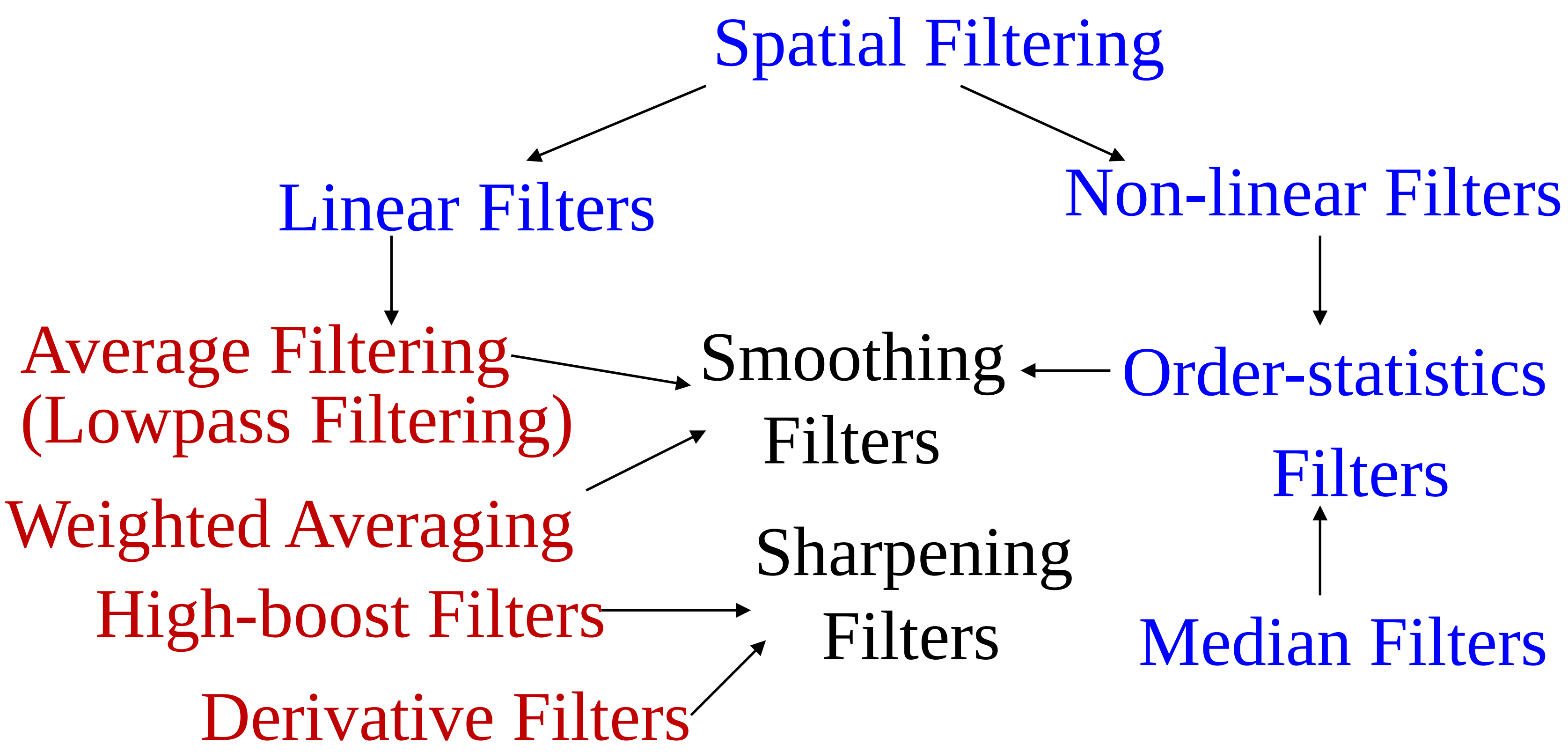
Image Manipulation
in
Spatial Domain

Image Enhancement II
(Spatial Filtering)

Filtering

- It is applied for noise removal or to perform some type of image enhancement
- Filtering can be done in the spatial domain as well as frequency domain
- The operators we will discuss here are called spatial filters since they operate on the raw image data in the spatial domain
- They operate on the image data by considering small neighborhoods in an image and returning the result based on a linear or non-linear operations; moving sequentially across and down the entire image

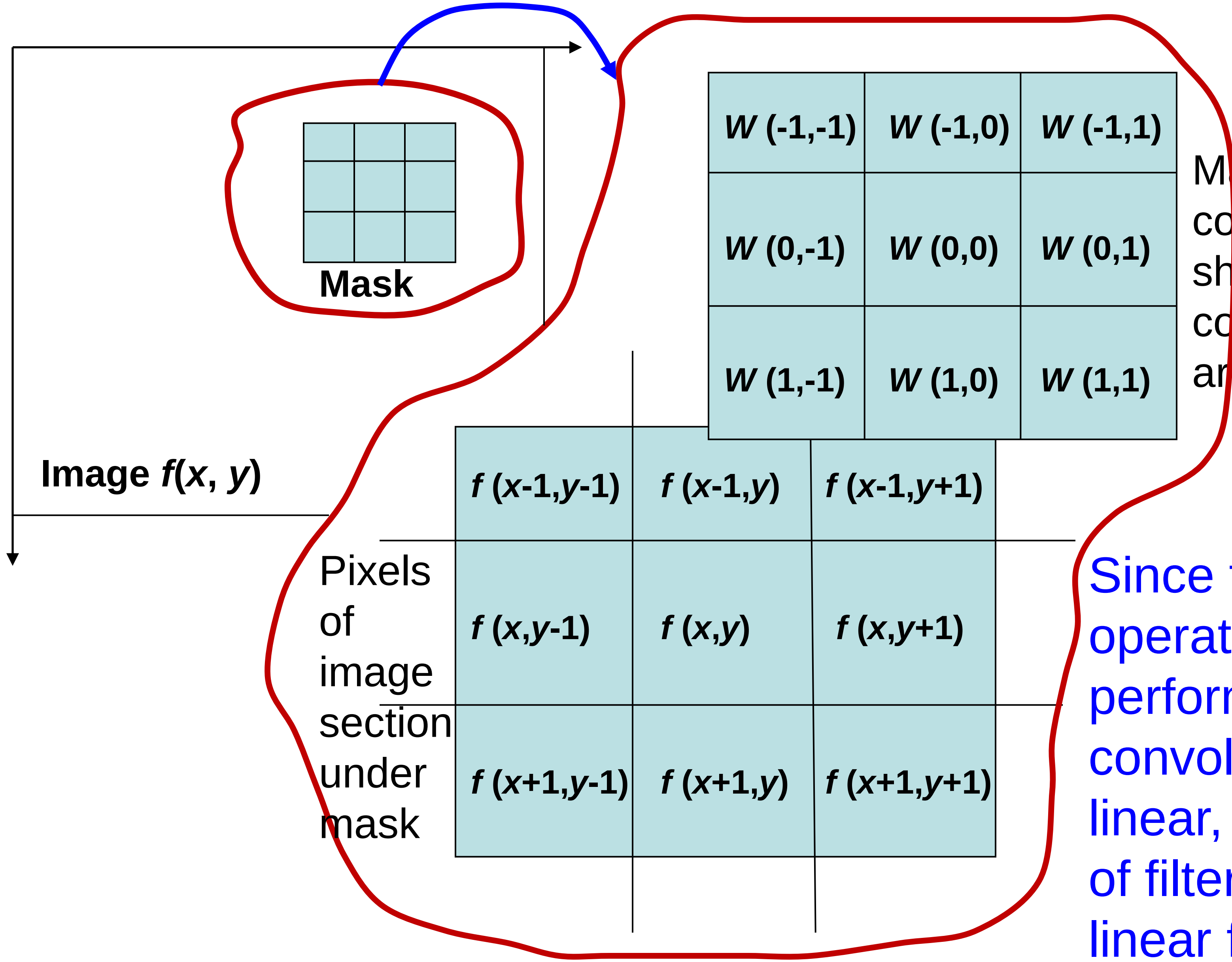
Spatial Filtering



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



Since the operations performed in the convolution are linear, these types of filters are called linear filters

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

Linear Spatial Filtering

- In general, Linear filtering of an image *f* of size *M*×*N* with a filter mask of size m×n 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 manner.

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

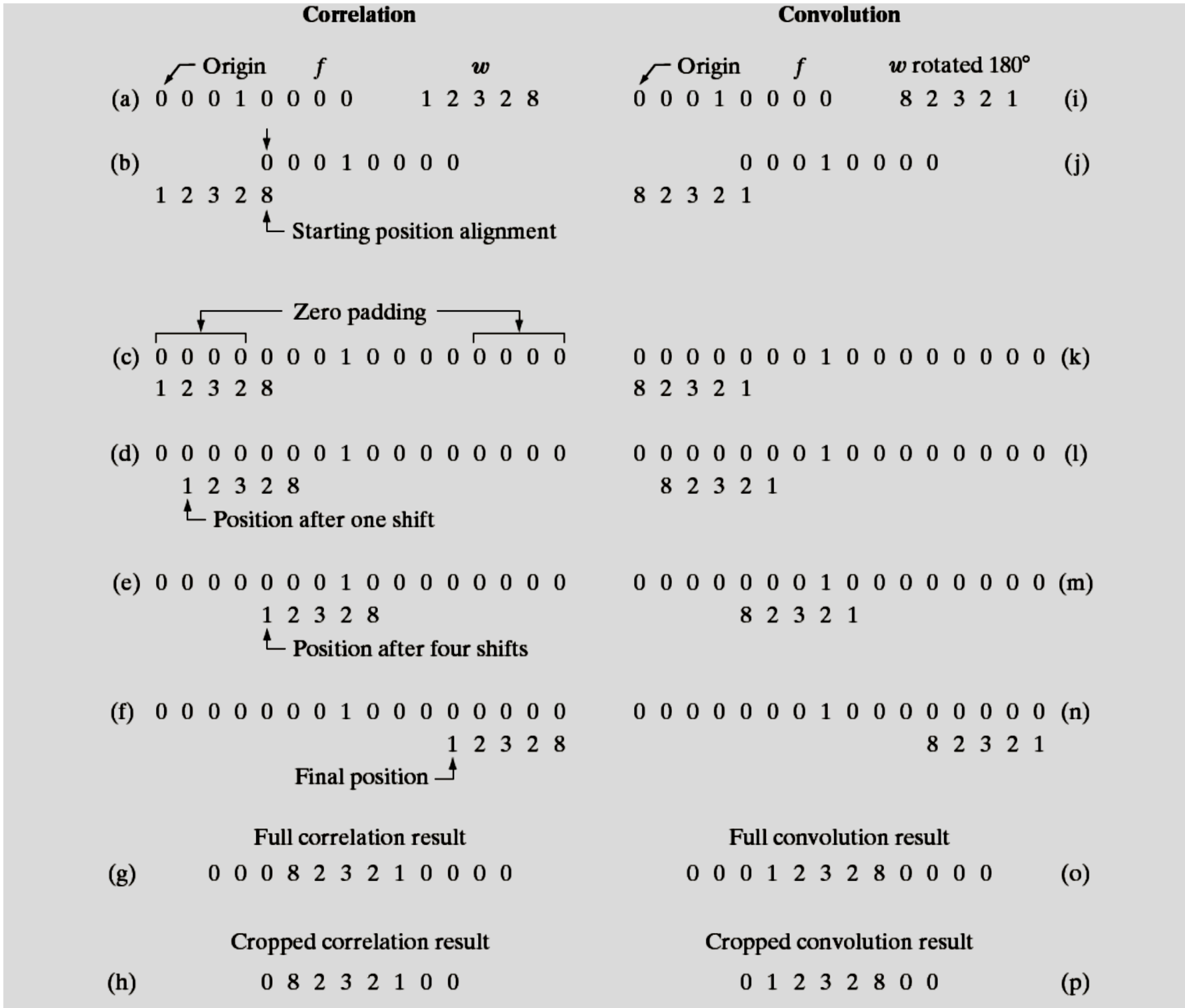


FIGURE 3.29 Illustration of 1-D correlation and convolution of a filter with a discrete unit impulse. Note that correlation and convolution are functions of *displacement*.

Spatial Correlation and Convolution

Origin $f(x, y)$

$$\begin{matrix}0&0&0&0&0\\0&0&0&0&0\\0&0&1&0&0\\0&0&0&0&0\\0&0&0&0&0\end{matrix}$$

$w(x, y)$

$$\begin{matrix}1&2&3\\4&5&6\\7&8&9\end{matrix}$$

(a)

Padded f

$$\begin{matrix}0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&1&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\end{matrix}$$

(b)

Initial position for w

$$\begin{matrix}1&2&3&0&0&0&0&0&0\\4&5&6&0&0&0&0&0&0\\7&8&9&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&1&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\end{matrix}$$

(c)

Full correlation result

$$\begin{matrix}0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&9&8&7&0&0&0\\0&0&0&6&5&4&0&0&0\\0&0&0&3&2&1&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\end{matrix}$$

(d)

Cropped correlation result

$$\begin{matrix}0&0&0&0&0\\0&9&8&7&0\\0&6&5&4&0\\0&3&2&1&0\\0&0&0&0&0\end{matrix}$$

(e)

Rotated w

$$\begin{matrix}9&8&7&0&0&0&0&0&0\\6&5&4&0&0&0&0&0&0\\3&2&1&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&1&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\end{matrix}$$

(f)

Full convolution result

$$\begin{matrix}0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&1&2&3&0&0&0\\0&0&0&4&5&6&0&0&0\\0&0&0&7&8&9&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\end{matrix}$$

(g)

Cropped convolution result

$$\begin{matrix}0&0&0&0&0\\0&1&2&3&0\\0&4&5&6&0\\0&7&8&9&0\\0&0&0&0&0\end{matrix}$$

(h)

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

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

- 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 correlation
- 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

Linear Filtering: Vector Representation

- It is convenient to write the response R of an m x n mask at any point (x, y) is:

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_{mn} z_{mn} = \sum_{i=1}^{mn} w_i z_i$$

w_s are the coefficients of an m x n filter and the z_s are the corresponding image intensities encompassed by the filter

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Generating Spatial Filter Mask

- Mask Coefficient
  - Proper selection of the coefficients and application of the mask at each pixel position in an image makes possible a variety of useful image operations
    - Noise reduction
    - Region thinning
    - Edge detection
  - Coefficients are selected based on what the filter is supposed to do, keeping in mind that all we can do with linear filtering is to implement a sum of products
- Applying a mask at each pixel location in an image is a computationally extensive task

Generating Spatial Filter Mask

- Example:

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

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 blur the image

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 'Lowpass Filters'

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 mask)

General Form : Smoothing Mask

- Filter of size $m \times n$ (m=2a+1 and n=2b+1)

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

summation of all coefficient of the mask (computed once)

The complete filtered image is obtained by applying the operation for x = 0, 1, 2, …., M – 1 and y = 0, 1, 2, …., N - 1

3x3 Smoothing Linear Filters

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

box filter

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

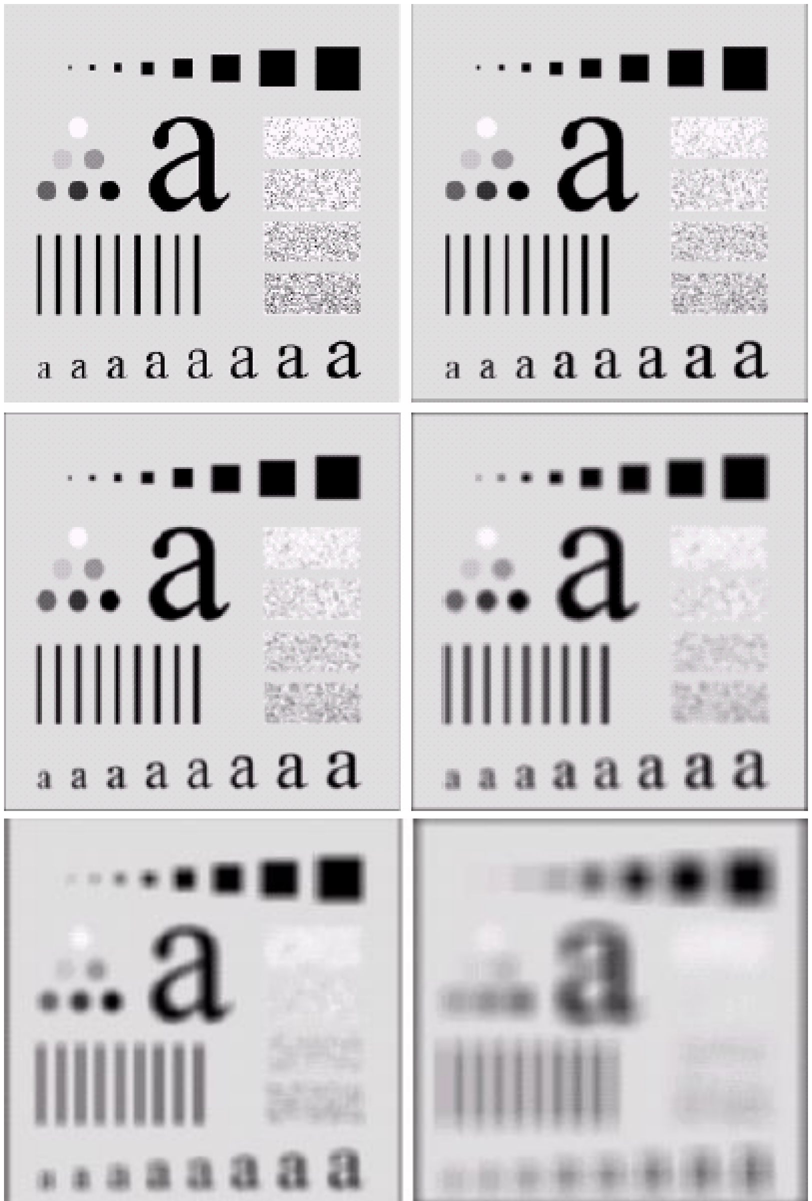
Weighted Average Filter

- 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

Example 1



a	b
c	d
e	f

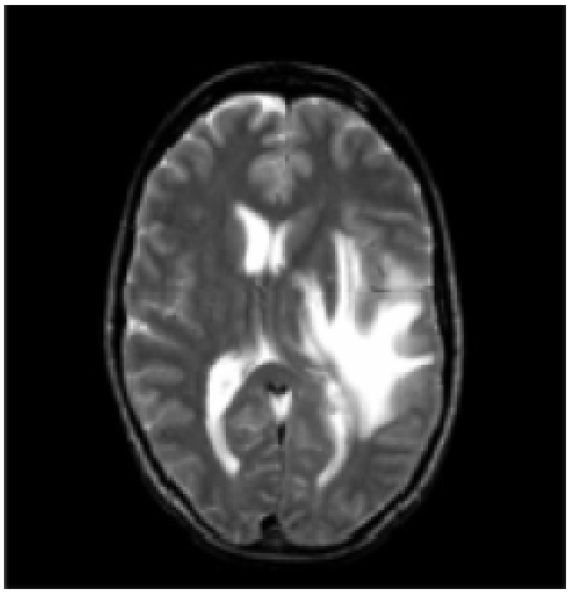
(a) Original image of size 500×500 pixel

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

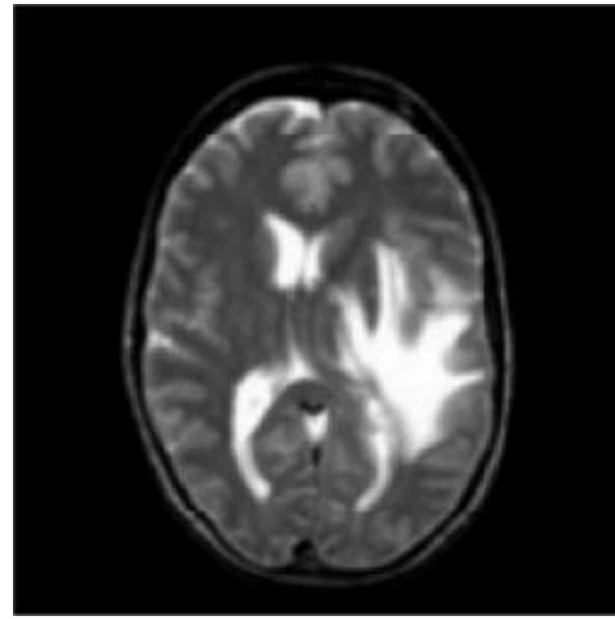
Note:

- 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



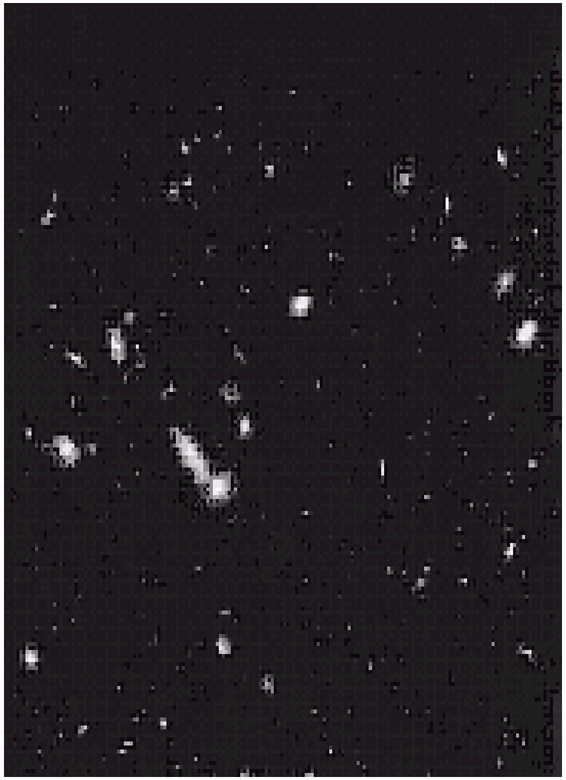
1/16	2/16	1/16
2/16	4/16	2/16
1/16	2/16	1/16

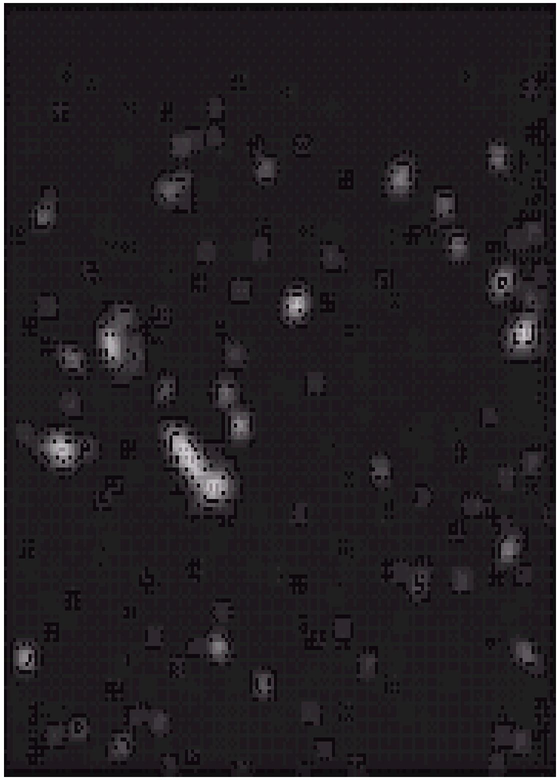


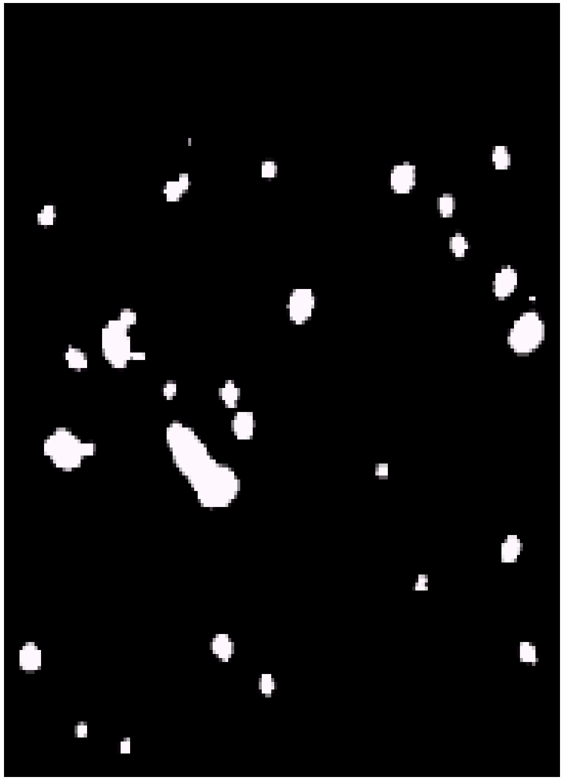
Before smoothing

After smoothing

Example 3







original image smoothing with 15x15 averaging mask thresholding

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

Order-Statistics Filters

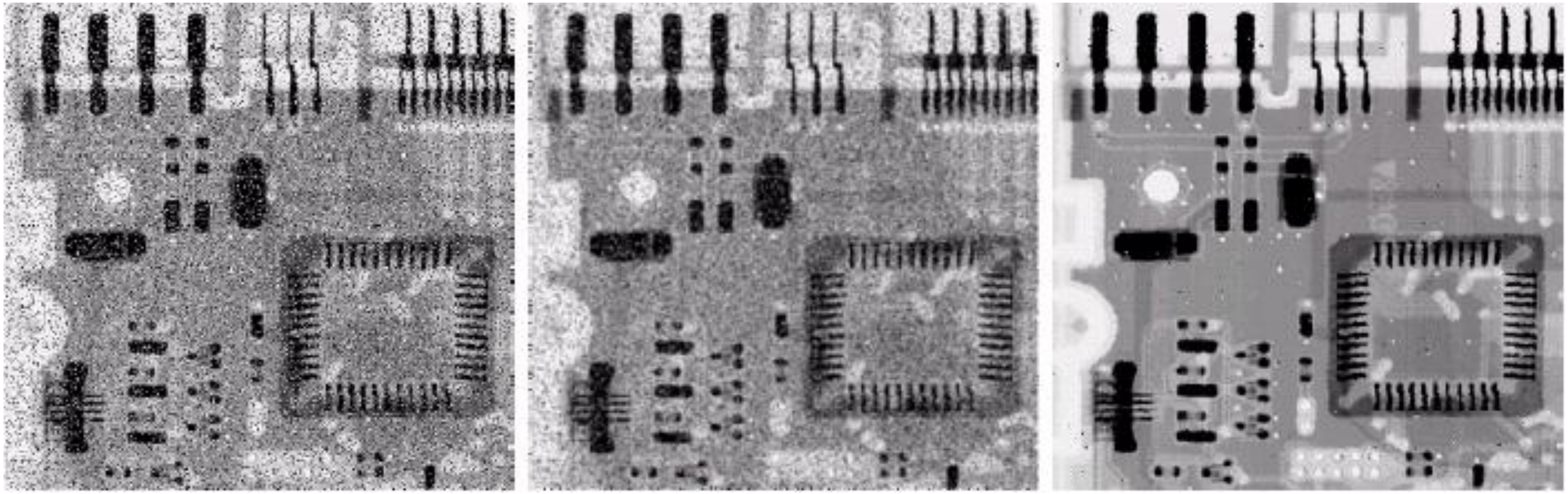
- Nonlinear spatial filters
- Do not have equivalent formulation in frequency space
- The response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and

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

Order-Statistics Filters

- It replaces the value of the center pixel by the Median/Min/Max value in the neighborhood pixels
  - Original value of the pixel is included in the computation
  - Median: 50th percentile of a ranked set of nos.
    - For ex: in a 3×3 neighborhood the median is the 5th largest value (5×5 neighborhood -> 13th largest value)
  - 100th percentile: max filter; 0th 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 Filtering: Example 1



(a) (b) (c)

X-ray image of circuit board

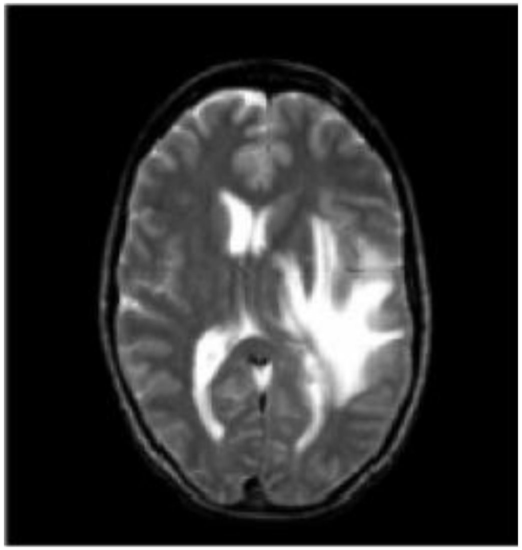
(a) Corrupted by salt-and-pepper noise

(b) Noise reduction with a 3×3 averaging mask

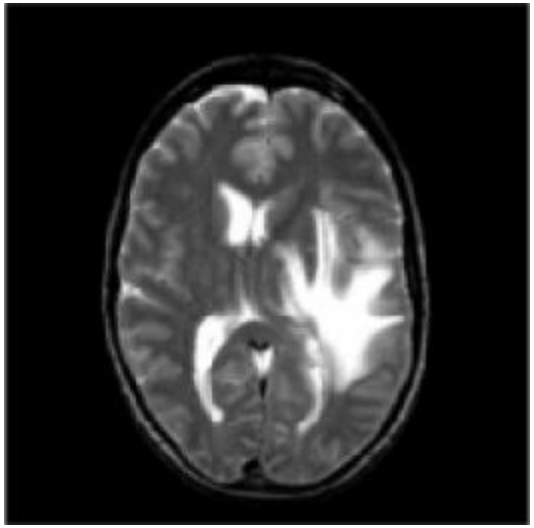
- less noise reduction but more blurring

(c) Noise reduction with a 3×3 median filter

Median Filtering: Example 2



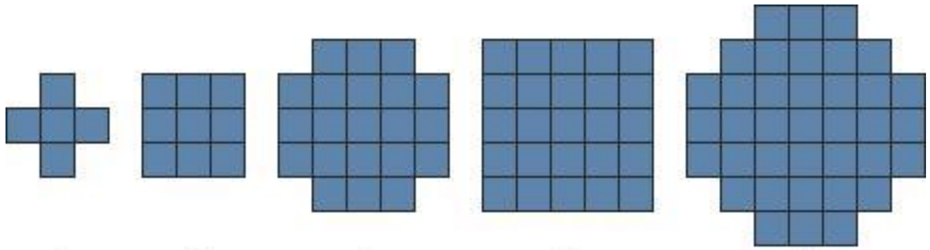
Before smoothing



After smoothing

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

Median Filters

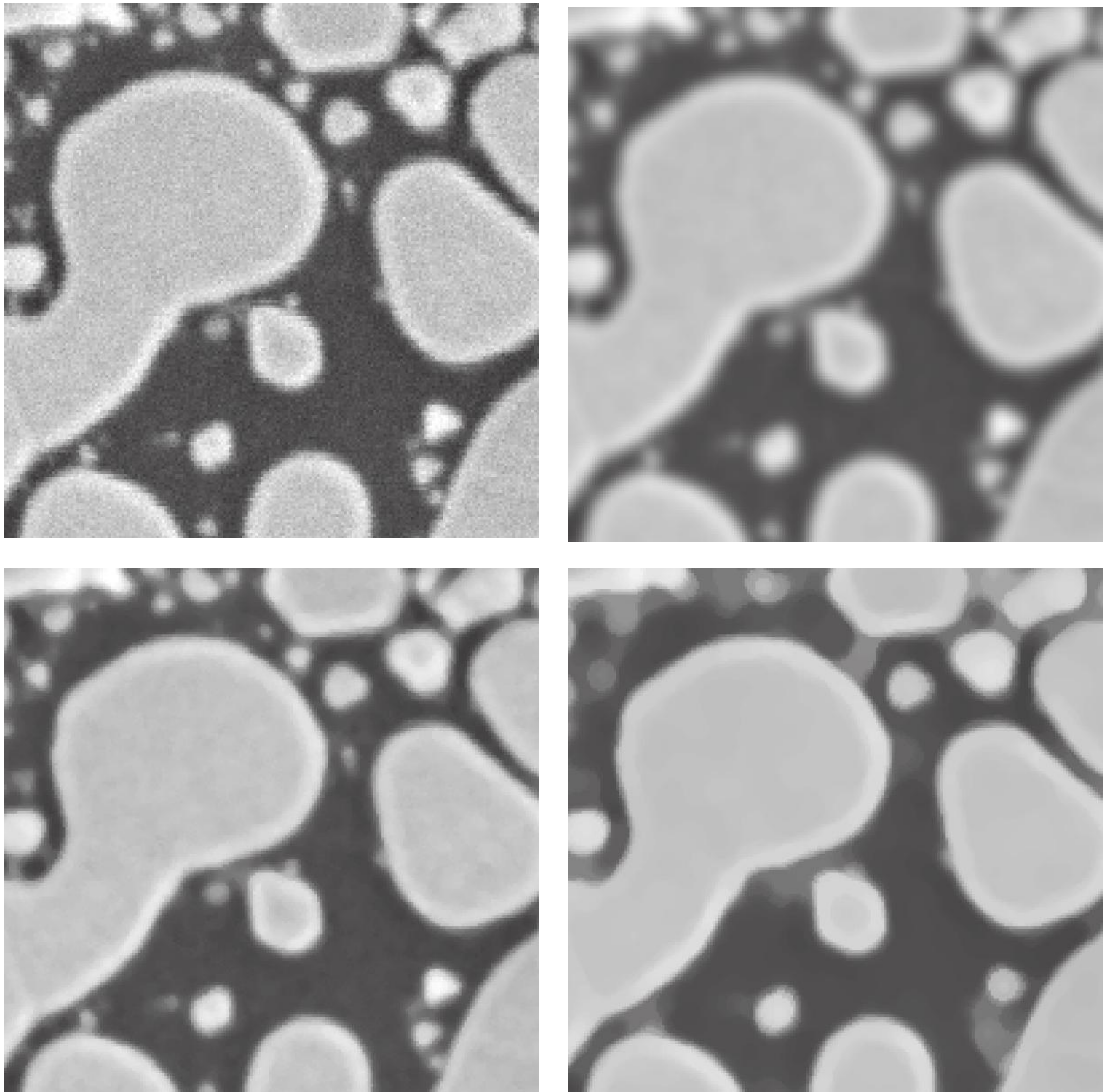


Neighborhood patterns used for median filtering

Advantages as compared to multiplication by weights

1. It does not reduce or blur the brightness difference across steps, because the values available are only those present in the neighborhood region, not an average between those values
2. It does not shift boundaries as averaging may, depending on the relative magnitude of values present in the neighborhood and the local curvature of the boundary

Median Filters



a	b
c	d

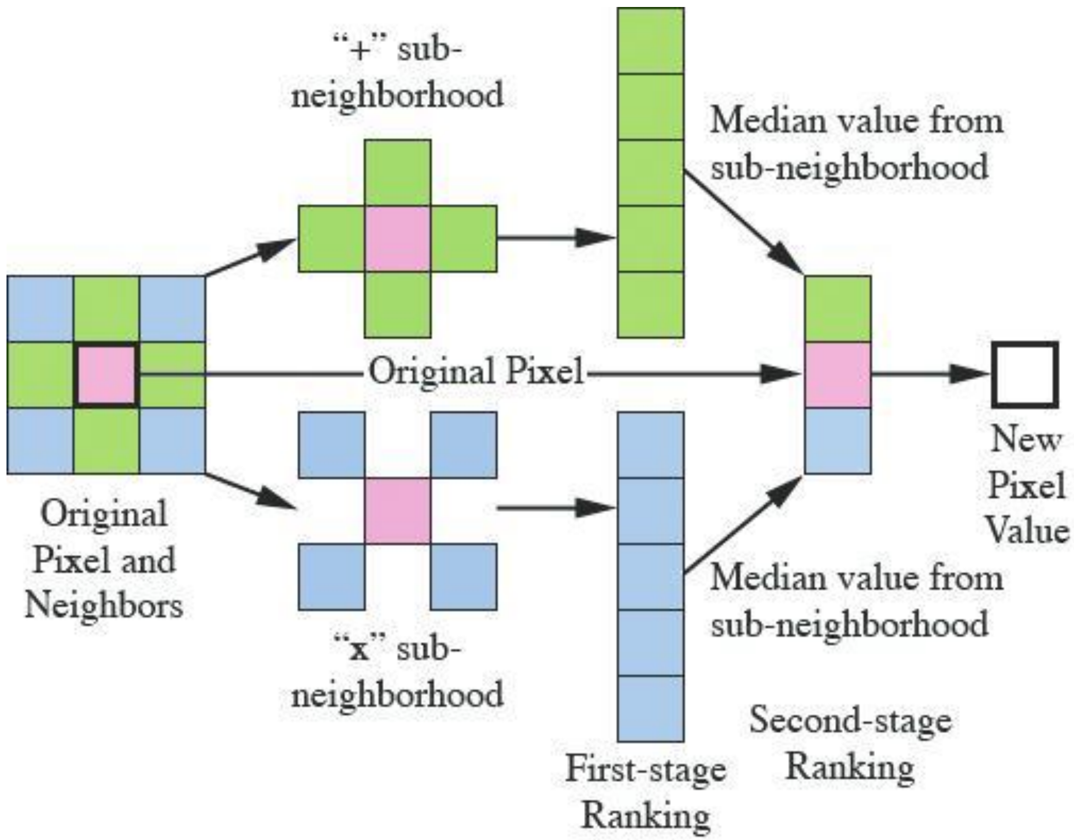
Comparison of noise reduction techniques (SEM image enlarged to show pixel detail):
(a) original;
(b) Gaussian filter, standard deviation = 2 pixels;
(c) median filter, radius = 2 pixels (5 × 5 Octagonal neighborhood);
(d) median filter, radius = 15 pixels.

Median Filters

- Forces the points with distinct gray levels to be more like their neighbors
- Tendency:
  - 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 $n \times n$ median filter
    - Eliminated ≡ forced to have the value equal the median intensity of the neighbors
    - Larger clusters are affected considerably less
  - Round corners
- Because of the minimal degradation to edges from median filtering, it is possible to apply the method repeatedly

Hybrid Median Filtering

How to overcome its tendency to erase lines which are narrower than the half-width of the neighborhood and to round corners ?



- Both groups include the central pixel and are ranked separately. The median of each group, and the central pixel, are then ranked again to select the final median value

Sharpening Spatial Filters

- To highlight transitions in intensity, i.e., fine details 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
  - blurring in spatial domain ~ pixel averaging in a neighborhood
  - averaging is analogous to integration
  - thus, 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

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

- Second-order derivative

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

Derivative Operator

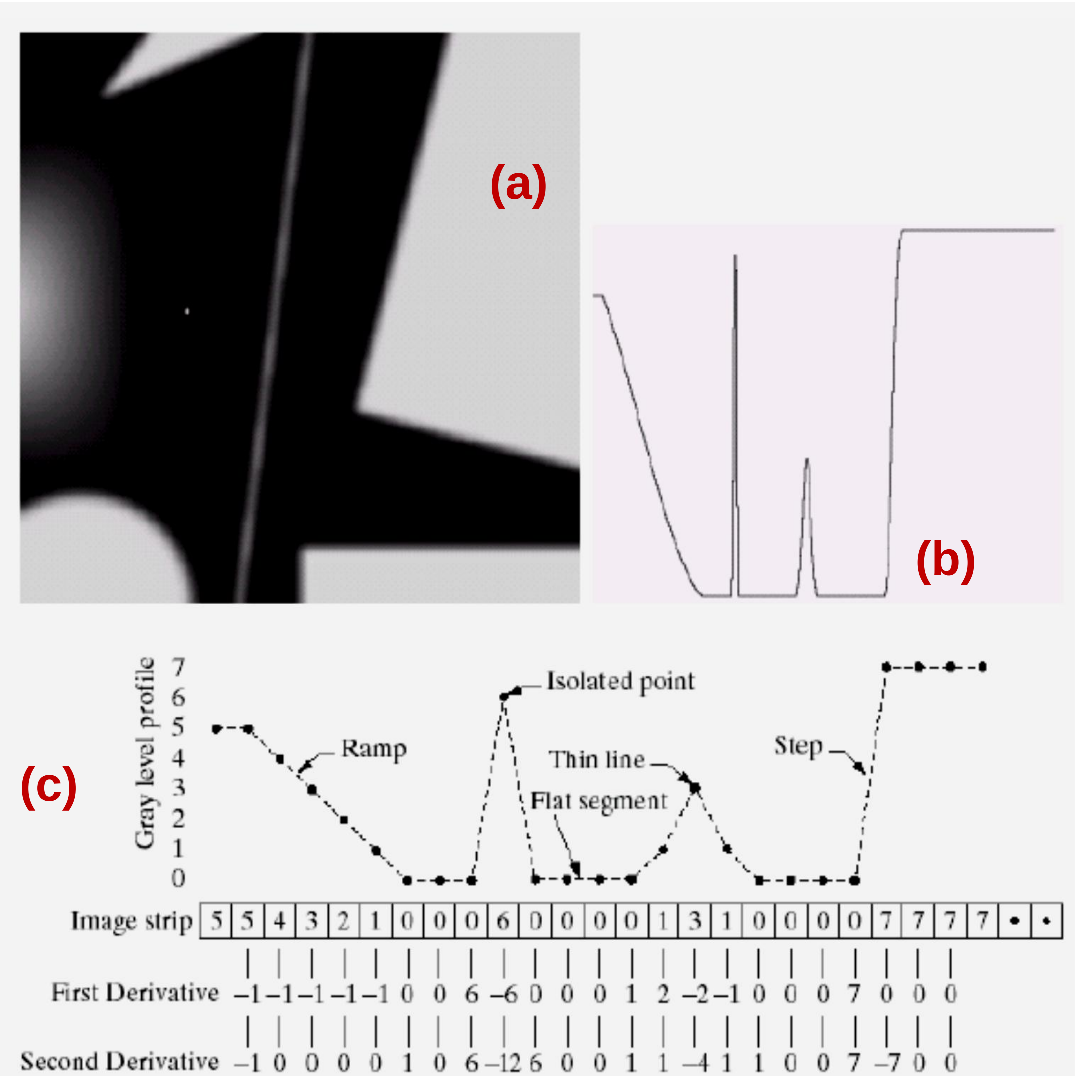
$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

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

- 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 (1)

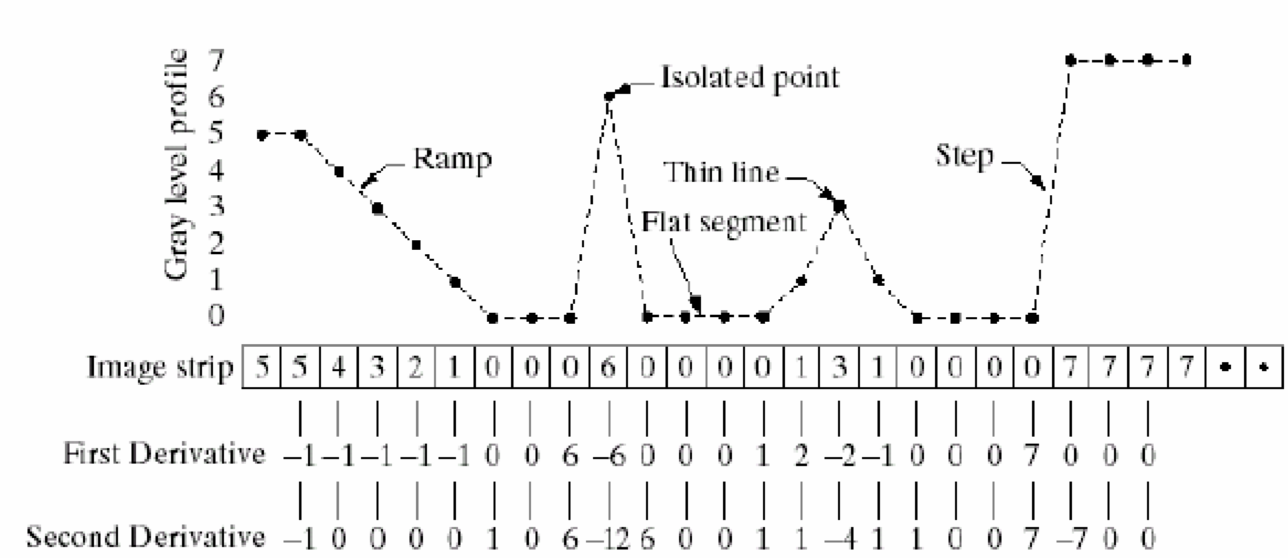


(a) A simple image

(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)

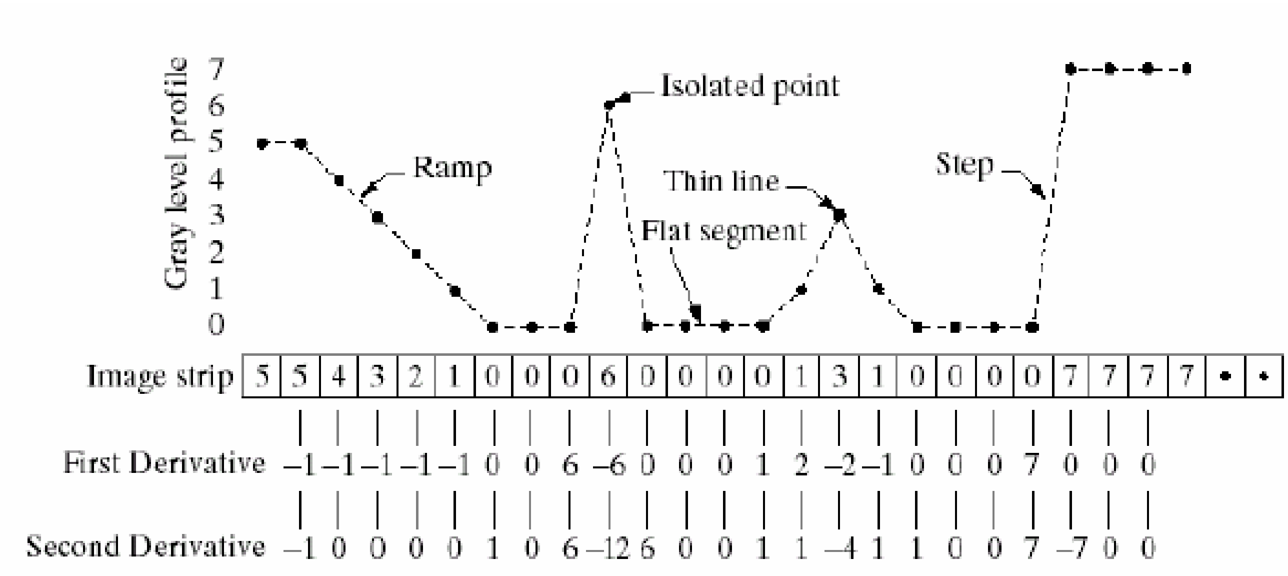
First- and Second- Order Derivatives in context of Image Processing (2)



Observations (1):

- 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 the first order derivatives produce “thick” edges and the second order derivatives produce finer edges

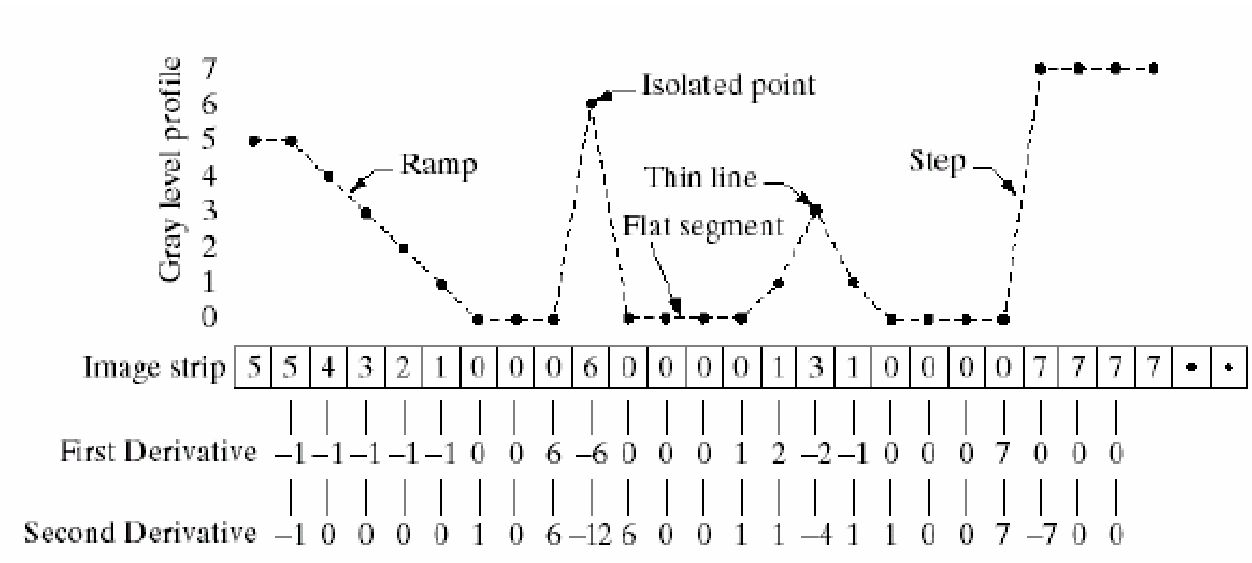
First- and Second- Order Derivatives in context of Image Processing (3)



Observations (2):

- 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 the second order derivative enhance fine detail much more than a first order derivative

First- and Second- Order Derivatives in context of Image Processing (4)



Observations (3):

- 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

Sharpening Filters: Conclusions (1)

1. First-order derivatives generally produce thicker edges in an image
2. Second order derivatives have a stronger response to fine details such as thin lines and isolated points
3. First-order derivatives generally have a stronger response to a gray-level step
4. Second order derivatives produce a double response at step changes in gray level
    - Also, for similar changes in grey-level values in an image, their response is stronger to
        - the line than to a step, and
        - to a point than to a line

Sharpening Filters: Conclusions (2)

- For image sharpening → Second order derivative

  The second derivative is more suited than the first derivative for image enhancement because of their ability to enhance fine detail

- First order derivatives are mainly used for edge extraction

- First order derivatives can be used in conjunction with the second derivative to obtain some impressive enhancement result

Second Derivative for Enhancement: The Laplacian

- Aim: defining a discrete formulation of the second-order derivative and then constructing a filter mask based on that formulation
- The filter is expected to be isotropic:
  - Response of the filter is independent of the direction of discontinuities in an image (it tends to enhance details in all directions equally)
  - Isotropic filters are rotation invariant
    - Rotating the image and then applying the filter gives the same result as applying the filter to the image first and then rotating the result

First and Second-order derivative of $f(x,y)$

- 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

Gradient operator

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

Laplacian operator

(linear operator)

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

simplest isotropic derivative operator

The equation needs to be expressed in discrete form

Discrete Form of Laplacian

from

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

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

summing the two components yield,

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

The equation can be implemented as mask

Result Laplacian mask

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

0	1	0
1	−4	1
0	1	0

This filter mask gives an isotropic result for rotations in increments of 90^0

Implementation is similar as linear smoothing filters. Only coefficients are different

The Laplacian

(a)

0	1	0
1	−4	1
0	1	0

(b)

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

(c)

−1	−1	−1
−1	8	−1
−1	−1	−1

(d)

(a) Filter mask used to implement the digital Laplacian as defined in the equation

(b) Mask used to implement an extension of this equation that includes the diagonal neighbors (this mask yields isotropic results for increment of 45^0)

(c) and (d) two other implementations of the Laplacian

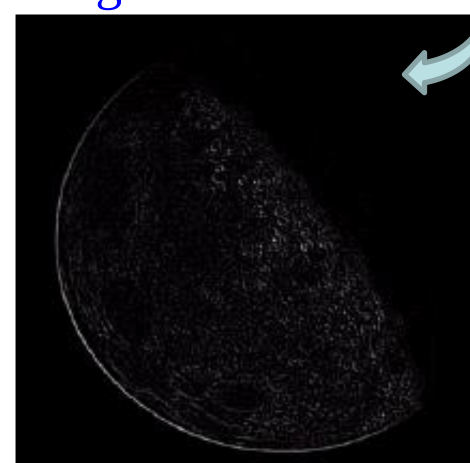
Figure (a) and (c) yield equivalent results, but the difference in sign must be kept in mind when combining (by addition or subtraction) a Laplacian-filtered image with another image

Effect of Laplacian Operator

- As it is a derivative operator,
  - highlights gray-level discontinuities in an image
  - deemphasizes regions with slowly varying gray levels
- Tends to produce images that have
  - grayish edge lines and other discontinuities, all superimposed on a dark, featureless background
- Background features can be recovered simply by adding original image to Laplacian output (still preserving the sharpening effect of the Laplacian operation)

Image of the North Pole of the moon

Laplacian filtered image with mask



Correct the Effect of Featureless Background

$$g(x,y) = f(x,y) + c\nabla^2 f(x,y)$$

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) \\ f(x,y) + \nabla^2 f(x,y) \end{cases}$$

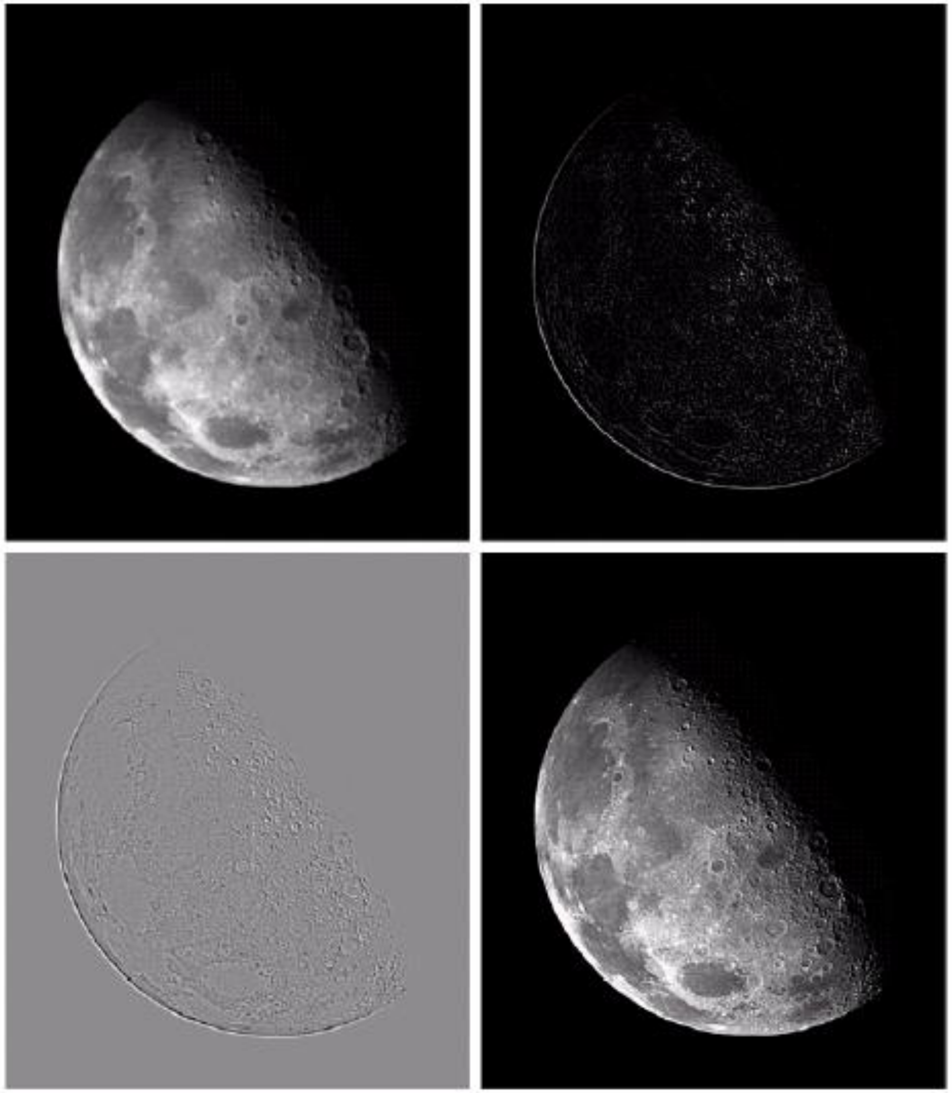
if the center coefficient of the Laplacian mask is negative

if the center coefficient of the Laplacian mask is positive

- be careful with the Laplacian filter used

The Laplacian: Example 1

(a) (b)



(c) (d)

(a) Image of the North Pole of the moon

(b) Laplacian filtered image with mask

1	1	1
1	-8	1
1	1	1

(c) Laplacian image scaled for display purposes

(d) Image enhanced by using the equation

$$g(x,y) = f(x,y) - \nabla^2 f(x,y)$$

Mask of Laplacian + Addition

- To simplify the computation, we can create a mask which does both operations, Laplacian Filter and Addition of the original image

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

0	1	0
1	- 4	1
0	1	0

$\Rightarrow$

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

The Laplacian: A Note

$$g(x,y)=\begin{cases} f(x,y)-\nabla^2 f(x,y) \\ f(x,y)+\nabla^2 f(x,y) \end{cases}$$

$$\begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 5 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} + \begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 4 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|}\hline -1 & -1 & -1 \\ \hline -1 & 9 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} + \begin{array}{|c|c|c|}\hline -1 & -1 & -1 \\ \hline -1 & 8 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 5 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} + \begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

The Laplacian: Example 2

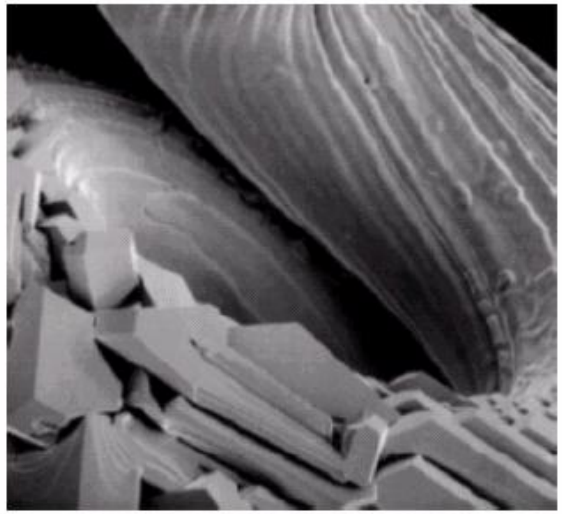
(a)

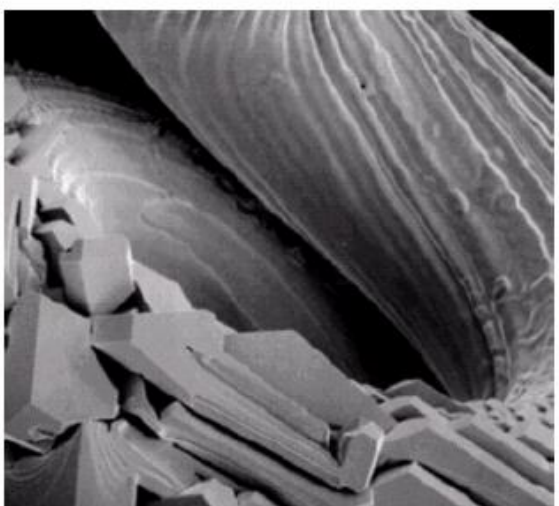
0	–1	0
–1	5	–1
0	–1	0

(b)

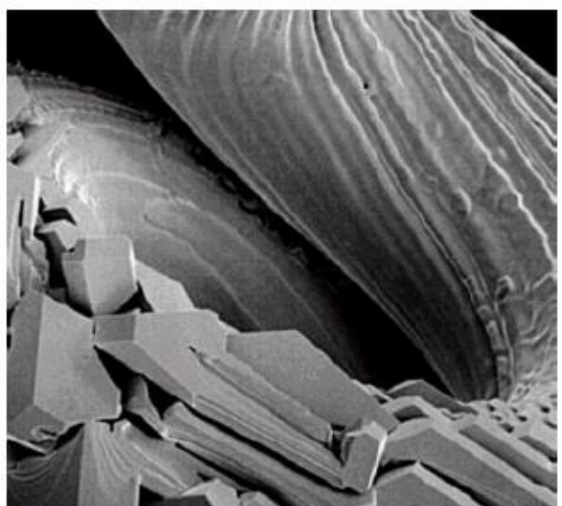
–1	–1	–1
–1	9	–1
–1	–1	–1

(c)





(d)



(e)

(a) Composite Laplacian mask

(b) A second composite mask

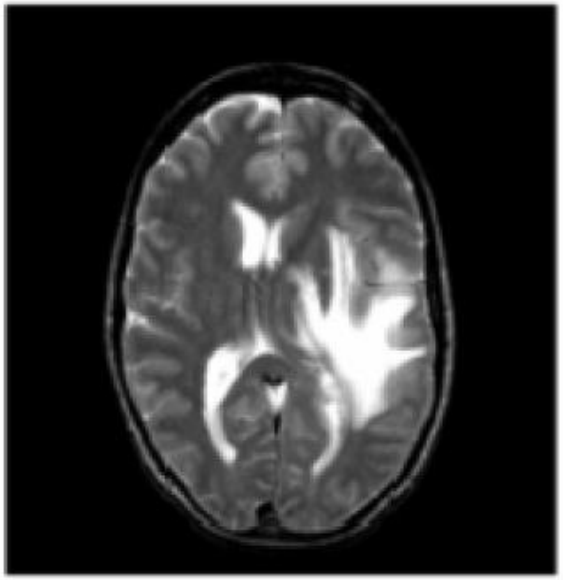
(c) Scanning electron microscope image

(d) Result of filtering with the mask in (a)

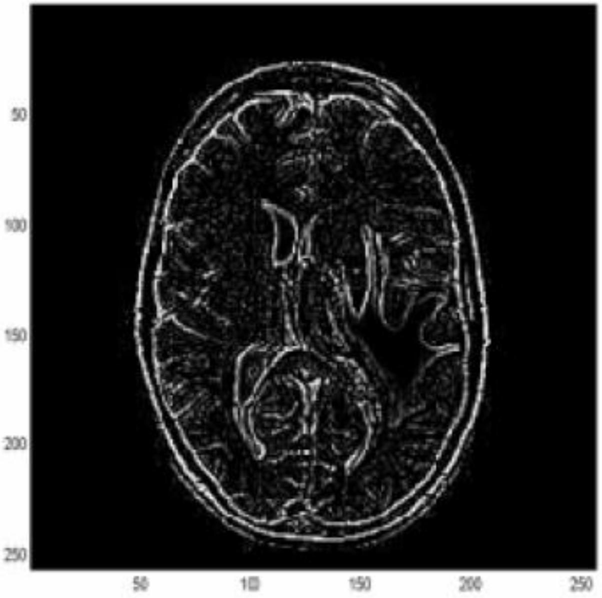
(e) Result of filtering with the mask in (b)

Note how much sharper (e) is than (d)

The Laplacian: Example 3



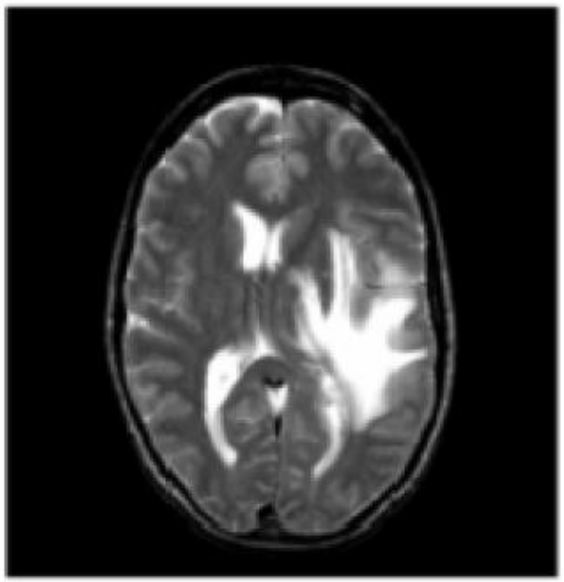
0	-1	0
-1	**8**	-1
0	-1	0



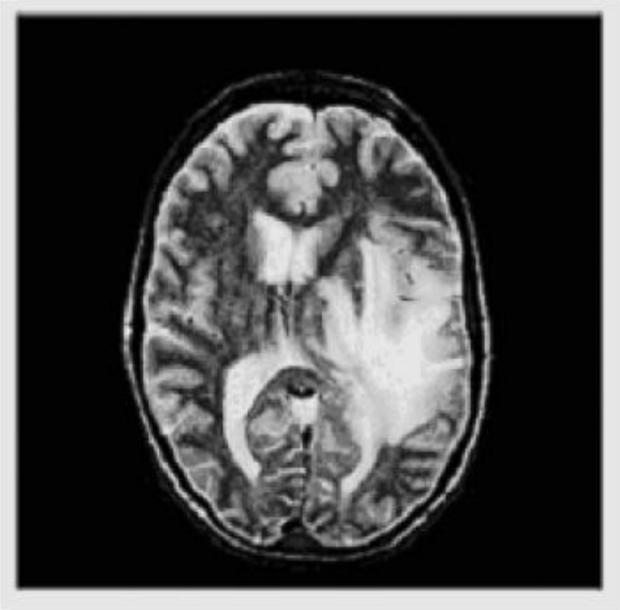
Before filtering

After filtering

The Laplacian: Example 4



-1	-1	-1
-1	9	-1
-1	-1	-1



Before filtering

After filtering

Unsharp Masking

- Subtracting a blurred version of an image from the original image to produce sharpen output image
- Origin: dark room photography → clamping together a blurred negative to a corresponding positive film and then developing this combination to produce a sharper image

- Unsharp Masking:
  1. Blur the original image
  2. Subtract the blurred image from the original
  3. Add the mask to the original

Unsharp Masking

$$g_{mask}(x,y) = f(x,y) - \bar{f}(x,y)$$

$\bar{f}(x,y)$: blurred version of original image

$$g(x,y) = f(x,y) + k * g_{mask}(x,y)$$

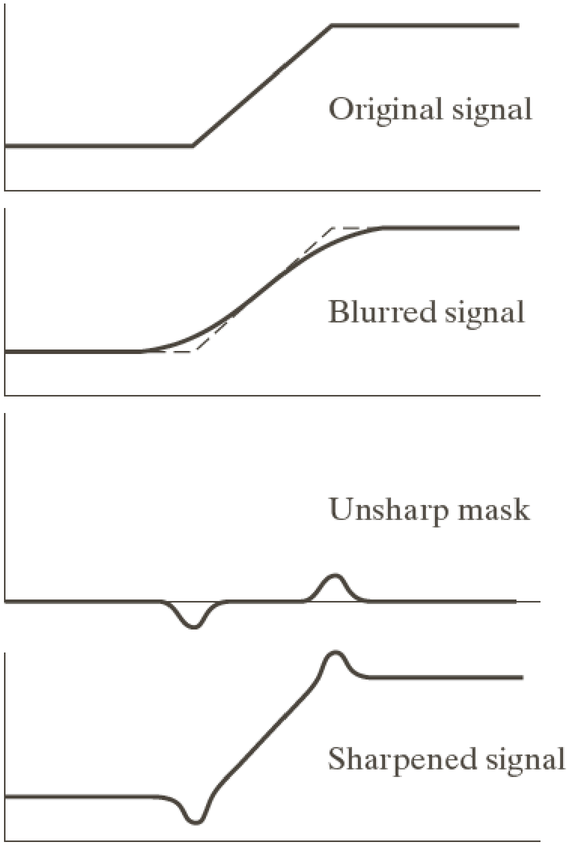
A weight, k ($k \geq 0$) is included for generality.

$k = 1$ → Unsharp Masking

$k > 1$ → Highboost Filtering

$k < 1$ → de-emphasizes the contribution of the unsharp mask

Working of Unsharp Masking



a
b
c
d

FIGURE 3.39 1-D illustration of the mechanics of unsharp masking. (a) Original signal. (b) Blurred signal with original shown dashed for reference. (c) Unsharp mask. (d) Sharpened signal, obtained by adding (c) to (a).

- The points at which a change of slope in the intensity occurs in the signal are now emphasized (sharpened)
- Observe that –ve values are added to the original

- Result may have –ve intensities if

- the original image has any zero values, or
- the value of *k* is chosen large enough to emphasize the peaks of the mask to a level larger than the min value in the original
  - -ve values would cause a dark halo around edges – not acceptable

Image Sharpening - Unsharp Masking

a
b
c
d
e

FIGURE 3.40 (a) Original image. (b) Result of blurring with a Gaussian filter. (c) Unsharp mask. (d) Result of using unsharp masking. (e) Result of using highboost filtering.

- Fig (d): $k = 1$
- Fig (e): $k = 4.5$, which is the largest possible value used and still keeps all the values +ve in the result
- Overall a significant improvement over original

The Gradient

- 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

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

- Geometrical property of gradient vector:

  it points in the direction of the greatest rate of change of f at location (x, y)

- 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

The Gradient

- 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

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

Use of Gradient Operators

Edge Detection through Gradient Operators

- Motivation: detect changes   change in the pixel value $\longrightarrow$ large gradient

(*a*) Original (*b*) Horizontal magnitude (*c*) Vertical magnitude

$$\text{image} \xrightarrow[f(m,n)]{} \boxed{\text{Gradient operator}} \xrightarrow[M(m,n)]{} \boxed{\text{Thresholding}} \xrightarrow[I(m,n)]{} \text{Edge map}$$

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

Gradient Computation

- 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

3×3 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

- **Roberts Cross-Gradient Operators:** (Sum of the magnitude of the differences of the diagonal neighbors)

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- $g_x : (z_9 - z_5)$, $\quad g_y : (z_8 - z_6)$
- Gradient image can be computed as:

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

- 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 × 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 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

3×3 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

3×3 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

- 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)

Gradient Masks

- Diagonal edges with Prewitt and Sobel masks

(to emphasize edges along the diagonal directions)

0	1	1
−1	0	1
−1	−1	0

−1	−1	0
−1	0	1
0	1	1

Prewitt

0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

Sobel

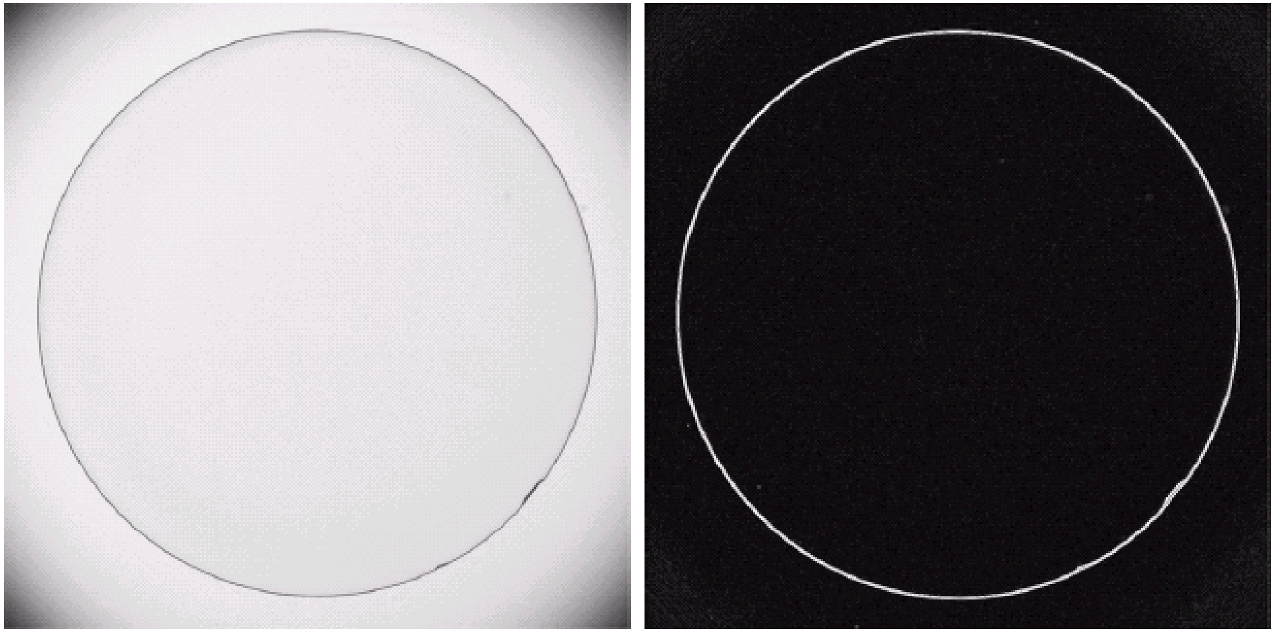
Effect of approximating the gradient

- Commonly approx. $$M(x,y) \approx |g_x| + |g_y|$$
- Advantage: computationally attractive and it still preserves relative changes in gray levels.
- Drawback: resulting filters will not be isotropic (invariant to rotation) in general.
- Not an issue when Prewitt and Sobel masks are used to compute g_x and g_y.
- These masks give isotropic results only for vertical and horizontal edges even if we use exact computation to calculate gradient.

The Gradient

Use of First Derivatives for Enhancement

- Gradient is used frequently in industrial inspection as:
  - an aid to humans in the detection of defects, or
  - a preprocessing step in automated inspection

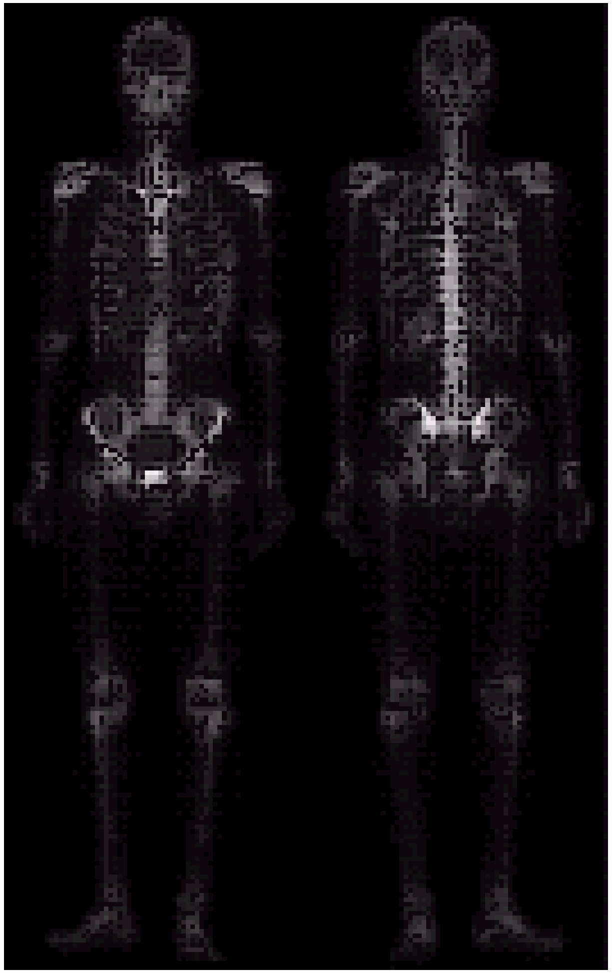


a b

FIGURE 3.45 Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock). (b) Sobel gradient. (Original image courtesy of Mr. Pete Sites, Perceptics Corporation.)

Example to show that a gradient can be used to enhance defects and eliminate slowly changing background features

Example - 1 of Combining Spatial Enhancement Methods

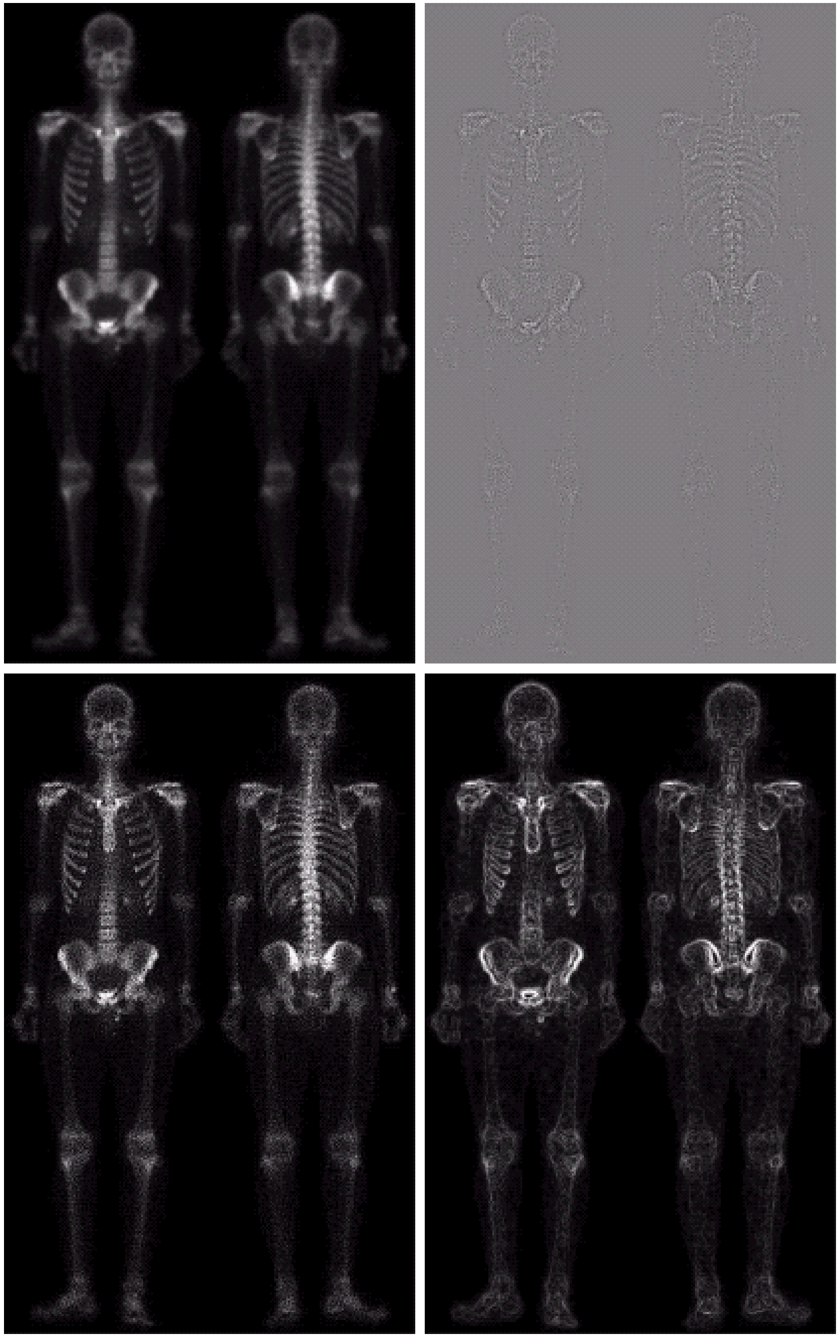


- Want to sharpen the original image and bring out more skeletal detail
- Problems: narrow dynamic range of gray level and high noise content makes the image difficult to enhance

Example 1 of Combining Spatial Enhancement Methods

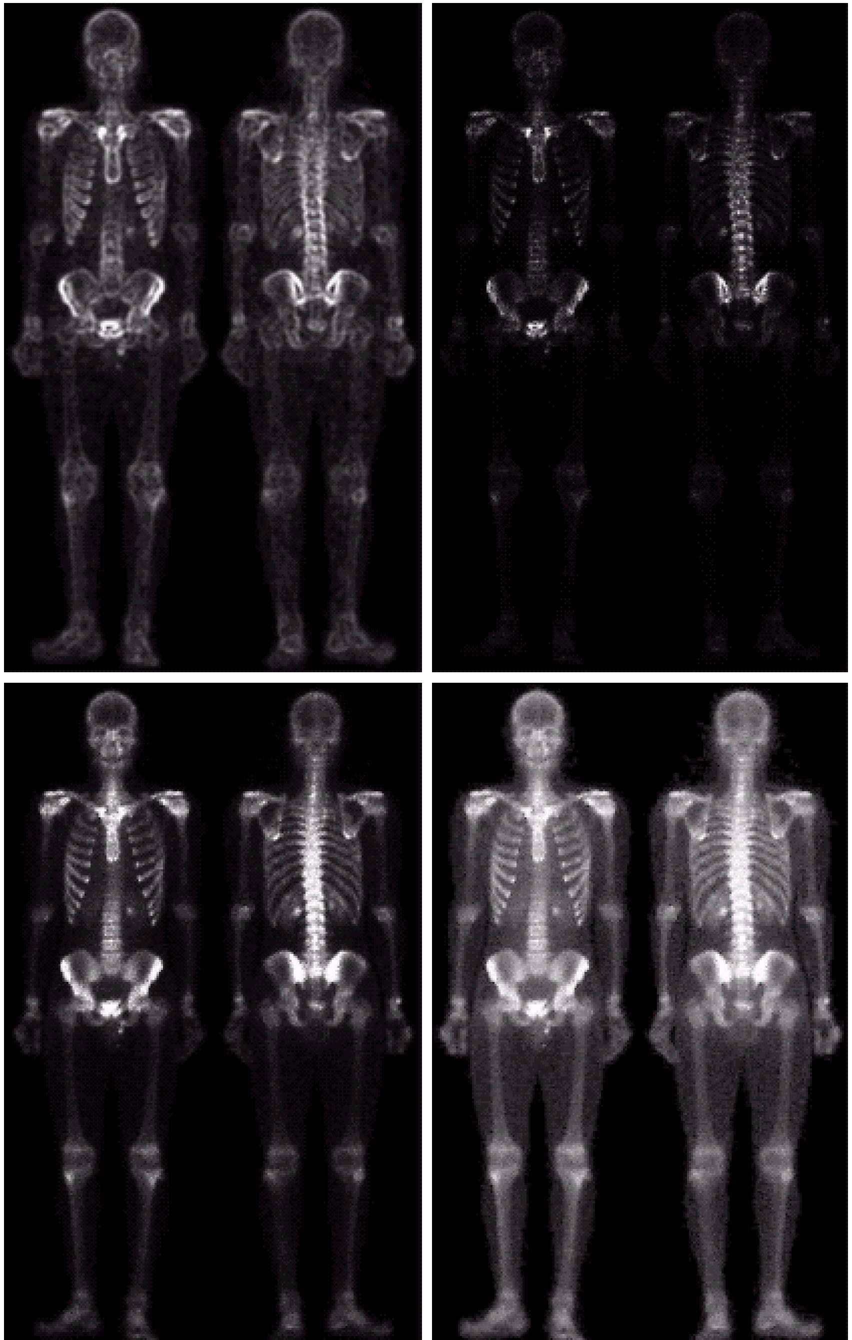
- Problems: narrow dynamic range of gray level and high noise content makes the image difficult to enhance
- Solution :

1. Laplacian to highlight fine detail
2. Gradient to enhance prominent edges
3. Gray-level (i.e., intensity) transformation to increase the dynamic range of gray levels



a	b
c	d

FIGURE 3.46
(a) Image of whole body bone scan. (b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b). (d) Sobel of (a).



e	f
g	h

FIGURE 3.46 *(Continued)* (e) Sobel image smoothed with a 5 × 5 averaging filter. (f) Mask image formed by the product of (c) and (e). (g) Sharpened image obtained by the sum of (a) and (f). (h) Final result obtained by applying a power-law transformation to (g). Compare (g) and (h) with (a). (Original image courtesy of G.E. Medical Systems.)

Example 2

A skilled medical technician is charged with the job of inspecting a certain class of images generated by an electron microscope. In order to simplify the inspection task, the technician decides to use digital image enhancement and, to this end, examines a set of representative images and finds the following problems:

1. bright, isolated dots that are of no interest;
2. not enough contrast in some images; and
3. shifts in the average gray-level value, when this value should be V to perform correctly certain intensity measurements.

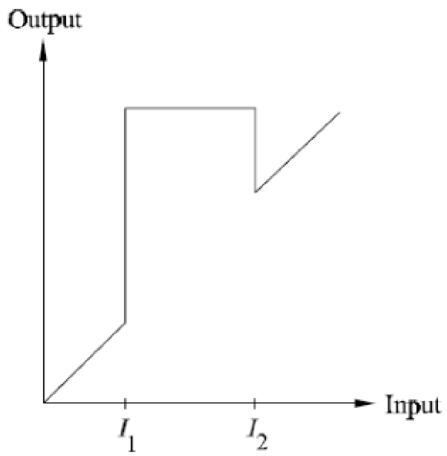
The technician wants to correct these problems and then display in white all gray levels in the band between I1 and I2, while keeping normal tonality in the remaining gray levels.

Propose a sequence of processing steps that the technician can follow to achieve the desired goal.

Example 2: Solution

The problem can be solved by carrying out the following steps:

1. Perform a median filtering operation.
2. Perform histogram equalization on the result of Step 1.
3. Compute the average gray level, say V0. Add the quantity (V – V0) to all pixels.
4. Perform the transition shown in figure:



References

- Medical Image Processing course at McMaster University, Canada
- Digital Image Processing course by Dr. W. Thongsongkrit wanasana@eng.cmu.ac.th
- Digital Image Processing course by Dr. Brian Mac Namee http://www.comp.dit.ie/bmacnamee/
- R. C. Gonzalez and R. E. Woods, Digital Image Processing, Third Edition, Pearson, 2012
- John C. Russ, The Image Processing Handbook, Sixth Edition, CRC Press, 2010