Image Processing

Lecture 03: Spatial Filtering

(Ch3 Intensity Transformation and Spatial Filtering – II)

Zhiguo Zhang

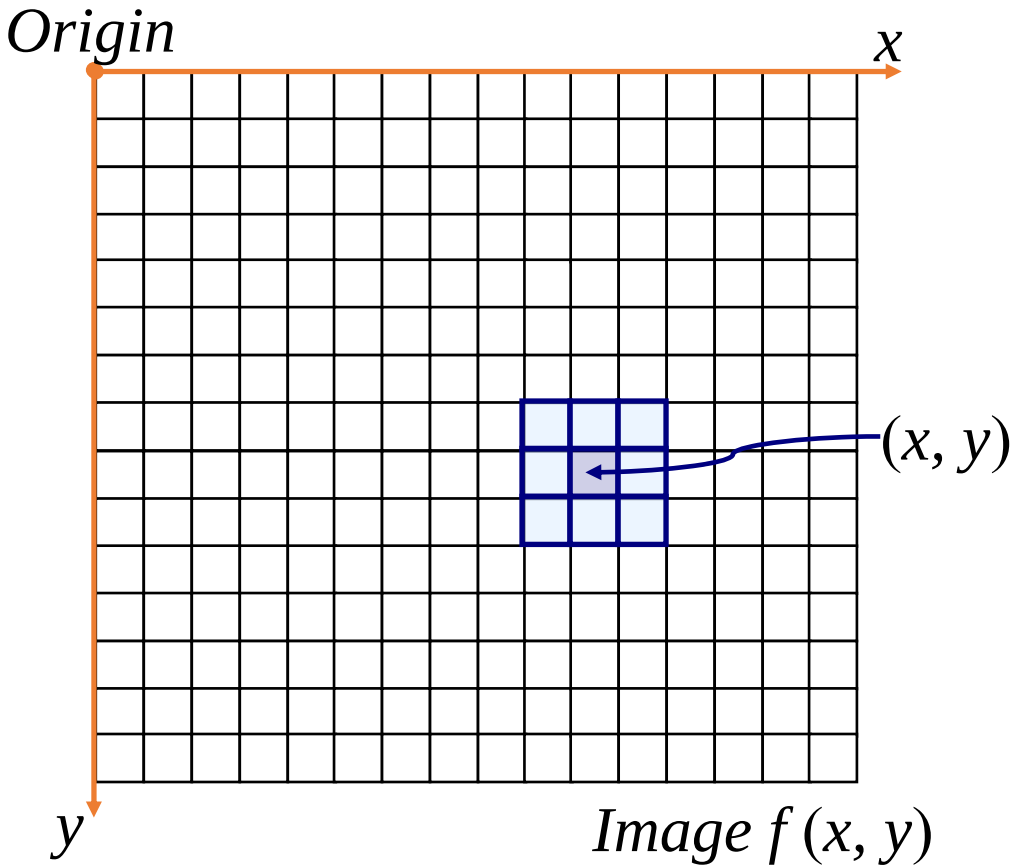
zhiguozhang@hit.edu.cn

Contents of This Lecture

- ✓ Basic of spatial filtering
- ✓ Smoothing spatial filters
  - Smoothing linear filters
  - Order-statistics filters
- ✓ Sharpening spatial filters
  - 1st derivative filters
  - 2nd derivative filters

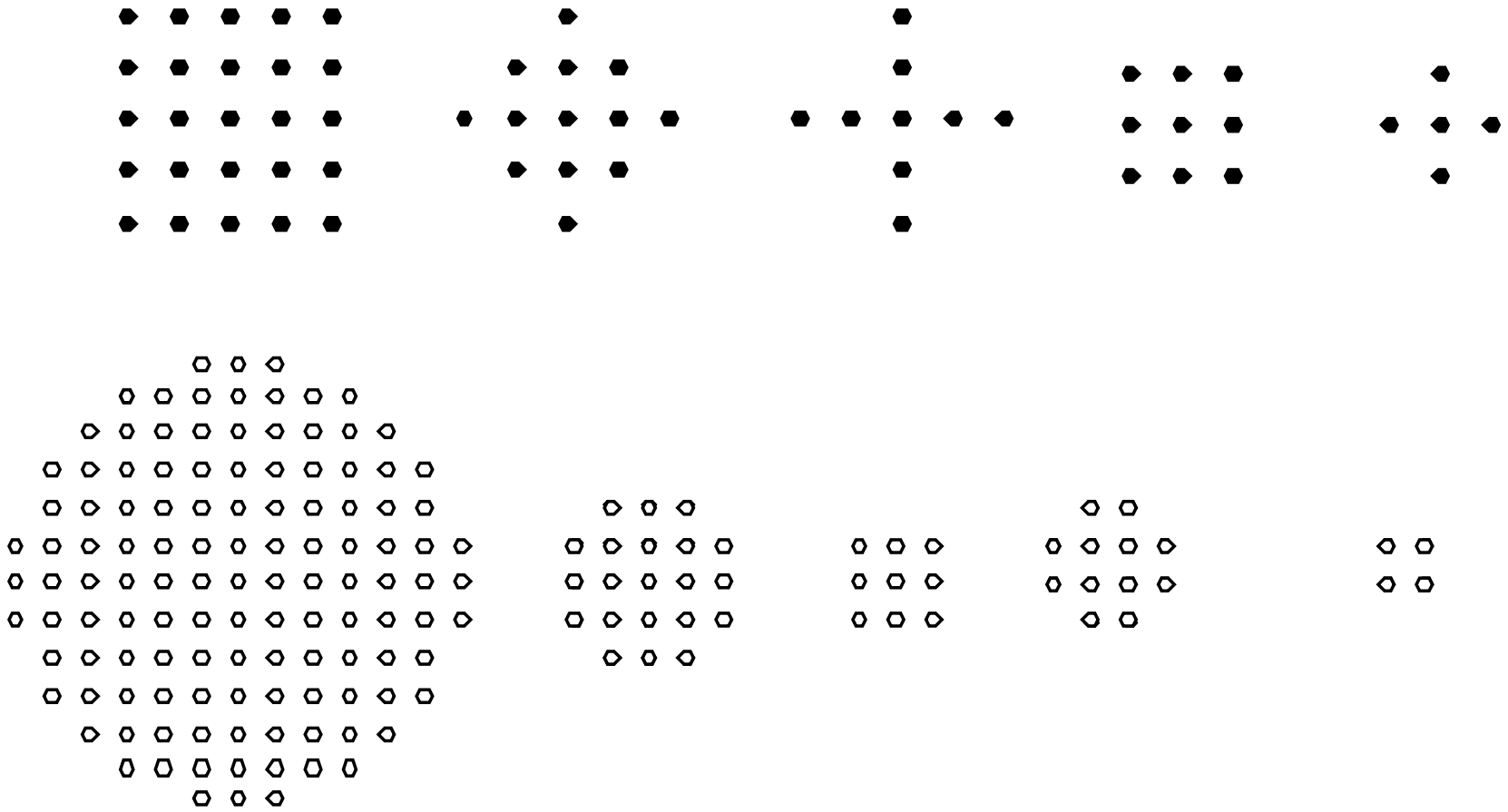
Basic of Spatial Filtering

- Neighbourhood processing simply operates on a larger neighbourhood of pixels than point processing.
- Neighbourhoods are mostly a rectangle around a central pixel.
- A neighbourhood of any size and any shape is possible.



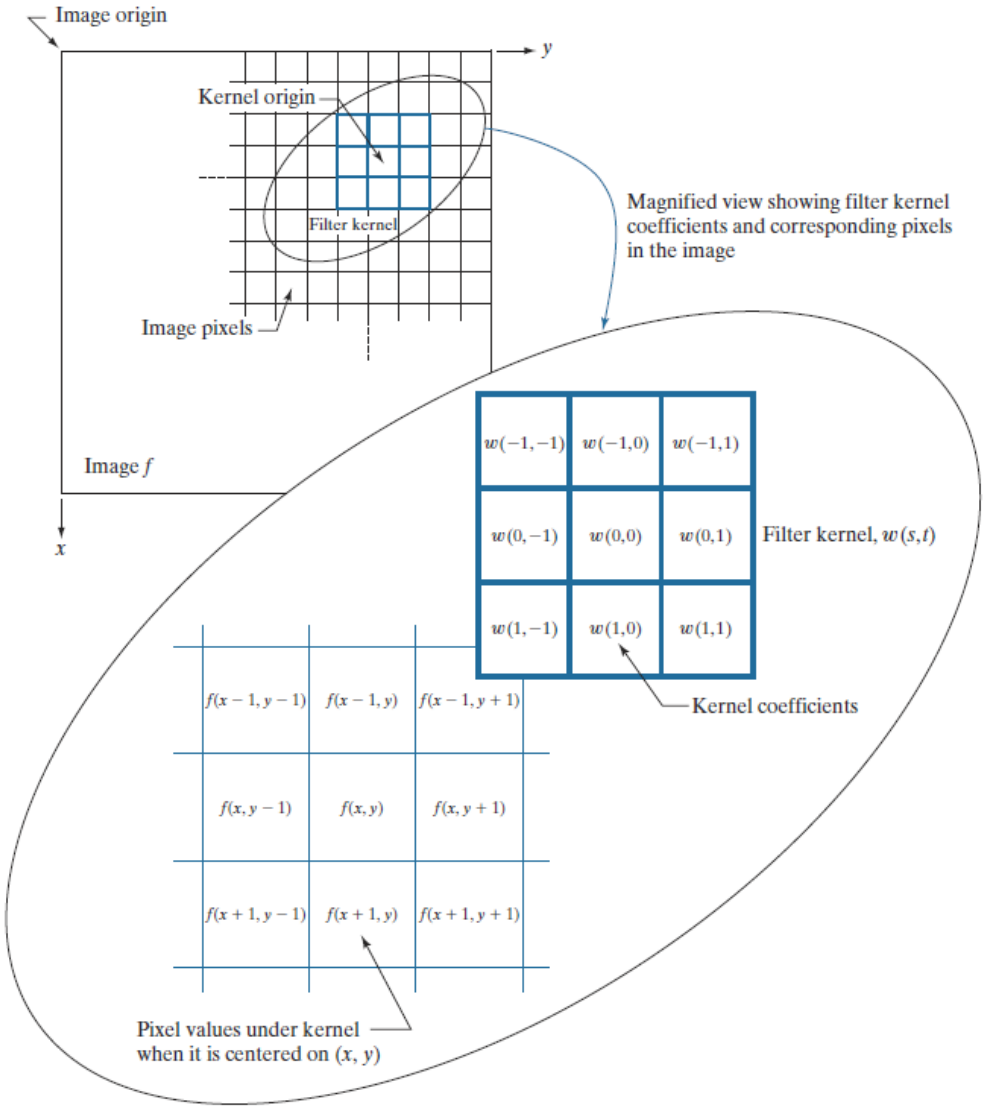
Basic of Spatial Filtering

- Typical Filter Masks



Basic of Spatial Filtering

- Use of spatial masks for image processing (spatial filters)
  - Linear filters
    - ➢Correlation
    - ➢Convolution
  - Nonlinear filters



The mechanics of linear spatial filtering using a 3 × 3 kernel. The pixels are shown as squares to simplify the graphics. Note that the origin of the image is at the top left, but the origin of the kernel is at its center. Placing the origin at the center of spatially symmetric kernels simplifies writing expressions for linear filtering.

Spatial Linear Filtering

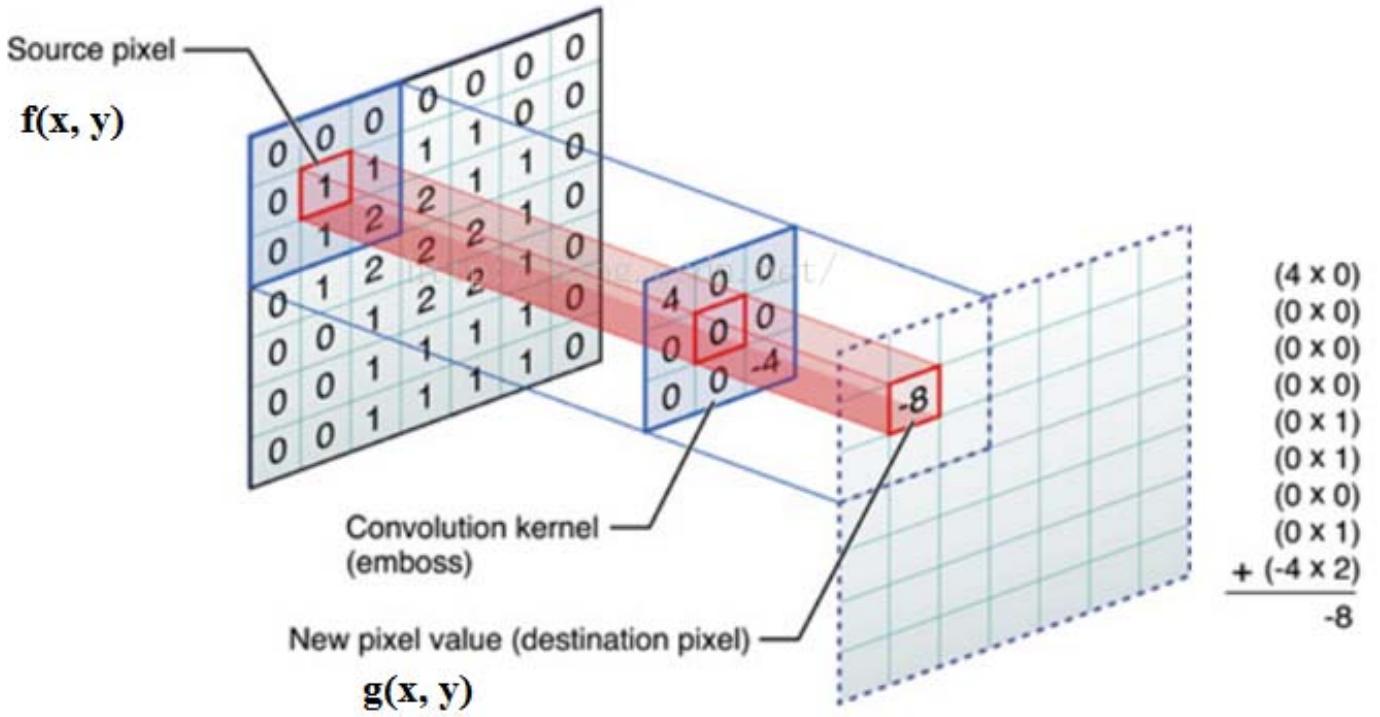
- Spatial Correlation:

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

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

$$R = w_1 z_1 + w_2 z_2 + \cdots + w_{mn} z_{mn}$$

$$= \sum_{i=1}^{mn} w_i z_i$$



Spatial Linear Filtering

- Spatial Convolution:
  - The mechanics of spatial convolution are the same, except that the correlation kernel is rotated by 180°.
  - Thus, when the values of a kernel are symmetric about its center, correlation and convolution yield the same result.
  - Because convolution is commutative, it is immaterial whether $\boldsymbol{w}$ or $\boldsymbol{f}$ is rotated, but rotation of the kernel is used by convention.

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

Spatial Linear Filtering

Correlation $$g(x,y) = \boldsymbol{w} \cdot \boldsymbol{f}(x,y) = \sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$ Eq. 1

Convolution $$g(x,y) = \boldsymbol{w} \cdot \boldsymbol{f}(x,y) = \sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t) f(x-s, y-t)$$ Eq. 2

- Often, spatial filtering algorithms are based on correlation and thus implement Eq. 1 instead. For correlation, we input $\boldsymbol{w}$; for convolution, we input $\boldsymbol{w}$ rotated by 180^0.
- The opposite is true for a correlation algorithm that implements Eq. 2.
- Thus, either Eq. 1 or Eq. 2 can be made to perform the function of the other by rotating the filter kernel.

Edge Padding

- At the edges of an image we are missing pixels to form a neighbourhood.

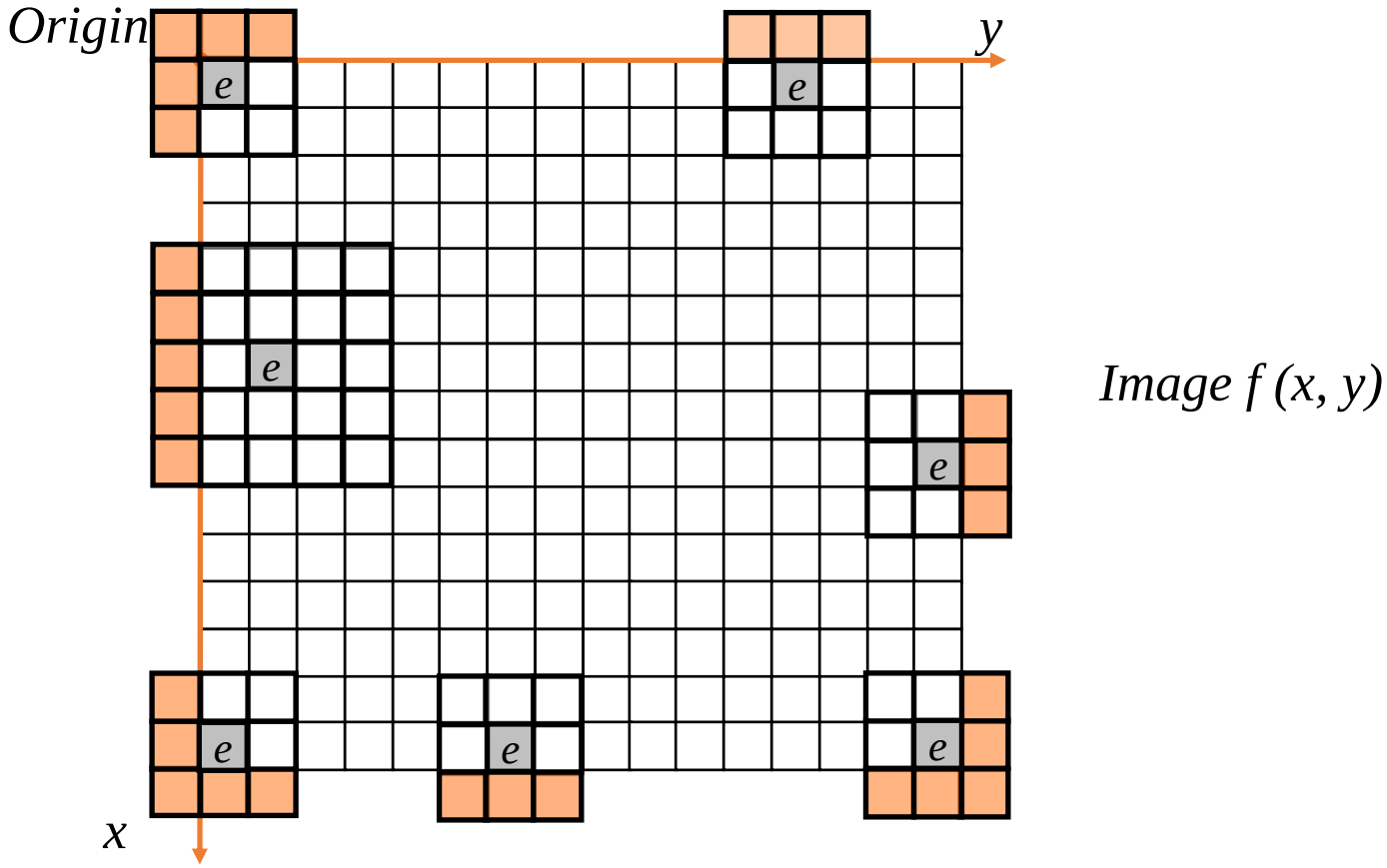


Image f (x, y)

Edge Padding (cont…)

- There are a few approaches to deal with missing edge pixels:
  - Pad the image
  - Omit missing pixels
    - ➢Can add extra code and slow down processing
  - Truncate the image

Edge Padding (cont…)

- Padding
  - Zero padding: values outside the boundary of the image are set to zero
  - Replicate padding: values outside the boundary are set equal to the nearest image border value
  - Mirror/symmetric padding: values outside the boundary of the image are obtained by mirror-reflecting the image across its border

Zero padding

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

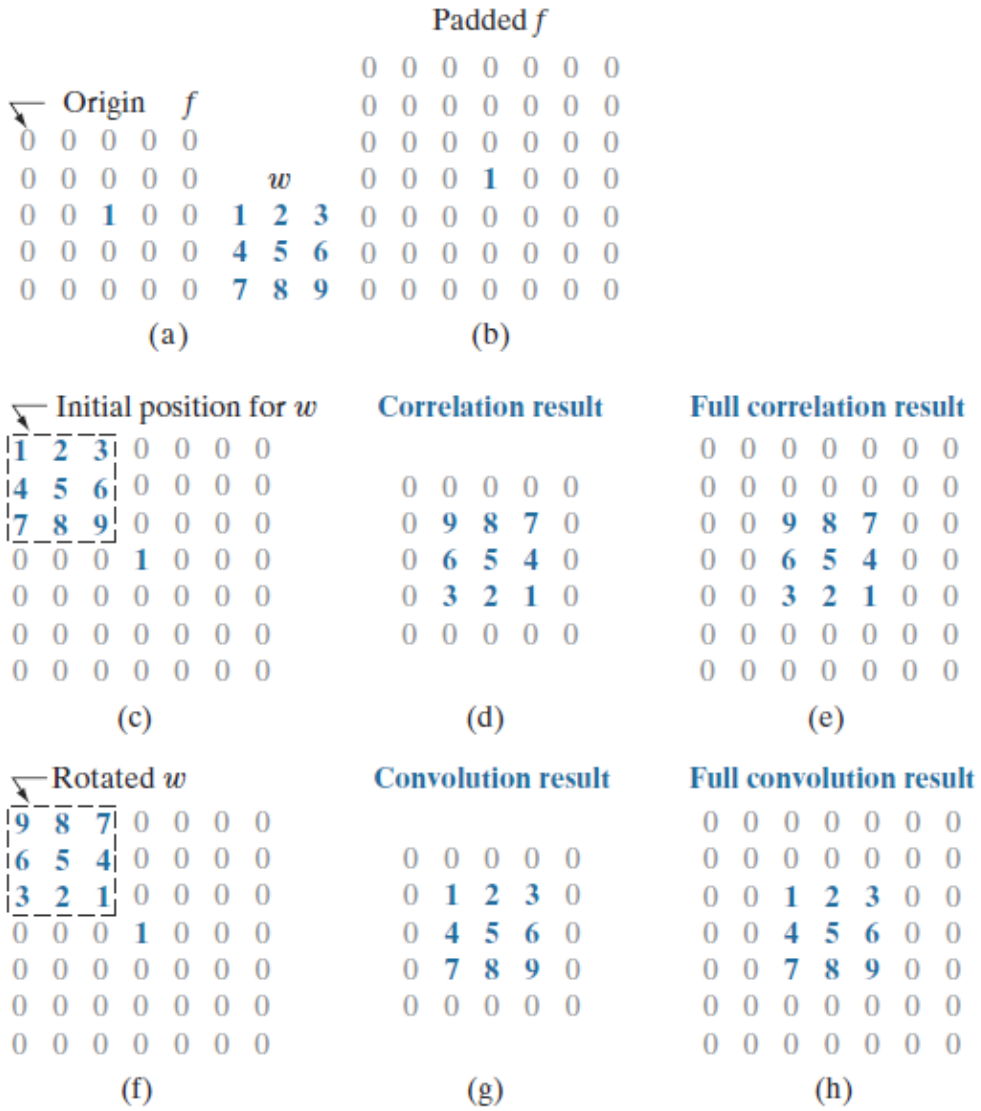
Replicate padding

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

Mirror padding

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

Correlation and Convolution: Example



- Correlation (middle row) and convolution (last row) of a 2-D kernel with an image consisting of a discrete unit impulse.
  - The 0's are shown in gray to simplify visual analysis.
  - Note that correlation and convolution are functions of x and y.
  - As these variable change, they *displace* one function with respect to the other.

Spatial Smoothing Filtering

- Used for noise reduction and blurring (removal of small details prior to large object extraction, bridging small gaps in lines).

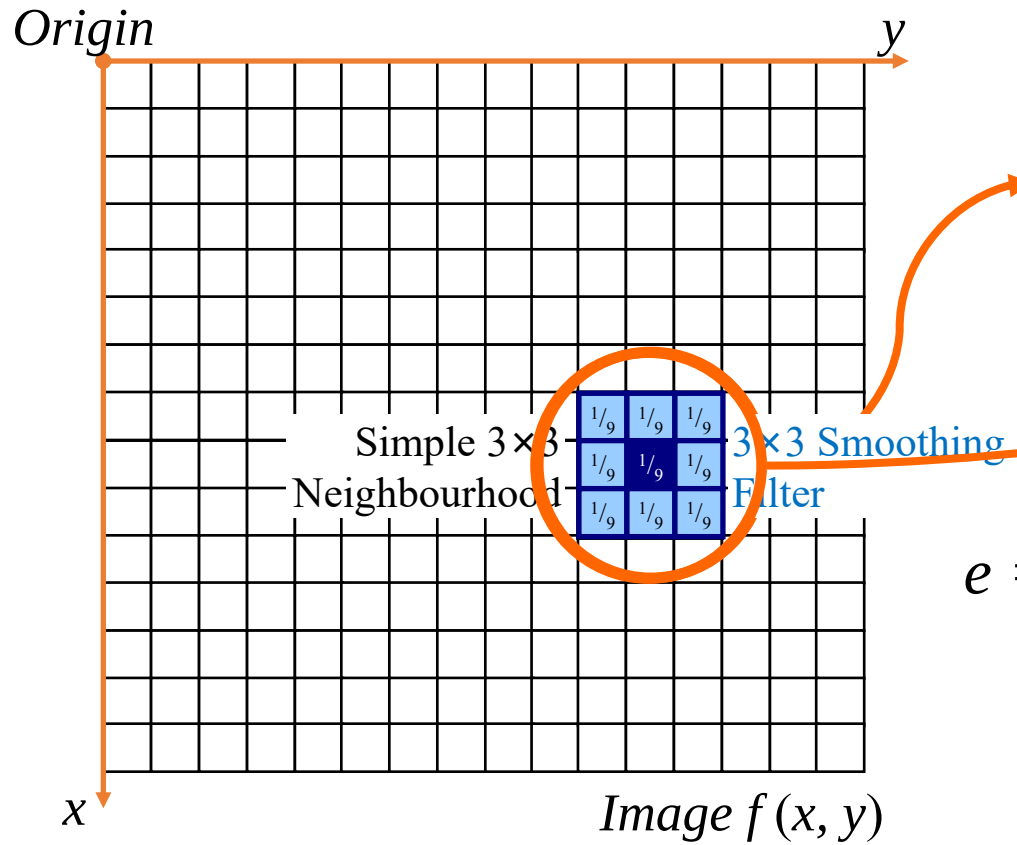
- Two types of spatial smoothing filtering methods
  1. Smoothing linear filters
  2. Order-statistics filters

Smoothing Linear Filters

- Smoothing linear filters are also called averaging filter
  - Simply average all of the pixels in a neighbourhood around a central value
  - Especially useful in removing noise from images
  - Also useful for highlighting gross features

$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$

Smoothing Linear Filters



104	100	108
99	106	98
95	90	85

Original Image Pixels

•

$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$

Spatial Filter

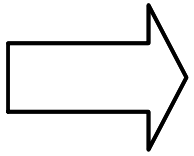
$$e = 1/9 \times (106 + 104 + 100 + 108 + 99 + 98 + 95 + 90 + 85)$$
$$= 98.3333 \approx 98$$

The above is repeated for every pixel in the original image to generate the smoothed image.

Smoothing Linear Filters

$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$

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



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

The results are rounded to the nearest integer.

Smoothing Linear Filters

- More effective smoothing filters can be generated by allowing different pixels in the neighbourhood to have different weights in the averaging function:
  - Pixels closer to the central pixel are more important.
  - The summation of all weights is 1.
  - Such an operation is often referred to as *weighted averaging*.

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

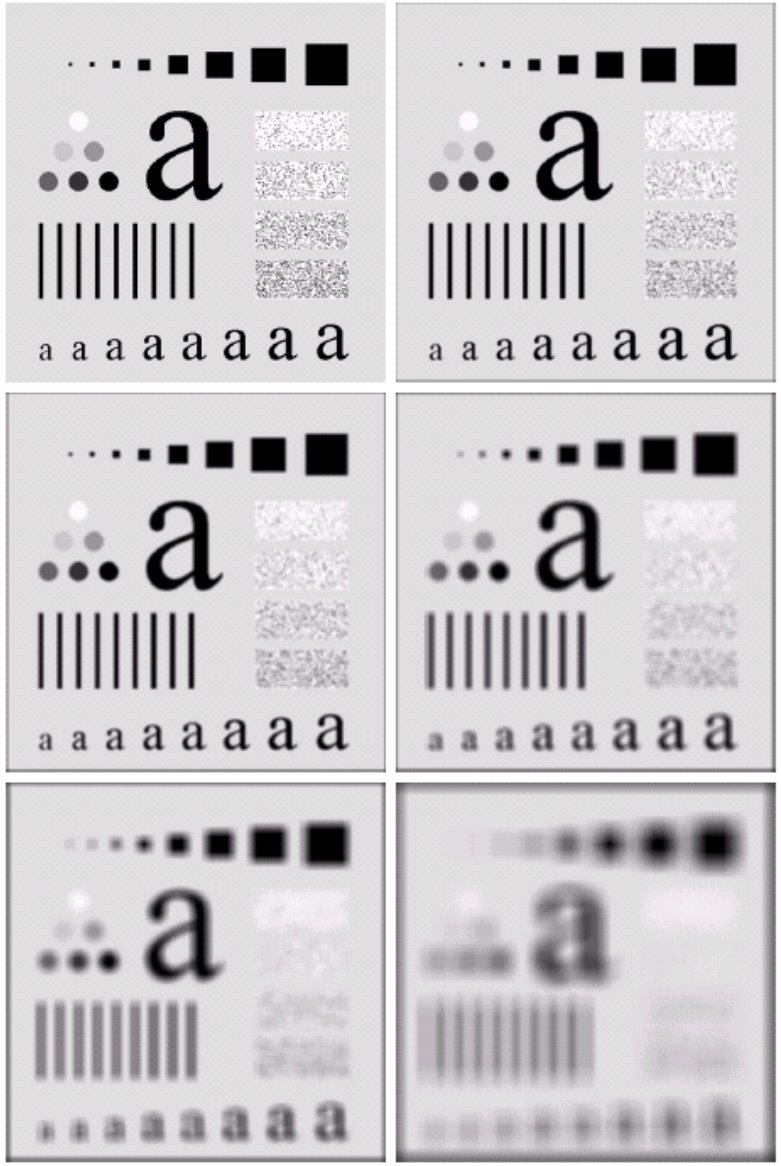
weighted averaging filter

Smoothing Linear Filters

- This process results in an image with reduced sharp transitions in gray levels.
- Because random noise typically consists of sharp transitions in gray levels, the most obvious application of smoothing is noise reduction.
- Averaging filters have the undesirable side effect that blurs edges.
- A major use of averaging filters is in the reduction of “irrelevant” detail in an image.

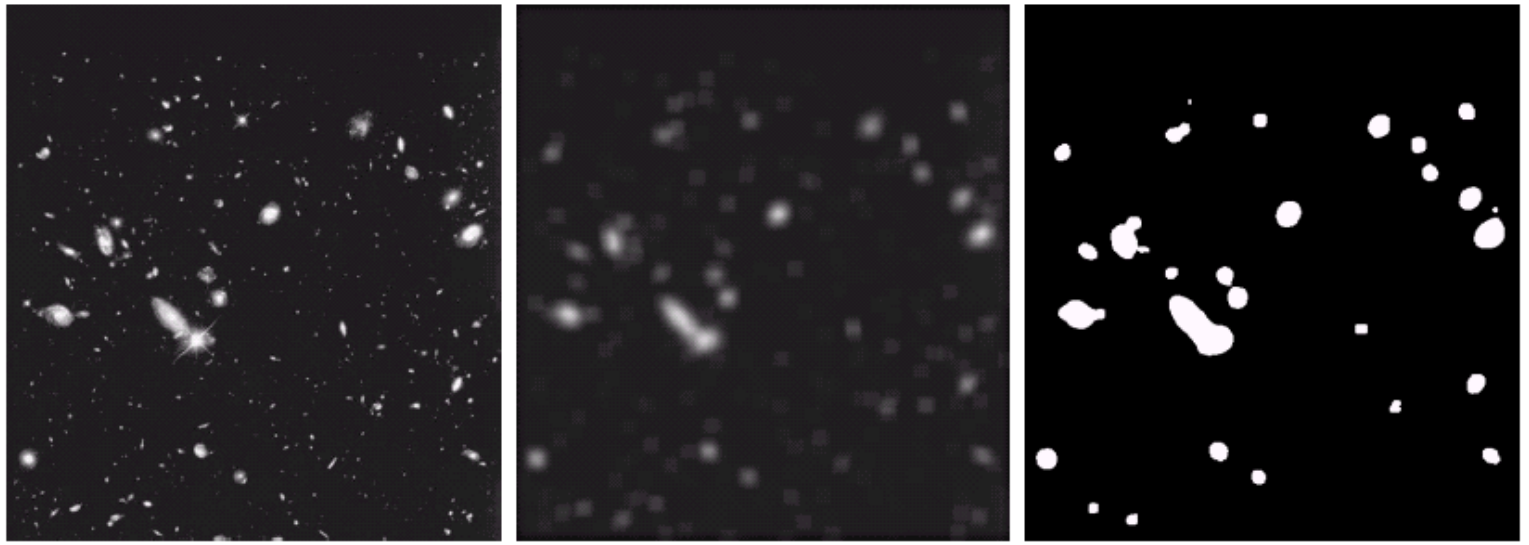
Smoothing Linear Filters

- The image at the top left is an original image of size 500×500 pixels.
- The subsequent images show the image after filtering with an averaging filter of increasing sizes: 3, 5, 9, 15 and 35.
- Notice how the detail begins to disappear.



Smoothing Linear Filters

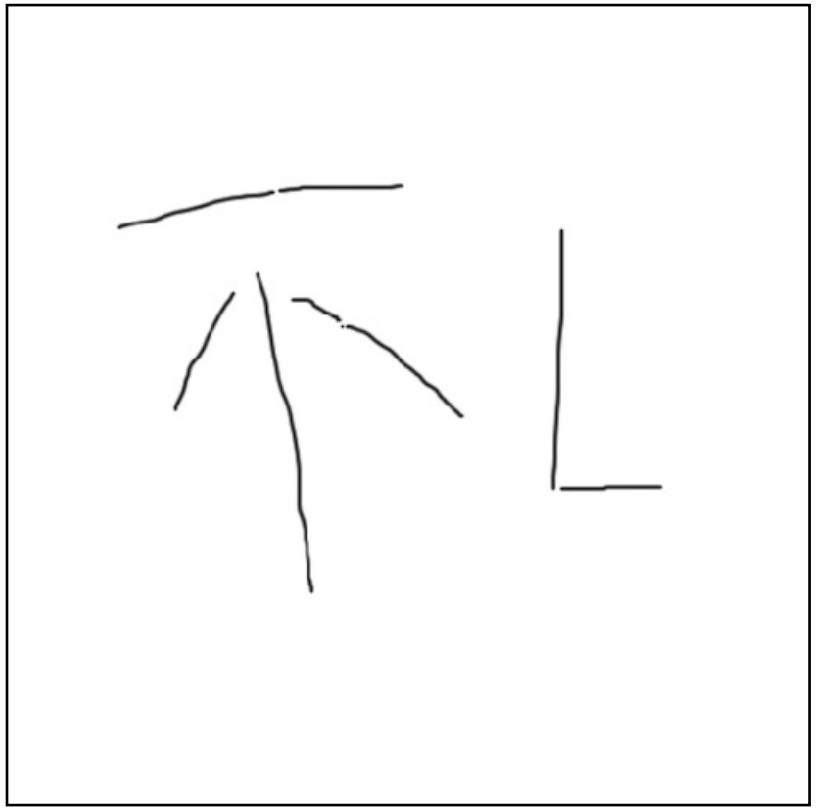
- By smoothing the original image we get rid of lots of the finer detail and only leave gross features for thresholding.



Original Image Smoothed Image Thresholded Image

Smoothing Linear Filters

- Smoothing filters blur edges and sharp transitions.



Order-Statistics Filters

- Order-statistics filters are ***nonlinear*** spatial filters whose response is based on ordering the pixels in the area of filter.
- Some typical operations include:
  - **Min:** Set the pixel value to the minimum in the neighbourhood
  - **Max:** Set the pixel value to the maximum in the neighbourhood
  - **Median:** Set the pixel value to the median in the neighbourhood
    - The median value of a set of numbers is the midpoint value in that set (e.g. for a 3×3 neighbourhood, with gray level (10,20,20,20,15,20,20,25,100), then we can order them in (10,15,20,20,20,20,20,25,100), 20 is the median. Sometimes the median works better than the average.

Order-Statistics Filters

- Sometimes a median filter works better than an averaging filter.
- Median filters are particularly effective in the presence of impulse noise, also called salt-and-pepper noise.

Salt-and-pepper noise presents itself as sparsely occurring white and black pixels.

- salt noise: extremely high values;
- pepper noise: extremely low values.

Order-Statistics Filters

a

b

c

d

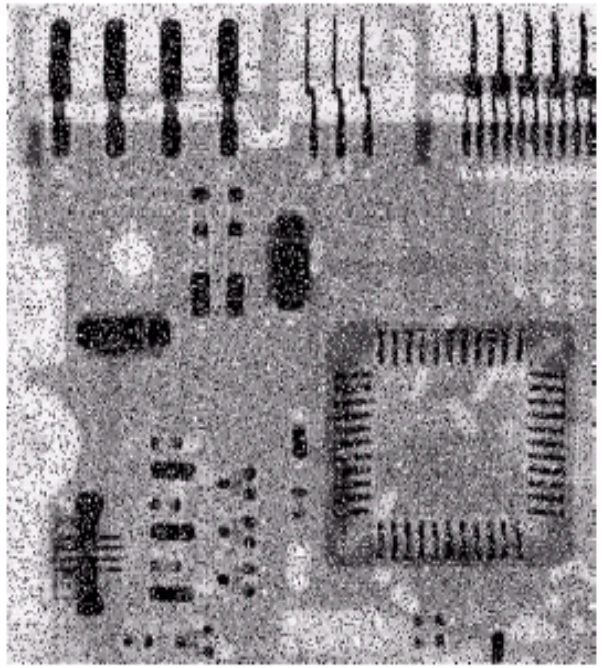
e

f

g

(a) Lena; (b) and (c): Lena with Gaussian noise and Salt-and-pepper noise, respectively;
(d) and (e): Smoothing with a square average filter on (b) and (c), respectively;
(f) and (g) smoothing with a 5×5 median filter on (b) and (c), respectively.

Order-Statistics Filters



Original Image
with Salt-and-Pepper Noise

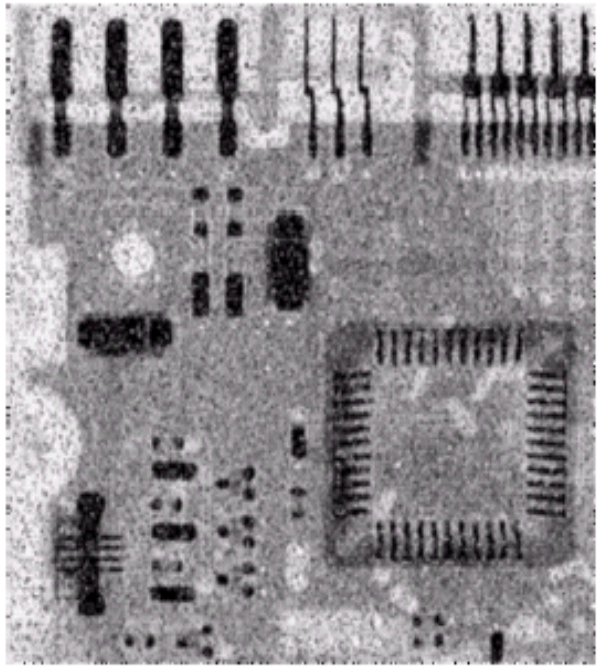


Image after
Averaging Filter

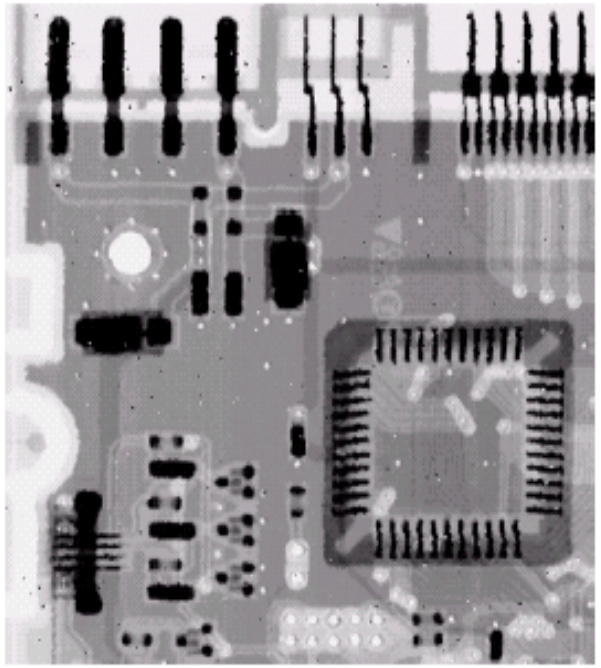


Image after
Median Filter

Max and Min Filter

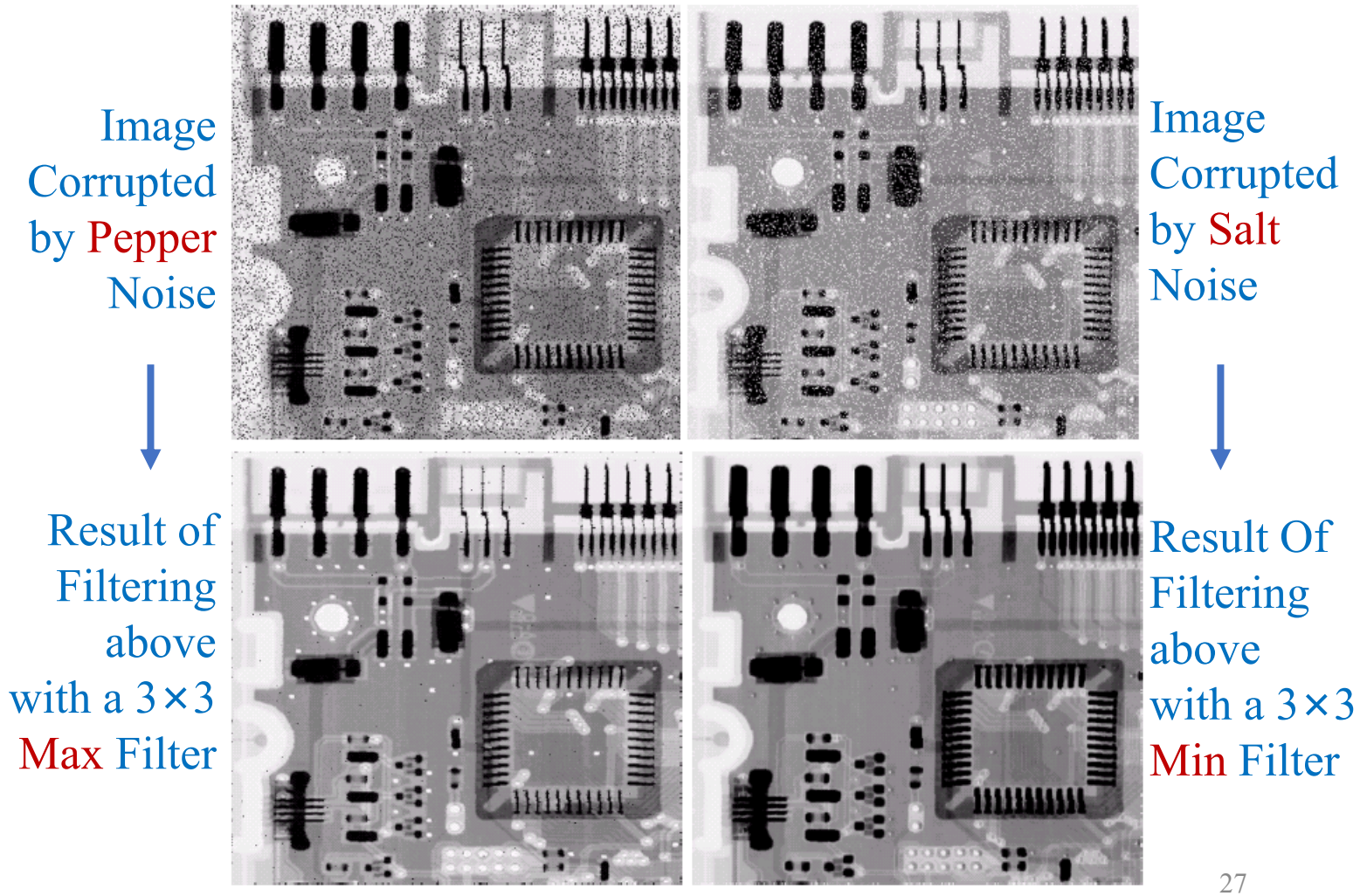
- **Max Filter:** $\hat{f}(x,y) = \max_{(s,t)\in S_{xy}} \{g(s,t)\}$
- **Min Filter:** $\hat{f}(x,y) = \min_{(s,t)\in S_{xy}} \{g(s,t)\}$

where S_{xy} is the set of pixels in the neighbourhood of (x, y).

- The max filter is good for removing pepper noise, while the min filter is good for removing salt noise.

Noise Removal Examples (cont…)

- The max filter removes pepper noise, but also removes some dark pixels from the borders of dark objects.
- Similarly, the min filter may remove some white points around the border of light objects.



Adaptive Filters

- The filters discussed so far are applied to an entire image without any regard for how image characteristics vary from one point to another.
- The behaviour of adaptive filters changes depending on the characteristics of the image inside the filter region.
- Adaptive filters increase the filter complexity.
- We will take a look at the adaptive median filter.

Adaptive Median Filtering

- The median filter performs relatively well on impulse noise as long as the spatial density of the impulse noise is not large (the probability of salt or pepper noise pixels is less than 0.2).
- The adaptive median filter can handle much more spatially dense impulse noise, and also performs some smoothing for non-impulse noise.
- An additional benefit of the adaptive median filter is that it seeks to preserve detail while smoothing non-impulse noise.
- The key insight in the adaptive median filter is that the filter size changes depending on the characteristics of the image.

Adaptive Median Filtering (cont…)

- Remember that filtering looks at each original pixel in turn and generates a new filtered pixel.
- First examine the following notation:
  - z_{min} : minimum gray level in S_{xy}
  - z_{max} : maximum gray level in S_{xy}
  - z_{med} : median of gray levels in S_{xy}
  - z_{xy} : gray level at coordinates (x, y)
  - S_{max} : maximum allowed size of S_{xy}

Adaptive Median Filtering (cont…)

- The key to understanding the algorithm is to remember that the adaptive median filter has three purposes:
  - Remove impulse noise
  - Provide smoothing of other noise
  - Reduce distortion, by not changing these intermediate-level points (not excessive thinning or thickening of object boundaries)

Adaptive Median Filtering (cont…)

- The adaptive median-filtering algorithm uses two processing levels:
  - Level A: $A1 = z_{med} - z_{min}$ — Z_{med} is an impulse?
    $A2 = z_{med} - z_{max}$
    If $A1 > 0$ and $A2 < 0$, Go to Level B
    Else increase the window size
    If window size $\leq S_{max}$ repeat Level A
    Else output z_{med}
  - Level B: $B1 = z_{xy} - z_{min}$ — Z_{xy} is an impulse?
    $B2 = z_{xy} - z_{max}$
    If $B1 > 0$ and $B2 < 0$, output z_{xy}
    Else output z_{med}

Adaptive Filtering Example

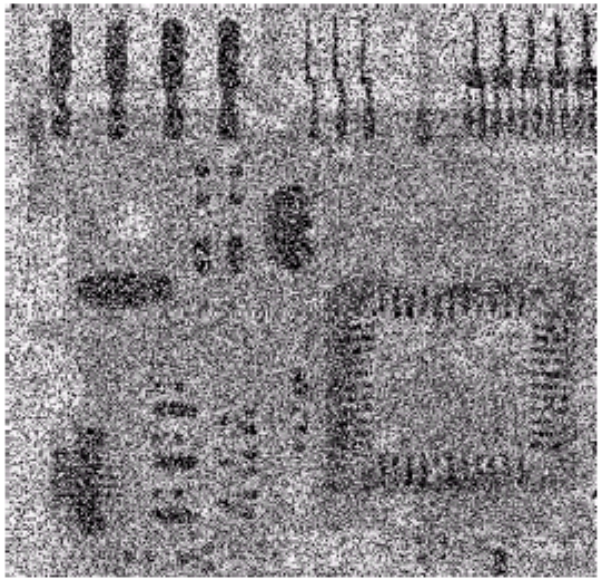
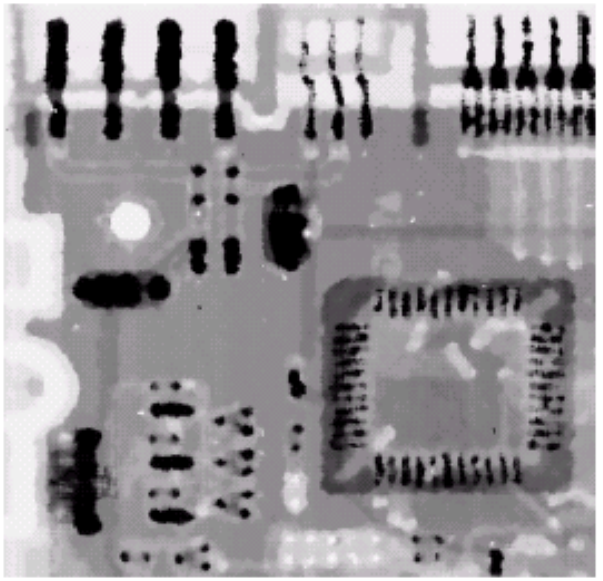
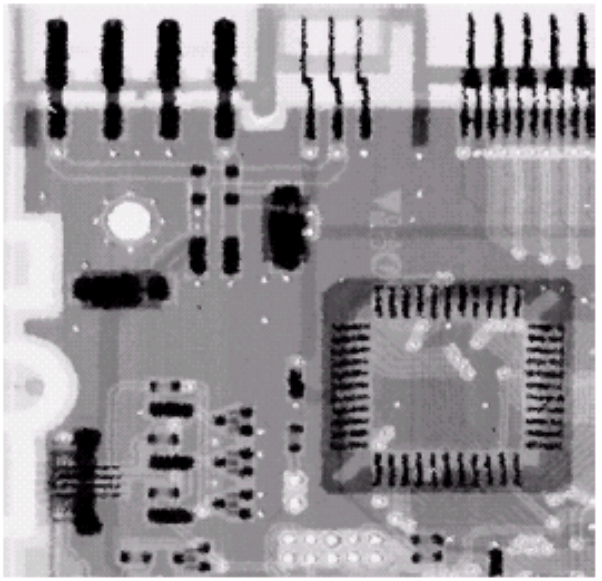


Image corrupted by salt-and-pepper noise with probabilities $P_s = P_p$=0.25



Result of filtering with a 7 × 7 median filter



Result of adaptive median filtering with $S_{max} = 7$

- Choice of maximum allowed window size depends on the application. It should be increased as the density of the impulse increases.
- A reasonable value can be estimated by experiments with various sizes of standard median filter.

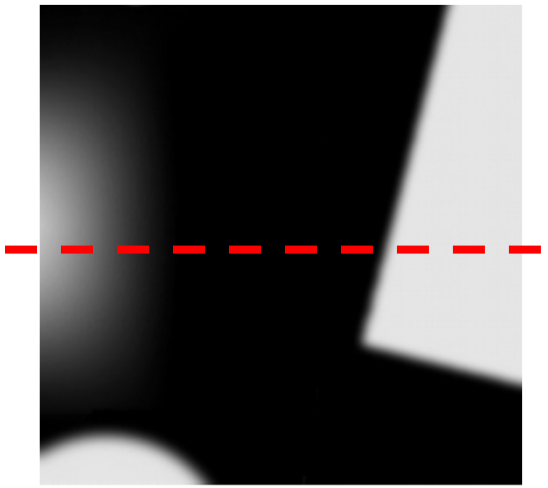
Sharpening Spatial Filters

- Previously we have looked at smoothing filters which remove fine detail.
- *Sharpening spatial filters* seek to highlight fine detail:
  - Remove blurring from images
  - Highlight edges
- Averaging is analogous to integration and causes blurring, so differentiation is expected to have opposite results and can sharpen an image.
  - smoothing ~ integration
  - sharpening ~ differentiation

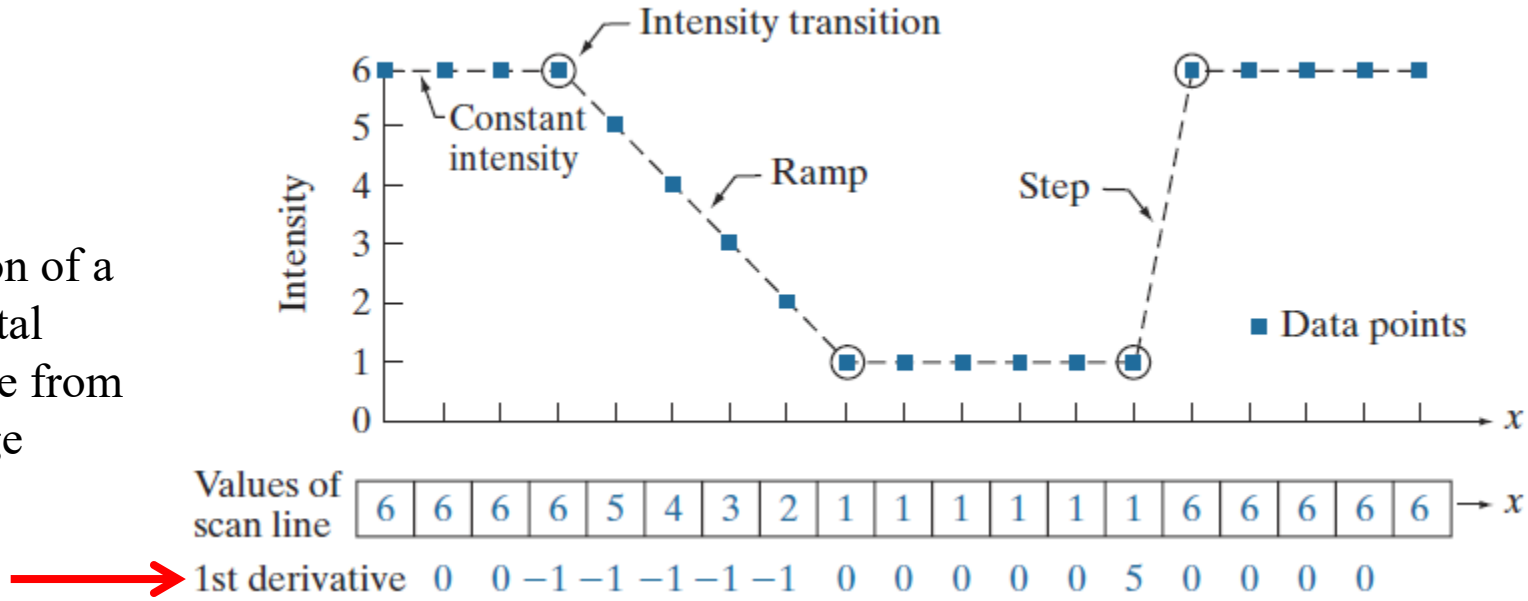
1st Derivative

- Differentiation measures the *rate of change* of a function.
- Let's consider a simple one dimensional example. The formula for the 1st derivative of a function is as follows:

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



A section of a horizontal scan line from an image

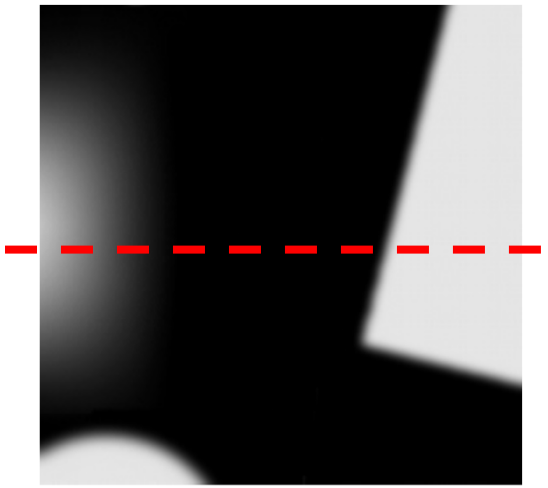


2nd Derivative

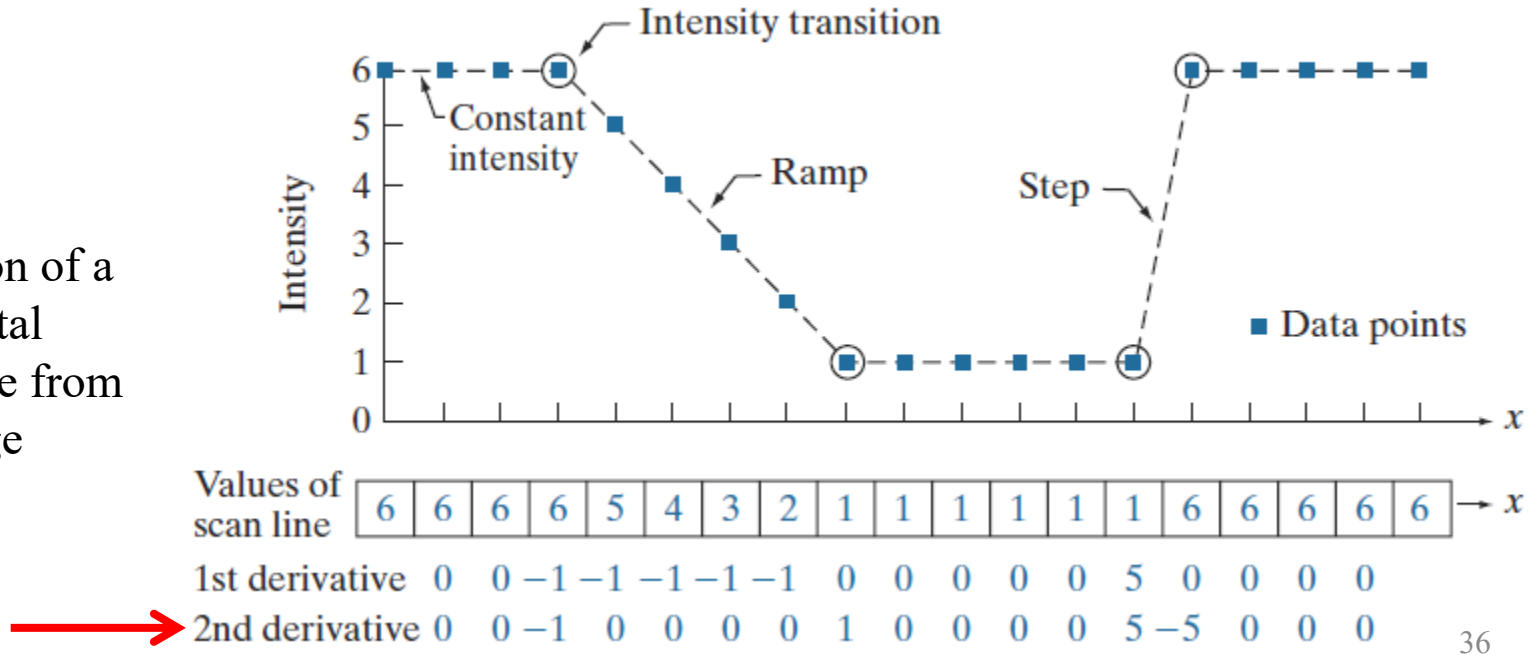
- The formula for the 2nd derivative of a function is as follows:

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

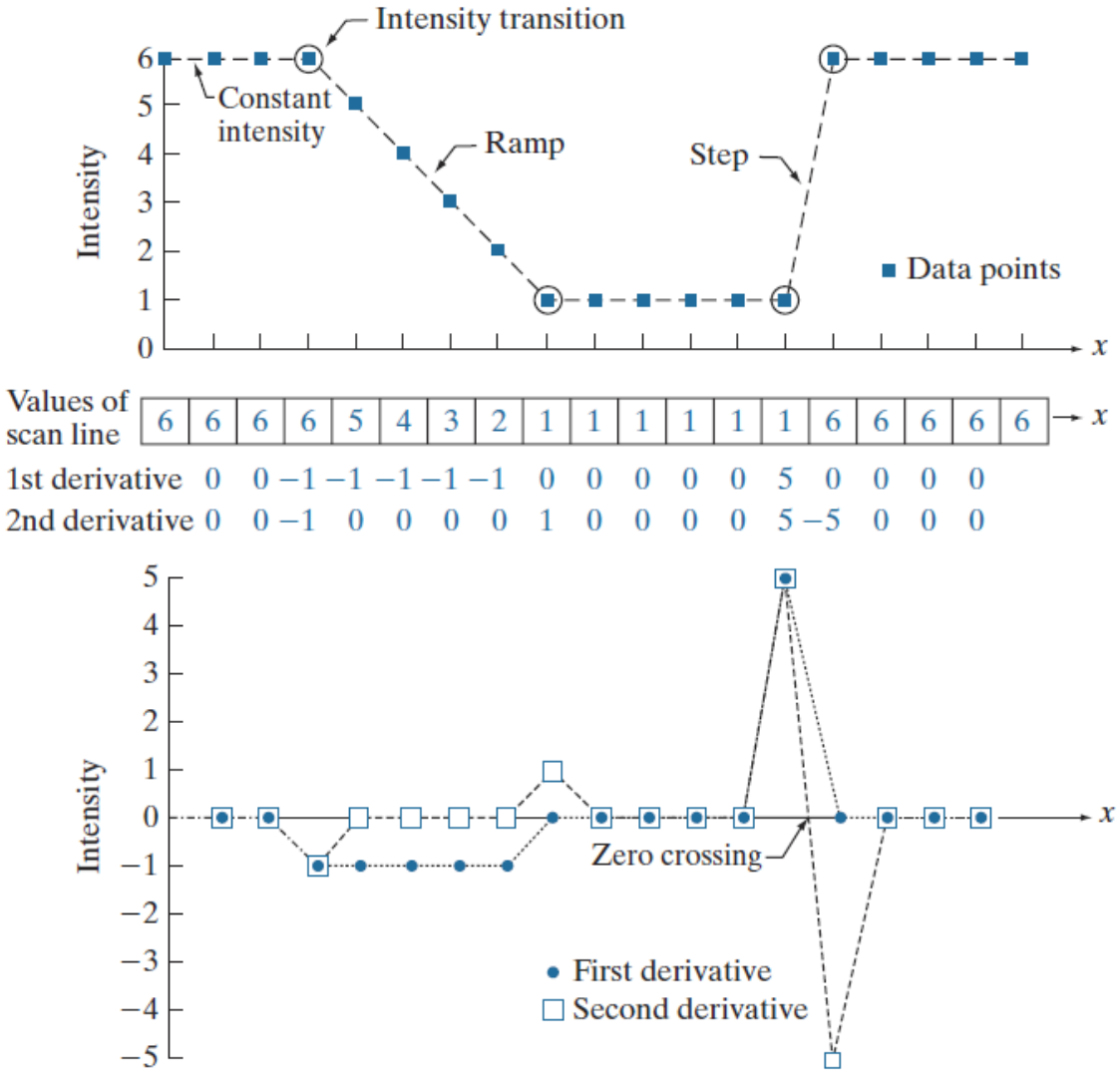
- Simply take into account the values both before and after the current value.



A section of a horizontal scan line from an image



1st & 2nd Derivatives



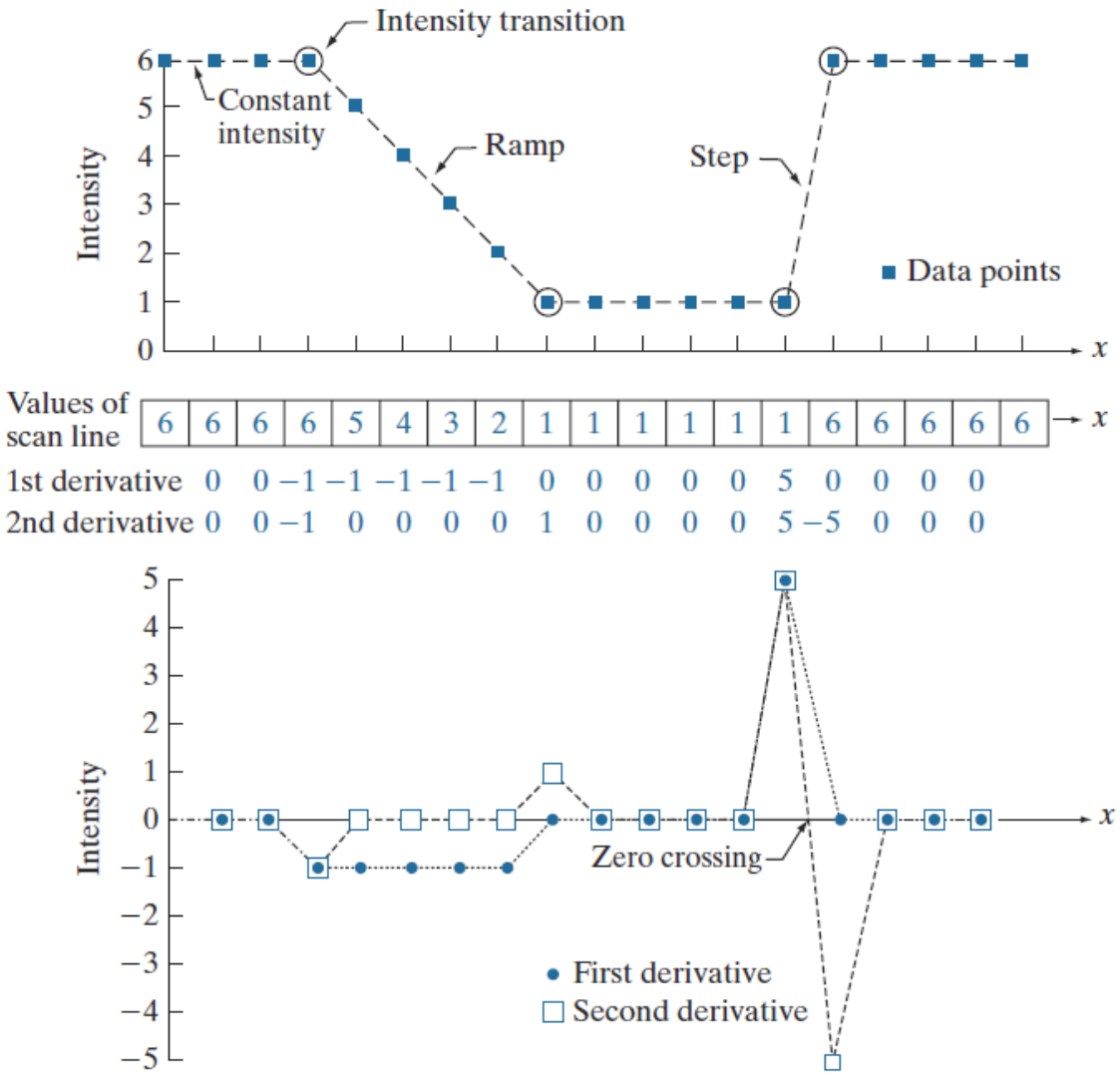
First derivative:

- 0 in constant gray segments
- Non-zero at the onset of steps or ramps
- Non-zero along ramps

Second derivative:

- 0 in constant gray segments.
- Non-zero at the onset and end of steps or ramps.
- 0 along ramps of constant slope.

1st & 2nd Derivatives



By comparing the 1st and 2nd derivatives we can conclude the following:

- 1st order derivatives generally produce thicker edges.
- 2nd order derivatives have a stronger response to fine detail, e.g. thin lines, and isolated points.
- 1st order derivatives have stronger response to gray level step.
- 2nd order derivatives produce a double response at step changes in gray level.

Using 2nd Derivatives for Image Enhancem

- The 2nd derivative is more useful for image enhancement than the 1st derivative
  - Stronger response to fine detail
  - Simpler implementation
- The first 2nd derivative we will look at is the *Laplacian*
  - Isotropic
  - One of the simplest sharpening filters

The Laplacian

- The Laplacian is defined as follows:

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

where the partial 2nd order derivative in the *x* direction is defined as follows:

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

and in the *y* direction as follows:

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

The Laplacian (cont…)

- So, the Laplacian can be given as follows:

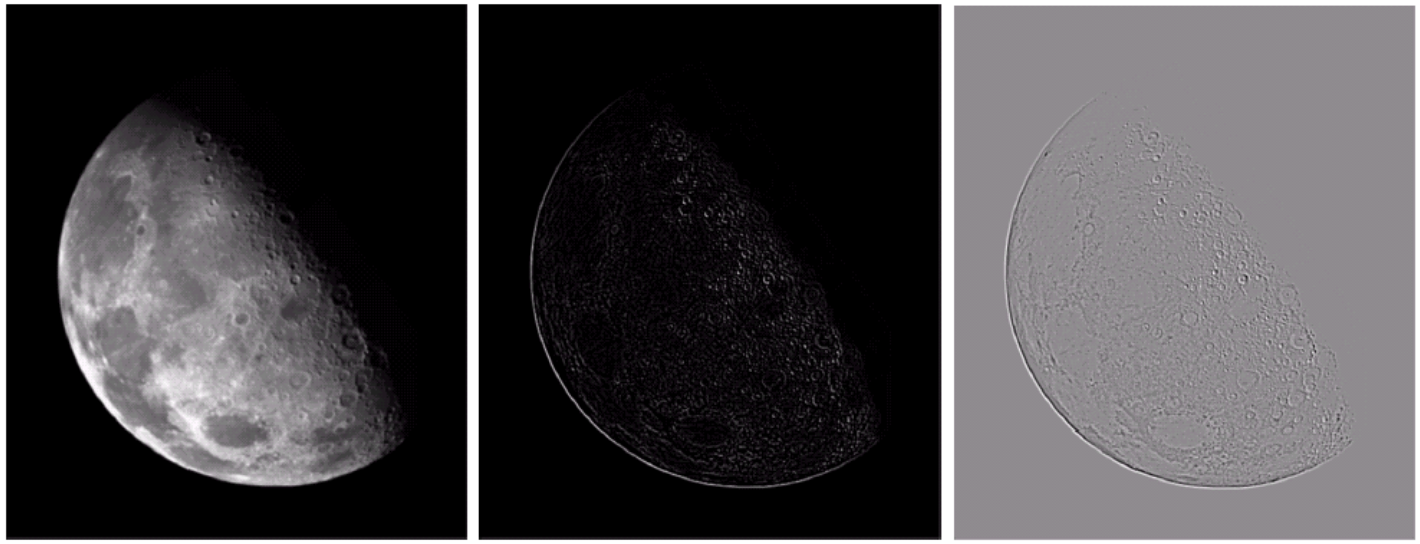
$$\nabla^2 f = [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)] - 4f(x,y)$$

- We can easily build a filter based on this

0	1	0
1	−4	1
0	1	0

The Laplacian (cont…)

- Applying the Laplacian to an image we get a new image that highlights edges and other discontinuities



Original Image

Laplacian Filtered Image

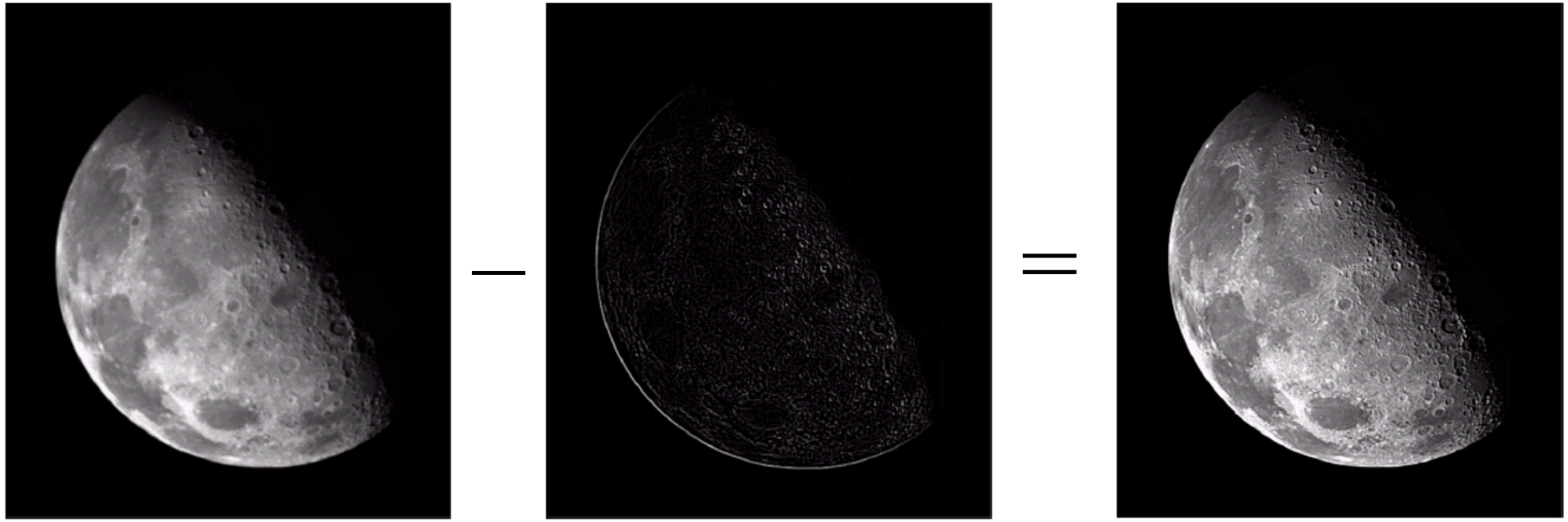
Laplacian Filtered Image
Scaled for Display

But That is Not Very Enhanced!

- The result of a Laplacian filtering is not an enhanced image.
- We have to do more work in order to get our final image.
- Subtract the Laplacian result from the original image to generate our final sharpened enhanced image. $g(x,y) = f(x,y) - \nabla^2 f$

Laplacian
Filtered Image
Scaled for Display

Laplacian Image Enhancement



Original Image — Laplacian Filtered Image = Sharpened Image

- In the final sharpened image, edges and fine detail are much more obvious.

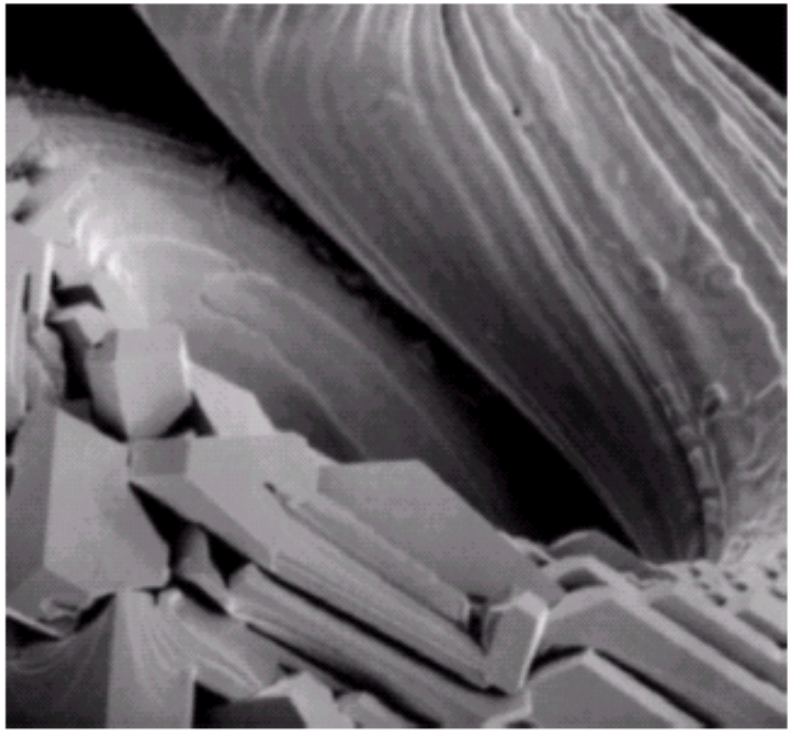
Simplified Image Enhancement

- The entire enhancement can be combined into a single filtering operation:

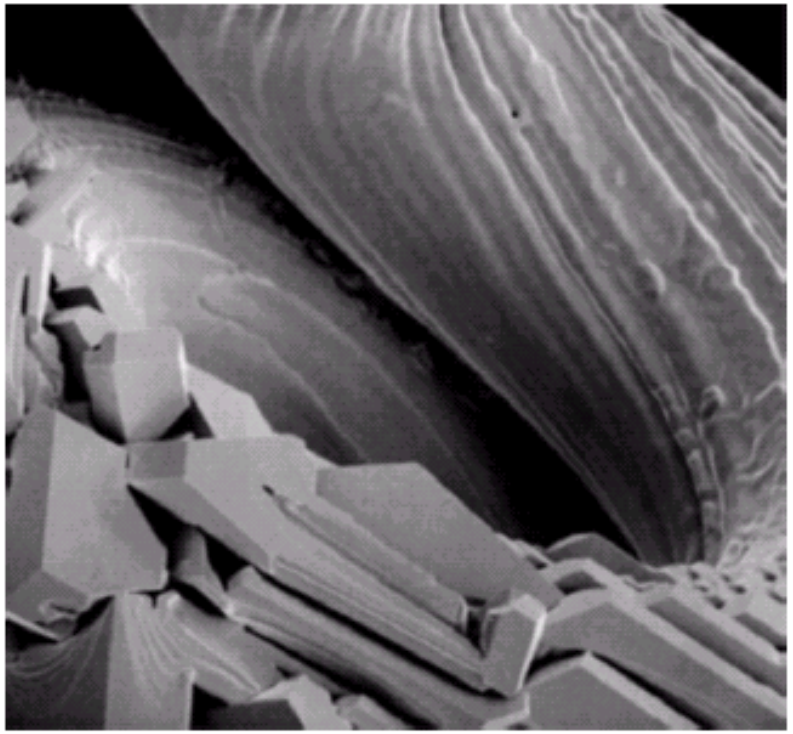
$$
\begin{aligned}
g(x,y) &= f(x,y) - \nabla^2 f \\
&= f(x,y) - [f(x+1,y) + f(x-1,y) \\
&\qquad + f(x,y+1) + f(x,y-1) \\
&\qquad - 4f(x,y)] \\
&= 5f(x,y) - f(x+1,y) - f(x-1,y) \\
&\qquad - f(x,y+1) - f(x,y-1)
\end{aligned}
$$

Simplified Image Enhancement (cont…)

- This gives us a new filter which does the whole job for us in one step.



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



Variants on the Simple Laplacian

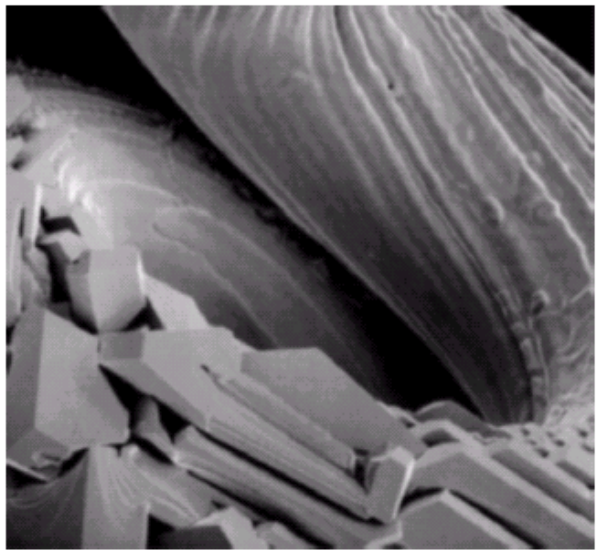
- There are lots of slightly different versions of the Laplacian that can be used:

0	1	0
1	–4	1
0	1	0

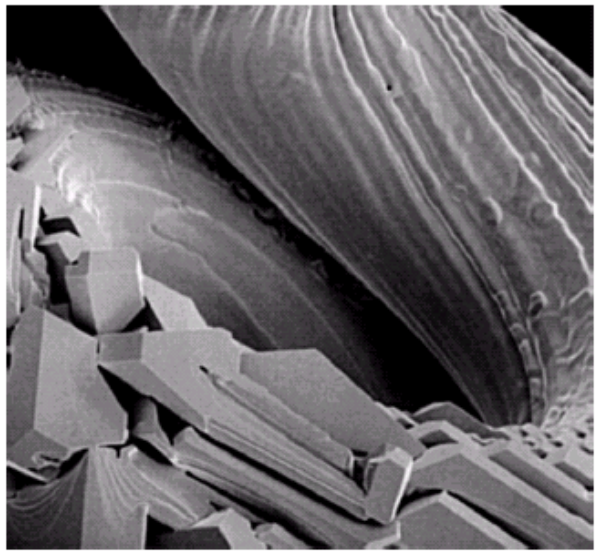
Simple Laplacian

1	1	1
1	– 8	1
1	1	1

Variant of Laplacian



–1	–1	–1
– 1	9	–1
– 1	– 1	–1



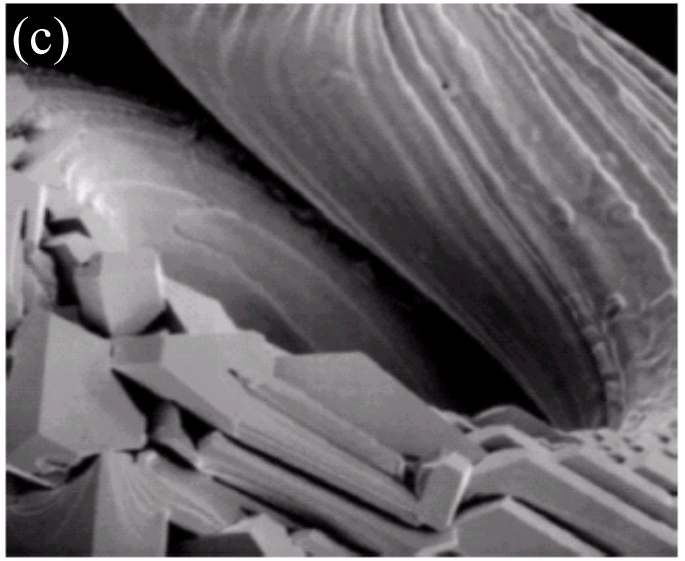
Variants on the Simple Laplacian

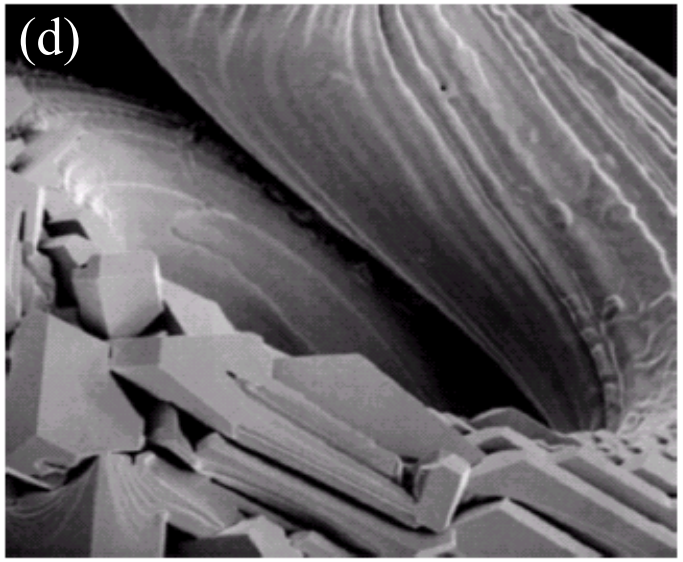
(a)

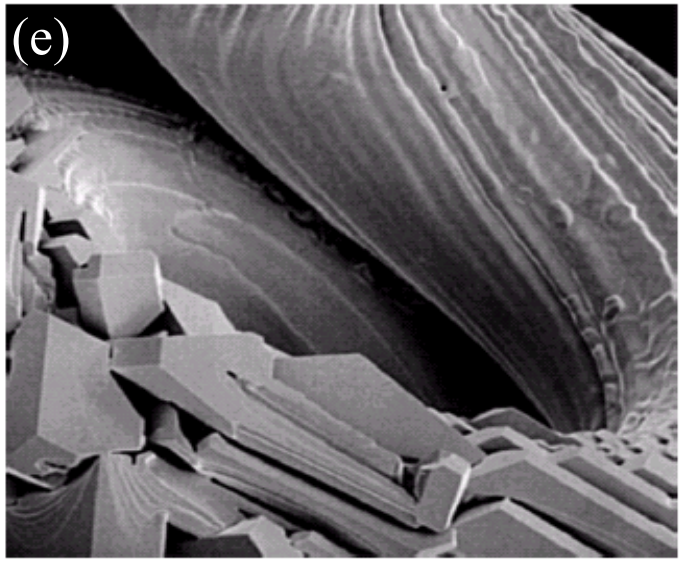
0	−1	0
−1	5	−1
0	−1	0

(b)

−1	−1	−1
−1	9	−1
−1	−1	−1







(a) Composite Laplacian mask.

(b) A second composite mask.

(c) Scanning electron microscope image.

(d) Results of filtering with the masks in (a).

(e) Results of filtering with the masks in (b).

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

Variants on the Simple Laplacian

- Two definitions of Laplacian: one is the negative of the other
- Accordingly, to recover background features:

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

- *I*: if the center of the mask is negative
- *II*: if the center of the mask is positive

I

0	1	0
1	−4	1
0	1	0

II

0	−1	0
−1	4	−1
0	−1	0

High Boost Filtering

- High boost filtering:

$$g_{hb}(x,y) = \begin{cases} Af(x,y) - \nabla^2 f(x,y) & (I) \\ Af(x,y) + \nabla^2 f(x,y) & (II) \end{cases} \qquad A \geq 1$$

  - *I*: if the center of the mask is negative
  - *II*: if the center of the mask is positive

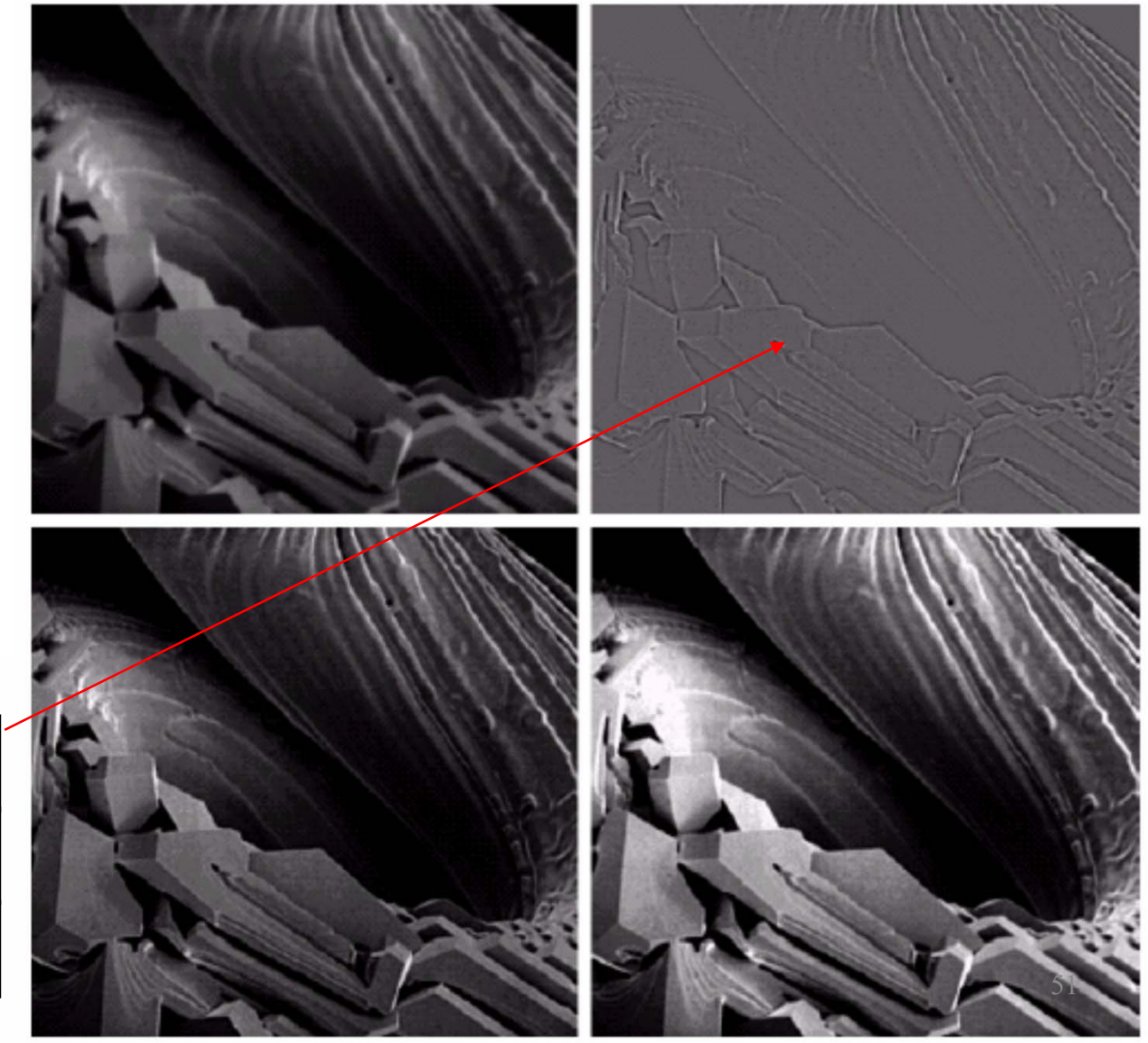
- When *A*=1, it is the standard Laplacian filtering, as the value of *A* increase past 1, the contribution of the sharpening process becomes less and less important.

High Boost Filtering

- One of the principal applications of boost filtering is when the input image is darker than desired.
- By varying the boost coefficient, it generally is possible to obtain an overall increase in average gray level of the image.

(a) Same as the figure in Slide 49, but darker.
(b) Laplacian of (a) computed with the mask using A=0.
(c) Laplacian enhanced image using the mask with A=1.
(d) Same as (c), but using A =1.7.

1	1	1
1	-8	1
1	1	1



1st Derivative Filtering

- Implementing 1st derivative filters is difficult in practice.
- For a function $f(x, y)$ the gradient of f at coordinates (x, y) is given as the column vector:

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

1st Derivative Filtering (cont…)

- The magnitude of this vector is given by:

$$M(x,y) = \|\nabla f\| = mag(\nabla f)$$
$$= \sqrt{g_x^2 + g_y^2}$$

- For practical reasons this can be approximated as:

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

1st Derivative Filtering (cont…)

- The simplest 1st derivative filtering:

$$g_x = (z_8 - z_5) \qquad \frac{\partial f}{\partial x} = f(x+1) - f(x)$$

$$g_y = (z_6 - z_5) \qquad \frac{\partial f}{\partial y} = f(y+1) - f(y)$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- Approximation:

$$M(x,y) \approx |z_8 - z_5| + |z_6 - z_5|$$

1st Derivative Filtering (cont…)

- Another definition (by Roberts [1965]):

$$g_x = (z_9 - z_5)$$
$$g_y = (z_8 - z_6)$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- Approximation (Roberts cross-gradient operators):

$$M(x, y) \approx |z_9 - z_5| + |z_8 - z_6|$$

−1	0
0	1

0	−1
1	0

1st Derivative Filtering (cont…)

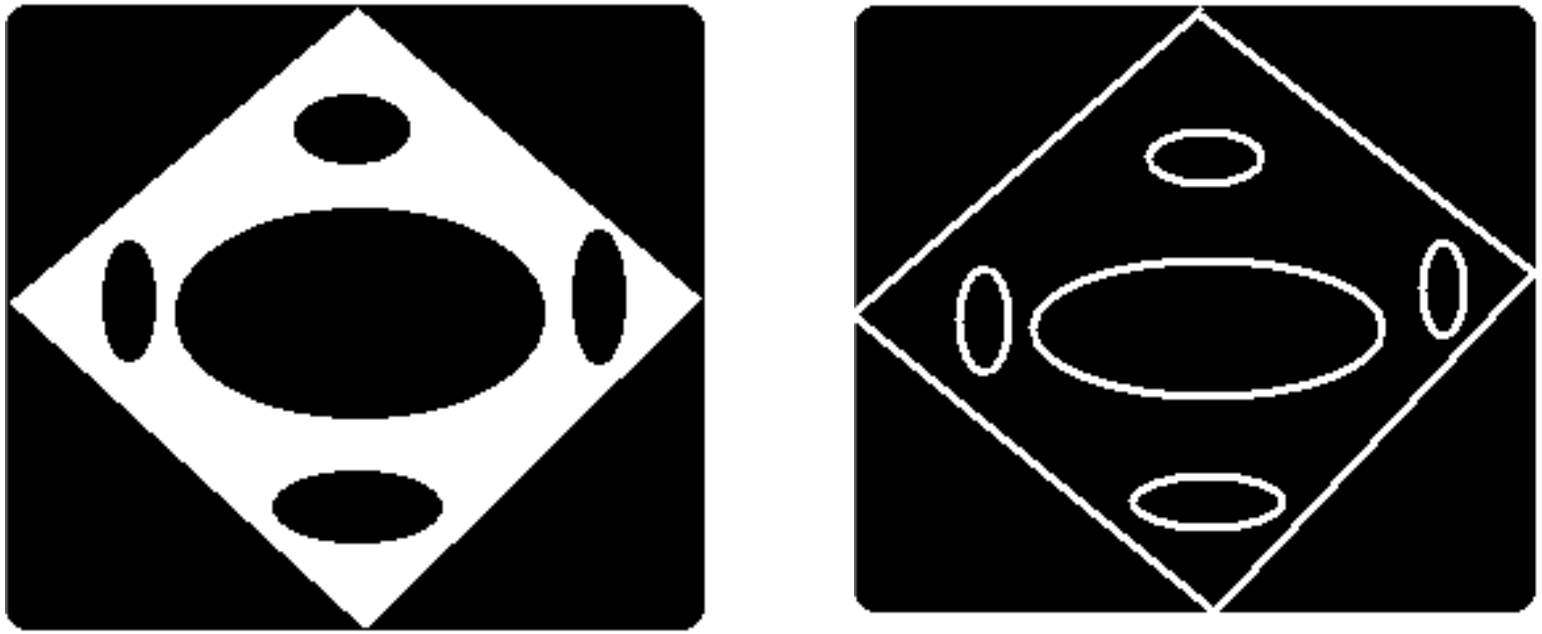


Image sharpening by gradient

1st Derivative Filtering (cont…)

- **Sobel operators:**

$$M(x,y) \approx \left|(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)\right| + \left|(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)\right|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel Example

- Sobel filters are typically used for edge detection

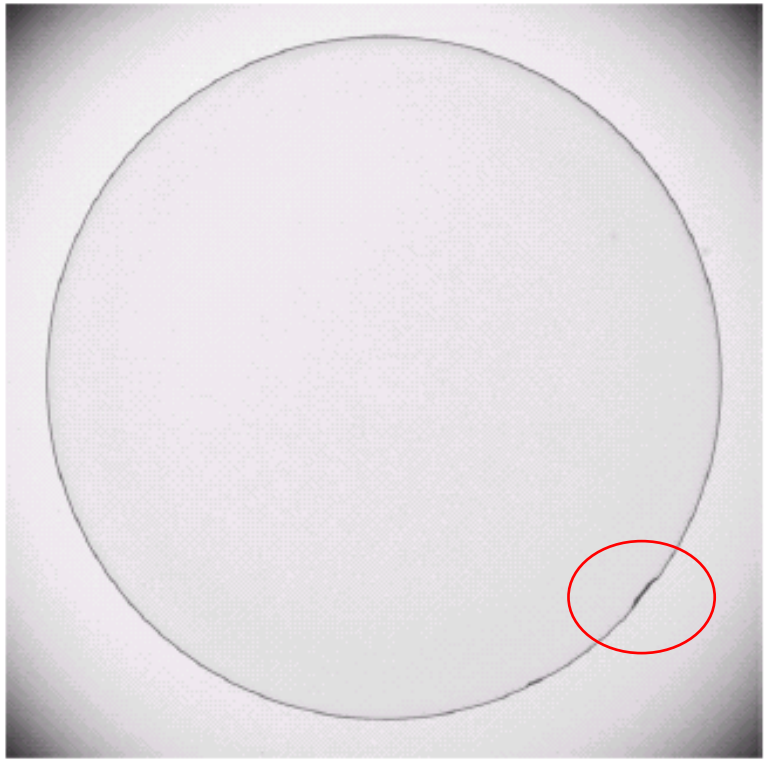
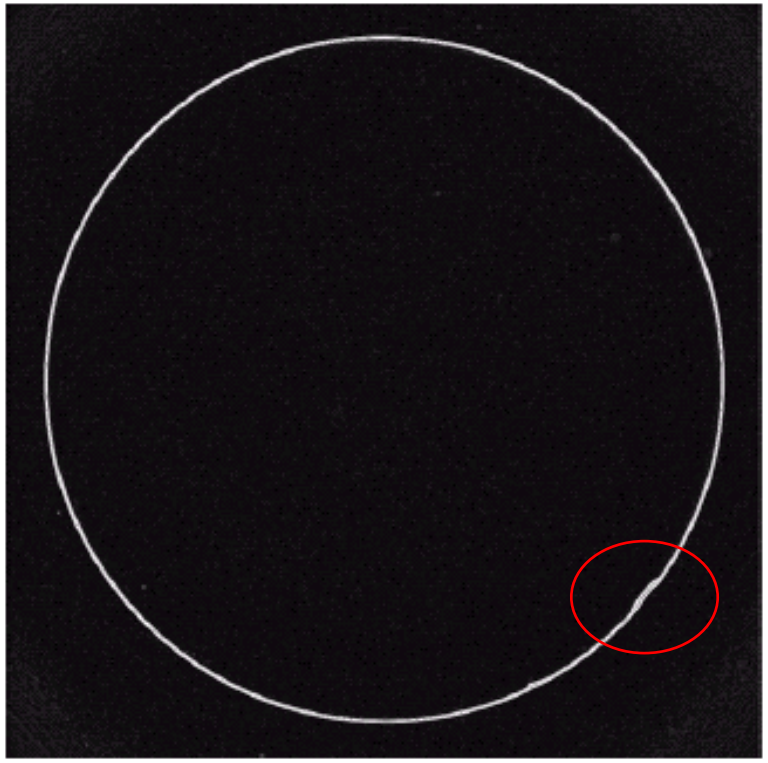


Image of a contact lens
(note defects on the boundary
at 4 and 5 o'clock)



Sobel gradient

Combining Spatial Enhancement Methods

- Successful image enhancement is typically *NOT* achieved using a single operation.
- We can combine a range of techniques in order to achieve a final result.
- This example will focus on enhancing the bone scan to the right.

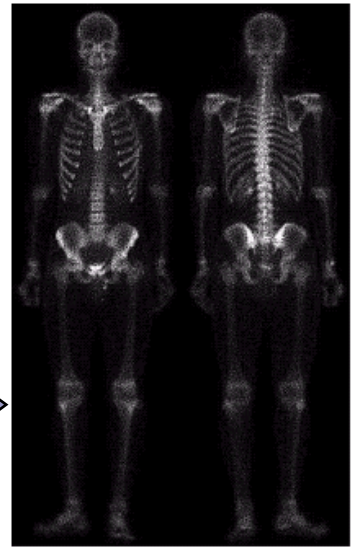
Combining Spatial Enhancement Methods (cont…)

(a)

Laplacian filter of bone scan (a)

(b)

Sharpened version of bone scan achieved by subtracting (a) and (b)



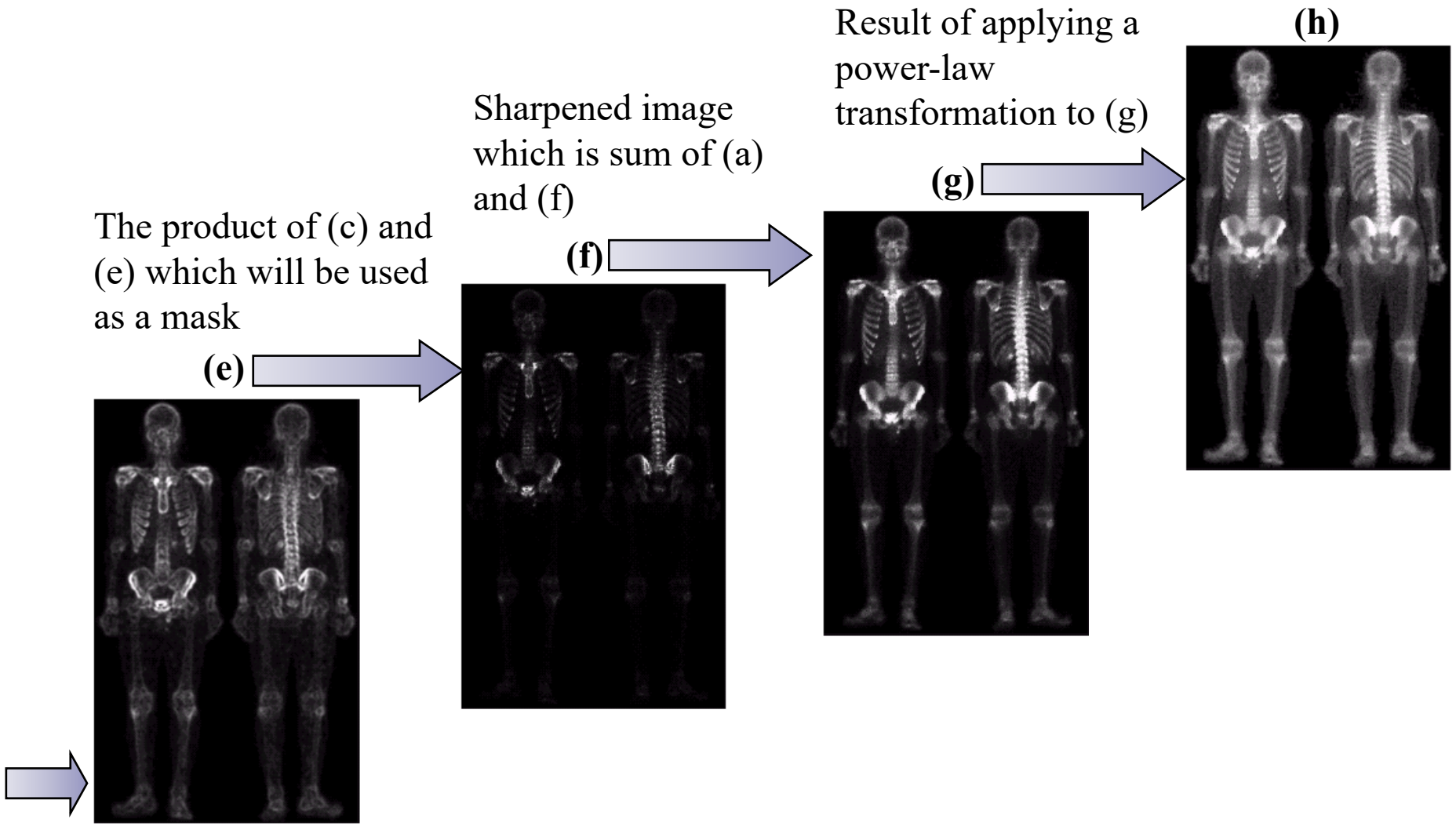
(c)

Image (d) smoothed with a 5×5 averaging filter

(d)

Sobel filter of bone scan (a)

Combining Spatial Enhancement Methods (cont…)



Combining Spatial Enhancement Methods (cont…)

- Compare the original and final images

Summary

- In this lecture we have learnt:
  - Basic of spatial filtering
  - Smoothing spatial filters
    - Smoothing linear filters
    - Order-statistics filters
  - Sharpening spatial filters
    - 1^{st} derivative filters
    - 2^{nd} derivative filters