring

(c) Noise reduction with a 3x3 median filter

Median Filtering: Example 2



Before smoothing



After smoothing

The smoothed MR brain image obtained by using median filtering over a fixed neighborhood of 3x3 pixels

Sharpening Spatial Filters

- To highlight fine detail in an image
- or to enhance detail that has been blurred, either in error or as a natural effect of a particular method of image acquisition
- Blurring vs. Sharpening
  - as we know that blurring can be done in spatial domain by pixel averaging in a neighbors
  - since averaging is analogous to integration
  - thus, we can guess that the sharpening must be accomplished by spatial differentiation

Derivative Operator

- The strength of the response of a derivative operator is proportional to the degree of intensity 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
- First-order derivative
- Second-order derivative

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

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

Derivative Operator

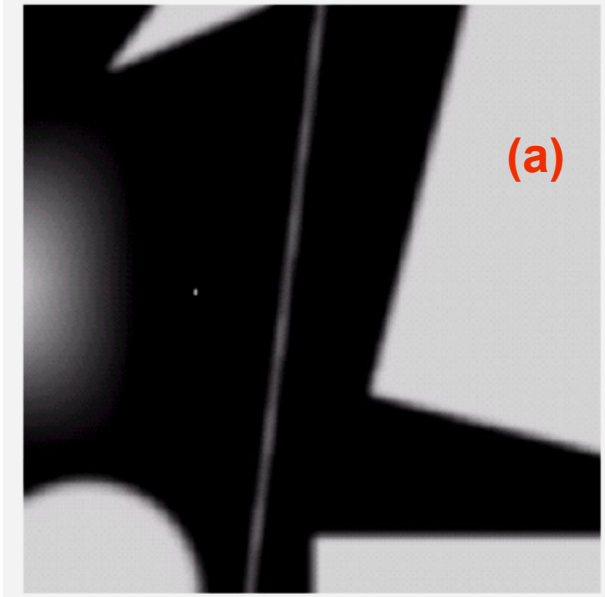
$$\frac{\partial f}{\partial} = f(x+1) - f(x)$$

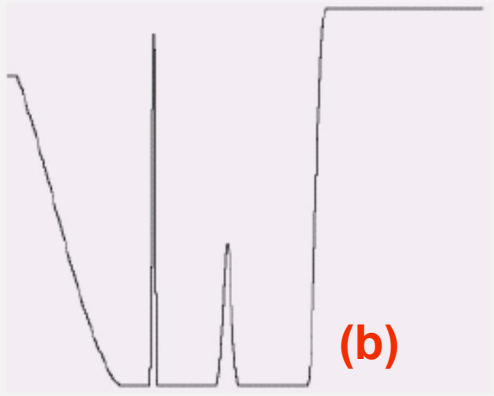
$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

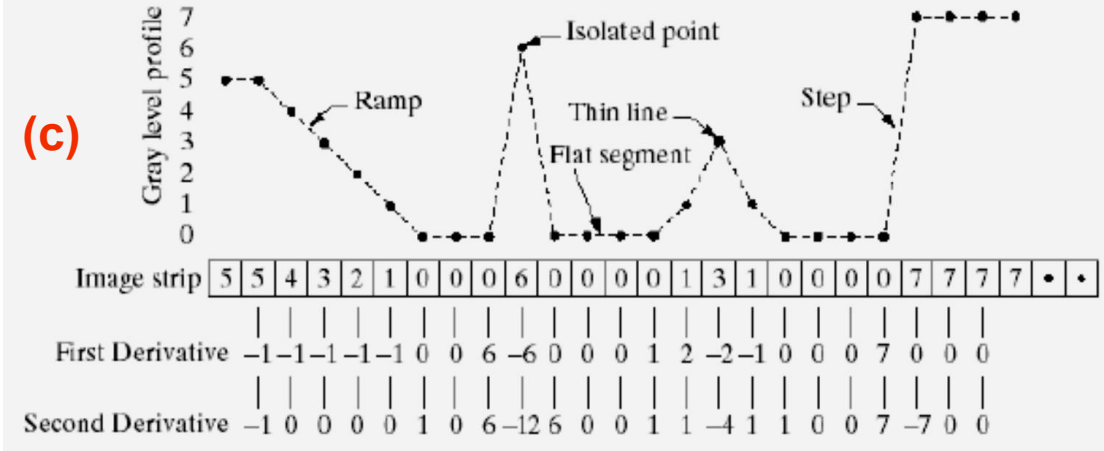
- Derivatives of a digital function are defined in terms of differences
- Various ways to define these differences

- Any definition for a first derivative
  - must be zero in flat segments (areas of constant gray-level values);
  - must be nonzero at the onset of a gray-level step or ramp; and
  - must be nonzero along ramps
- Any definition for a second derivative
  - must be zero in flat areas;
  - must be nonzero at the onset & end of a gray-level step or ramp; &
  - must be zero along ramps of constant slope

First- and Second- Order Derivatives in the context of Image Processing





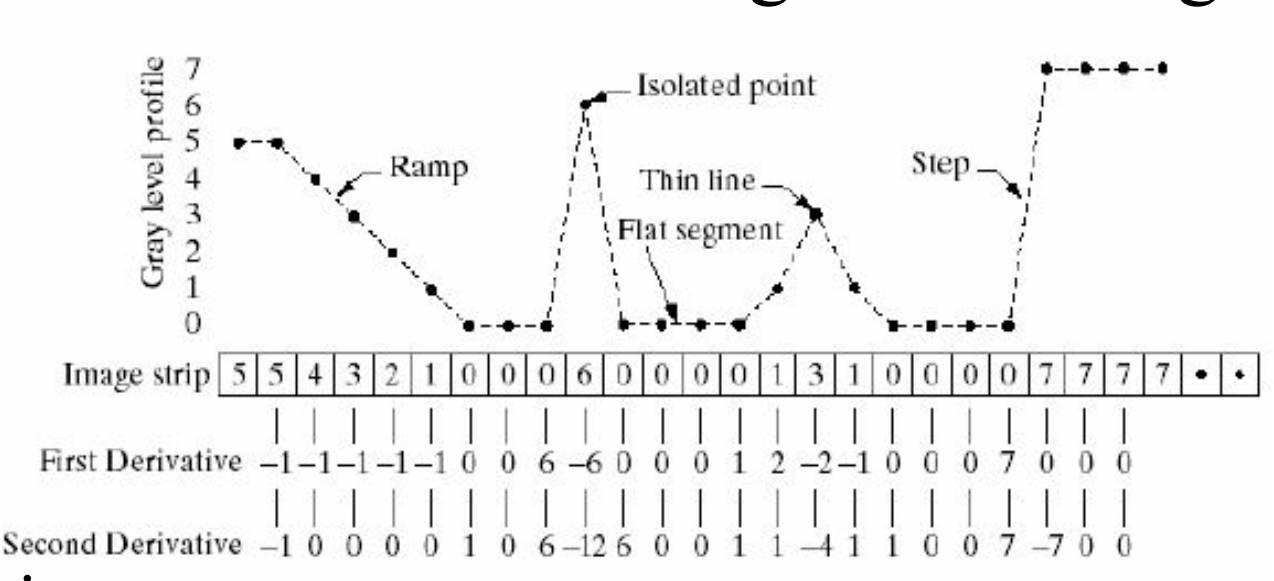


(a) A simple image

(b) 1-D horizontal gray level profile along the center of the image and including the isolated noise point

(c) Simplified profile (the points are joined by dashed lines to simplify interpretation)

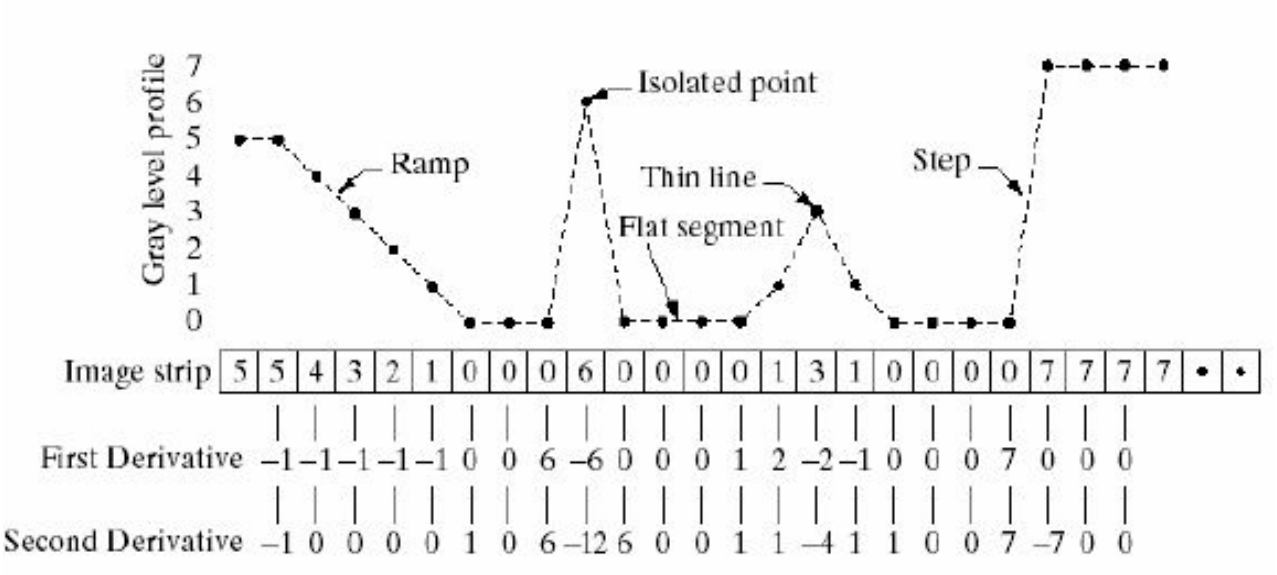
First- and Second- Order Derivatives in context of Image Processing



Observations:

- First order derivative is nonzero along the entire ramp, while the second order derivative is nonzero only at the onset and end of the ramp
- Because edges in an image resemble this type of transition, we can say that First order derivatives produce “thick” edges and second order derivatives produce finer edges

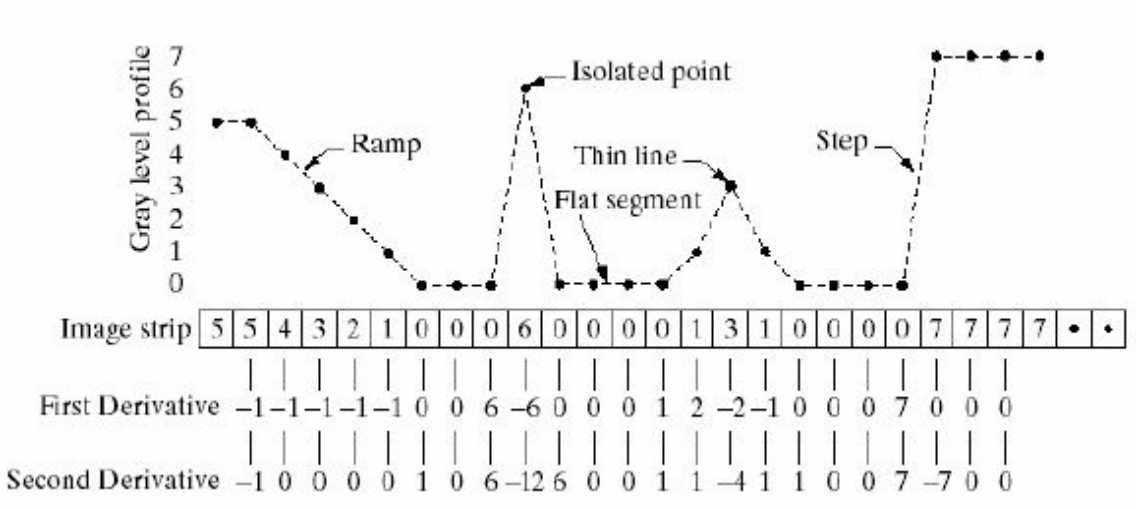
First- and Second- Order Derivatives in context of Image Processing



Observations:

- Isolated noise point
  - The response at and around the point is much stronger for the second order derivative than for the First order derivative
  - So second order derivative enhance fine detail much more than a first order derivative

First- and Second- Order Derivatives in context of Image Processing



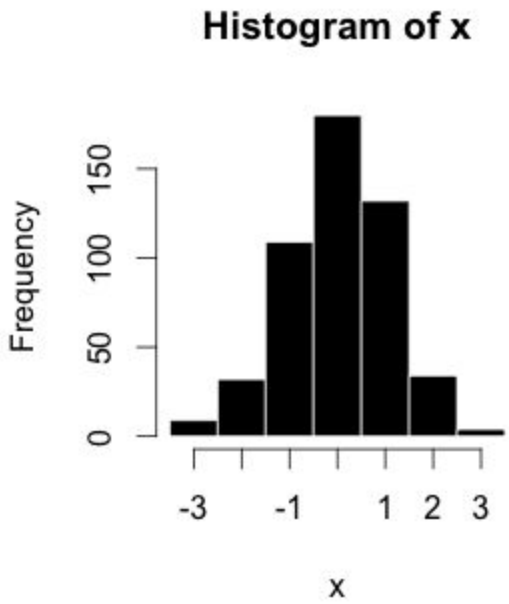
Observations:

- The response of the two derivatives is the same at the gray-level step
- In most cases when the transition into a step is not from zero, the second derivative will be weaker
- The sign of the second derivative changes at the onset and end of a step or ramp (zero crossing – useful property for locating edges)
  - The second derivative has a transition from positive back to negative
  - It would produce a double edge one pixel thick, separated by zeros

Histogram based Image Enhancement

What are histograms

Bin/Interval	Count/Frequency
−3.5 to −2.51	9
−2.5 to −1.51	32
−1.5 to −0.51	109
−0.5 to 0.49	180
0.5 to 1.49	132
1.5 to 2.49	34
2.5 to 3.49	4



Histograms in Image

Histogram Processing

- Histogram of a digital image with gray levels in the range [0, L-1] is a discrete function
  - $h(r_k) = n_k$
- Where
  - r_k : the k^{th} 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

Normalized Histogram

- Dividing each of histogram at gray level r_k by the total number of pixels in the image, n

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

- For $k = 0,1,\ldots,L\text{-}1$
- $p(r_k)$ gives an estimate of the probability of occurrence of gray level r_k
- The sum of all components of a normalized histogram is equal to 1
- Total number of pixels $n = N*M$

Histogram Processing

- Example: 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

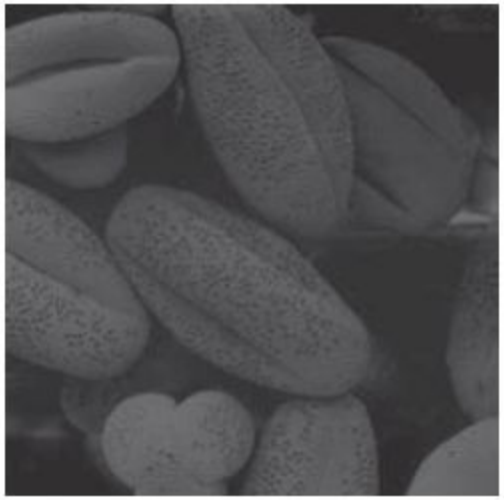
0: 1/16 4: 3/16

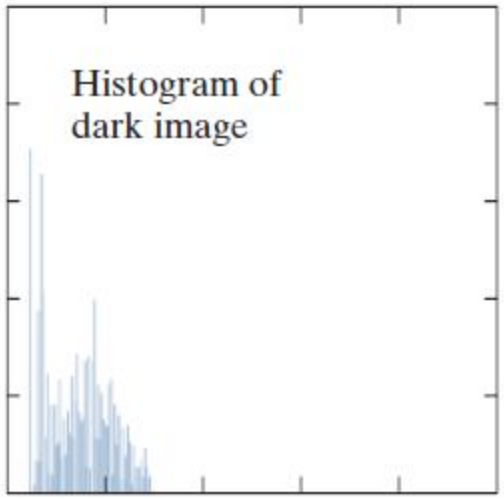
1: 3/16 5: 0/16

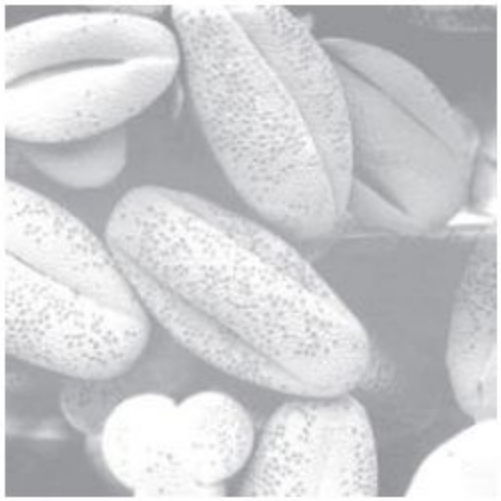
2: 2/16 6: 3/16

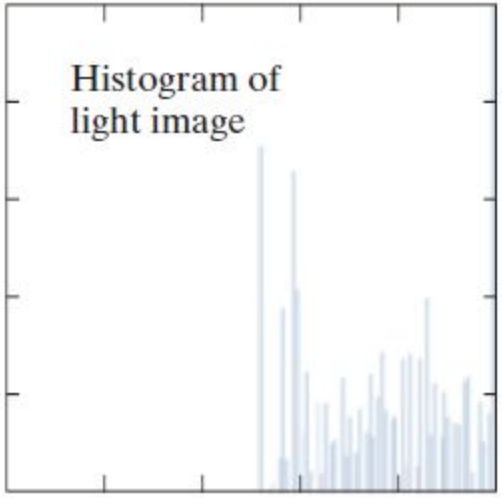
3: 3/16 7: 1/16

Histogram Example





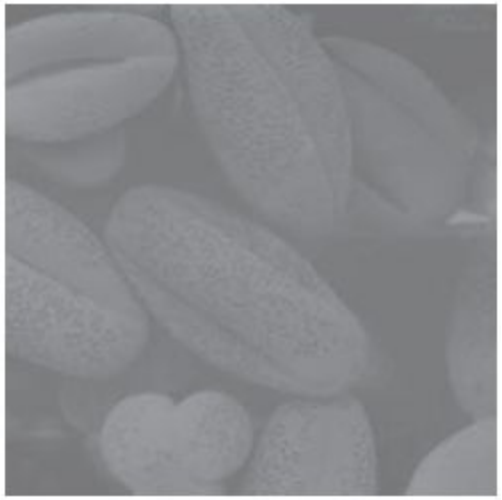


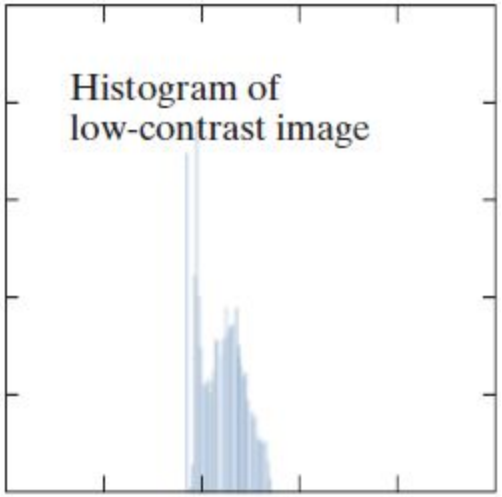


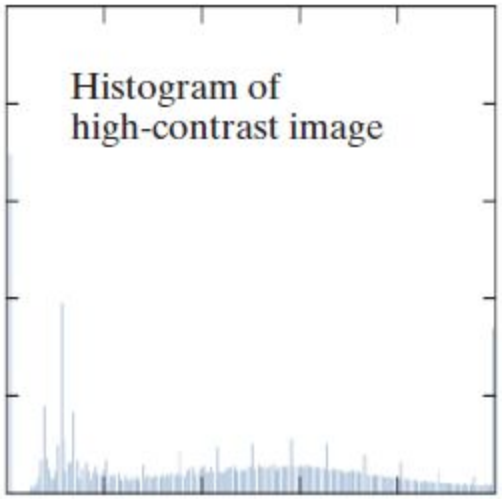
Dark Image:
Components of histogram are concentrated on the low side of the gray scale

Light Image:
Components of histogram are concentrated on the high side of the gray scale.

Histogram Example







Low-Contrast Image: Histogram is narrow and centered toward the middle of the gray scale

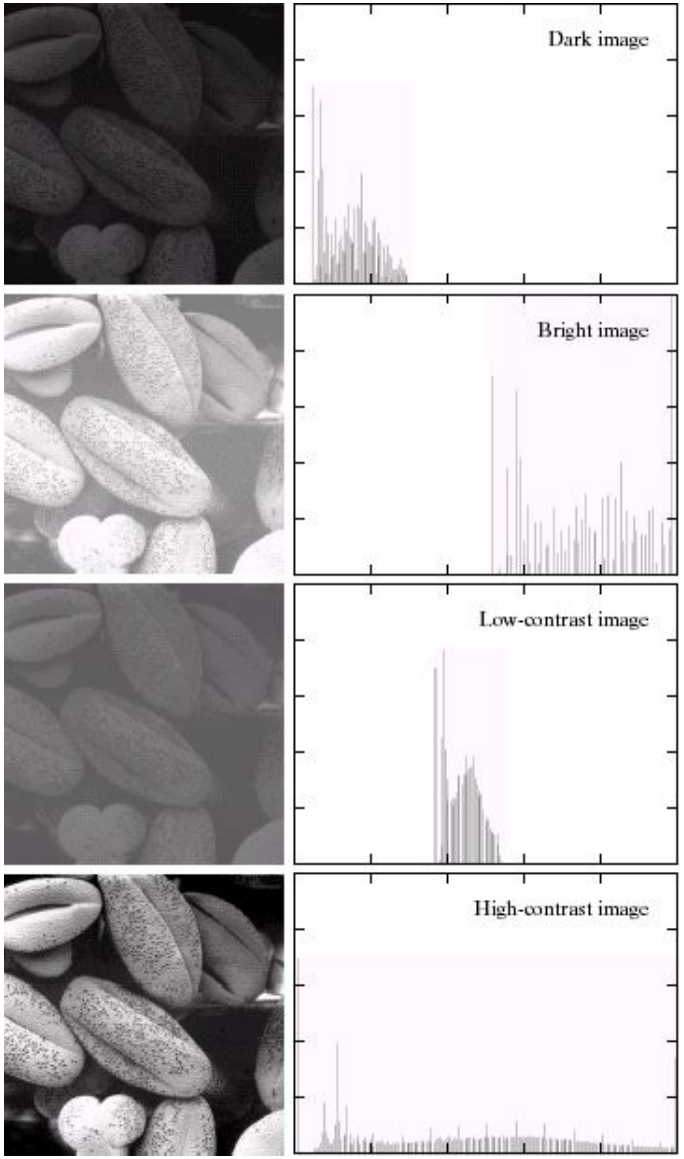
High-Contrast Image: Histogram covers broad range of the gray scale and the distribution of pixels is not too far from uniform (very few vertical lines being much higher than the others)

Histogram Example

A selection of images and their histograms

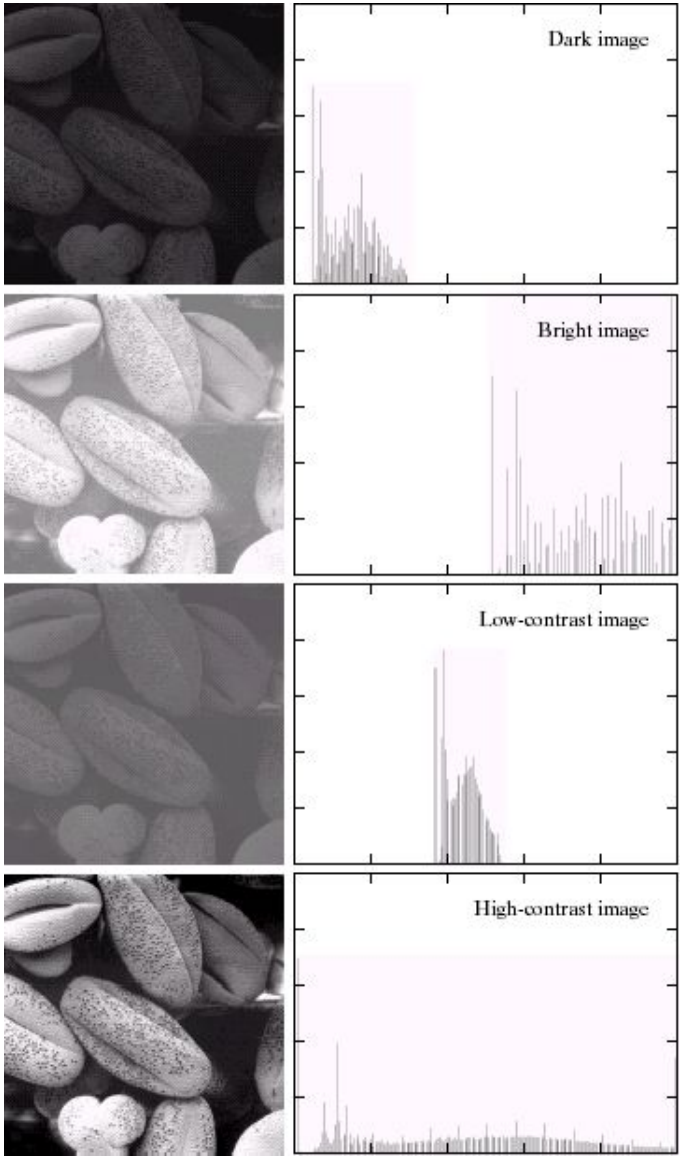
Notice the relationships between the images and their histograms

Note that the high contrast image has the most evenly spaced histogram



Histogram Example

- Observation from these histograms
  - –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

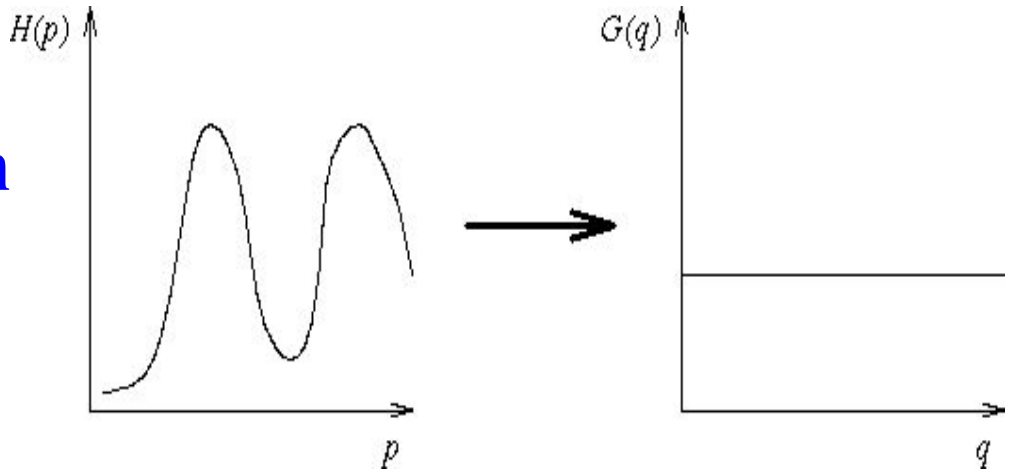


Histogram Equalization

- Increases dynamic range of an image
- Enhances contrast of image to cover all possible grey levels
- Ideal histogram = flat (equalized)
  - same no. of pixels at each grey level

Ideal no. of pixels at each grey level= $\frac{N*M}{L}$

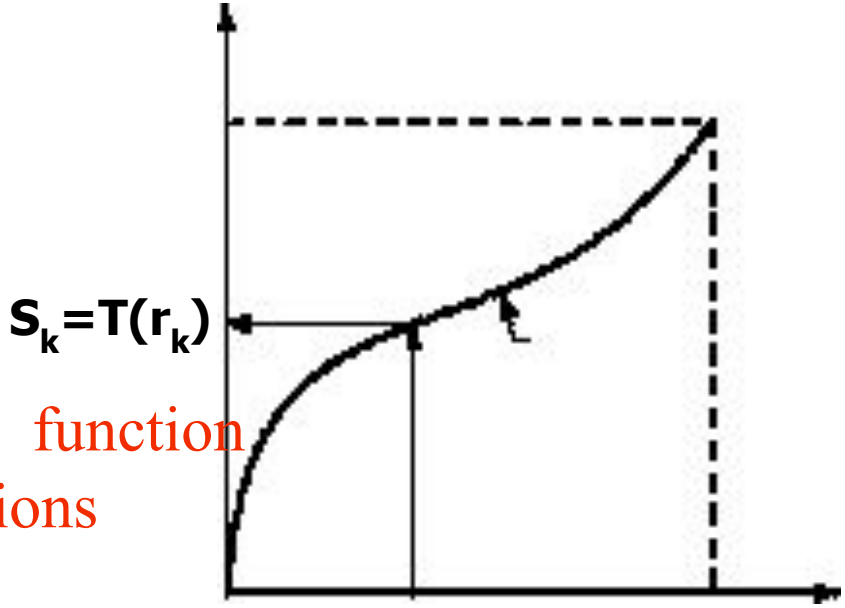
Typical histogram



Ideal histogram

Histogram Equalization

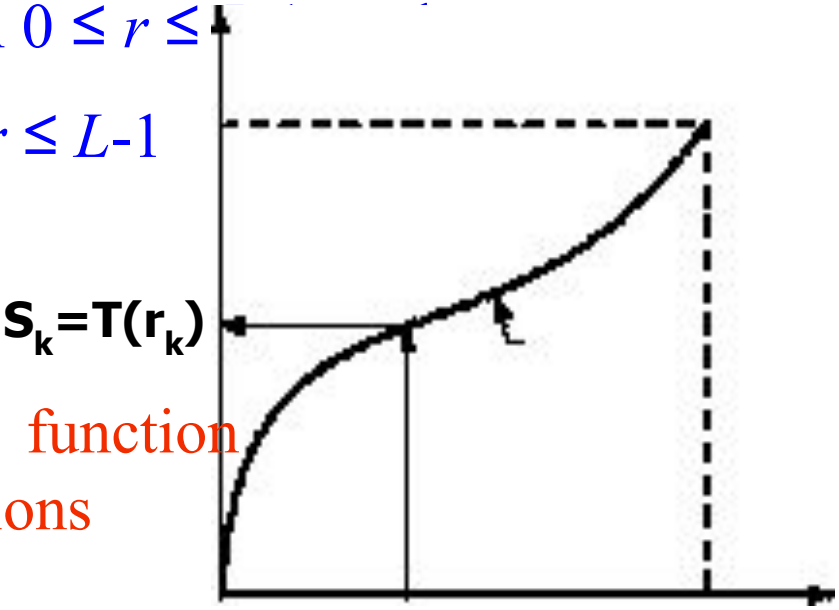
- To get the mapping of the form $s = T(r)$, analyze ideal, continuous case first, assuming
  - $0 \leq r \leq L\text{-}1$ and $0 \leq s \leq L\text{-}1$



A gray level transformation function satisfying two conditions

Histogram Equalization

- To get the mapping of the form $s = T(r)$, analyze ideal, continuous case first, assuming
  - $0 \le r \le L\text{-}1$ and $0 \le s \le L\text{-}1$
  - Assume that
    - (a). $T(r)$ is single-valued and monotonically increasingly in the interval $0 \le r \le$
    - (b). $0 \le T(r) \le L\text{-}1$ for $0 \le r \le L\text{-}1$



A gray level transformation function satisfying two conditions

Meaning of Two Conditions of $T(r)$

- Single-valued (one-to-one relationship) guarantees that the inverse transformation will exist
- Monotonicity condition preserves the increasing order from black to white in the output image thus it won't cause a negative image
- Not monotonically increasing could result in at least a section of the intensity range being inverted, this produces some inverted gray levels in the output image
- $0 \leq T(r) \leq L\text{-}1$ for $0 \leq r \leq L\text{-}1$ guarantees that the output gray levels will be in the same range as the input levels
- The inverse transformation from s back to r is
  - $r = T^{-1}(s)$: $0 \leq s \leq 1$

Probability Density Function

- The intensity levels in an image may be viewed as random variables in the interval [0, L-1]
- PDF (probability density function) is one of the fundamental descriptors of a random variable

Random Variables

- If a random variable x is transformed by a monotonic transformation function $T(x)$ to produce a new random variable y
- The probability density function of y can be obtained from knowledge of $T(x)$ and the probability density function of x, as follows:

$$p_y(y) = p_x(x) \left| \frac{dx}{dy} \right|$$

absolute value

- A function $T(x)$ is monotonically increasing if $T(x_1) < T(x_2)$ for $x_1 < x_2$, and
- A function $T(x)$ is monotonically decreasing if $T(x_1) > T(x_2)$ for $x_1 < x_2$
- The preceding equation is valid if $T(x)$ is an increasing or

Applied to Image

- If $p_r(r)$ and $T(r)$ are known and $T^{-1}(s)$ satisfies condition (a) then $p_s(s)$ can be obtained using a formula :
- where

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

  - $p_r(r)$ denote the PDF of random variable r
  - $p_s(s)$ denote the PDF of random variable s

The PDF of the transformed variable s is determined by the PDF of the input image and by the chosen transformation function

Transformation Function

- A transformation function is a cumulative distribution function (CDF) of random variable *r*:

$$s = T(r) = (L-1)\int_{0}^{r} p_r(w)dw$$

where *w* is a dummy variable of integration

The right side is recognized as the cumulative distribution function of random variable *r*

Note: $T(r)$ depends on $p_r(r)$

Cumulative Distribution Function

$$s = T(r) = (L-1)\int_0^r p_r(w)dw$$

- CDF is an integral of a probability function (always positive) is the area under the function
- Thus, CDF is always single valued and monotonically increasing
  - Thus, CDF satisfies the condition (a)
- Similarly, the integral of a probability density function for variables in the range [0, L-1] also is in the range [0, L-1]
  - Thus, CDF satisfies the condition (b) also

Finding $p_s(s)$ from given $T(r)$

$$\frac{ds}{(r)} = \frac{dT}{dr}$$

$$\frac{dr}{dr} = (L-1)\frac{d}{dr}\left[\int_0^r p_r(w)dw\right]$$

$$= (L-1)\, p_r(r)$$

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

$$= p_r(r)\left|\frac{1}{(L-1)\, p_r(r)}\right|$$

$$= \frac{1}{(L-1)} \quad \text{where } 0 \le s \le L-1$$

Substitute and yield

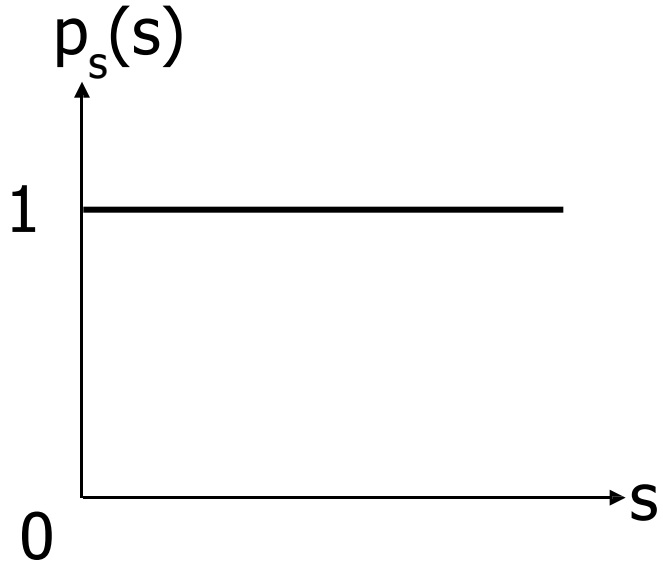
$$p_s(s)$$

- As $p_s(s)$ is a probability function, it must be zero outside the interval [0, L-1] in this case because its integral over all values of s must equal L-1
- Called $p_s(s)$ as **a uniform probability density function**
- $p_s(s)$ is always a uniform, independent of the form of $p_r(r)$

$$s = T(r) = (L-1)\int_0^r p_r(w)dw$$

yields

a random variable *s* characterized by a uniform probability function



Discrete Transformation Function

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

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

- The discrete version of transformation

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

Histogram Equalization

- Thus, an output image is obtained by mapping each pixel with level r_k in the input image into a corresponding pixel with level s_k in the output image
- The transformation (mapping) is called Histogram Equalization or Histogram Linearization
- In discrete space, it cannot be proved in general that this discrete transformation will produce the discrete equivalent of a uniform probability density function, which would be a uniform histogram

Histogram Equalization: Example 1

- Suppose that a 3-bit image ($L = 8$) of size 64×64 pixels ($MN = 4096$) has the intensity distribution in Table.

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

Histogram Equalization: Example 2

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.

Formula: $S_k = (L-1) \sum_0^k p_r(r_k)$

$S_0 = 7p_r(r_0) = 1.33$

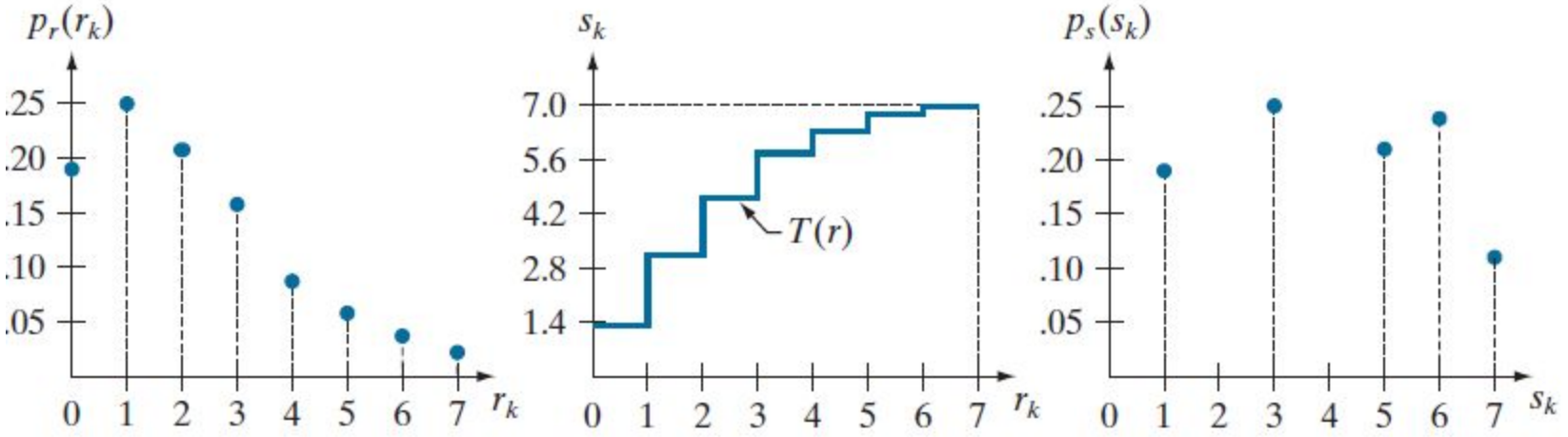
$S_1 = 7\,[p_r(r_0) + p_r(r_1)] = 3.08$

$S_2 = 7\,[p_r(r_0) + p_r(r_1) + p_r(r_2)] = 4.55,$

and so on……

$s_0 = 1.33 \rightarrow 1$ $s_2 = 4.55 \rightarrow 5$ $s_4 = 6.23 \rightarrow 6$ $s_6 = 6.86 \rightarrow 7$

$s_1 = 3.08 \rightarrow 3$ $s_3 = 5.67 \rightarrow 6$ $s_5 = 6.65 \rightarrow 7$ $s_7 = 7.00 \rightarrow 7$



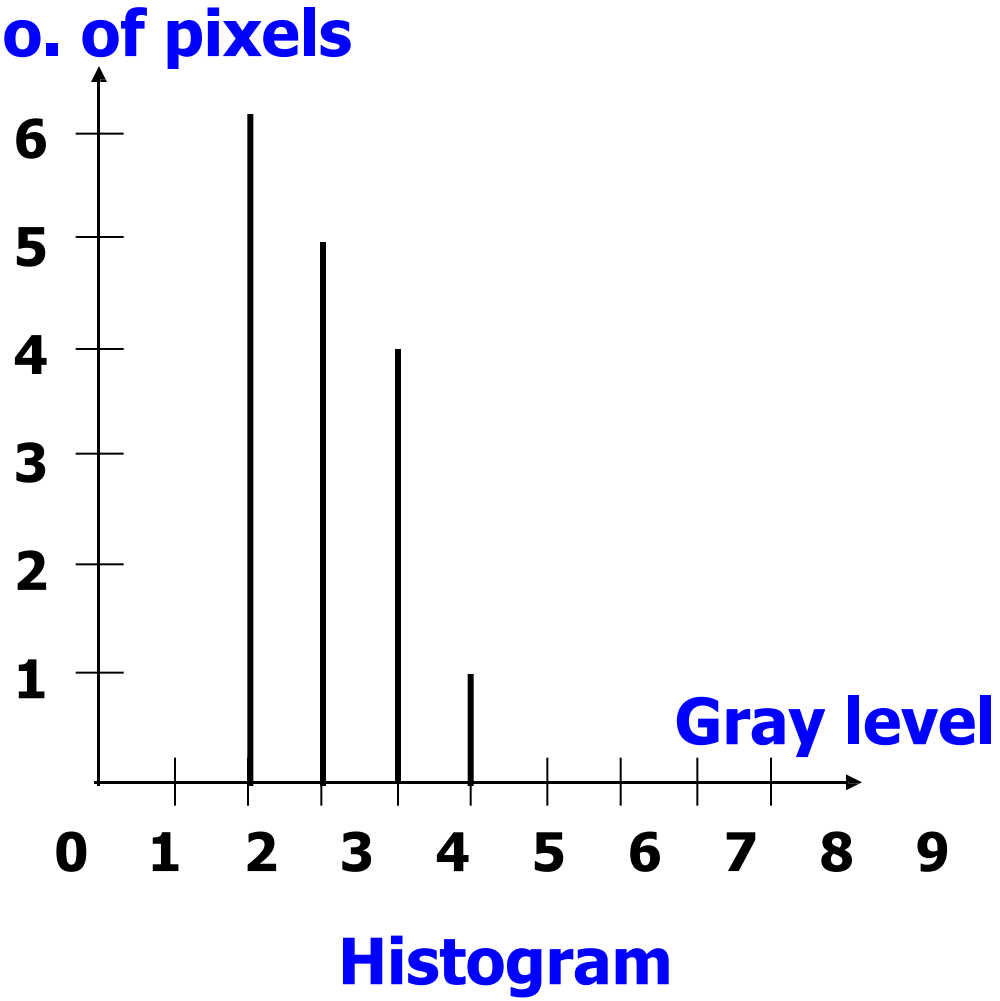
Histogram equalization. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

Histogram Equalization: Example 2

2	3	3	2
4	2	4	3
3	2	3	5
2	4	2	4

4x4 image

Gray scale = [0,9]



Histogram Equalization: Example 2

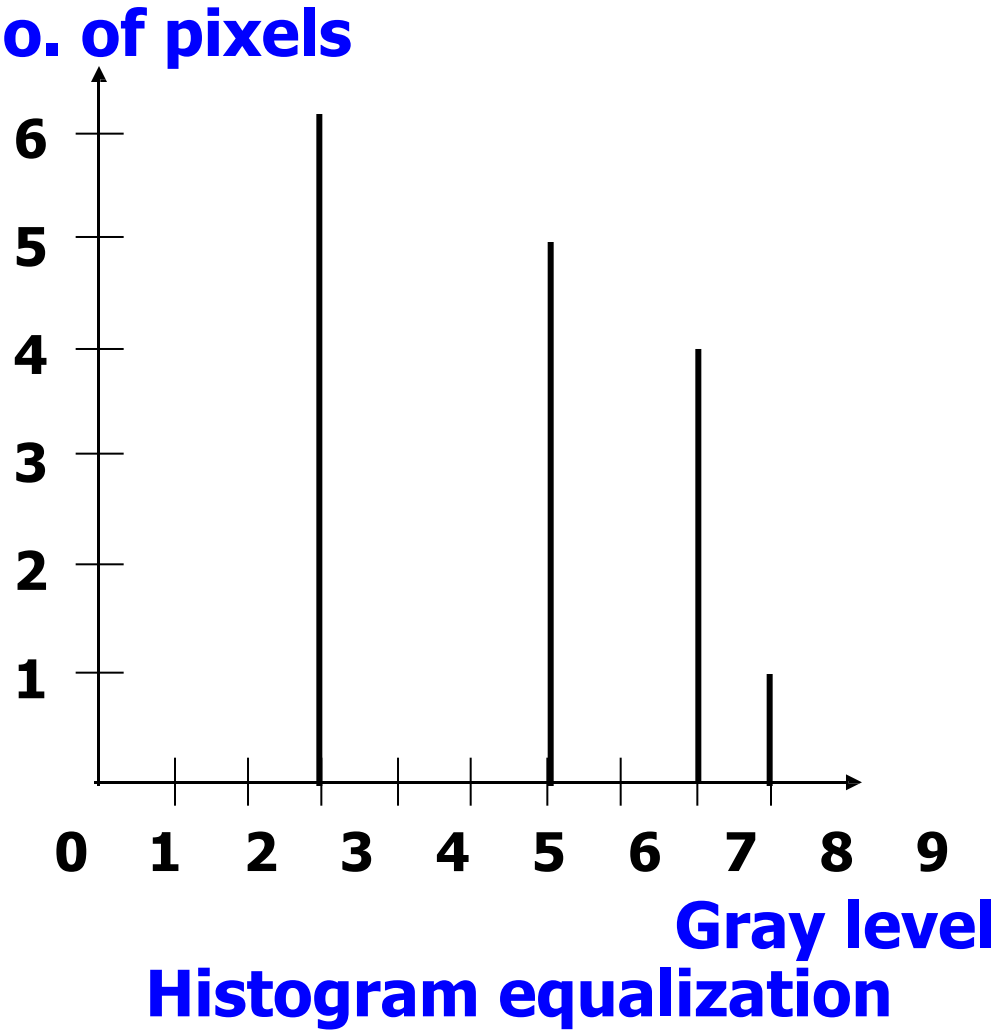
Gray Level (r)	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
No. of pixels	**0**	**0**	**6**	**5**	**4**	**1**	**0**	**0**	**0**	**0**
$\sum_{j=0}^{k} n_j$	**0**	**0**	**6**	**11**	**15**	**16**	**16**	**16**	**16**	**16**
$s = \sum_{j=0}^{k} \frac{n_j}{n}$	**0**	**0**	**6 / 16**	**11 / 16**	**15 / 16**	**16 / 16**	**16 / 16**	**16 / 16**	**16 / 16**	**16 / 16**
			3.3	**6.1**	**8.4**					

Histogram Equalization: Example 2

3	6	6	3
8	3	8	6
6	3	6	9
3	8	3	8

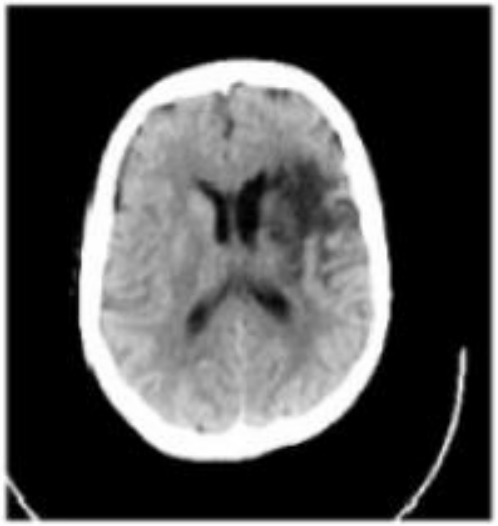
Output image

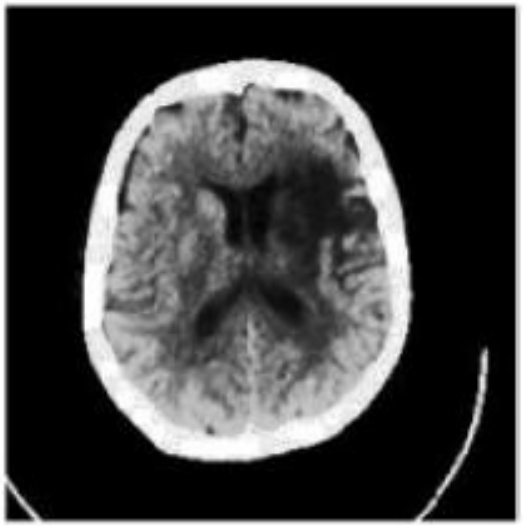
Gray scale = [0,9]

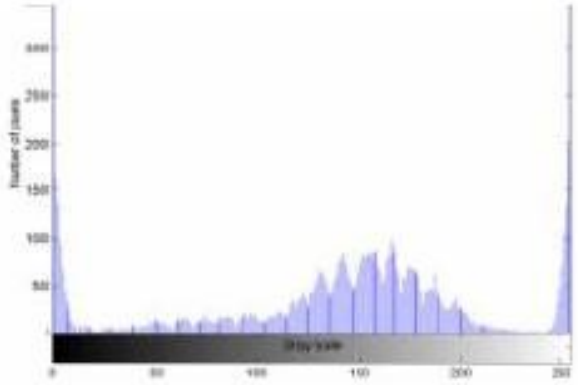


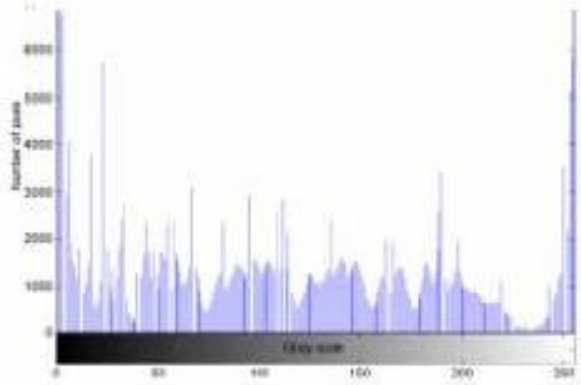
Histogram equalization

Histogram Equalization









Before HE

After HE

Histogram Matching (Specification)

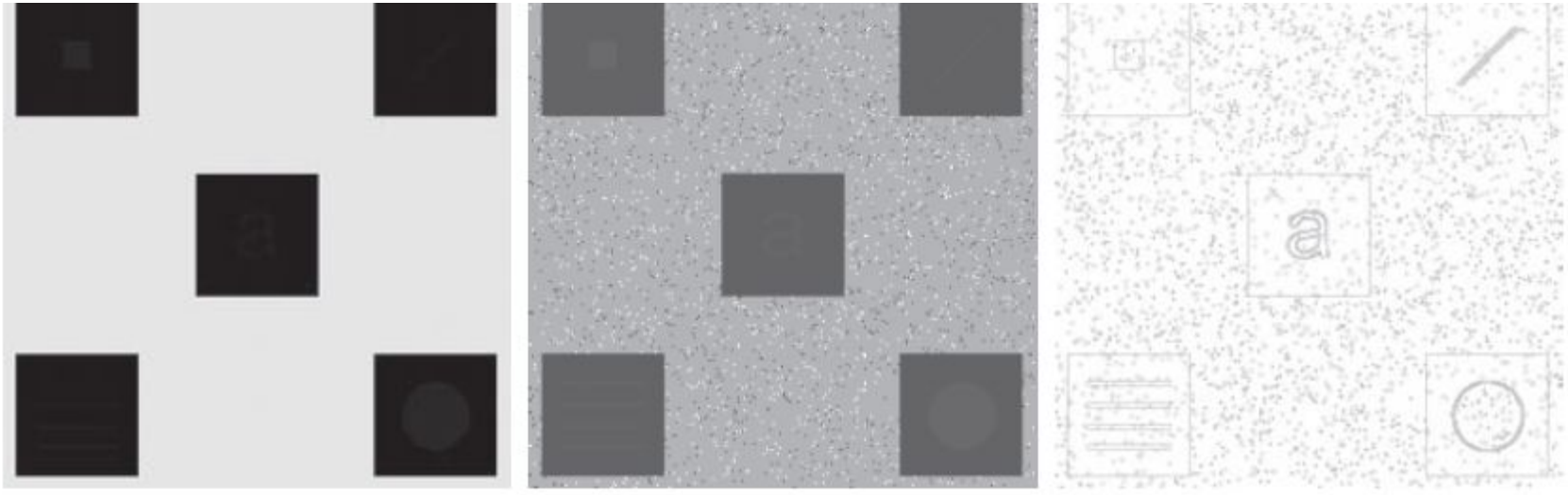
- Histogram equalization has a disadvantage which is that it can generate only one type of output image.
- With Histogram Specification, we can specify the shape of the histogram that we wish the output image to have.
- It doesn't have to be a uniform histogram

Global Vs Local Enhancement

- Histogram processing: global approach suitable for overall enhancement
  - Pixels are modified by a transformation function based on the gray-level content of an entire image
- Sometimes, we may need to enhance details over small areas in an image, which is called a local enhancement
  - Pixels in these areas may have negligible influence on the computation of a global transformation; and hence it is not necessary to get the desired local enhancement
- For local enhancement, transformation should be based on gray-level distribution in the neighborhood of every pixel

Local HE Example

- 8-bit, 512 × 512 image consisting of five black squares on a light gray background.



(a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization (3x3 neigbourhood).

Use of Histogram Statistics for Image Enhancement

- 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***

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

$$m = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)$$

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

$$\sigma^2 = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x,y) - m]^2$$

Use of Histogram Statistics for Image Enhancement

- Two uses of the mean and variance for enhancement purposes
  - Global mean and variance are measured over an entire image and are used primarily for gross adjustments of overall intensity and 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 in the image

Use of Histogram Statistics for Image Enhancement

- Local mean and variance

$$m_{S_{xy}} = \sum_{i=0}^{L-1} r_i \, p_{S_{xy}}(r_i) \qquad \sigma_{S_{xy}}{}^2 = \sum_{i=0}^{L-1} ((r_i - m_{S_{xy}})^2 \, p_{S_{xy}}(r_i)$$

- where
  - (x, y): coordinates of a pixel in an image
  - S_{xy}: neighborhood (sub image of specified size), centered at (x, y)
  - $p_{S_{xy}}(r_i)$ is the histogram of pixels in region S_{xy}

Use of Histogram Statistics for Image Enhancement

- Local enhancement method

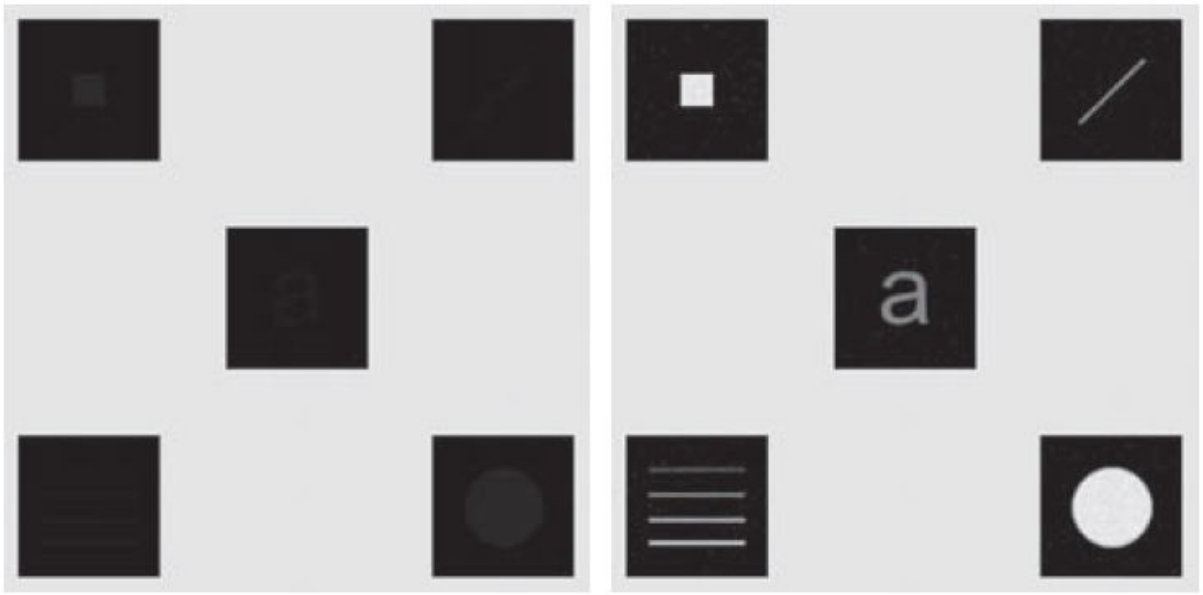
$$g(x,y) = \begin{cases} E \cdot f(x,y) & \text{if } m_{xy} \le k_0 m_G, \quad k_1 \sigma_G \le \sigma_{xy} \le k_2 \sigma_G \\ f(x,y) & \text{otherwise} \end{cases}$$

m_G: Global mean $\quad \sigma_G$: Global variance

Local enhancement method

$$g(x,y) = \begin{cases} C\,f(x,y) & \text{if } k_0 m_G \le m_{S_{xy}} \le k_1 m_G \text{ AND } k_2 \sigma_G \le \sigma_{S_{xy}} \le k_3 \sigma_G \\ f(x,y) & \text{otherwise} \end{cases}$$

- For example, if our focus is on areas that are darker than one-quarter of the mean intensity, we would choose $k0 = 0$ and $k1 = 0.25$
- To enhance a dark area of low contrast, we might choose $k2 = 0$ and $k3 = 0.1$

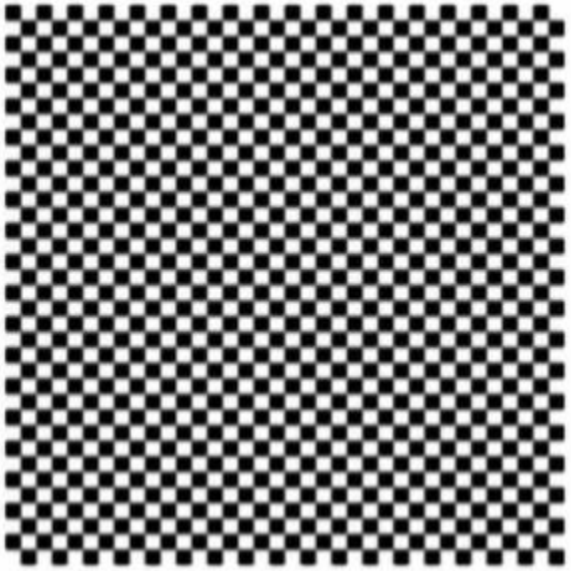


(a) Original image. (b) Result of local enhancement based on local histogram statistics.

Spatial Filtering

Spatial Frequencies

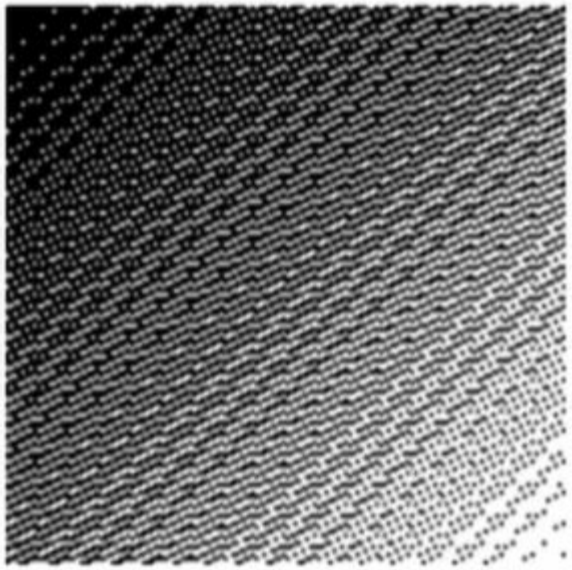
Radical change in intensity



High frequency Image

Changes in tone is abrupt over small areas

Slowly varying change in intensity

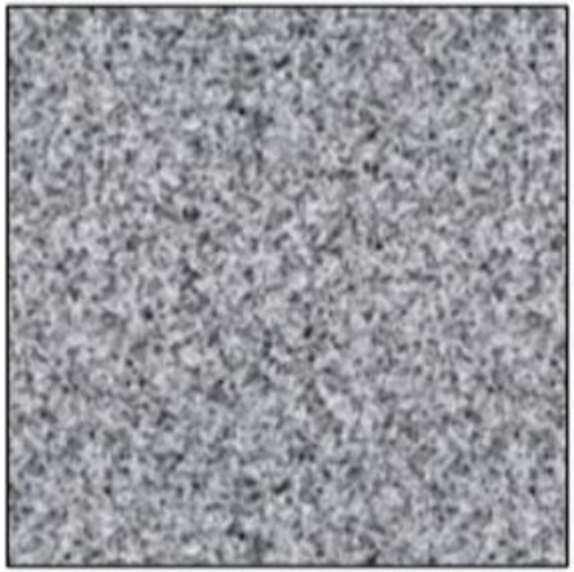


Low frequency Image

Little variation in tone over several pixels

Spatial Frequencies

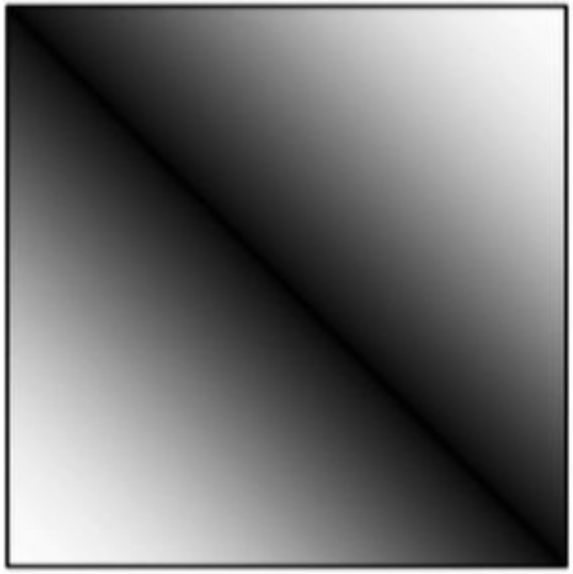
Radical change in intensity



High frequency Image

Changes in tone is abrupt over small areas

Slowly varying change in intensity



Low frequency Image

Little variation in tone over several pixels

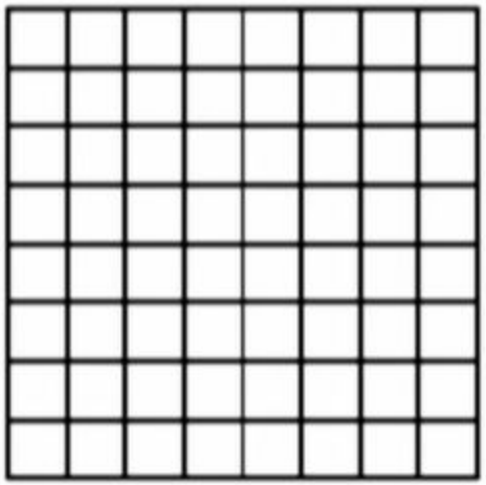
Spatial Frequency Definitions

- Number of changes in intensity value per unit distance for any particular part of the image.

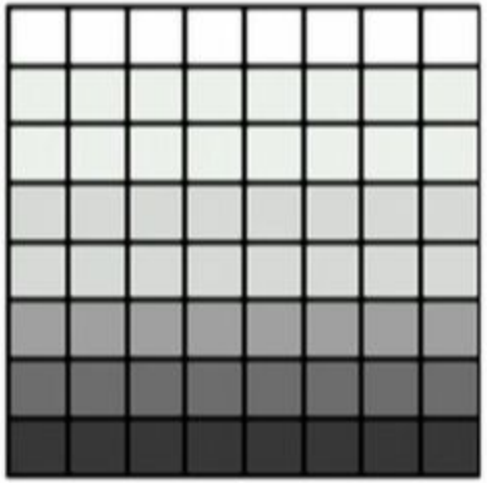
- Image is composed of a
  - Low frequency details- basic details, few changes over a given area
  - High Frequency details – fine details, dramatic changes over a given area

Spatial Frequency Examples

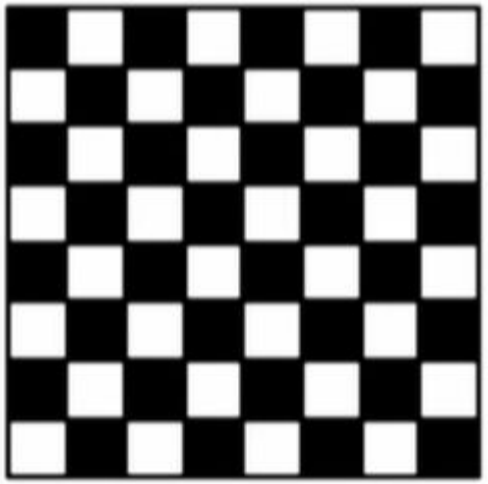
- Zero spatial frequency - flat image, all pixels same value.
- Low spatial frequency - smoothly varying grayscale image.
- High spatial frequency- an image consisting of check board fashion



Zero spatial frequency

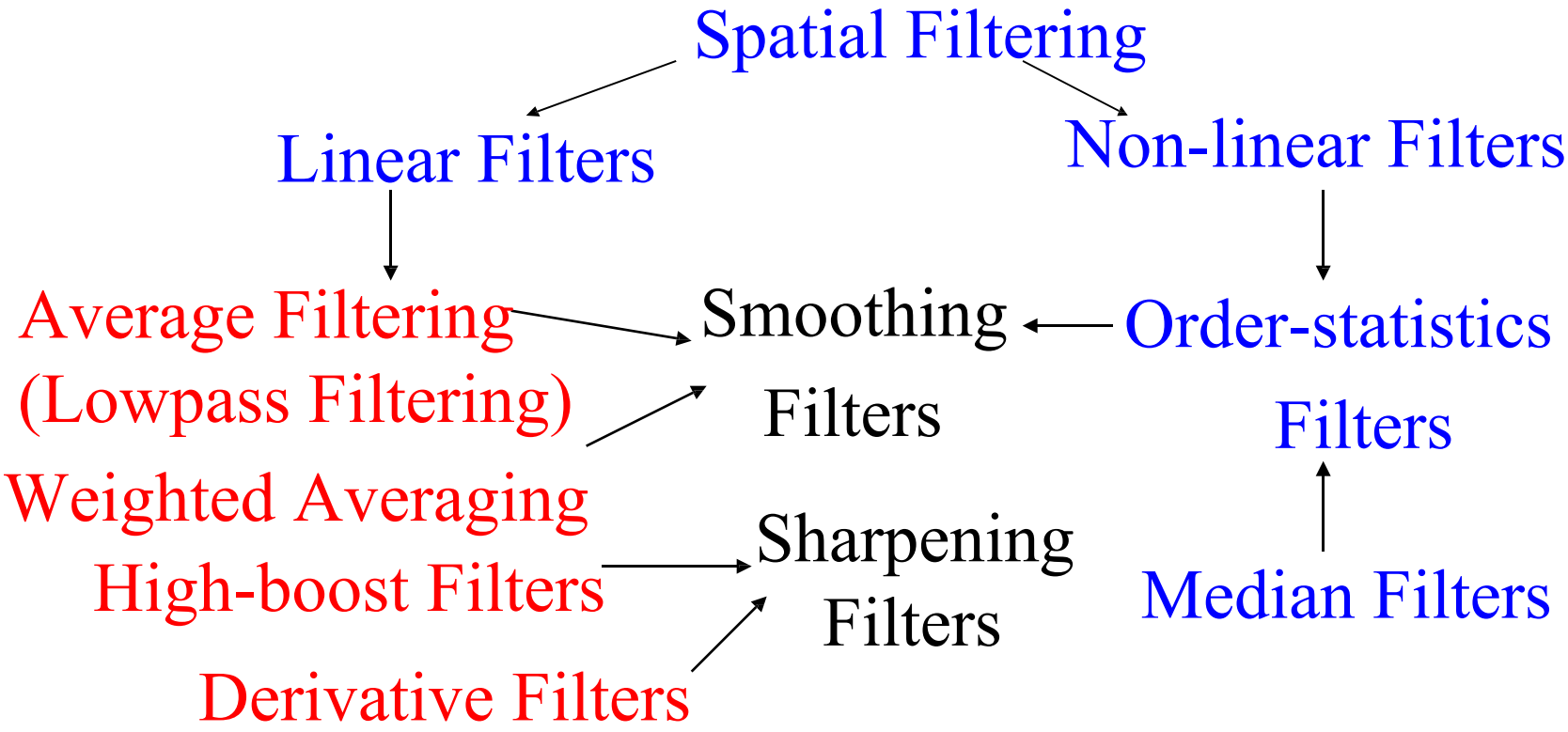


Low spatial frequency



High spatial frequency-

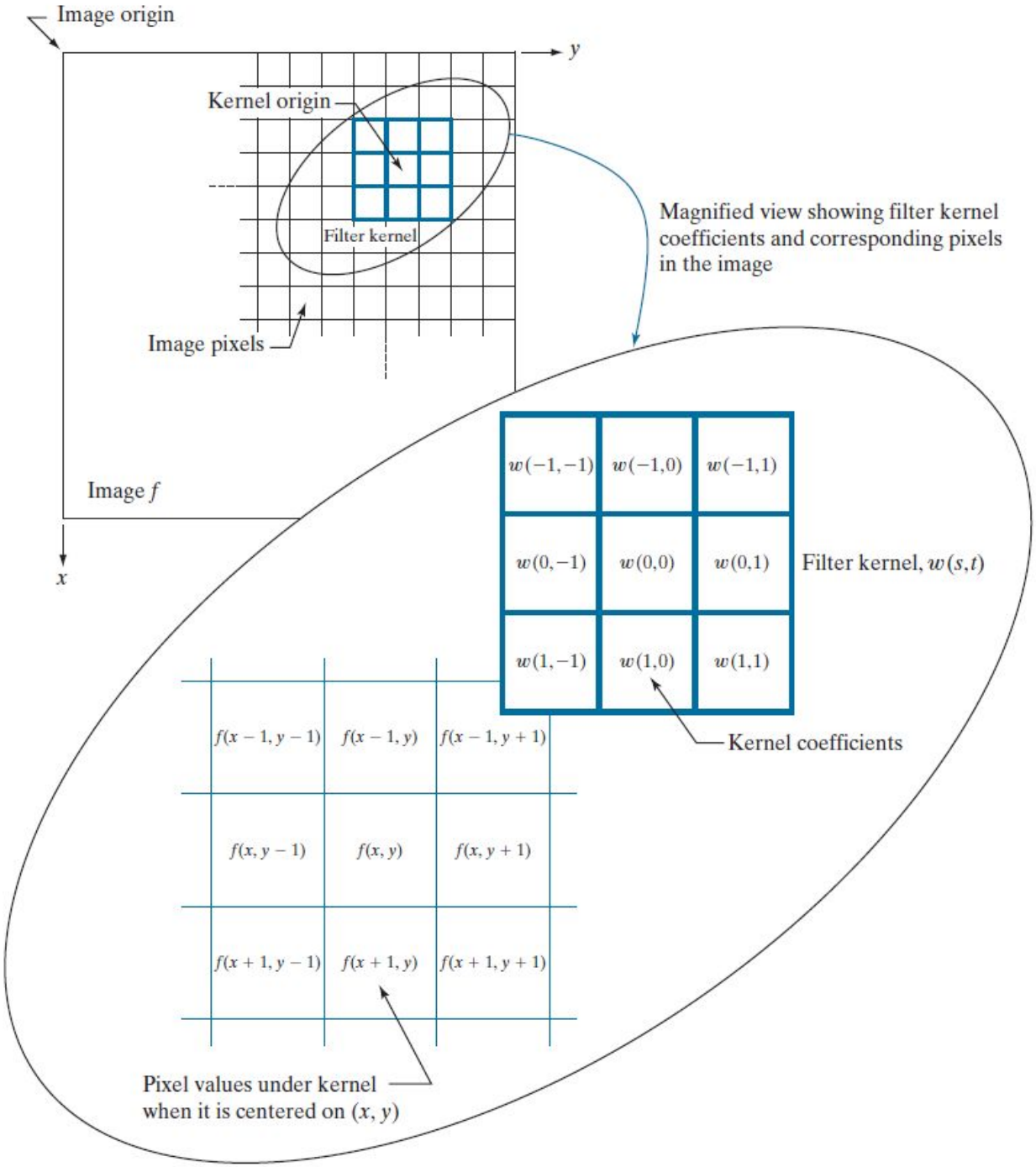
Spatial Filtering



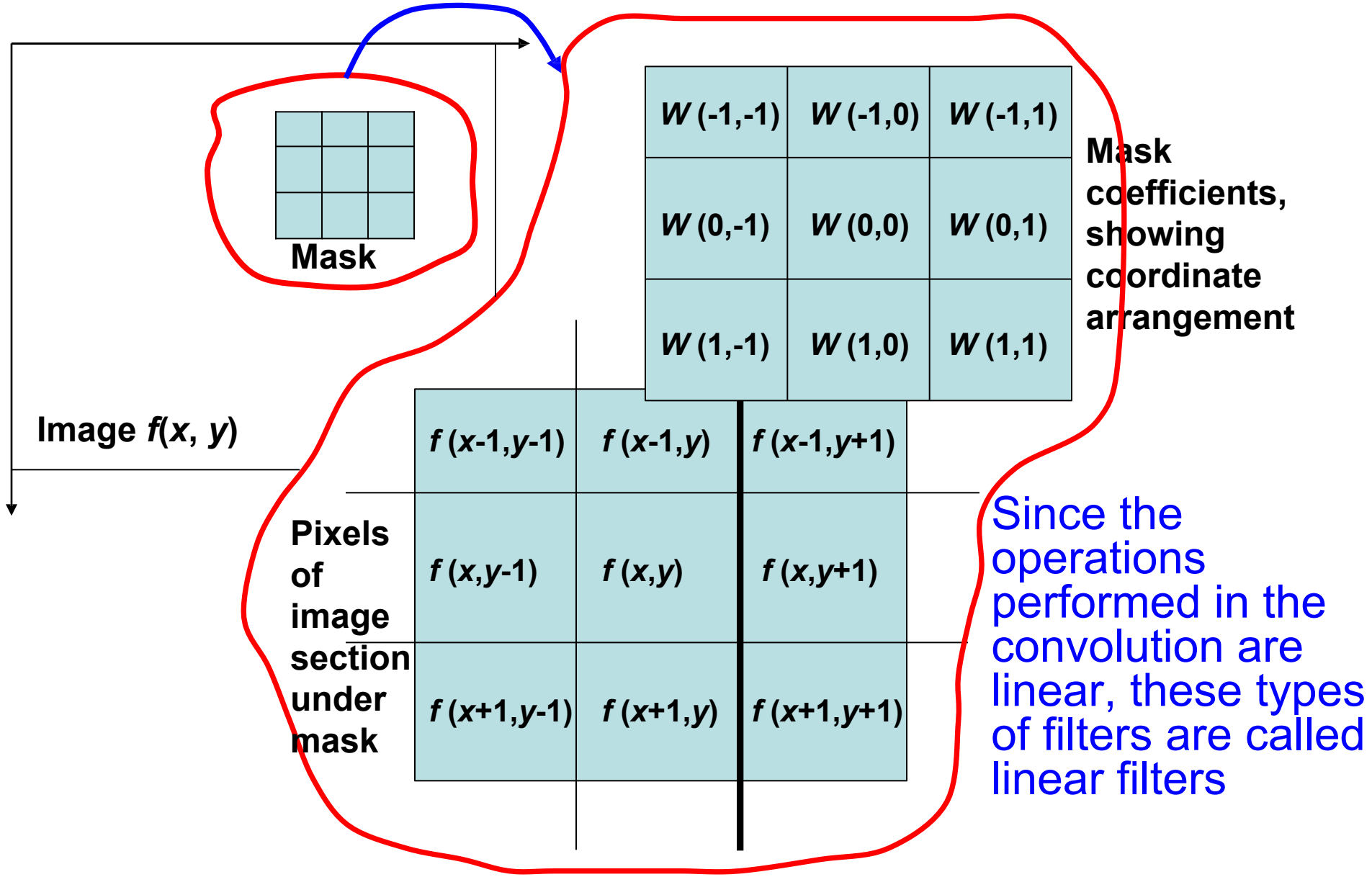
Smoothing filters: used for blurring and for noise reduction

Sharpening filters: used to highlight fine detail in an image or to enhance detail that has been blurred, either in error or as a natural effect of a particular method of image acquisition

Linear Spatial Filtering



Linear Spatial Filtering



Linear Spatial Filtering

- The process consists simply of moving the filter mask from point to point in an image
- At each point (x, y), the response of the filter at that point is calculated using a predefined relationship
- The response is given by a sum of products of the filter coefficients and the corresponding image pixels in the area spanned by the filter mask
- The coefficient w (0,0) coincides with image value f (x, y), indicating that the mask is centered at (x, y) when the computation of the sum of products takes place

Linear Spatial Filtering

- For a 3x3 mask shown in figure, the response, R, of linear filtering with the filter mask at a point (x, y) in the image

$$R = w(-1,-1)\,f(x-1,y-1) + w(-1,0)\,f(x-1,y) + \ldots\ldots$$
$$+ w(0,0)\,f(x,y) + \ldots + w(1,0)\,f(x+1,y)$$
$$+ w(1,1)\,f(x+1,y+1)$$

$w(s,t)$: $(2a+1) \times (2b+1)$ mask

Where, „a“ & „b“ are nonnegative integers

- For a mask of size mxn, we assume that $m = 2a+1$ and $n = 2b+1$
- Focus: mask of odd sizes, smallest meaningful size: 3x3

Linear Spatial Filtering

- In general, Linear filtering of an image *f* of size *M*x*N* with a filter mask of size mxn is given by:

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

where $a = (m-1)/2$ and $b = (n-1)/2$

- To generate a complete filtered image this equation must be applied for $x = 0, 1, 2, \ldots, M-1$ and $y = 0, 1, 2, \ldots, N-1$
  - In this way, we are assured that the mask processes all pixels in the image
- Nonlinear spatial filters also operate on neighborhoods, and the mechanics of sliding a mask past an image are the same. They do not explicitly use coefficients in the sum-of-products

Linear Spatial Filtering

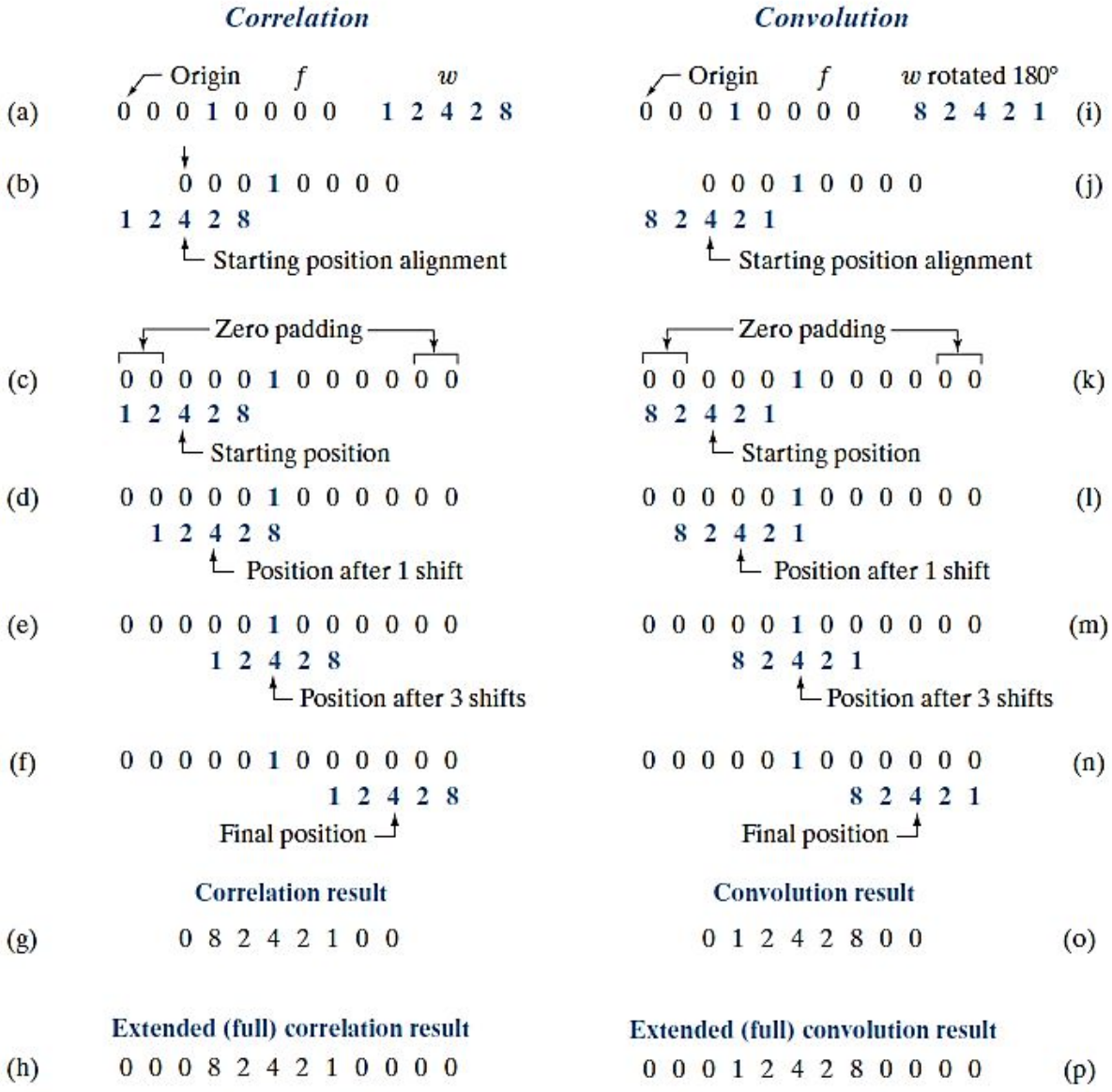
- Border Effects- Special Consideration
  - For a small value of *m* and *n*, it is not that much problem; otherwise we may loose substantial amount of data
  - Common procedures:
    - Border strips are added and set to zero
    - Rows and columns are considered to wrap around
    - Limit the excursion of the center of the filter mask to a distance no less than (*n*-1)/2 pixels from the border of the original image
      - The resulting filtered image will be smaller than the original, but all the pixels in the filtered image will have been processed with the full mask

Spatial Correlation and Convolution

- Correlation is the process of moving a filter mask over the image and computing the sum of products at each location, as explained

- The mechanics of convolution are the same, except that the filter is first rotated by 180^0

Spatial Correlation and Convolution



Spatial Correlation and Convolution

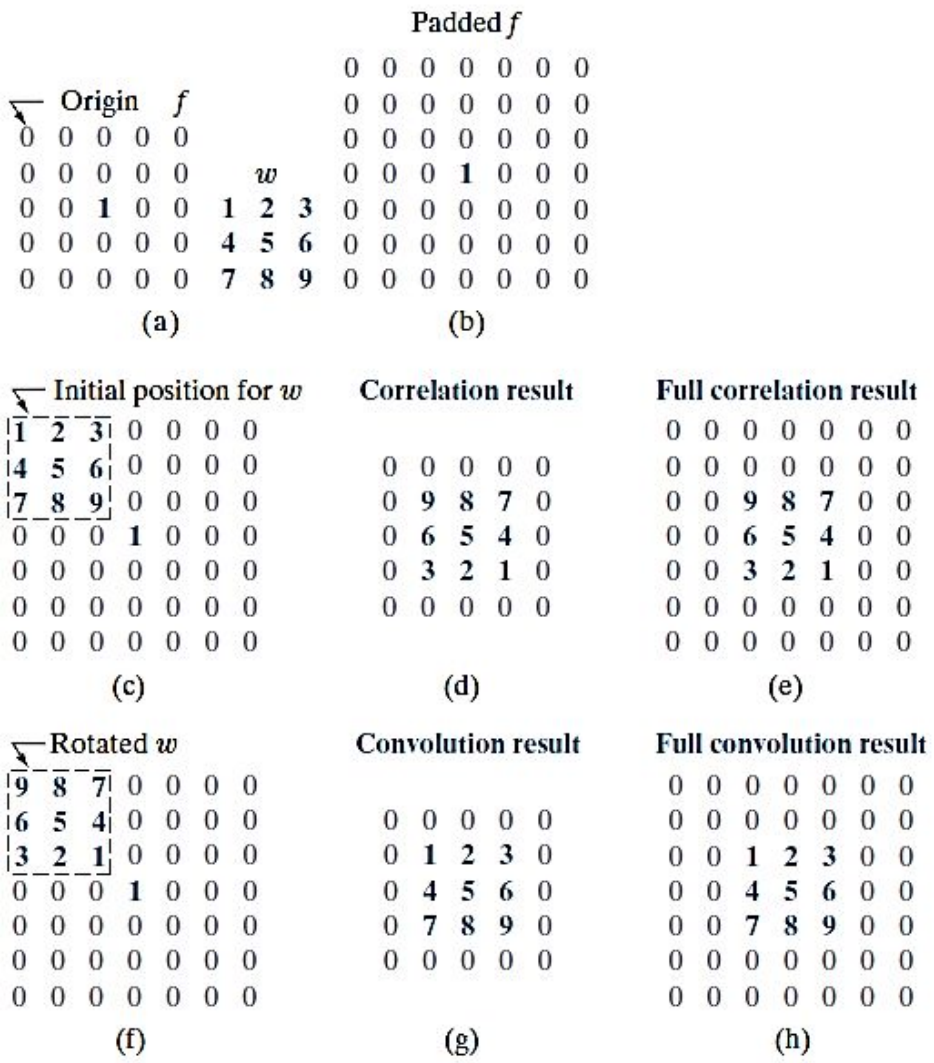


FIGURE 3.30 Correlation (middle row) and convolution (last row) of a 2-D filter with a 2-D discrete, unit impulse. The 0s are shown in gray to simplify visual analysis.

- See that convolution of a function with an impulse copies the function at the location of the impulse
- If the filter mask is symmetric, correlation and convolution yield the same result

Spatial Correlation and Convolution

- Correlation

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

- Convolution

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

where $a = (m-1)/2$ and $b = (n-1)/2$

Minus sign denotes 180^0 rotation of f, which has been done for notational simplicity and to follow convention

Spatial Correlation and Convolution

- Using correlation or convolution to perform spatial filtering is a matter of preference
  - Either can perform the function of the other by a simple rotation of the filter
  - Important is the filter mask used in a given filtering task be specified in a way that corresponds to the intended operation
- The linear spatial filtering discussed here are based on convolution
- Convolution filter, Convolution mask, Convolution kernel spatial filter (not necessarily used for true convolution)
- Convolving a mask with an image is often used to denote the sliding, sum-of-products process

Generating Spatial Filter Mask

- Example:

Consider the image: $\begin{bmatrix} 0 & 1 & 0 \\ & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$

1

Suppose the second row of the image is to be highlighted, what is the mask that is needed?

Generating Spatial Filter Mask

- One interesting observation:
  - Overall effect can be predicted on the general pattern, if one uses convolution mask
- If the sum of coefficients of the mask is one, average brightness of the image will be retained
- If the sum of coefficients of the mask is zero, average brightness of the image will be lost and will return a dark image
- If coefficients are alternating positive and negative, the mask is a filter that will sharpen an image
- If coefficients are all positive, it is a filter that will

Smoothing Spatial Filters

- Used for blurring and for noise reduction
- Blurring is used in preprocessing steps, such as
  - Removal of small details from an image prior to large object extraction, and
  - Bridging of small gaps in lines or curves
- Noise reduction can be accomplished by blurring with a linear filter and also by nonlinear filtering
- The output/response of a smoothing, linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask
- Also called „Averaging Filters“ or

Smoothing Linear Filters

- 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 intensity transitions
  - random noise in the image
  - edges of objects in the image
- Thus, smoothing can reduce noises (desirable) and blur edges (undesirable)
- Other uses
  - smoothing of false contours resulting from using an insufficient number of intensity levels
  - reduction of irrelevant detail in an image (here, "irrelevant" means pixel regions that are small wrt the size of filter

General Form : Smoothing Mask

- Filter of size m x n (m and n odd)

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

summation of all coefficient of the mask (computed once)

The complete filtered image is obtained by applying the operation for $x = 0, 1, 2, \ldots, M - 1$ and $y = 0, 1, 2, \ldots, N - 1$

Weighted Average Filter

- 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.

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

Weighted Average Filter

3x3 Smoothing Linear Filters

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

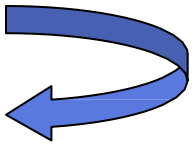
1	1	1
1	1	1
1	1	1

box filter

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

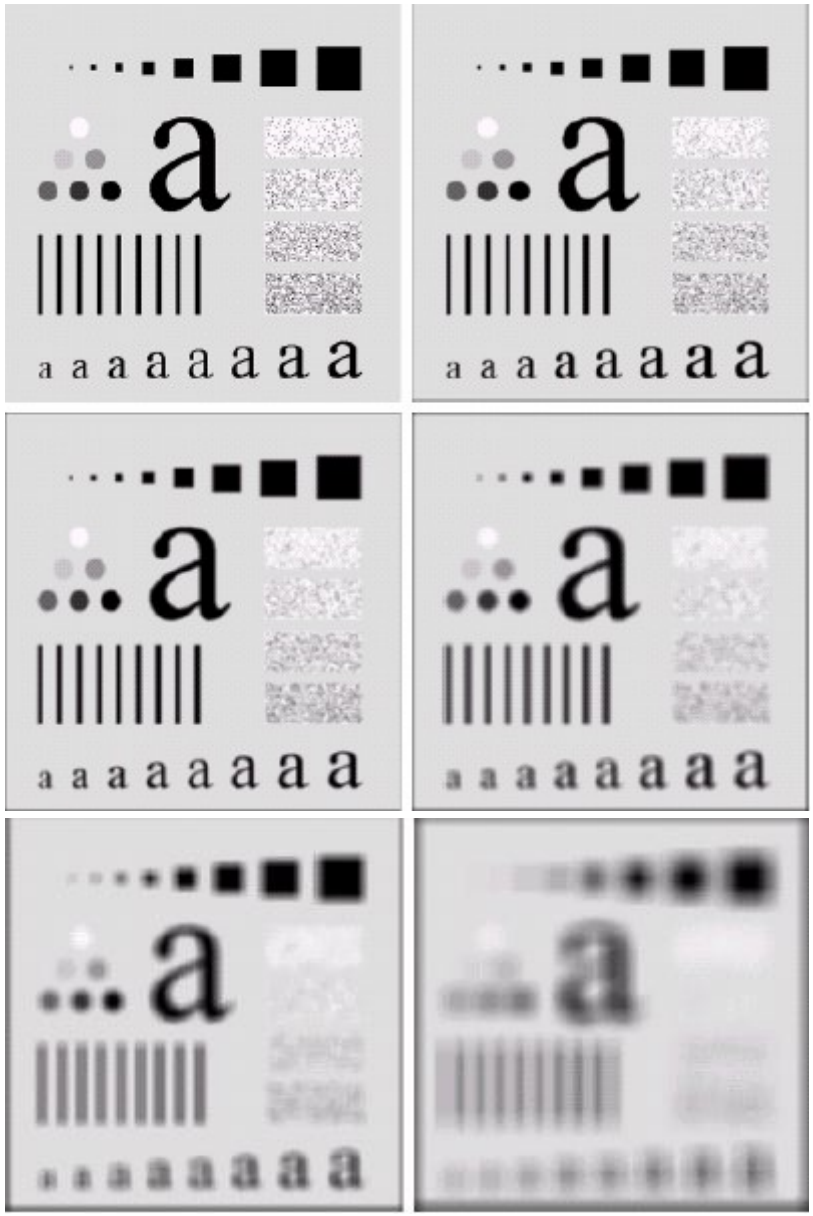
1	2	1
2	4	2
1	2	1

weighted average



the center is the most important and other pixels are inversely weighted as a function of their distance from the center of the mask

Example 1



a	b
c	d
e	f

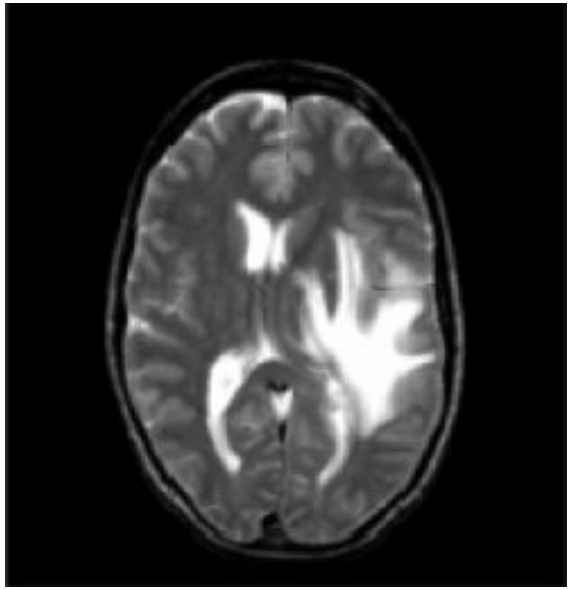
(a) Original image of size 500x500 pixel

(b)-(f) Results of smoothing with square averaging filter masks of sizes *n* = 3, 5, 9, 15, and 35 respectively

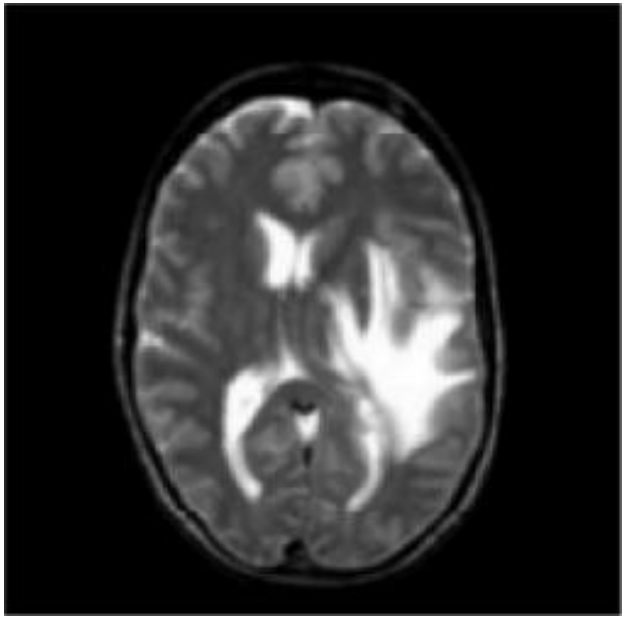
Note:

- details that are of approx. the same size as the filter mask are affected considerably more
- the size of the mask establishes the relative size of the objects that will be blended with the background
- big mask is used to eliminate small objects from an image

Example 2



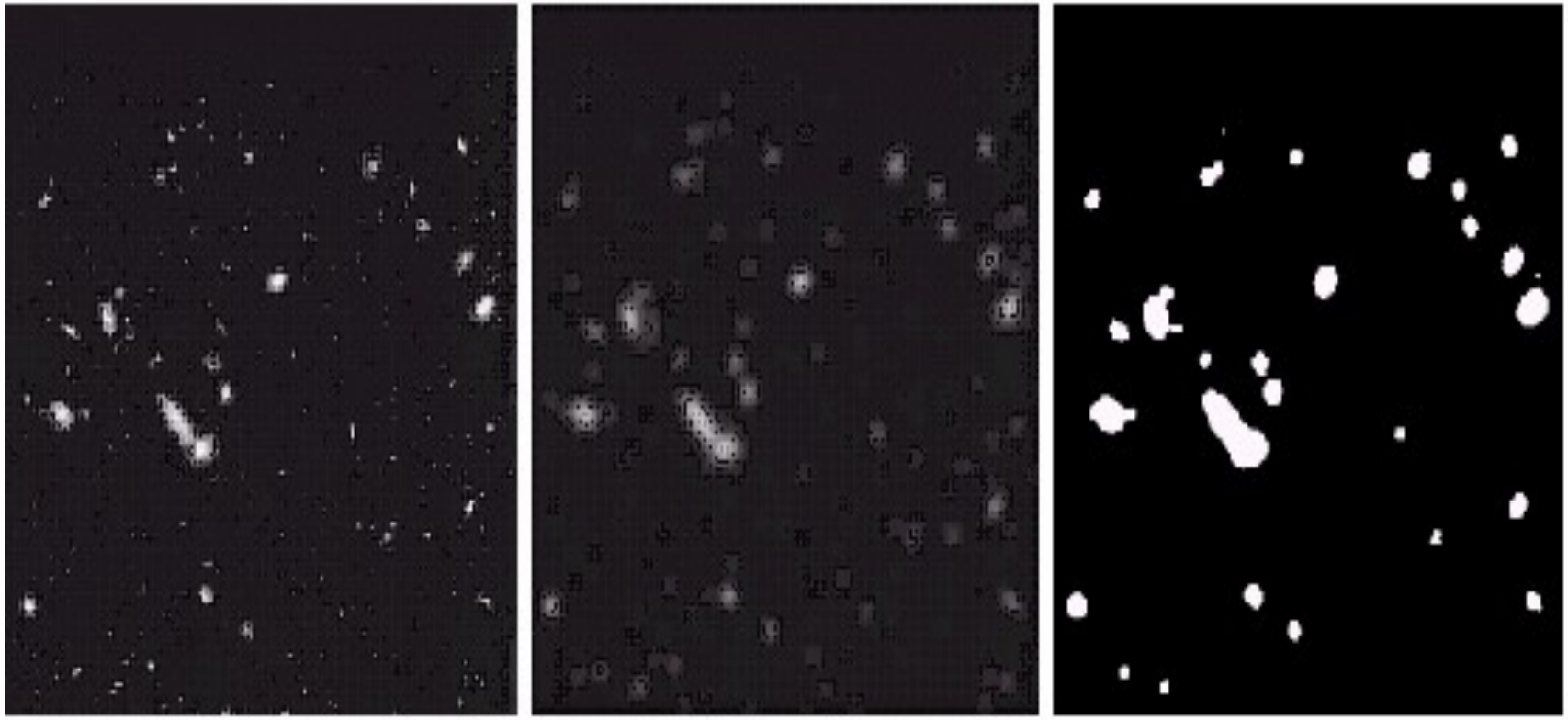
1/16	2/16	1/16
2/16	4/16	2/16
1/16	2/16	1/16



Before smoothing

After smoothing

Example 3



original image

result after smoothing with 15x15 averaging mask

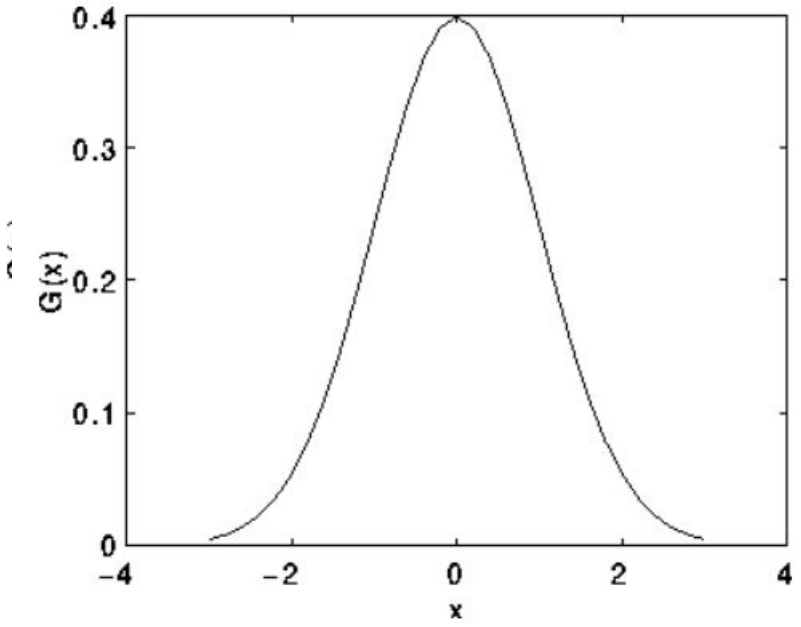
result of thresholding

we can see that the result after smoothing and thresholding, the remains are the largest and brightest objects in the

Gaussian Filters

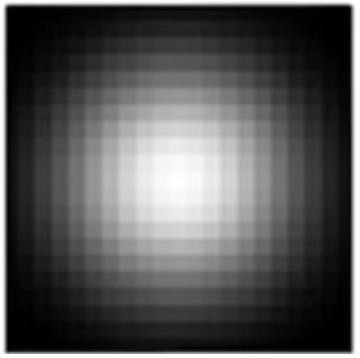
The Gaussian distribution in 1-D has the form:

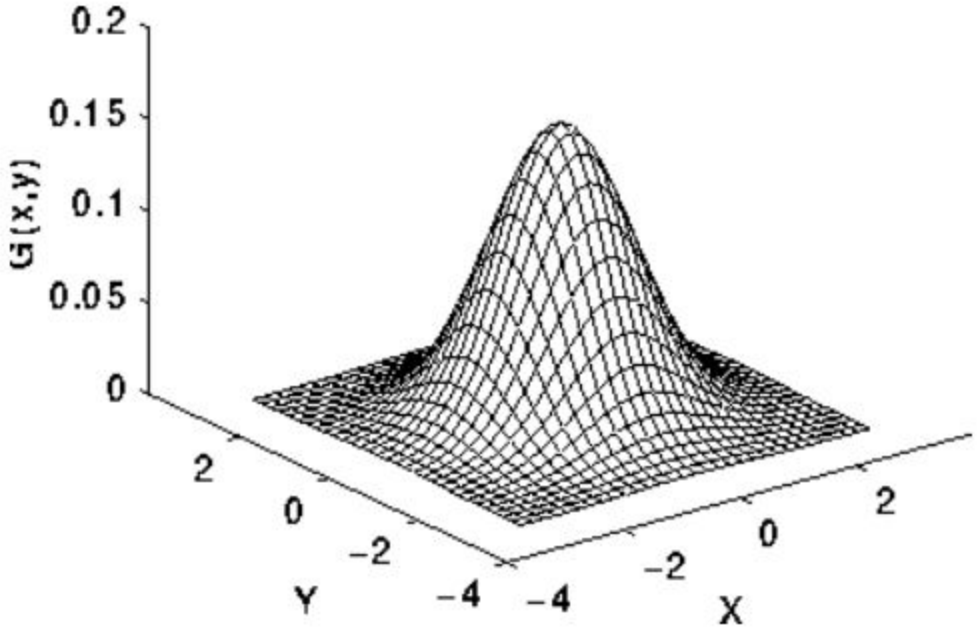
$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$



Gaussian Filters

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



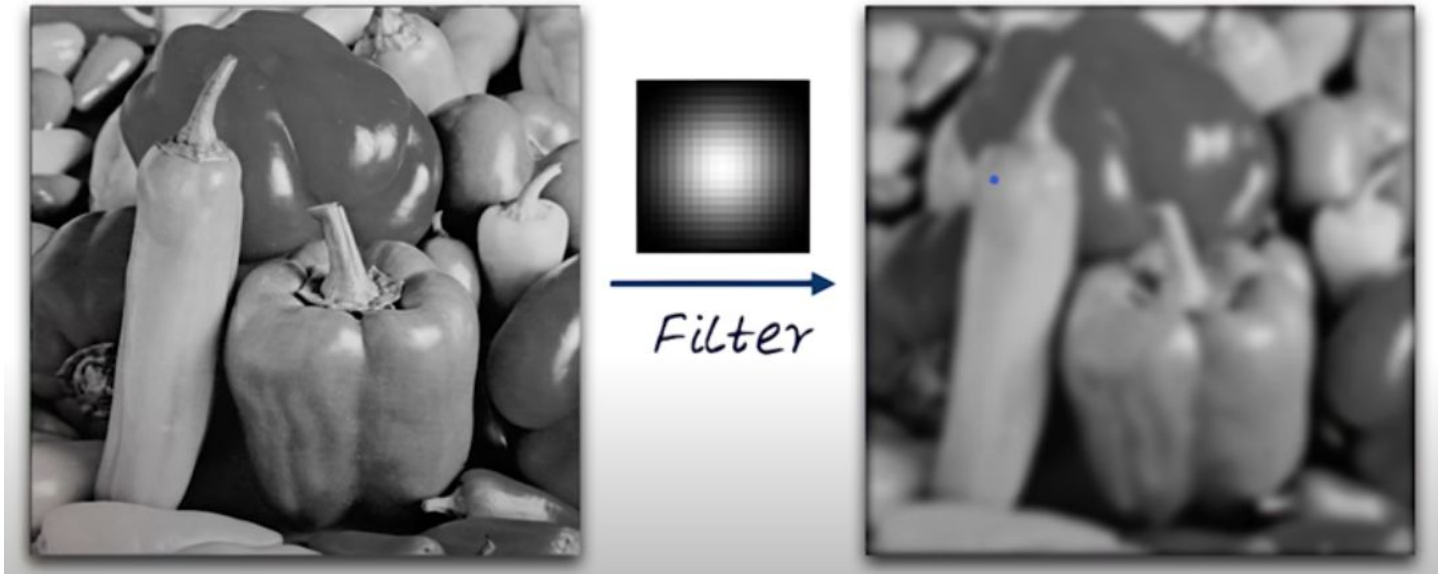


$$\frac{1}{273}$$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Figure 3 Discrete approximation to Gaussian function with σ=1.0

Gaussian Filters



Gaussian Filters

Order-Statistics Filters

- Nonlinear spatial filters
- The response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and

  then replacing the value of the center pixel with the value determined by the ranking results
- Example (n x n is the size of the mask)
  - median filter : $R = \text{median}\{z_k | k = 1, 2, \ldots, n \text{ x } n\}$
  - max filter : $R = \max\{z_k | k = 1, 2, \ldots, n \text{ x } n\}$
  - min filter : $R = \min\{z_k | k = 1, 2, \ldots, n \text{ x } n\}$
- Most popular: median filter

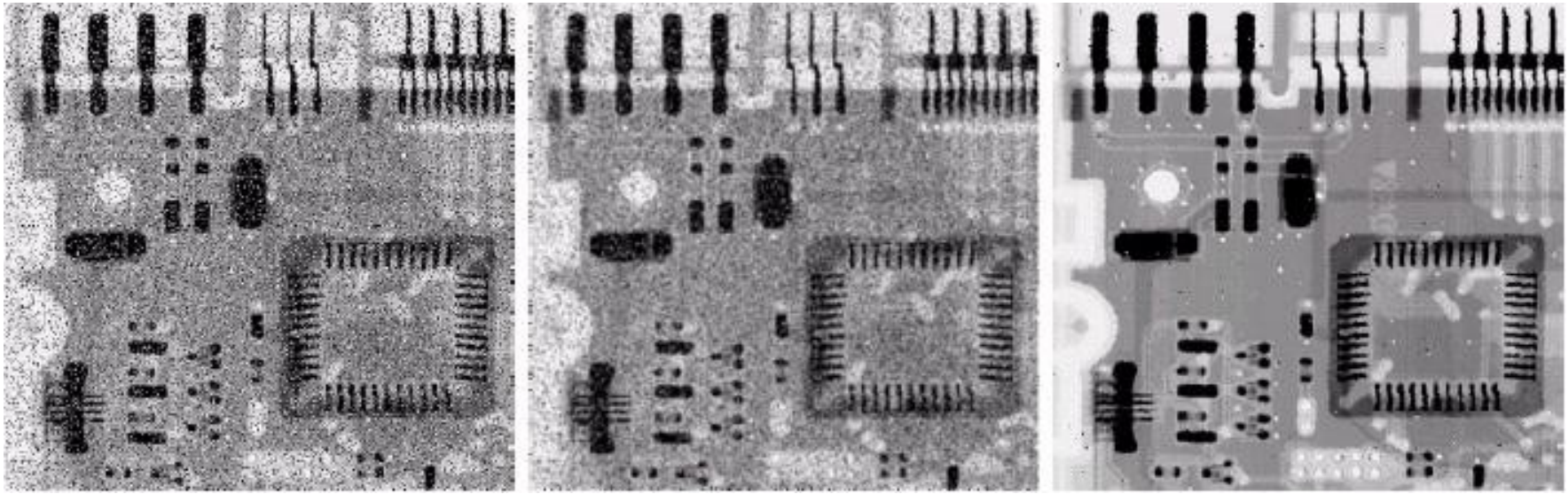
Median Filters

- It replaces the value of the centre pixel by the Median/Min/Max value in the neighborhood pixels
  - Original value of the pixel is included in the computation
  - Median: 50^{th} percentile of a ranked set of nos.
    - For ex: in a 3x3 neighborhood the median is the 5th largest value (5x5 neighborhood -> 13th largest value)
  - 100^{th} percentile: max filter; 0^{th} percentile: min filter
- 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

Median Filters

- 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 Filtering: Example 1



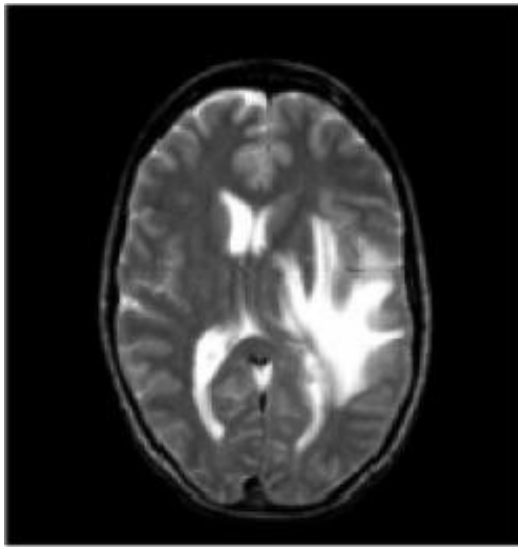
(a) (c) (b)

(a) X-ray image of circuit board corrupted by salt-and-pepper noise

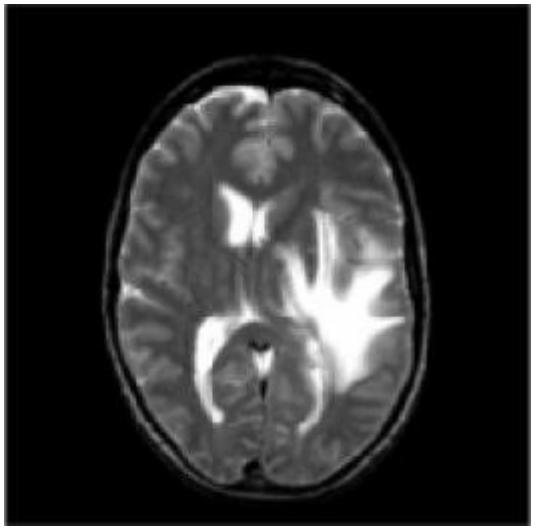
(b) Noise reduction with a 3x3 averaging mask
-less noise reduction but more blurring

(c) Noise reduction with a 3x3 median filter

Median Filtering: Example 2



Before smoothing



After smoothing

The smoothed MR brain image obtained by using median filtering over a fixed neighborhood of 3x3 pixels

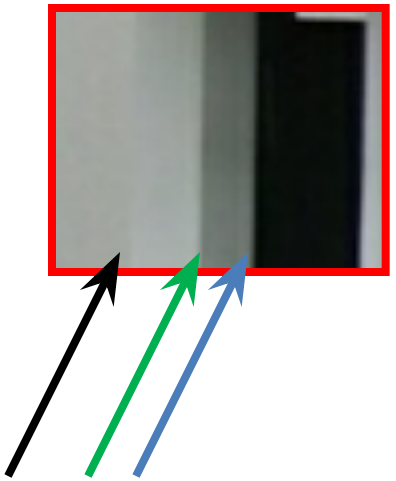
Sharpening Spatial Filters

- To highlight fine detail in an image
- or to enhance detail that has been blurred, either in error or as a natural effect of a particular method of image acquisition
- Blurring vs. Sharpening
  - as we know that blurring can be done in spatial domain by pixel averaging in a neighbors
  - since averaging is analogous to integration
  - thus, we can guess that the sharpening must be accomplished by spatial differentiation

Closeup of edges

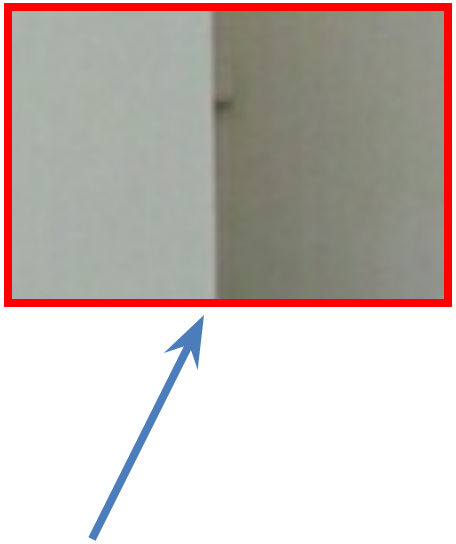
Source: D. Hoiem

Closeup of edges



Source: D. Hoiem

Closeup of edges



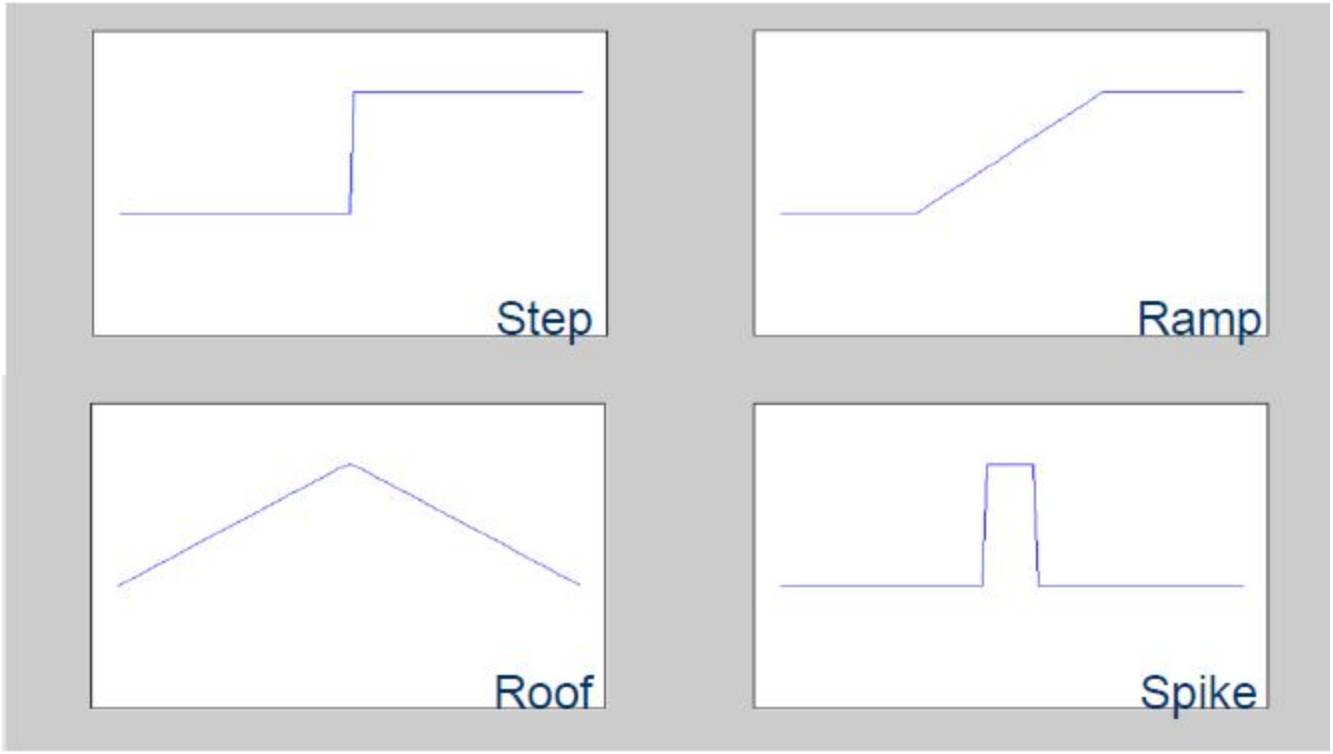
Source: D. Hoiem

Closeup of edges

Source: D. Hoiem

Edge

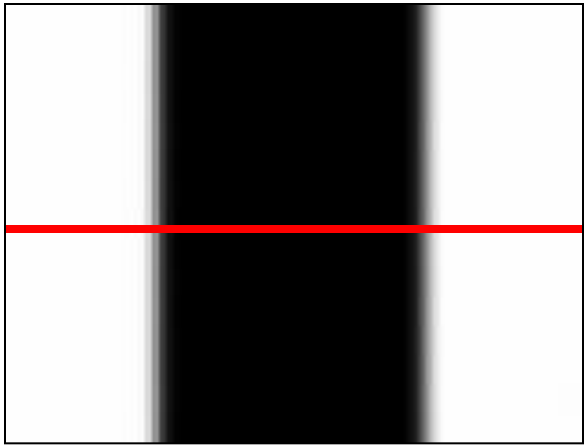
- Discontinuity of intensities in the image



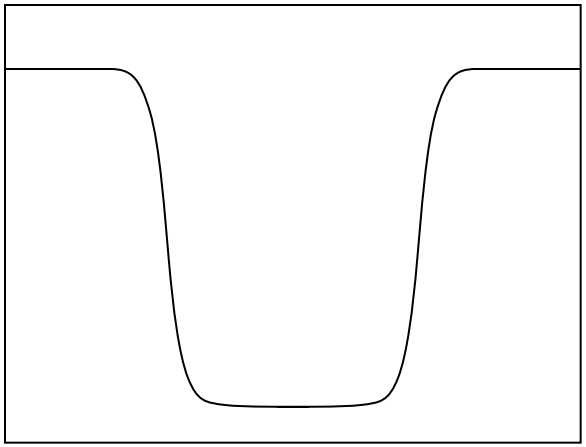
Characterizing edges

- place of rapid change in the image intensity function

image



intensity function
(along horizontal scanline)

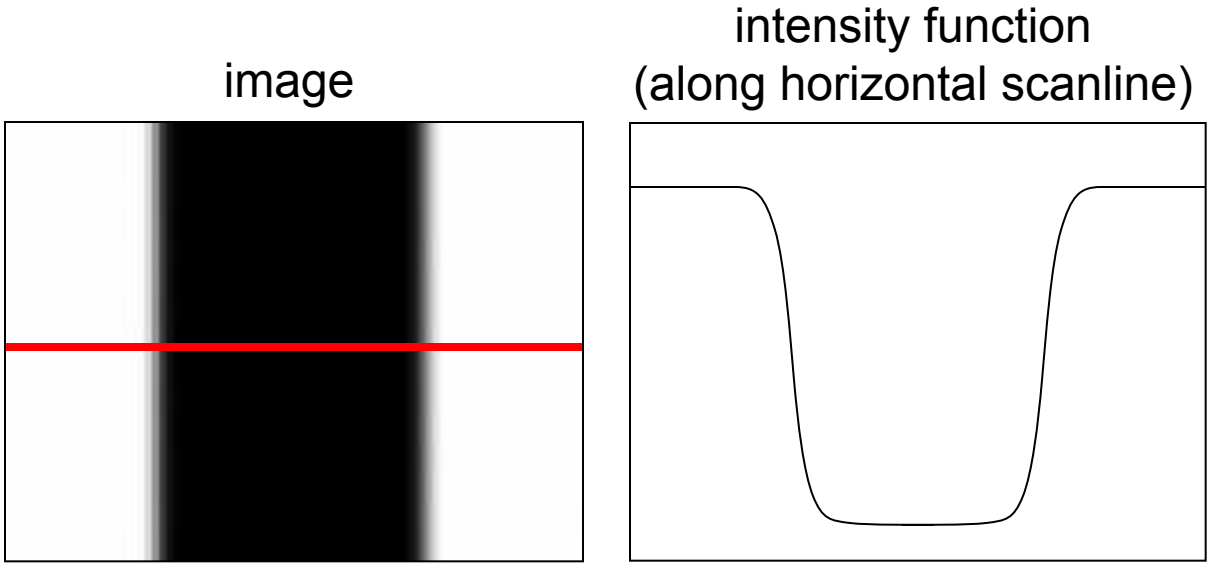


Derivative Operator

- The strength of the response of a derivative operator is proportional to the degree of intensity 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

Characterizing edges

- place of rapid change in the image intensity function



(a)	f(x)	10	10	10	10	10	20	20	20	20	20
(b)	f'(x)	0	0	0	0	0	10	0	0	0	0
(c)	f"(x)	0	0	0	0	0	10	-10	0	0	0

Derivative Operator

- The strength of the response of a derivative operator is proportional to the degree of intensity 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
- First-order derivative
- Second-order derivative

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

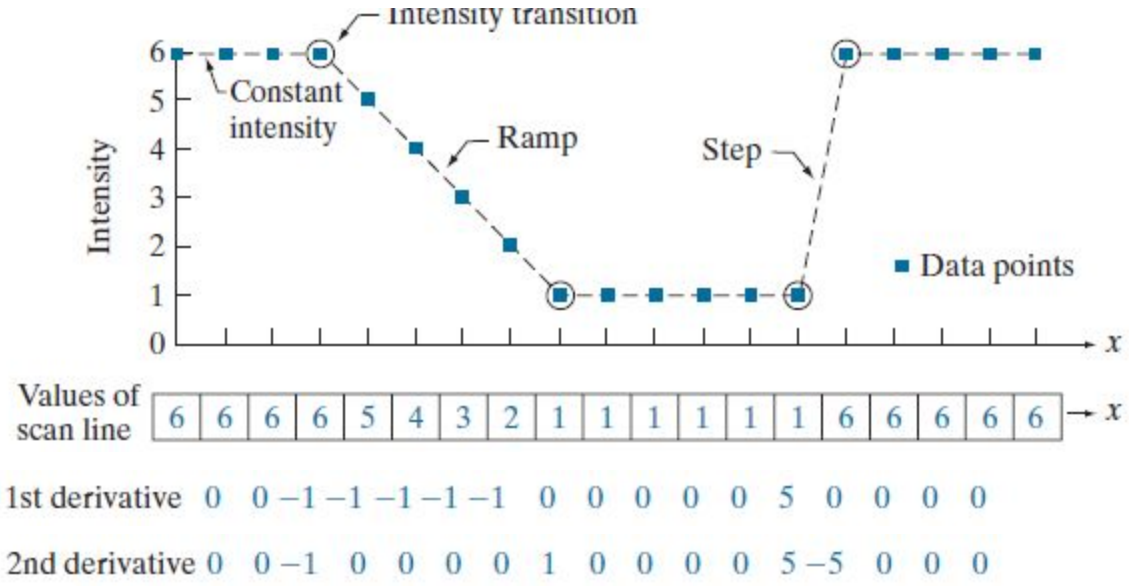
$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

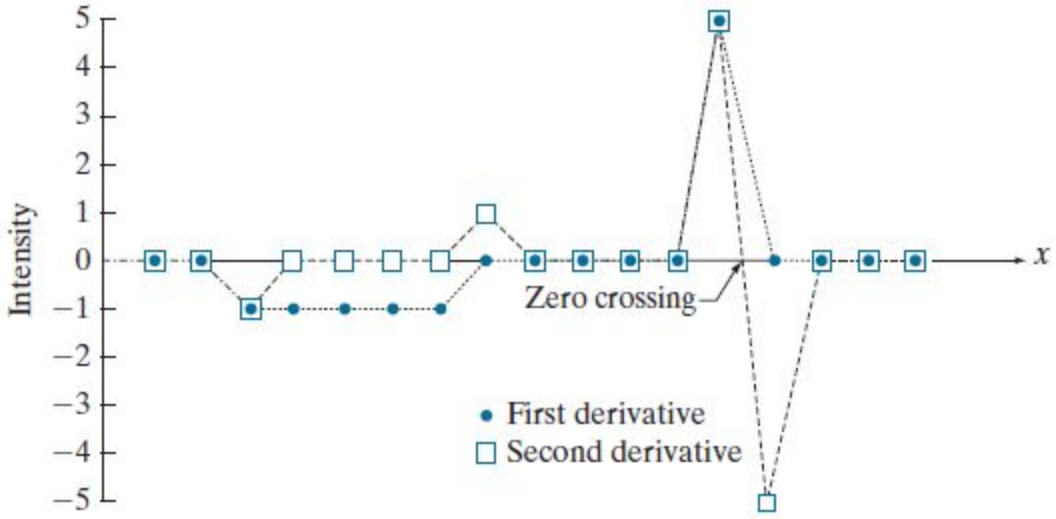
Derivative Operator

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

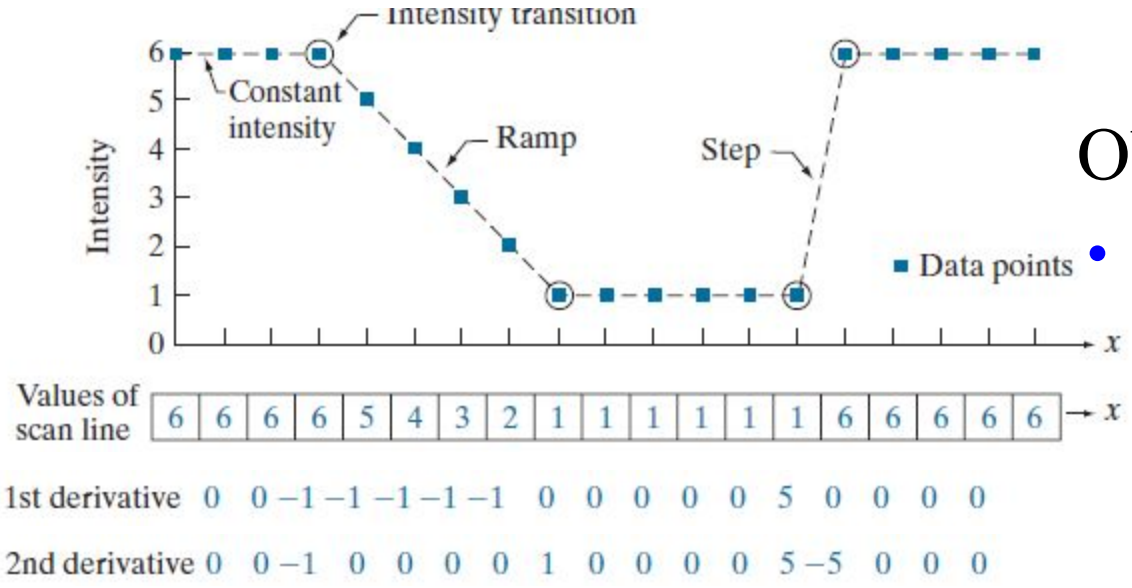
$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

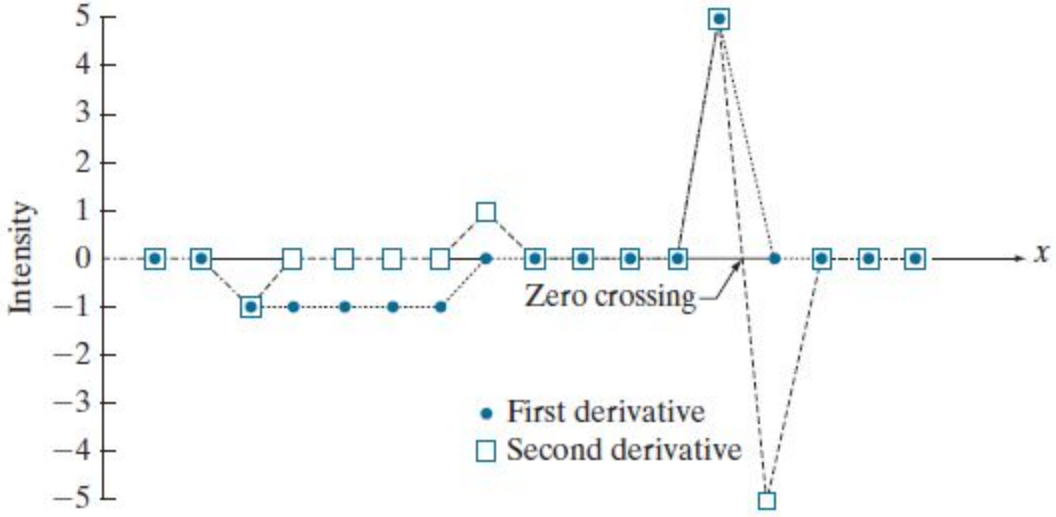
- Derivatives of a digital function are defined in terms of differences
- Various ways to define these differences





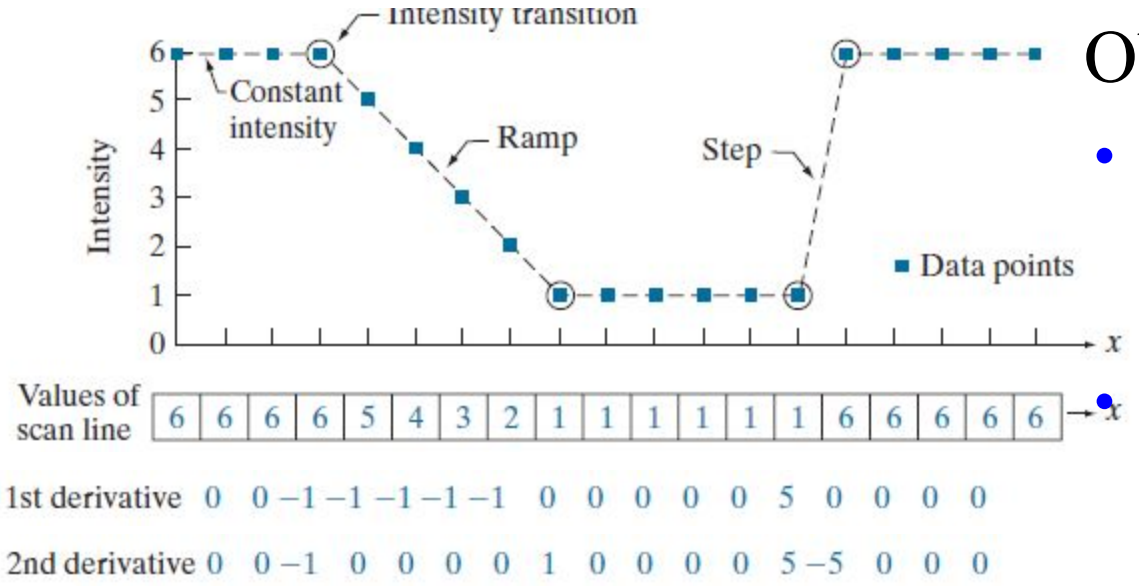
- for a first derivative
  - must be zero in flat segments (areas of constant gray-level values);
  - must be nonzero at the onset of a gray-level step or ramp; and
  - must be nonzero along ramps
- for a second derivative
  - must be zero in flat areas;
  - must be nonzero at the onset & end of a gray-level step or ramp; &
  - must be zero along ramps of constant slope

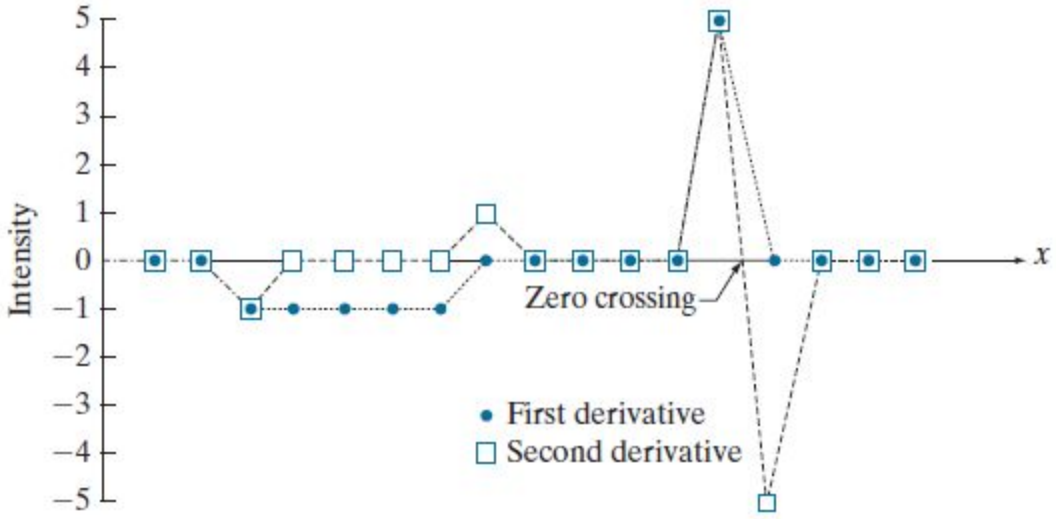




Observations:

- First order derivative is nonzero along the entire ramp, while the second order derivative is nonzero only at the onset and end of the ramp
- Because edges in an image resemble this type of transition, we can say that First order derivatives produce “thick” edges and second order derivatives produce finer edges





Observations:

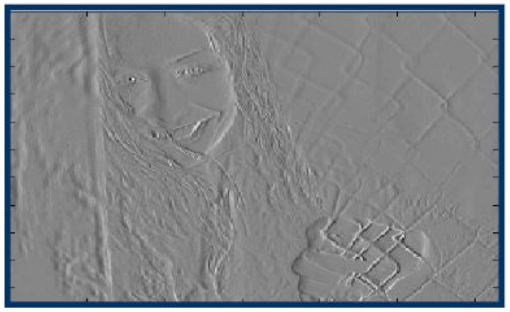
- The response of the two derivatives is the same at the gray-level step
- The sign of the second derivative changes at the onset and end of a step or ramp (zero crossing – useful property for locating edges)
  - The second derivative has a transition from positive back to negative
  - It would produce a double edge one pixel thick, separated by zeros

Edge Detection

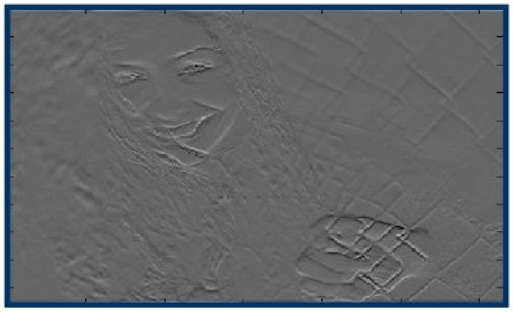
- First Order Derivatives
- Second Order Derivatives
- Smoothing+ derivatives

First Order Image Derivatives

Image I



$$I_x = I * \begin{bmatrix} 1 & -1 \end{bmatrix}$$



$$I_y = I * \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

First Order Image Derivatives

- First derivatives are implemented using the magnitude of the gradient.
- For a function $f(x, y)$, the gradient of f at coordinates (x, y) is defined as a 2-d column vector
- Geometrical property of gradient vector:
  it points in the direction of the greatest rate of change of f at location (x, y)

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

Let $\alpha(x, y)$ represent the direction angle of the vector ∇f at (x, y)

$$\alpha(x, y) = \tan^{-1}\left(\frac{g_y}{g_x}\right)$$

angle is measured wrt the X-axis

The direction of an edge at (x, y) is perpendicular to the direction of the gradient vector at that point

First Order Image Derivatives

- The magnitude of this vector is the value of the rate of change in the direction of the gradient vector at (x, y)

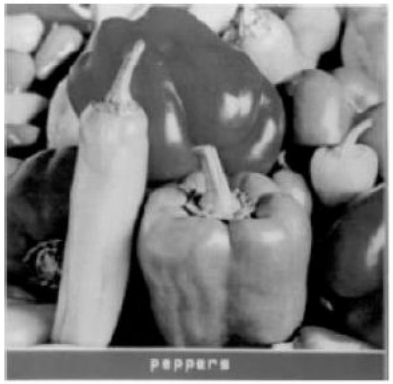
$$M(x,y) = mag(\nabla f) = [g_x^2 + g_y^2]^{1/2} = \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}$$

- $M(x, y)$ is an image of the same size as the original, created when x and y are allowed to vary over all pixel locations in f → "gradient image" or simply "gradient"
- Gradient vector is a linear operator but its magnitude is not
- Computationally it is more suitable to approximate the magnitude by absolute values

$$M(x,y) \approx |g_x| + |g_y|$$

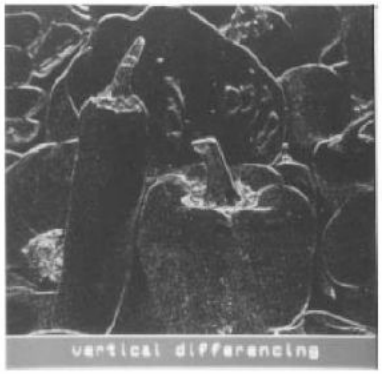
Edge Detection using Gradient

- Motivation: detect changes change in the pixel value ⟶ large gradie

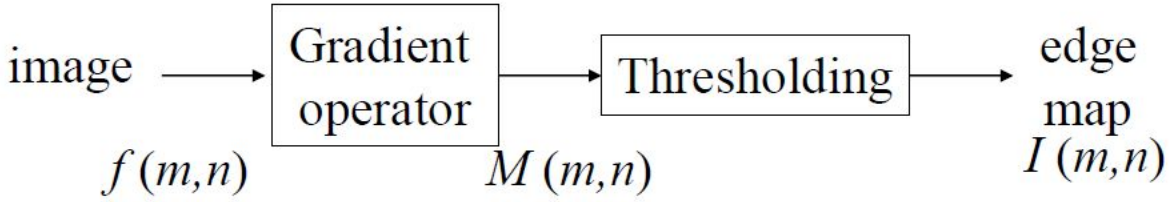


(a) Original

(b) Horizontal magnitude



(c) Vertical magnitude



$$I(m,n) = \begin{cases} 1 & |g(m,n)| > th \\ 0 & otherwise \end{cases}$$

Gradient Masks

- Computation of the gradient of an image is based on obtaining partial derivatives g_x and g_y at every pixel location
- It is always possible to implement the derivatives in digital form in different ways

- Follow the following notation:

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

3x3 area representing the gray levels in a neighborhood of an image

Center point z_5 denotes $f(x, y)$ at an arbitrary location (x, y);
z_1 denotes $(x - 1, y - 1)$; and so on..

Gradient Masks

- **Prewitt Operators:**
- $g_x : (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$
- $g_y : (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

3x3 area representing the gray levels in a neighborhood of an image

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Prewitt operators

Gradient Masks

- **Sobel Operators:**
- $g_x : (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$
- $g_y : (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

3x3 area representing the gray levels in a neighborhood of an image

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel operators

- A weight value of 2 is used to achieve some smoothing by giving more importance to the center point.

Gradient Masks

- **Roberts Cross-Gradient Operators:** (Sum of the magnitude of the differences of the diagonal neighbors)
- $g_x : (z_9 - z_5)$, $g_y : (z_8 - z_6)$
- Gradient image can be computed as:

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$$M(x,y) = [(z_9 - z_5)^2 + (z_8 - z_6)^2]^{1/2}$$
$$M(x,y) \approx |z_9 - z_5| + |z_8 - z_6|$$

- These derivatives can be implemented for an entire image by using the mask

−1	0
0	1

0	−1
1	0

Roberts operators

Gradient Masks

- **Roberts Cross-Gradient Operators:**

−1	0
0	1

0	−1
1	0

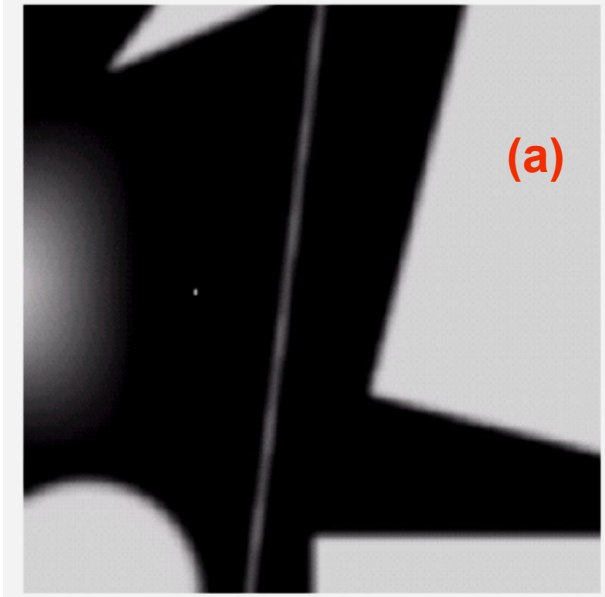
Roberts operators

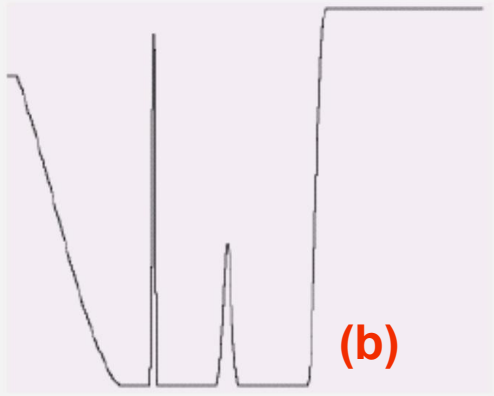
- Masks of size 2 x 2 are awkward to implement because they do not have a clear center.
- It makes the edge point only, NOT the information about the edge orientation.

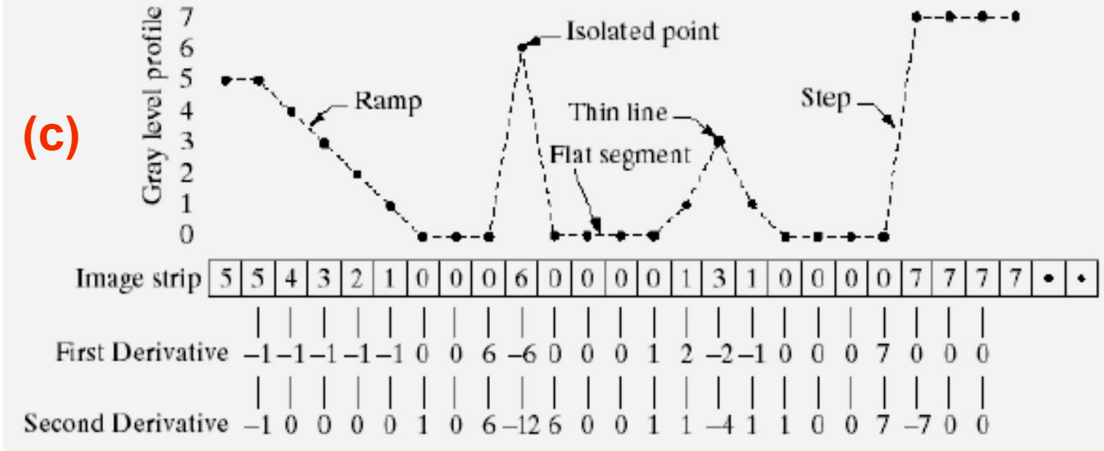
Gradient Masks

- Prewitt and Sobel operators: mostly used in practice for computing digital gradients.
- Prewitt masks: simpler to implement.
- Sobel masks have slightly superior noise-suppression characteristics
  - an important issue when dealing with derivatives
  - Reason: a weight value of 2 is used to achieve some smoothing by giving more importance to the center point
- The sum of coefficients in all gradient masks is 0:
  - They give a response of 0 in areas of constant gray level (as expected from a derivative operator)

First- and Second- Order Derivatives in the context of Image Processing





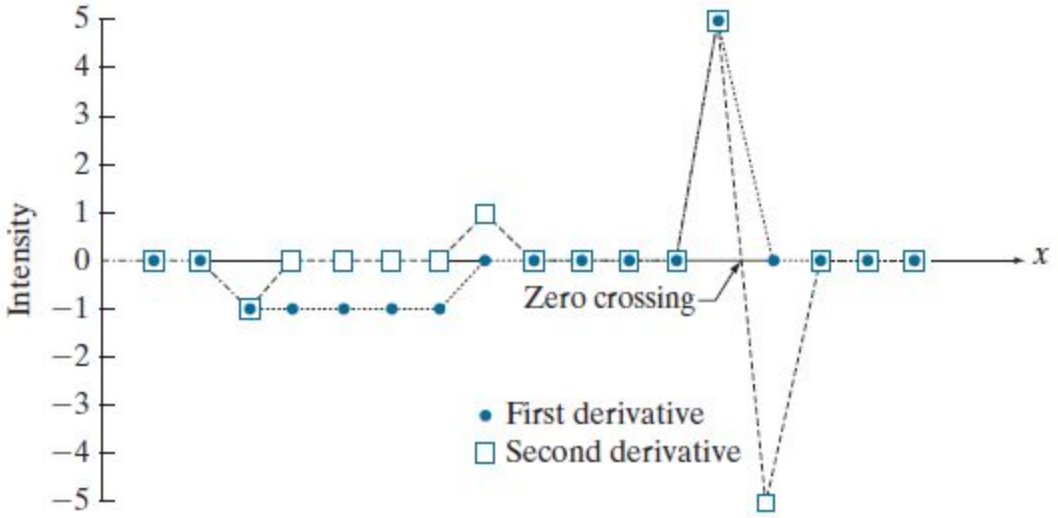


(a) A simple image

(b) 1-D horizontal gray level profile along the center of the image and including the isolated noise point

(c) Simplified profile (the points are joined by dashed lines to simplify interpretation)

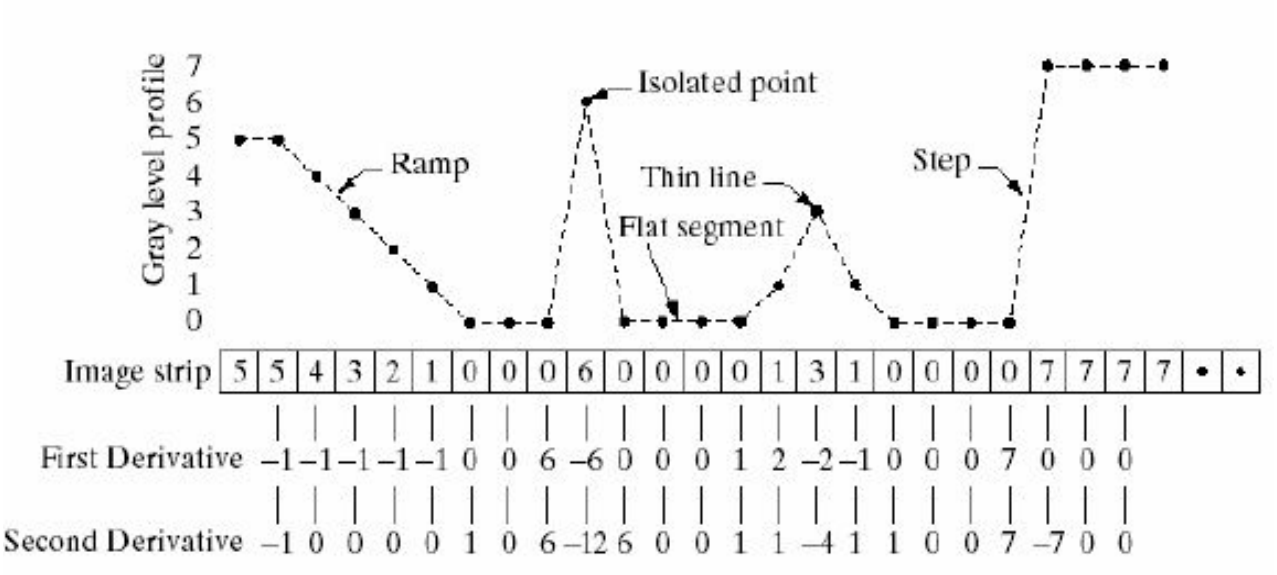
First- and Second- Order Derivatives in context of Image Processing



Observations:

- First order derivative is nonzero along the entire ramp, while the second order derivative is nonzero only at the onset and end of the ramp
- Because edges in an image resemble this type of transition, we can say that First order derivatives produce “thick” edges and second order derivatives produce finer edges

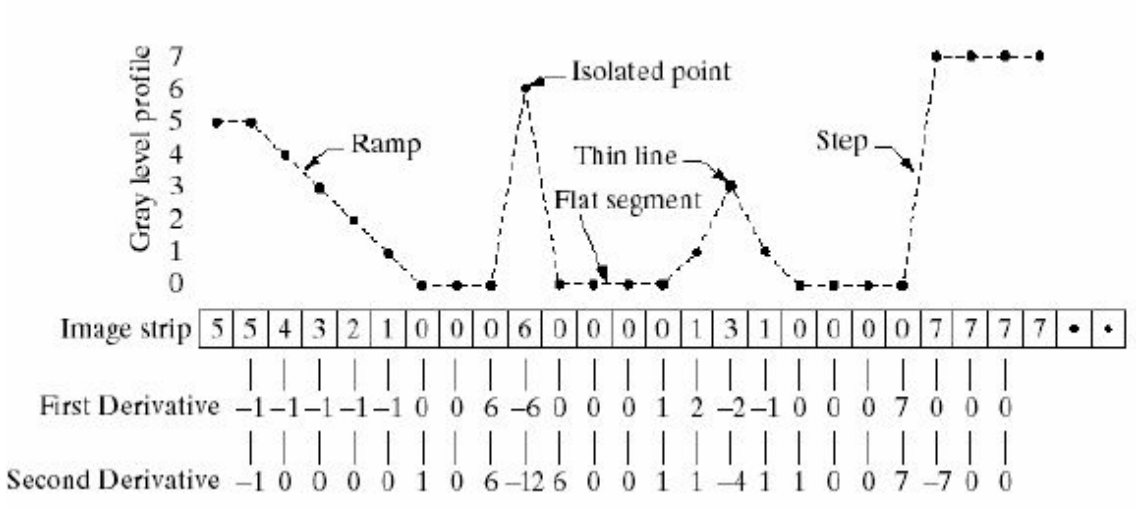
First- and Second- Order Derivatives in context of Image Processing



Observations:

- Isolated noise point
  - The response at and around the point is much stronger for the second order derivative than for the First order derivative
  - So second order derivative enhance fine detail much more than a first order derivative

First- and Second- Order Derivatives in context of Image Processing



Observations:

- The response of the two derivatives is the same at the gray-level step
- In most cases when the transition into a step is not from zero, the second derivative will be weaker
- The sign of the second derivative changes at the onset and end of a step or ramp (zero crossing – useful property for locating edges)
  - The second derivative has a transition from positive back to negative
  - It would produce a double edge one pixel thick, separated by zeros