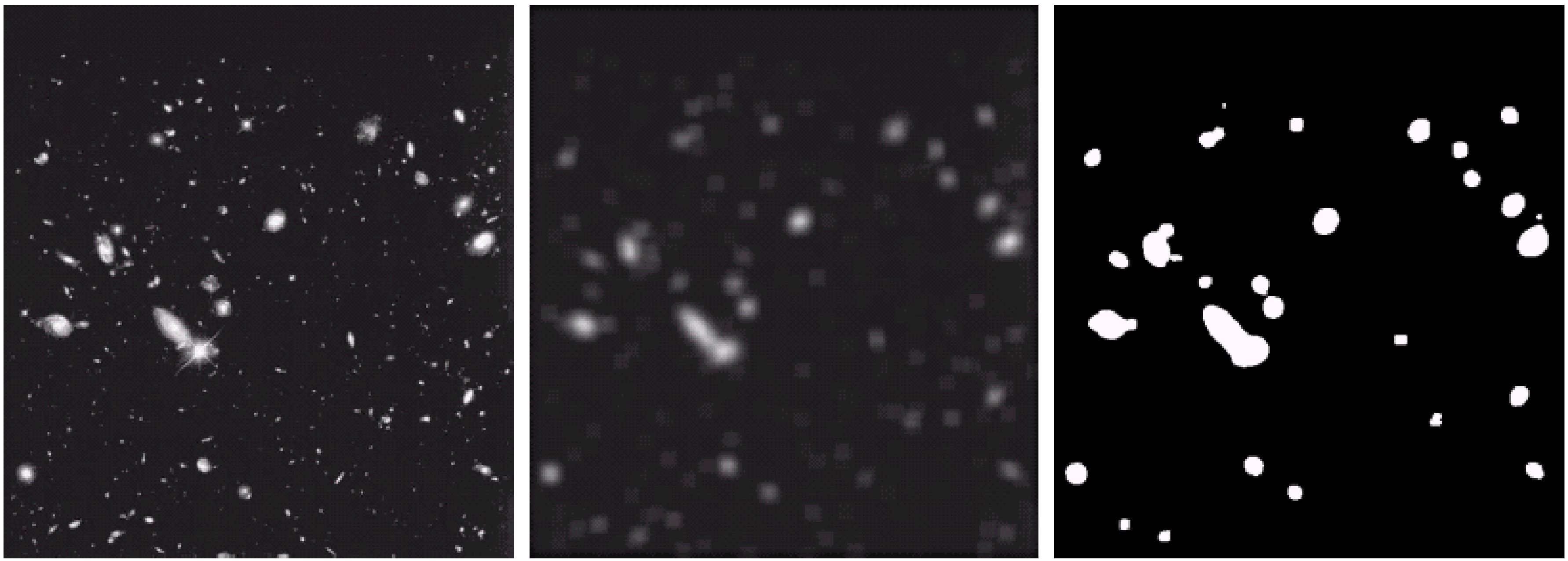
Digital Image Processing

Lecture # 4
Spatial Filtering

Spatial Filtering

Spatial Filtering



a b c

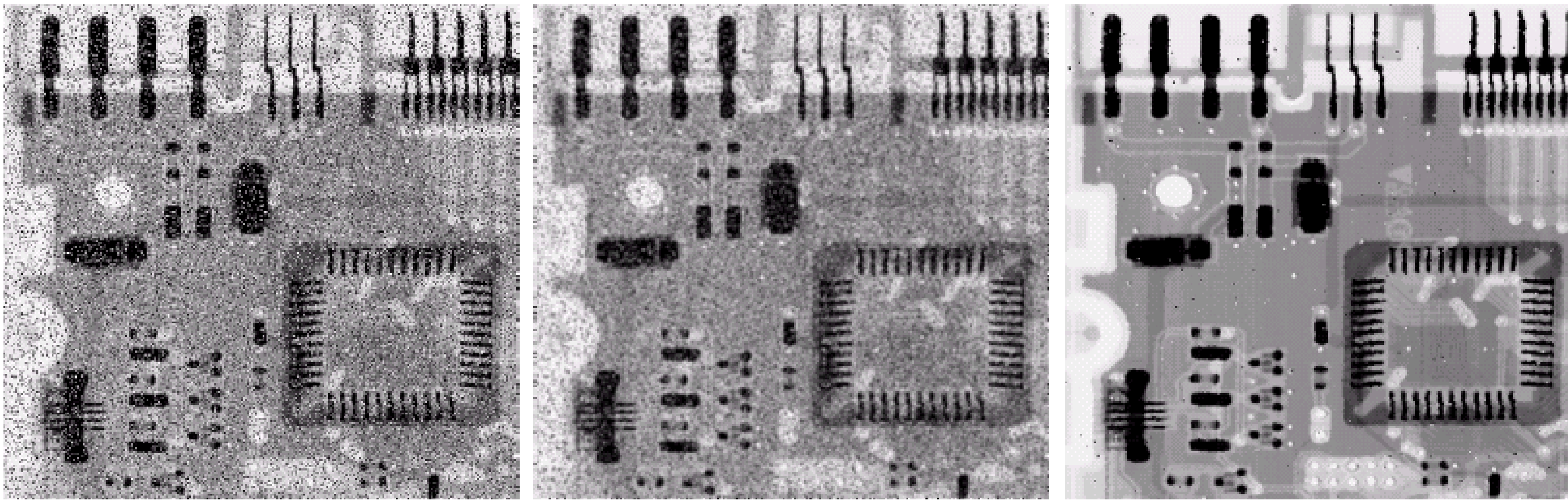
FIGURE 3.36 (a) Image from the Hubble Space Telescope. (b) Image processed by a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

Spatial Filtering

a b c

FIGURE 4.50 (a) Original image (784 × 732 pixels). (b) Result of filtering using a GLPF with $D_0 = 100$. (c) Result of filtering using a GLPF with $D_0 = 80$. Note the reduction in fine skin lines in the magnified sections in (b) and (c).

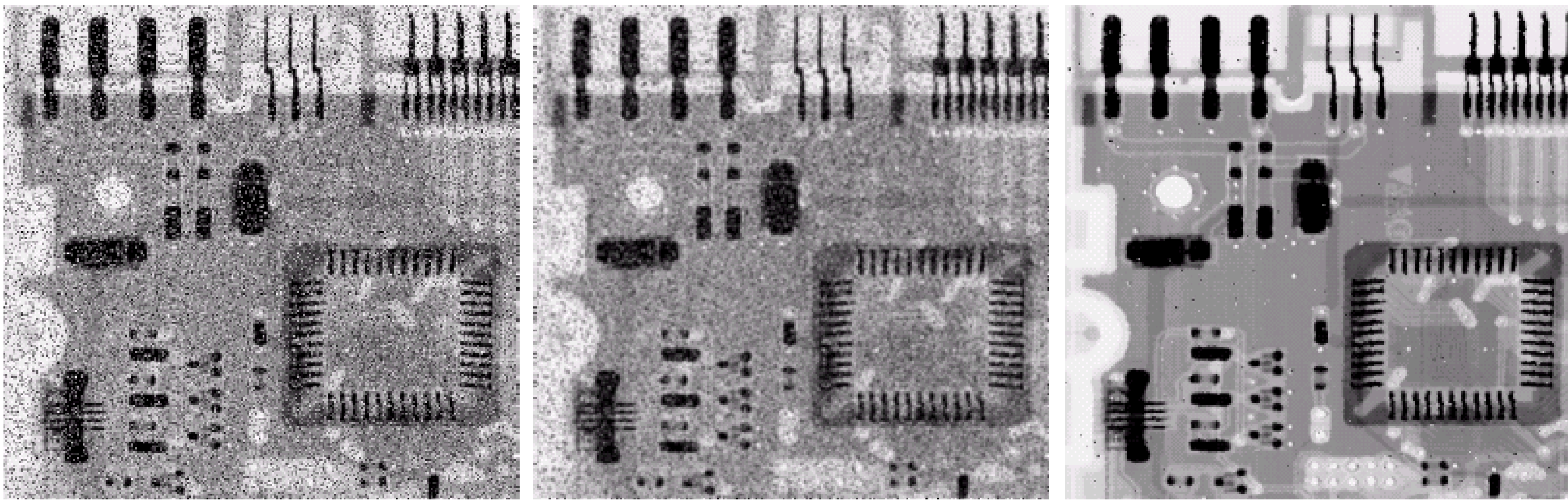
Spatial Filtering



a b c

FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Spatial Filtering



a b c

FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3 × 3 averaging mask. (c) Noise reduction with a 3 × 3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Spatial Filtering

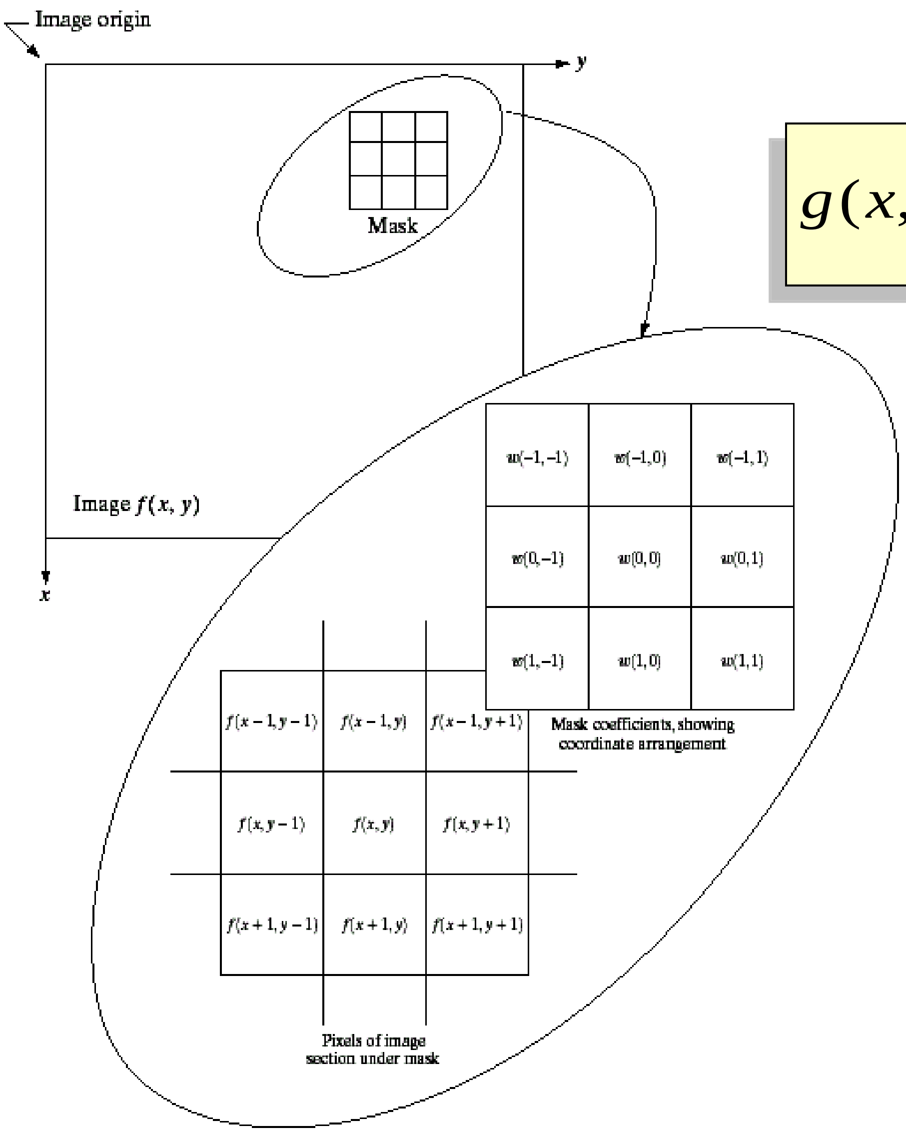
a b

FIGURE 4.58 (a) Original, blurry image. (b) Image enhanced using the Laplacian in the frequency domain. Compare with Fig. 3.38(e).

Spatial Filtering: Basics

- The output intensity value at *(x,y)* depends not only on the input intensity value at *(x,y)* but also on the specified number of neighboring intensity values around *(x,y)*
- Spatial masks (also called window, filter, kernel, template) are used and **convolved** over the entire image for local enhancement (spatial filtering)
- The size of the masks determines the number of neighboring pixels which influence the output value at *(x,y)*
- The values (coefficients) of the mask determine the nature and properties of enhancing technique

Spatial Filtering: Basics



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

$$\text{where a} = \frac{m-1}{2},\quad \text{b} = \frac{n-1}{2}$$

$$x=0,1,2,....,M-1,\ y=0,1,2,...,N-1$$

Filtering can be given in equation form as shown above

Spatial Filtering: Basics

- Given the 3×3 mask with coefficients: $w_1, w_2, \ldots, w_9$
- The mask cover the pixels with gray levels: $z_1, z_2, \ldots, z_9$

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

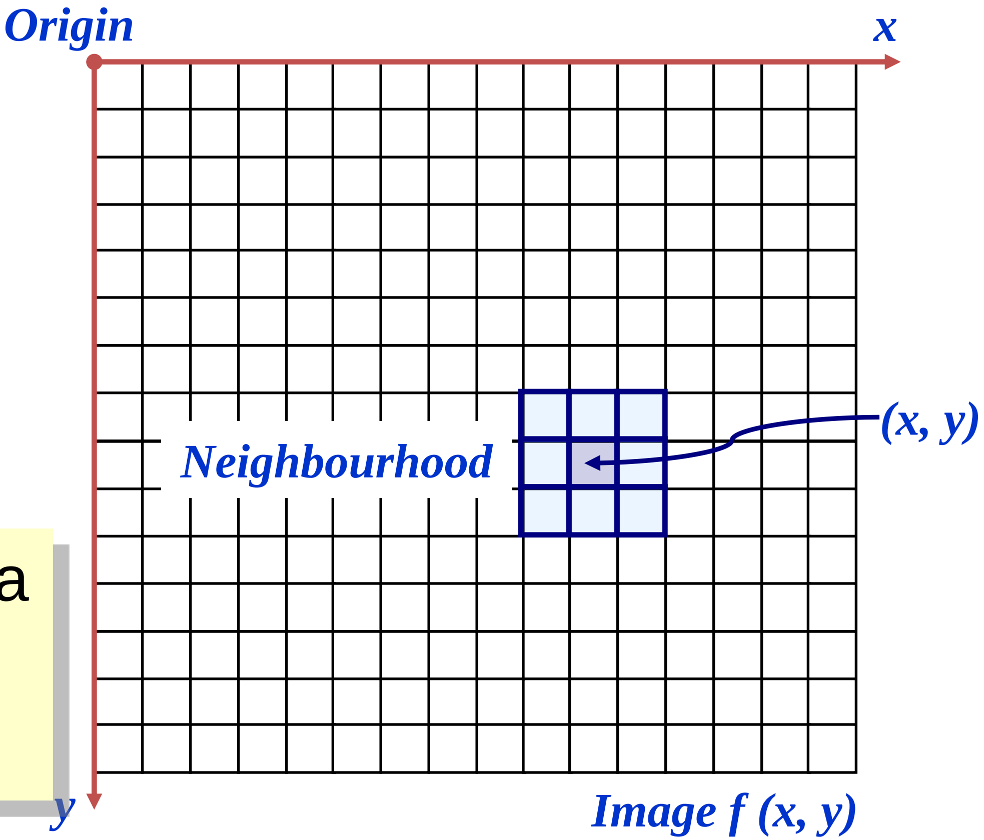
$$z \longleftarrow z_1 w_1 + z_2 w_2 + z_3 w_3 + \cdots + z_9 w_9 = \sum_{i=1}^{9} z_i w_i$$

- z gives the output intensity value for the processed image (to be stored in a new array) at the location of z_5 in the input image

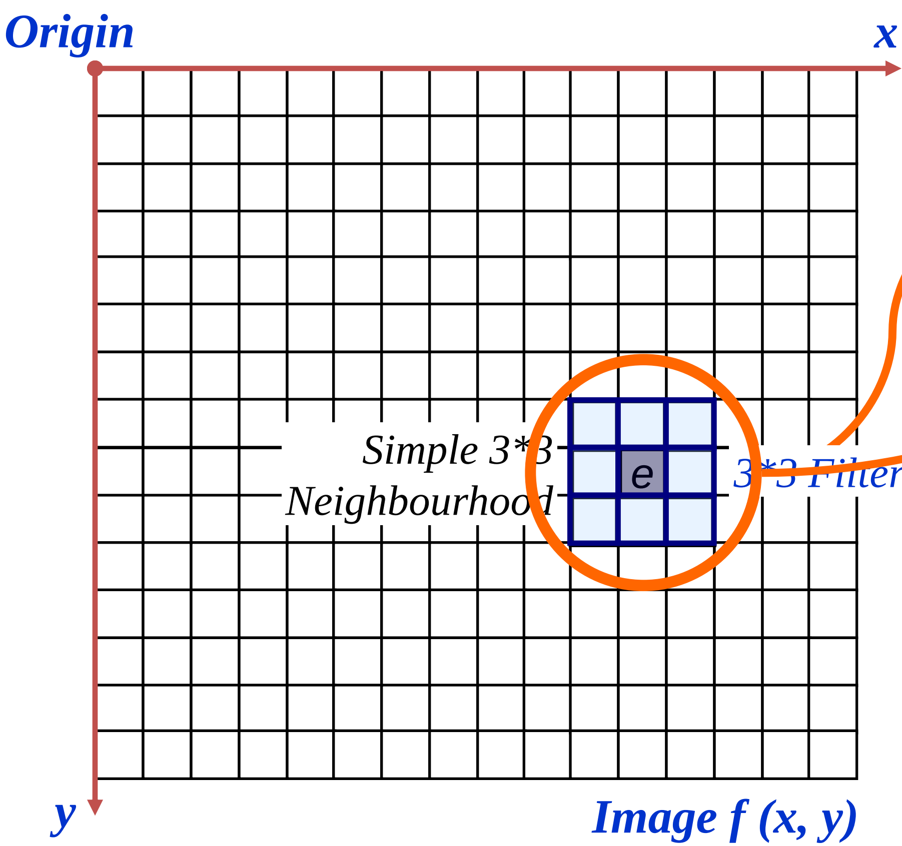
Spatial Filtering: Basics

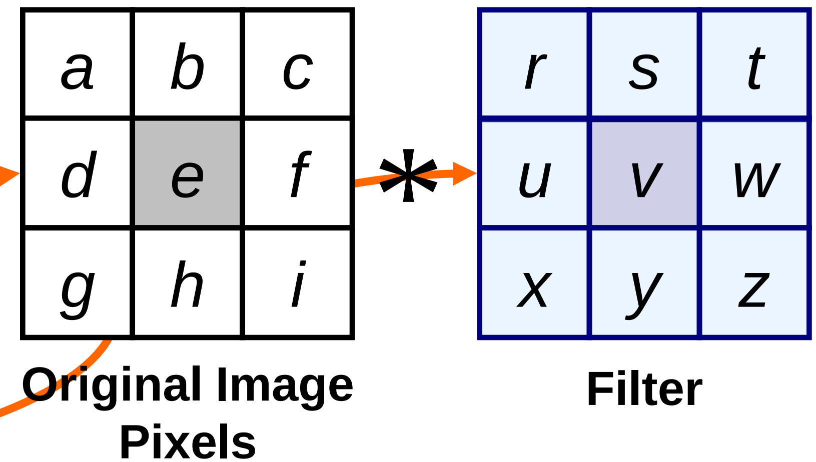
Neighbourhood operations: Operate on a larger neighbourhood of pixels than point operations

Neighbourhoods are mostly a rectangle around a central pixel



Spatial Filtering: Basics





$$e_{processed} = v*e + r*a + s*b + t*c + u*d + w*f + x*g + y*h + z*i$$

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

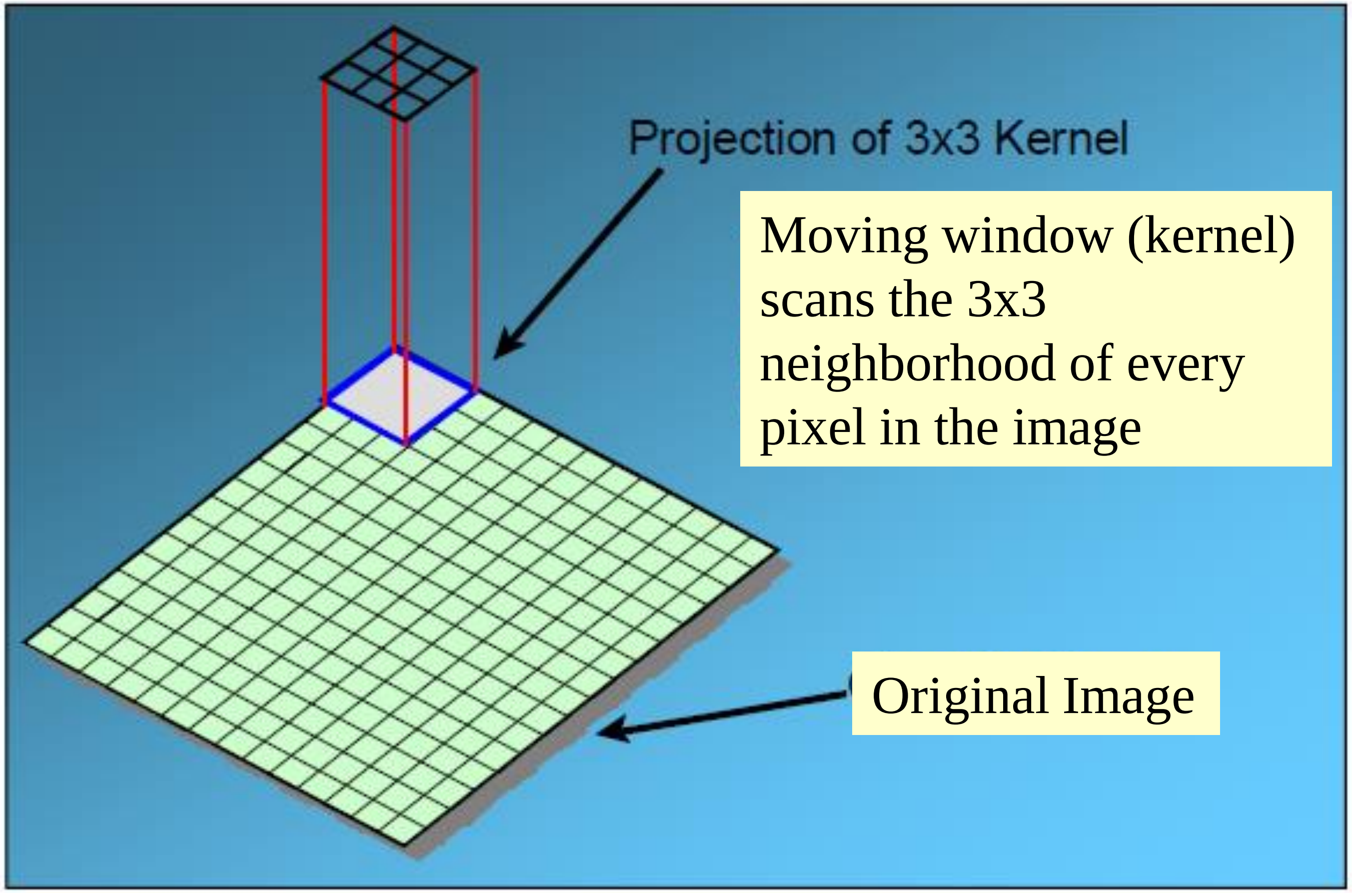
Spatial Filtering: Basics

Original Image

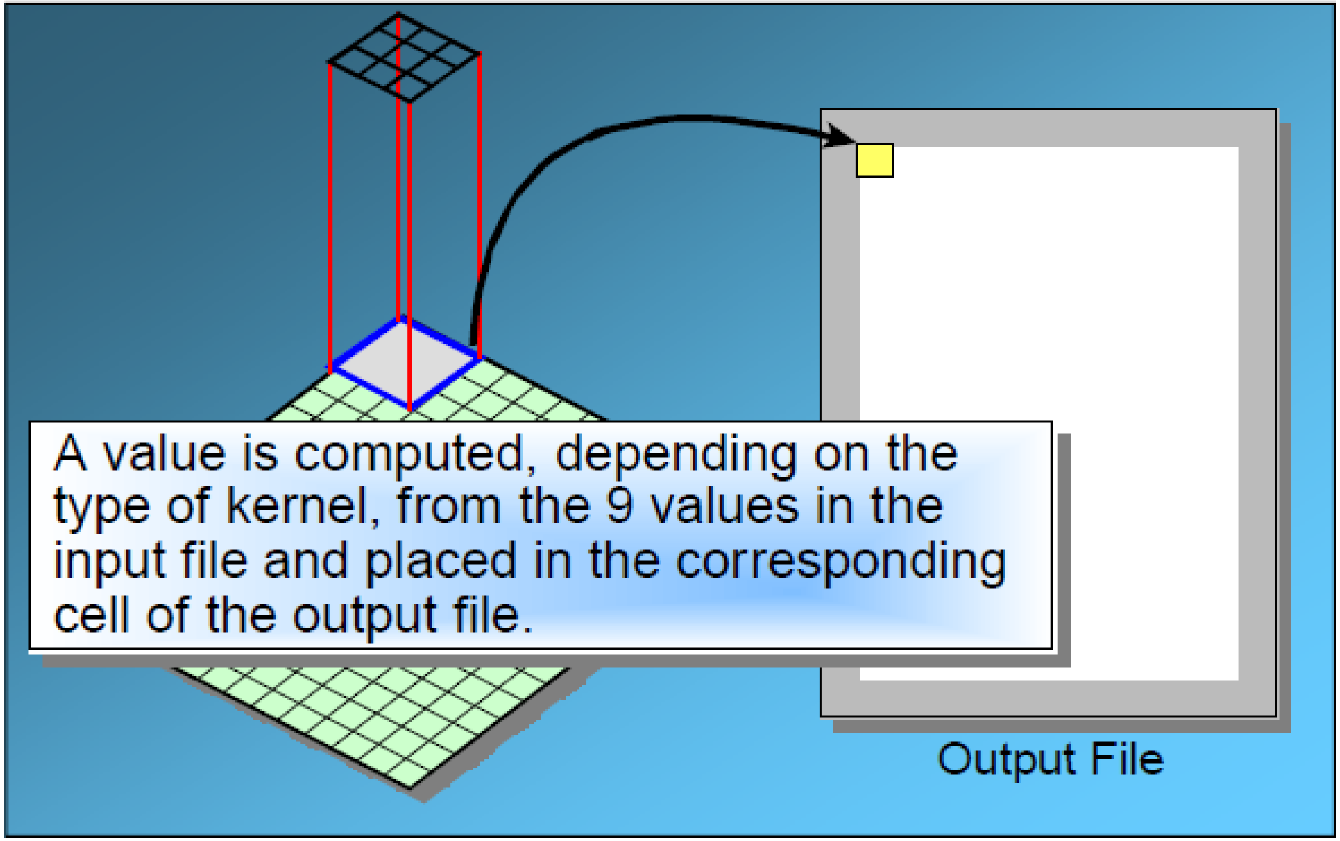
123	127	128	119	115	130
140	145	148	153	167	172
133	154	183	192	194	191
194	199	207	210	198	195
164	170	175	162	173	151

Enhanced Image

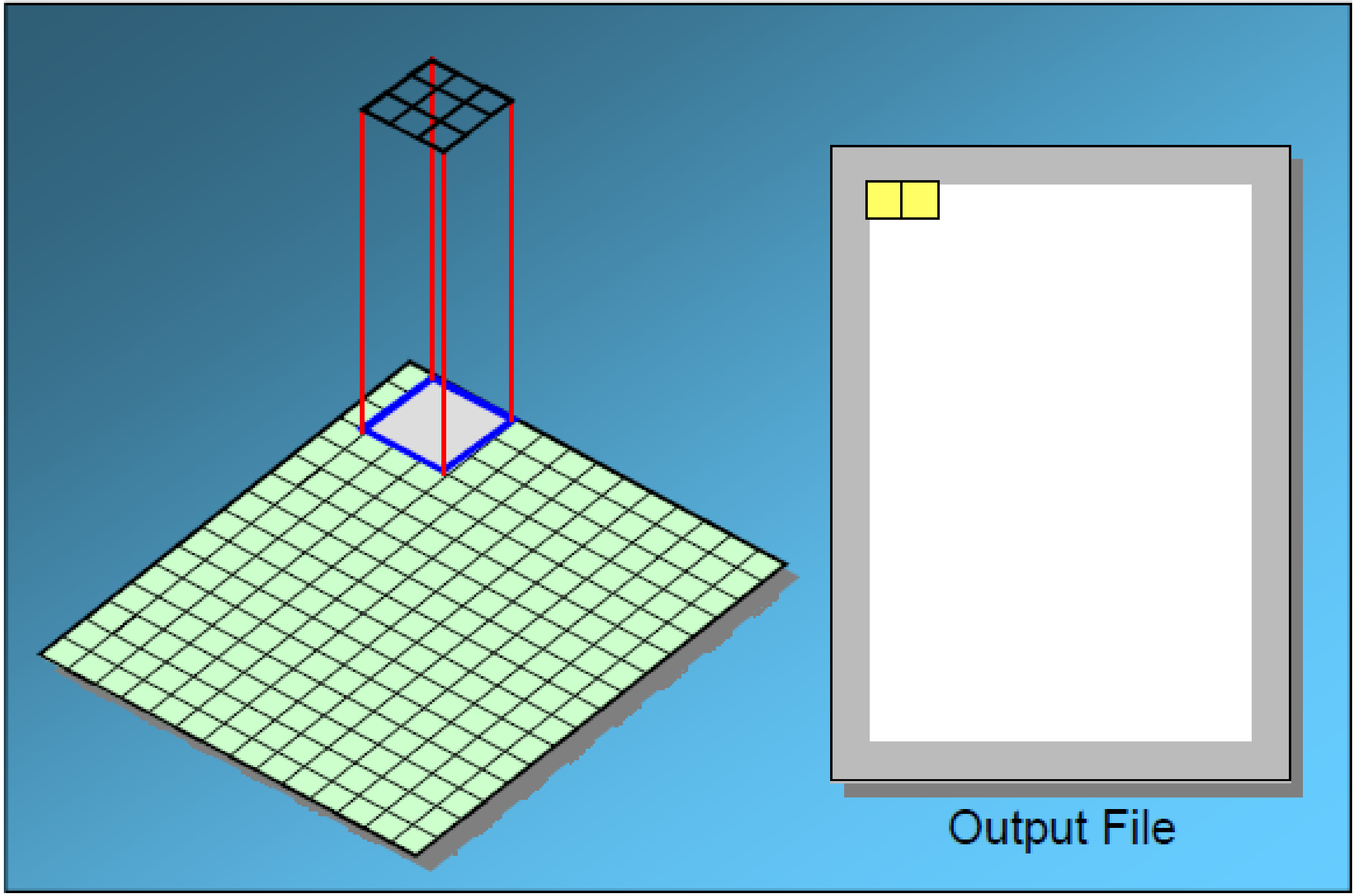
Spatial Filtering: Basics



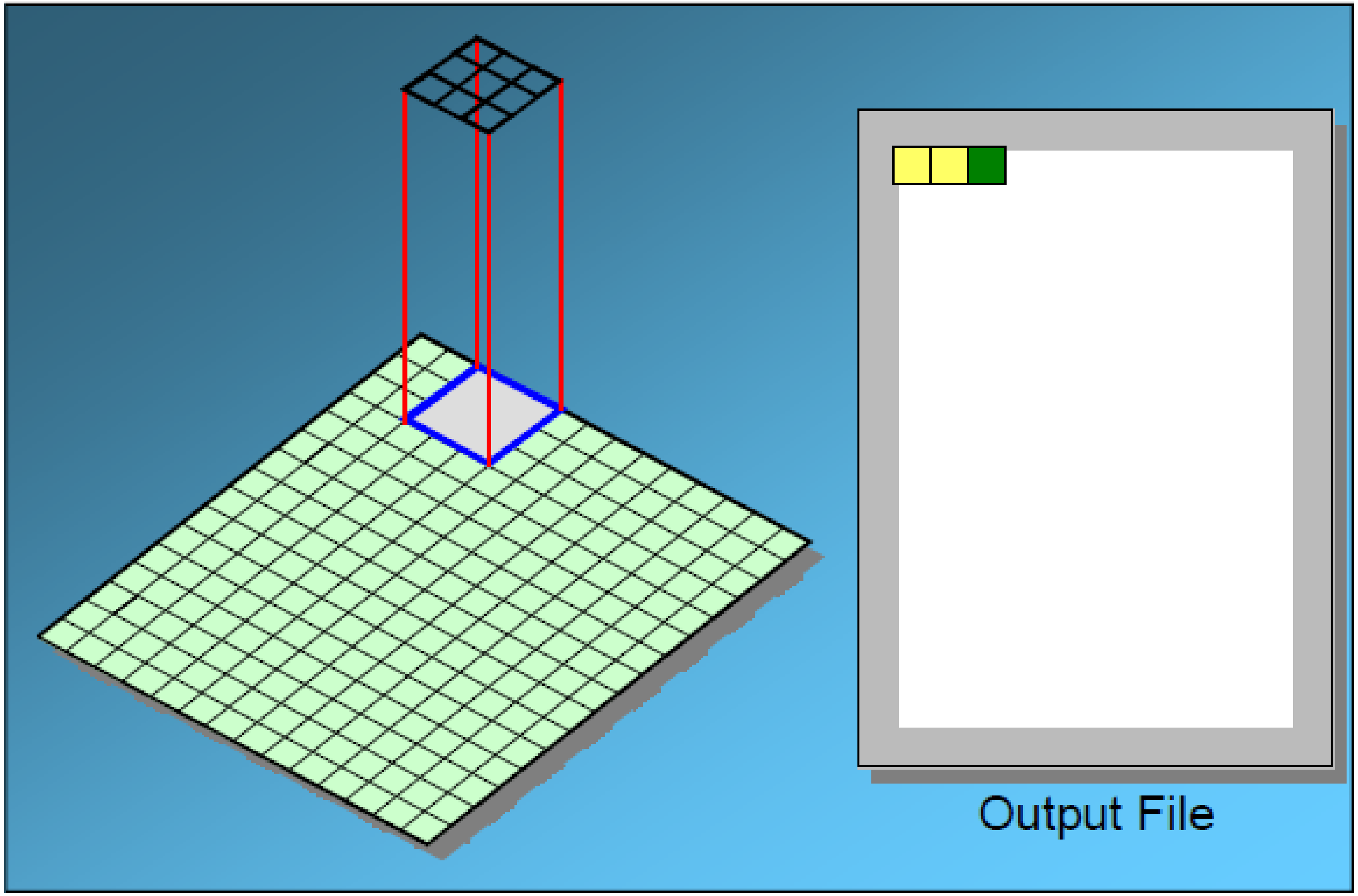
Spatial Filtering: Basics



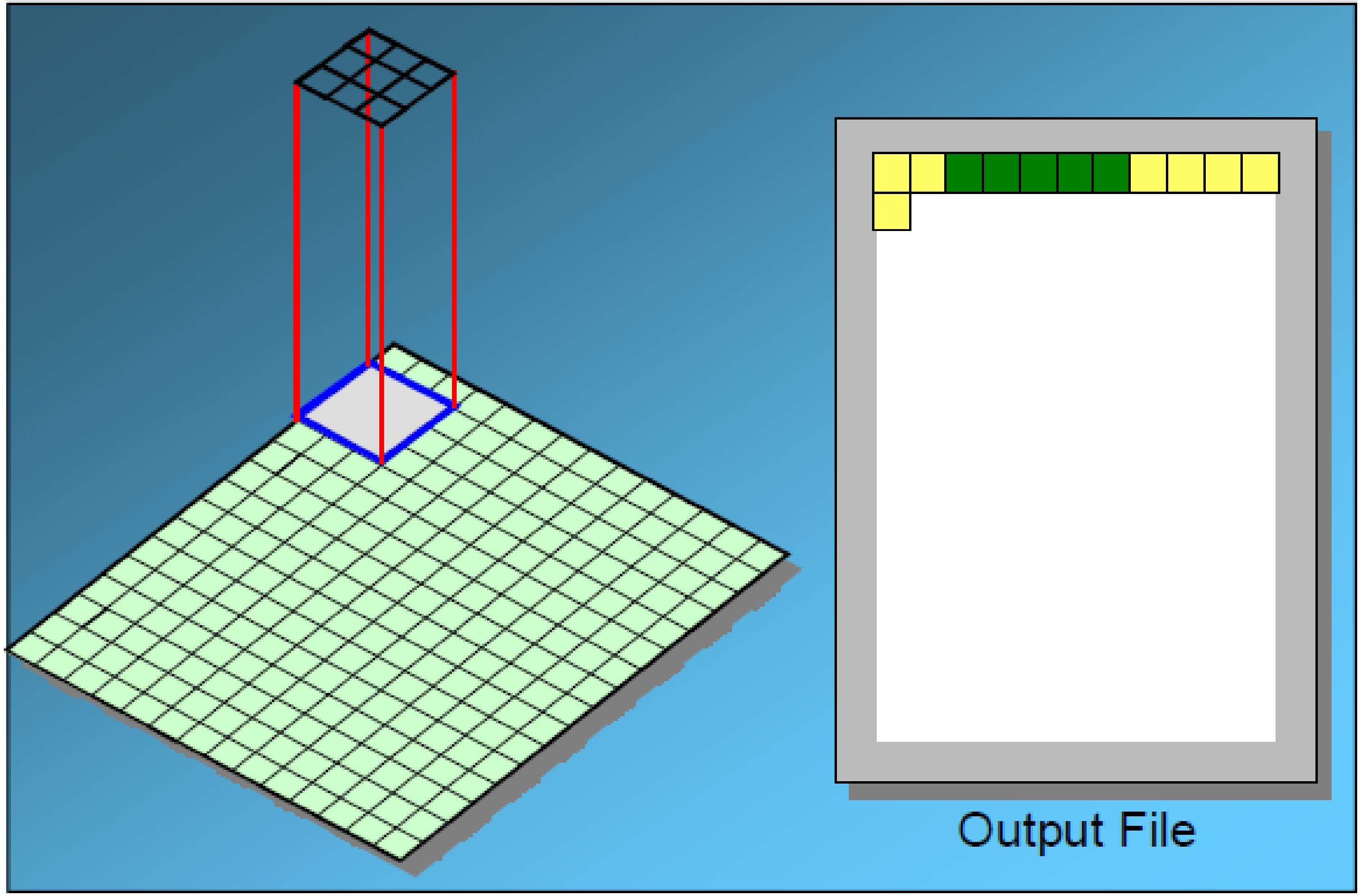
Spatial Filtering: Basics



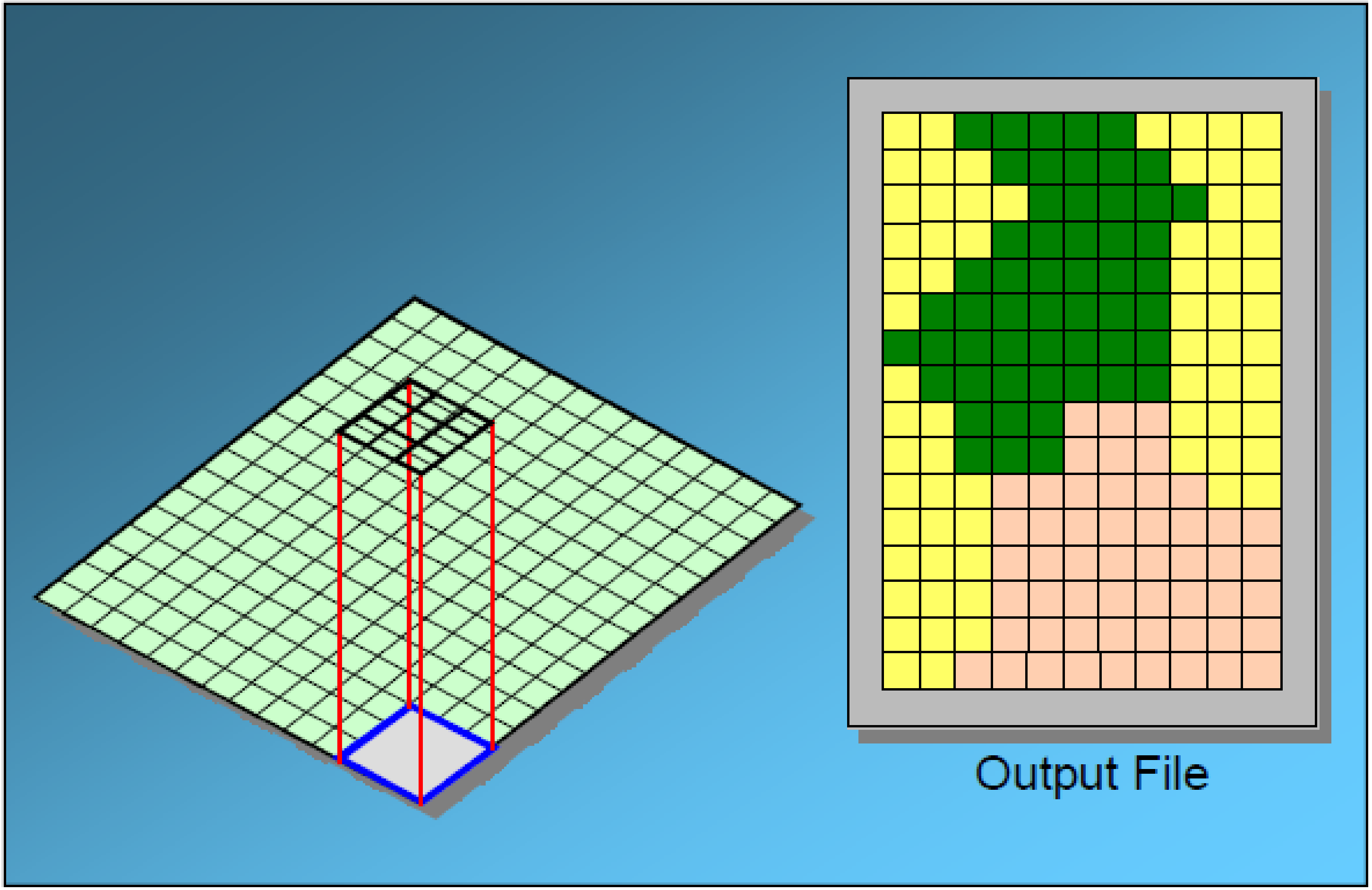
Spatial Filtering: Basics



Spatial Filtering: Basics



Spatial Filtering: Basics



Spatial Filtering: Basics

Mask operation near the image border: Problem arises when part of the mask is located outside the image plane

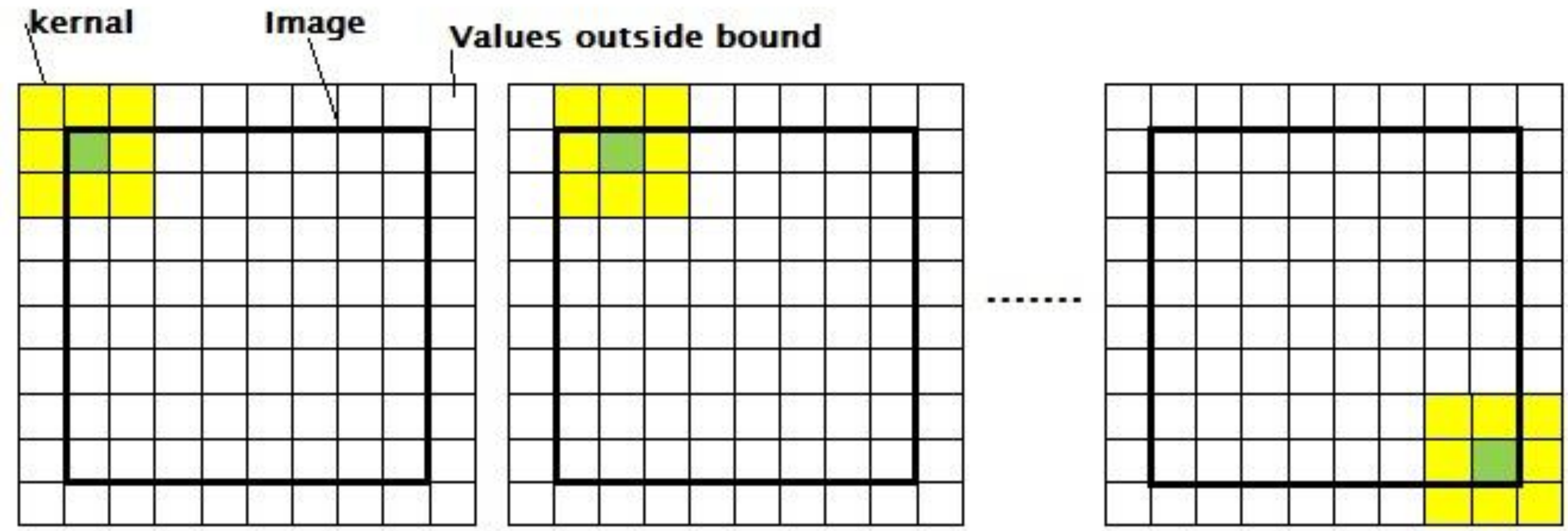
Discard the problem pixels (e.g. 512x512 input 510x510 output if mask size is 3x3)

Zero padding: Expand the input image by padding zeros (512x512 original image, 514x514 padded image, 512x512 output)

Zero padding is not recommended as it creates artificial lines or edges on the border

Pixel replication: We normally use the gray levels of border pixels to fill up the expanded region (for 3x3 mask). For larger masks a border region equal to half of the mask size is mirrored on the expanded region.

Spatial Filtering: Basics



Mask operation near the border: Pixel replication

102	102	130	143	123	115	...	...
102	102	130	143	123	115	...	...
93	93	...	...	...			
98	98	...					
82	82	...					
65	65						
...	...						
...	...						

Expanded area

Original image size (shaded area)

Smoothing Spatial Filters

Simply average all of the pixels in a neighbourhood around a central value

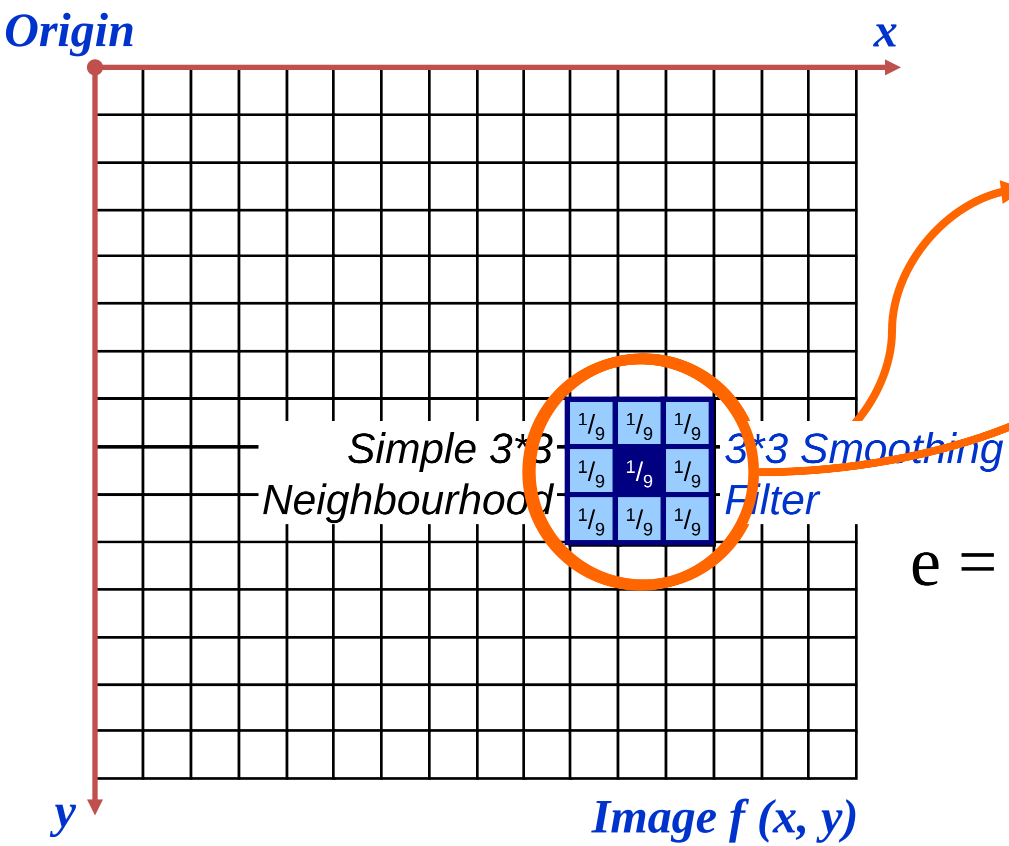
$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$

Simple averaging filter

Smoothing Spatial Filters

- For blurring/noise reduction
- Blurring is usually used in preprocessing steps, e.g., to remove small details from an image prior to object extraction, or to bridge small gaps in lines or curves
- Equivalent to Low-pass spatial filtering in frequency domain because smaller (high frequency) details are removed based on neighborhood averaging (averaging filters)

Smoothing Spatial Filters



104	100	108
99	106	98
95	90	85

Original Image Pixels

*

$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$

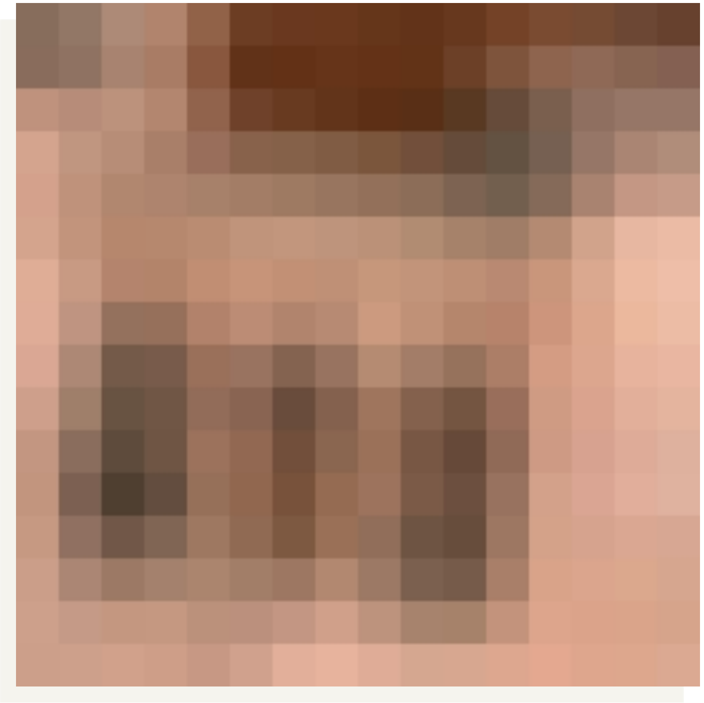
Filter

$$
\begin{aligned}
e = {} & {}^1/_9*106 + \\
& {}^1/_9*104 + {}^1/_9*100 + {}^1/_9*108 + \\
& {}^1/_9*99 + {}^1/_9*98 + \\
& {}^1/_9*95 + {}^1/_9*90 + {}^1/_9*85 \\
= {} & 98.3333
\end{aligned}
$$

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

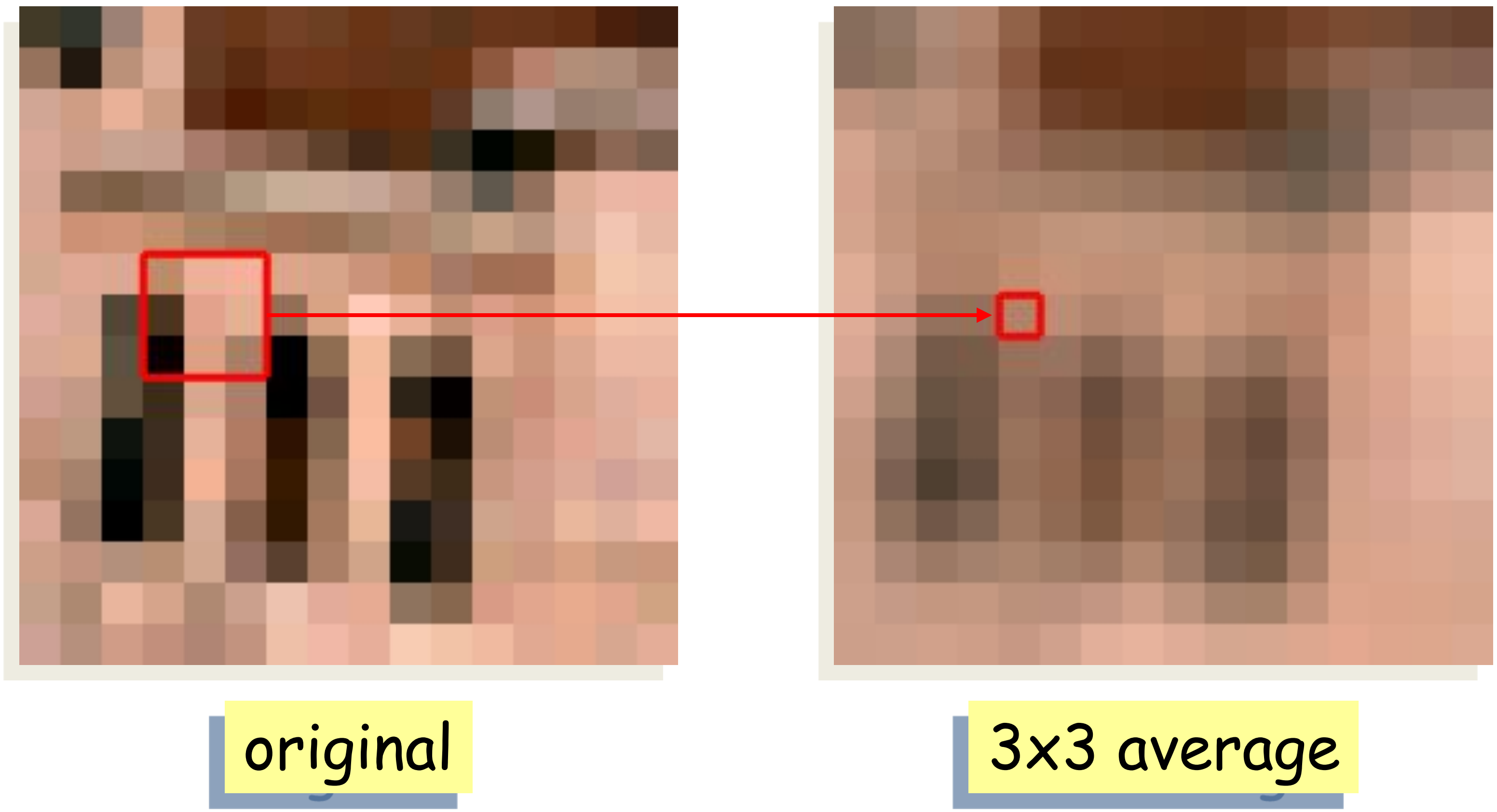
Smoothing Filter: Example

original

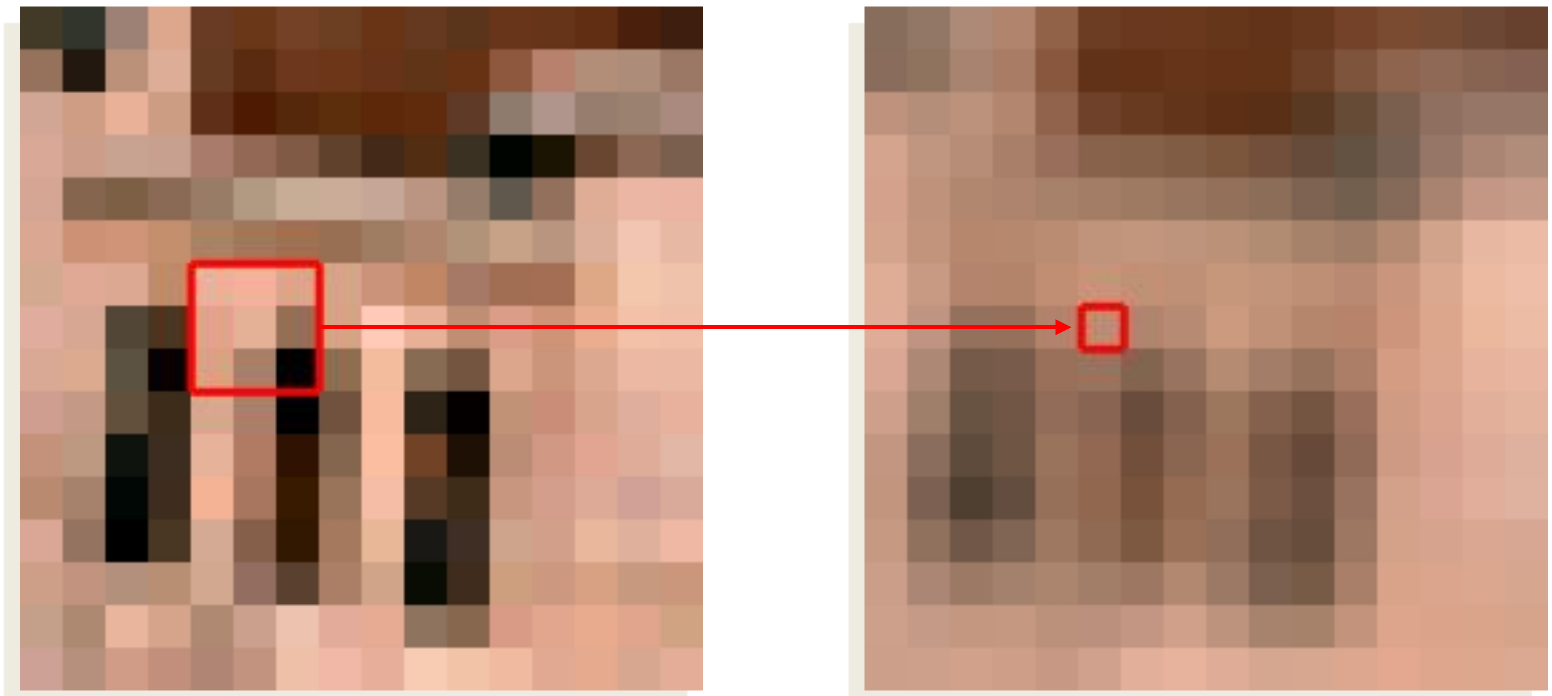


3x3 average

Smoothing Filter: Example



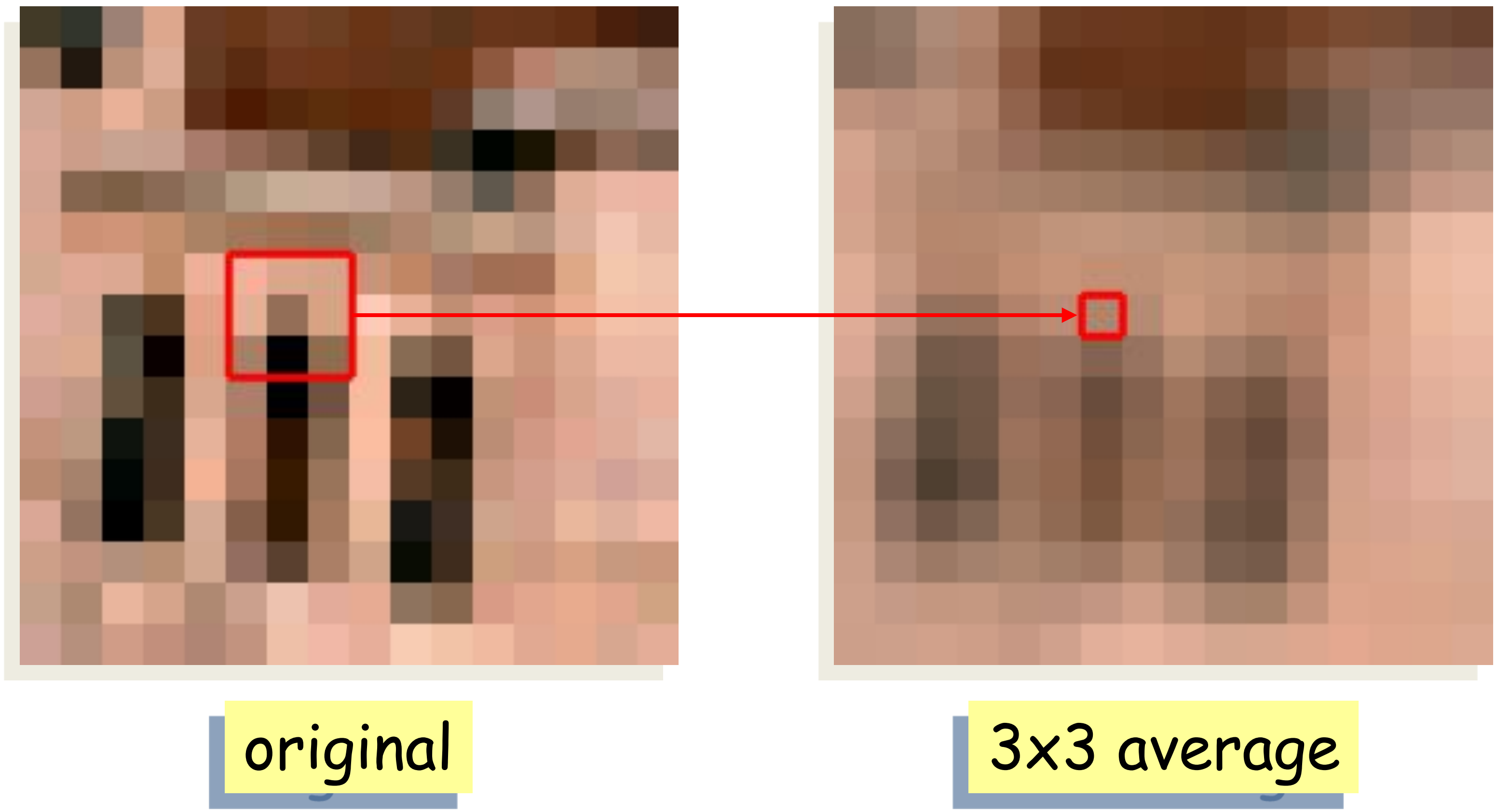
Smoothing Filter: Example



original

3x3 average

Smoothing Filter: Example



Original image

Size: 500x500

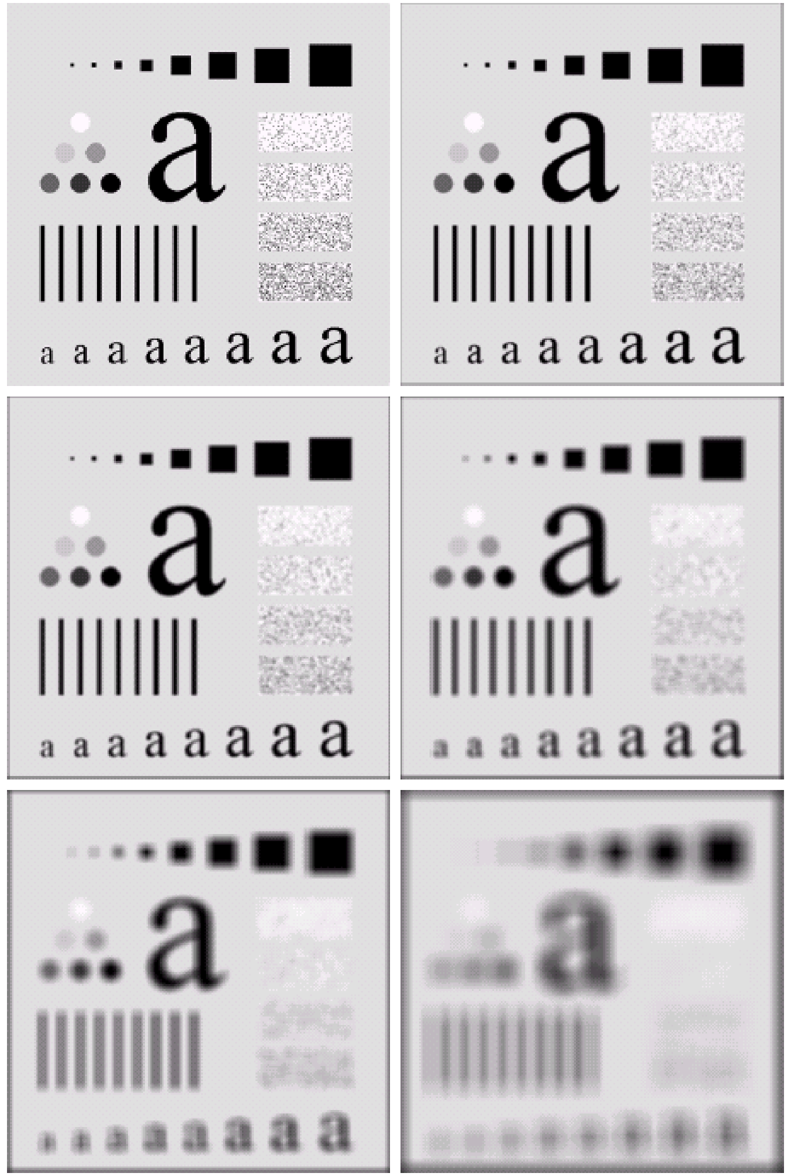
Smooth by 3x3 box filter

Smooth by 5x5 box filter

Smooth by 9x9 box filter

Smooth by 15x15 box filter

Smooth by 35x35 box filter



Notice how detail begins to disappear

Smoothing Spatial Filters

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

Consider the output pixel is positioned at the center

Box Filter all coefficients are equal

Weighted Average give more (less) weight to near (away from) the output location

Sharpening Spatial Filters

Previously we have looked at smoothing filters which remove fine detail

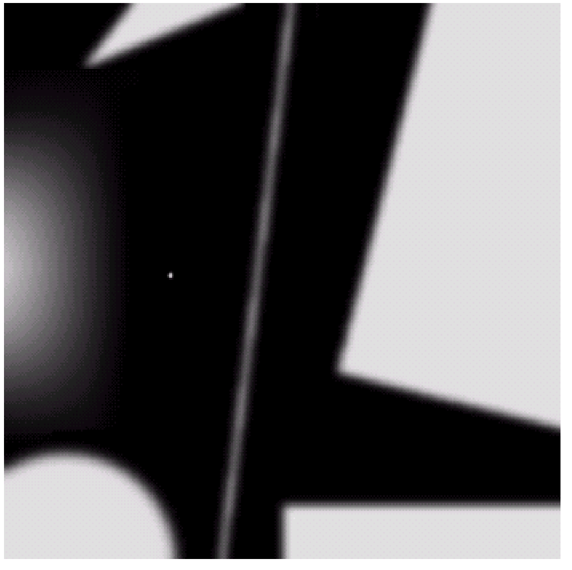
Sharpening spatial filters seek to highlight fine detail

- Remove blurring from images
- Highlight edges

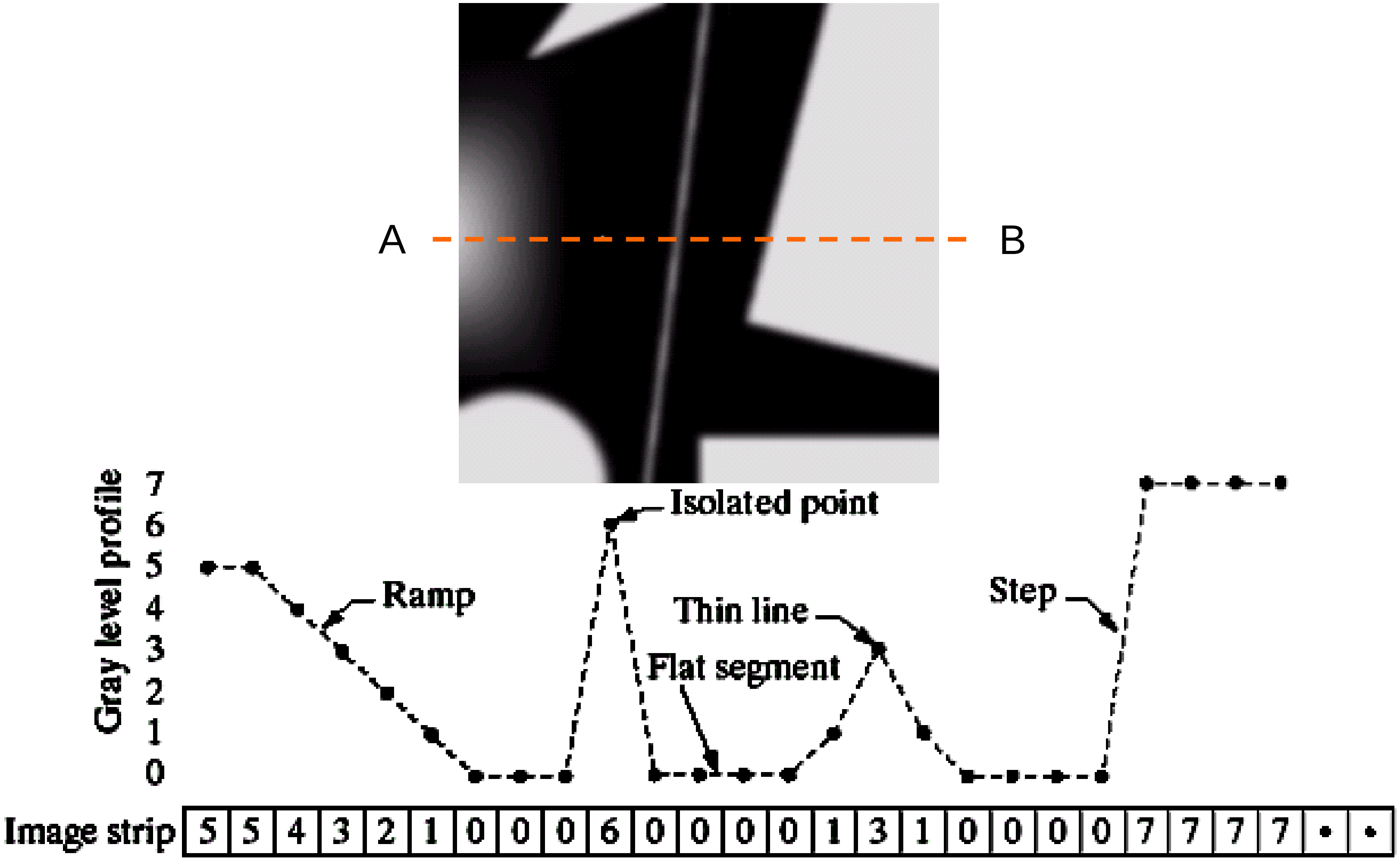
Sharpening filters are based on *spatial differentiation*

Spatial Differentiation

- Let’s consider a simple 1 dimensional example



Spatial Differentiation



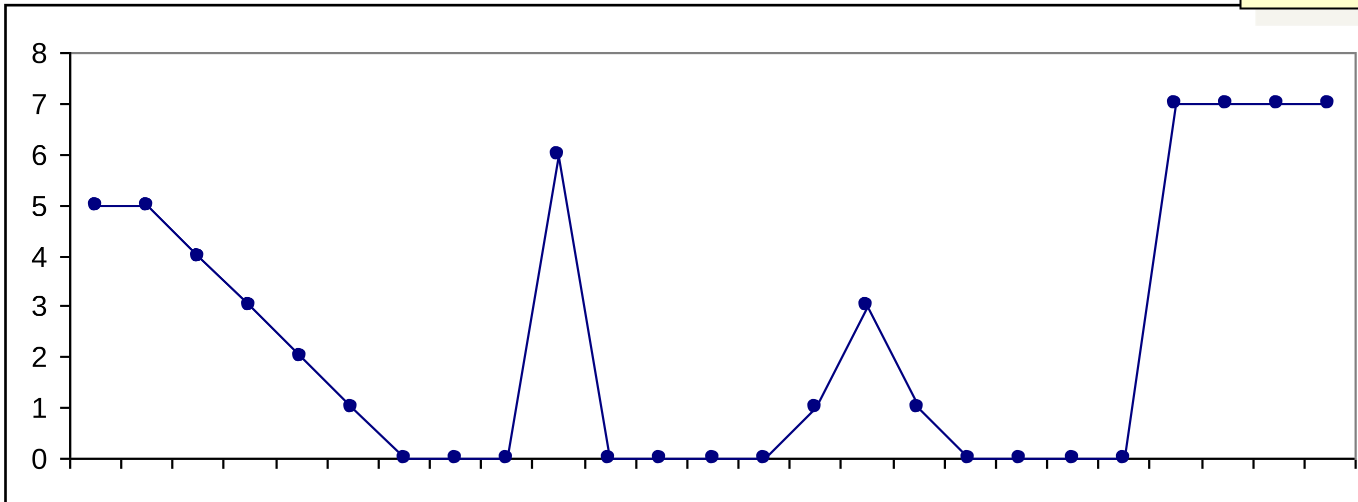
1^{st} Derivative

The 1^{st} derivative of a function is given by:

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

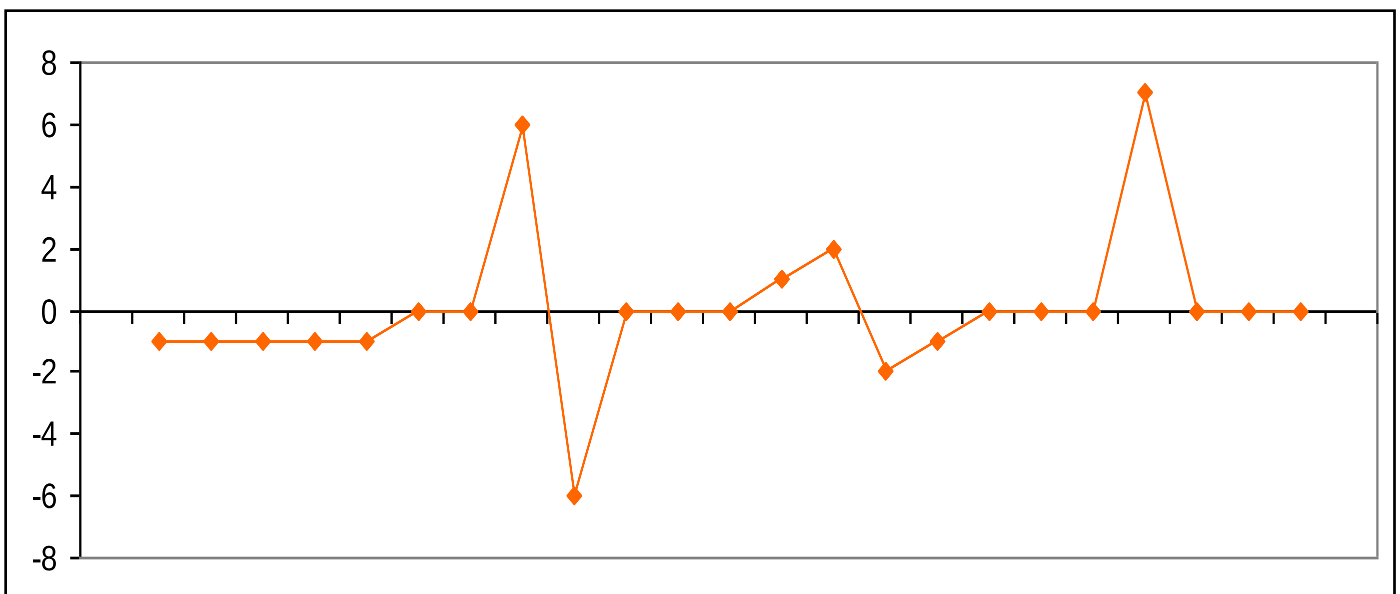
Its just the difference between subsequent values and measures the rate of change of the function

1^{st} Derivative



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

	-1	-1	-1	-1	-1	0	0	6	-6	0	0	0	1	2	-2	-1	0	0	0	7	0	0	0	



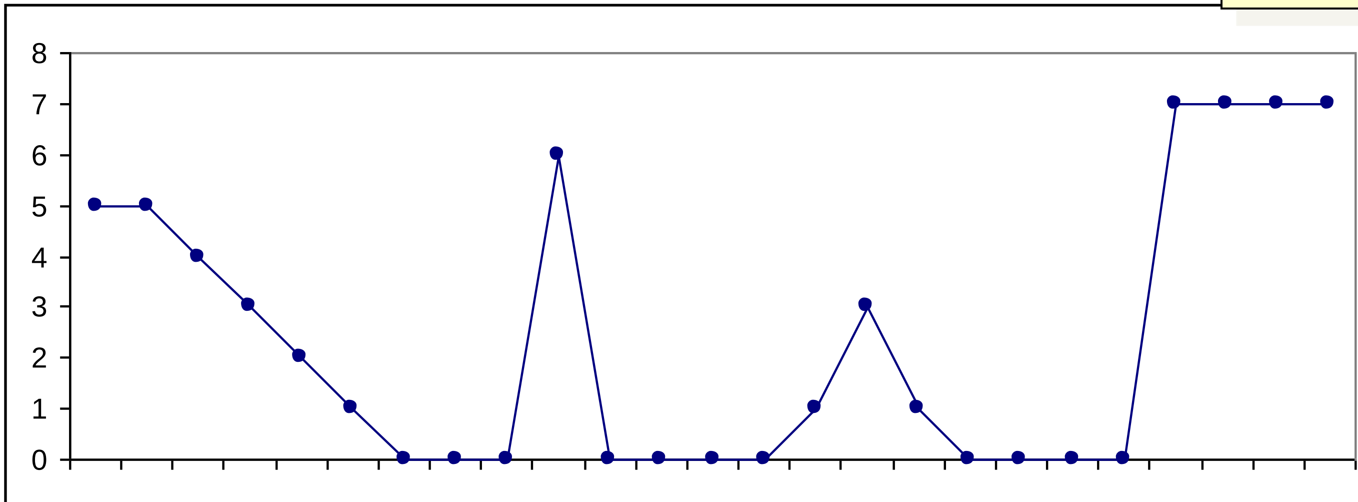
2^{nd} Derivative

The 2nd derivative of a function is given by:

Simply takes into account the values both before and after the current value

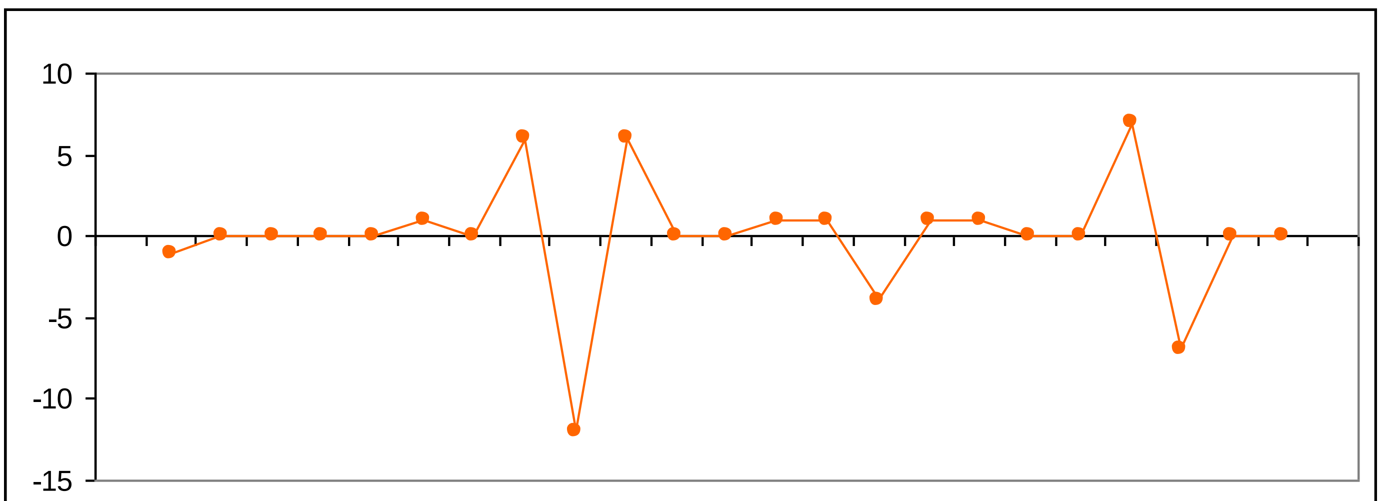
$$\frac{\partial^2 f}{\partial^2 x} = f(x+1) + f(x-1) - 2f(x)$$

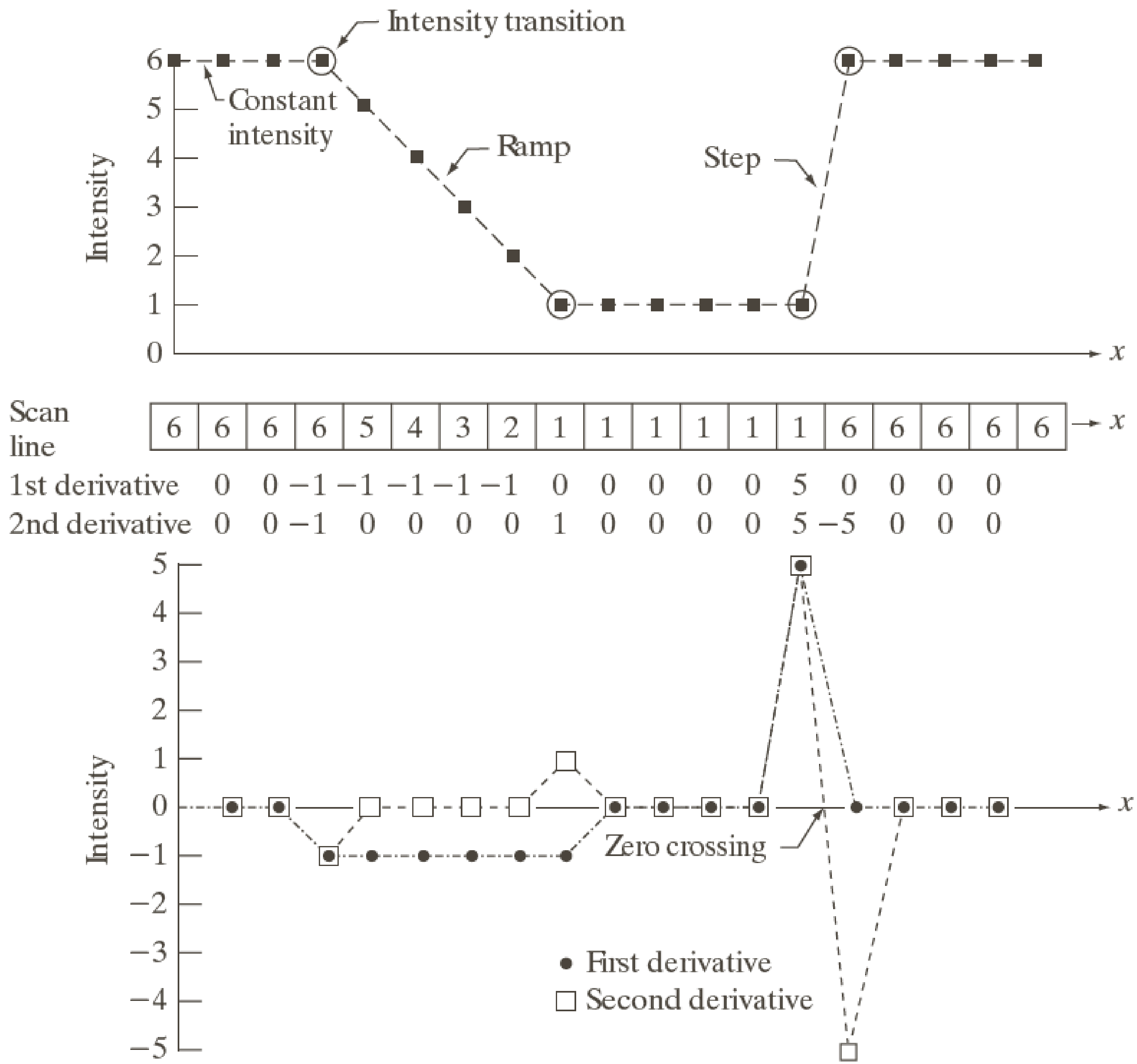
2^{nd} Derivative



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

	-1	0	0	0	0	1	0	6	-12	6	0	0	1	1	-4	1	1	0	0	7	-7	0	0	




Intensity transition
Constant intensity
Ramp
Step
Intensity
x
Scan line
6 6 6 6 5 4 3 2 1 1 1 1 1 1 6 6 6 6 6
1st derivative 0 0 −1 −1 −1 −1 −1 0 0 0 0 0 5 0 0 0 0
2nd derivative 0 0 −1 0 0 0 0 1 0 0 0 0 5 −5 0 0 0
Zero crossing
First derivative
Second derivative

2^{nd} Derivative for Image Enhancement

The 2nd derivative is more useful for image enhancement than the 1st derivative - *Stronger response to fine detail*

We will come back to the 1st order derivative later on

The first sharpening filter we will look at is the *Laplacian*

Laplacian Filter

The Laplacian is defined as follows:

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

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

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

Laplacian Filter

So, the Laplacian can be given as follows:

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

Can we implement it using a filter/ mask?

0	1	0
1	-4	1
0	1	0

Laplacian Filter

(a)

0	1	0
1	−4	1
0	1	0

(b)

1	1	1
1	−8	1
1	1	1

(c)

0	−1	0
−1	4	−1
0	−1	0

(d)

−1	−1	−1
−1	8	−1
−1	−1	−1

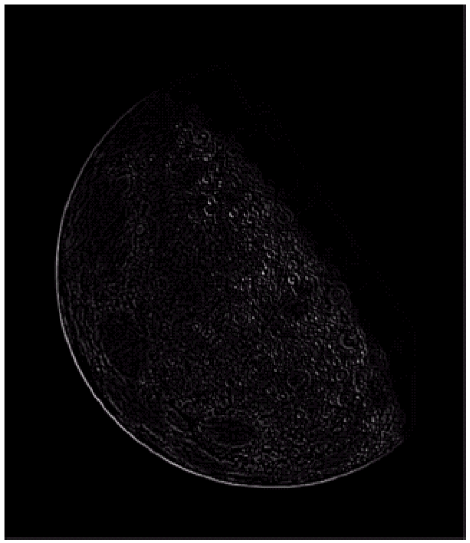
a b
c d

FIGURE 3.39 (a) Filter mask used to implement the digital Laplacian, as defined in Eq. (3.7-4). (b) Mask used to implement an extension of this equation that includes the diagonal neighbors. (c) and (d) Two other implementations of the Laplacian.

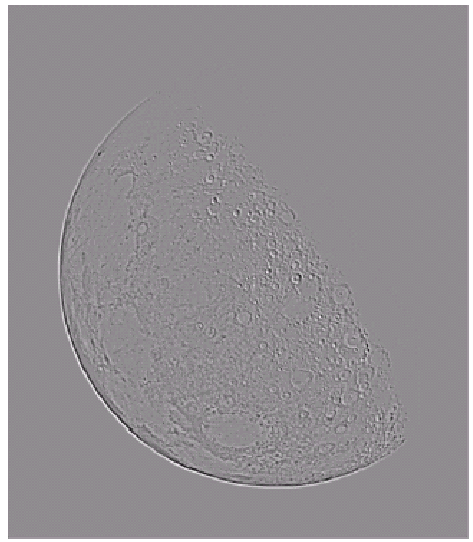
Laplacian Filter

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

Original
Image

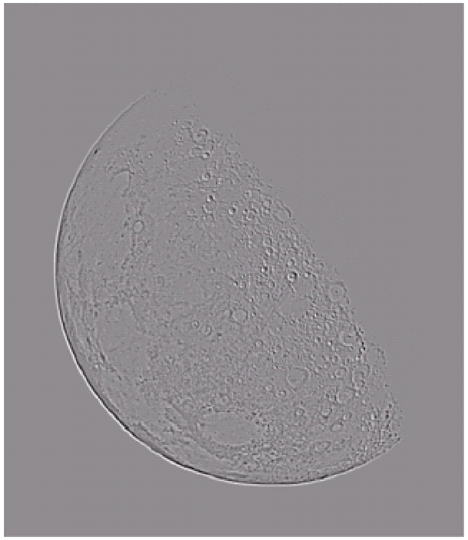


Laplacian
Filtered Image



Laplacian
Filtered Image
Scaled for Display

Laplacian Image Enhancement



Laplacian
Filtered Image
Scaled for Display

The result of a Laplacian filtering is not an enhanced image

To generate the final enhanced image

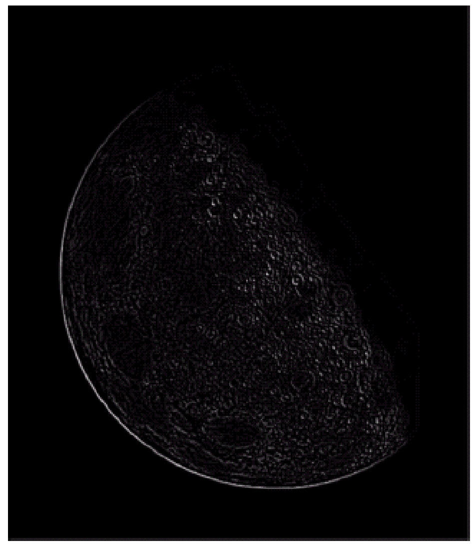
w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f, & w_5 < 0 \\ f(x,y) + \nabla^2 f, & w_5 > 0 \end{cases}$$

Laplacian Image Enhancement

Original
Image

-



Laplacian
Filtered Image

=

Sharpened
Image

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

Laplacian Image Enhancement

Simplified Image Enhancement

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

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

Simplified Image Enhancement

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

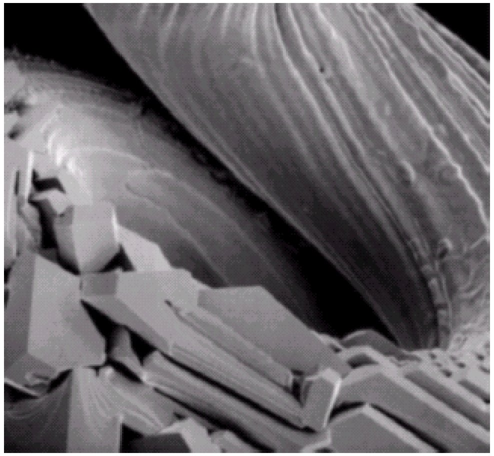
$$g(x,y) = f(x,y) - \nabla^2 f$$

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

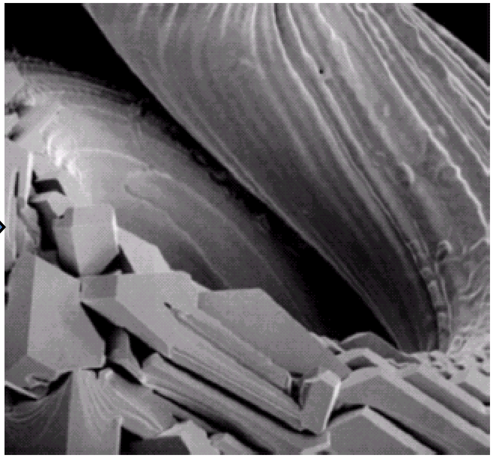
0	-1	0
-1	5	-1
0	-1	0

Simplified Image Enhancement

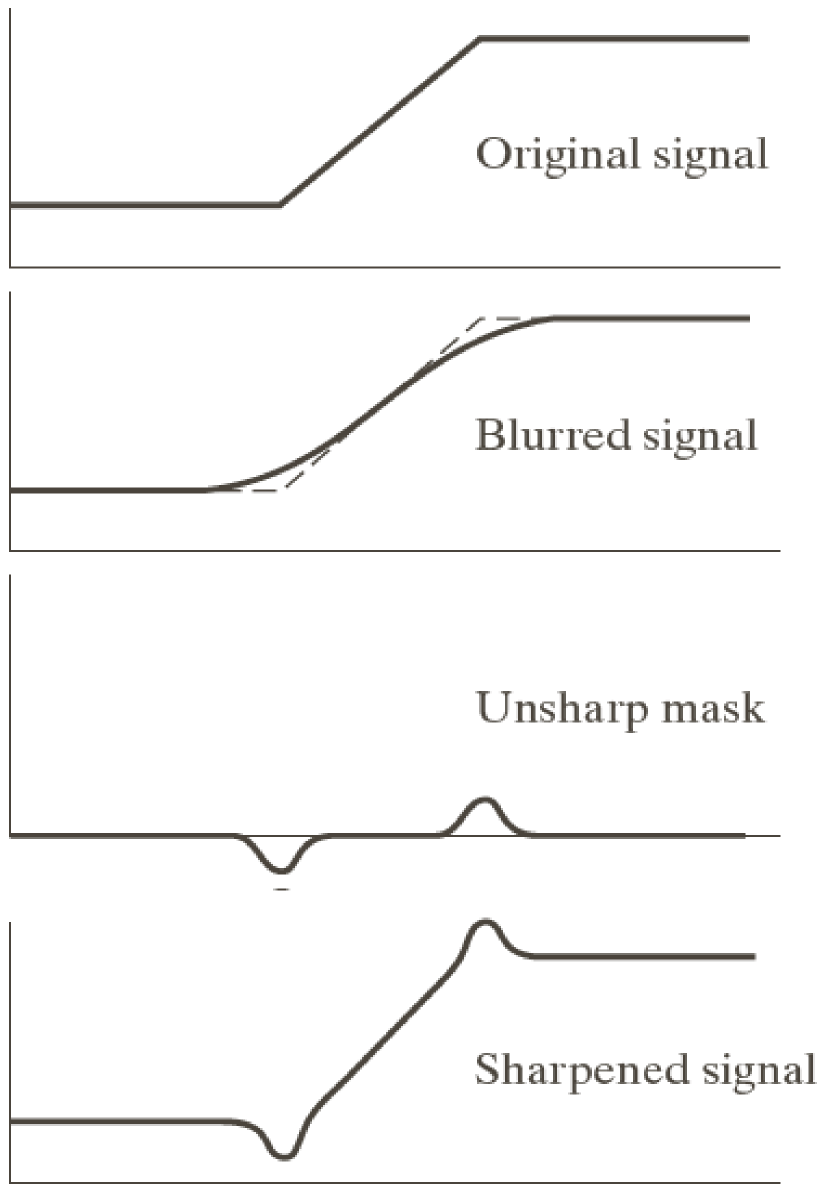
- This gives us a new filter which does the whole job for us in one step



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



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.

Use of first derivatives for image enhancement: The Gradient

- The **gradient** of a function f(x,y) is defined as

$$\nabla \mathrm{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

Gradient Operators

- Most common differentiation operator is the gradient vector.

$$\nabla f(x,y) = \begin{bmatrix} \dfrac{\partial f(x,y)}{\partial x} \\ \dfrac{\partial f(x,y)}{\partial y} \end{bmatrix} = \begin{bmatrix} G_x \\ G_y \end{bmatrix}$$

Magnitude:

$$\left|\nabla f(x,y)\right| = \left[G_x{}^2 + G_y{}^2\right]^{1/2} \approx \left|G_x\right| + \left|G_x\right|$$

Direction:

$$\angle f(x,y) = \tan^{-1}\left[\frac{G_y}{G_x}\right]$$

Gradient Operators

Sobel Operator

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

Extract horizontal edges

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

Extract vertical edges

Emphasize more the current point (y direction)

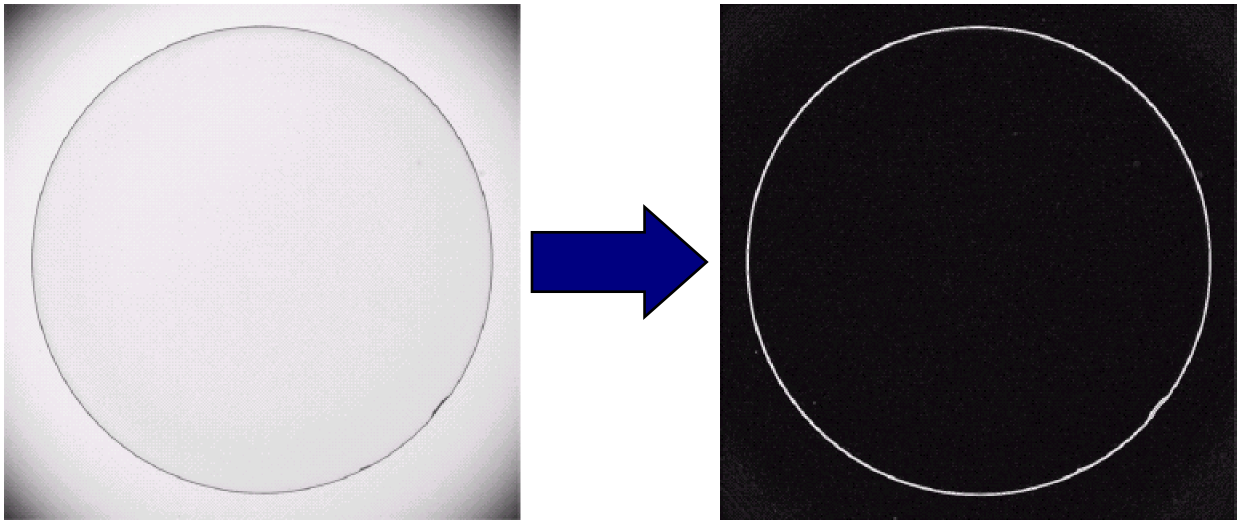
$$\nabla f \approx \left|(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)\right| + \left|(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)\right|$$

Emphasize more the current point (x direction)

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Pixel Arrangement

Sobel Operator: Example



An image of a contact lens which is enhanced in order to make defects more obvious

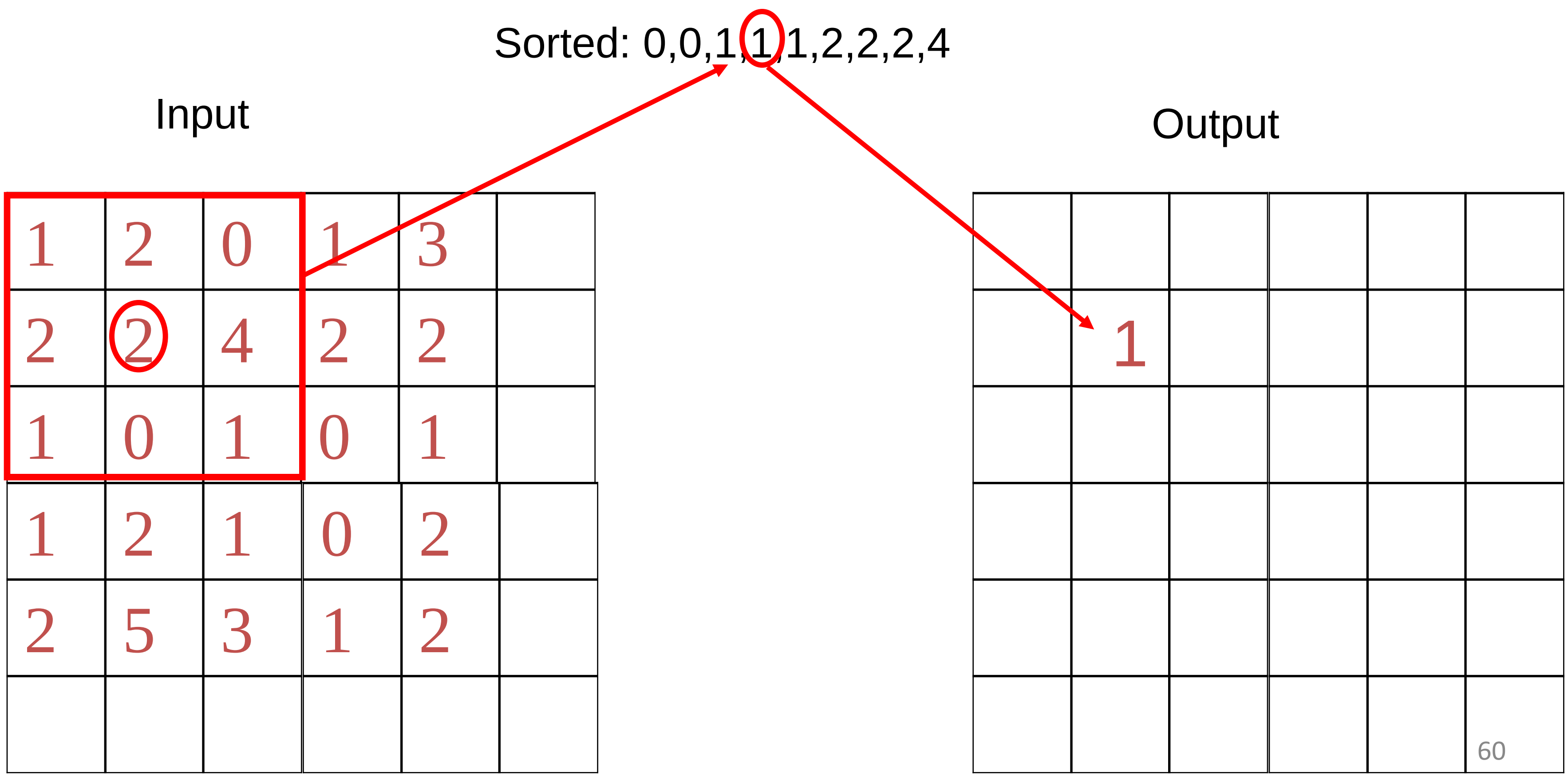
Sobel filters are typically used for edge detection

Order-Statistic Filtering

- Output is based on order of gray levels in the masked area
- Some simple neighbourhood 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:** The median value of a set of numbers is the midpoint value in that set

Median Filter

- For an image, mask symmetric: 3x3, 5x5, etc.



Median Filtering

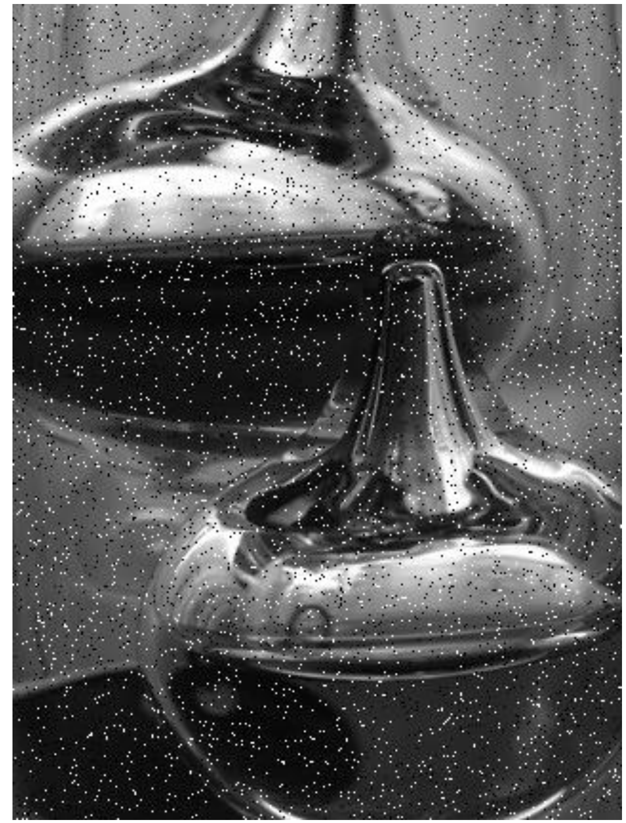
10	20	20
20	15	20
20	25	100

Sort the values
Determine the median

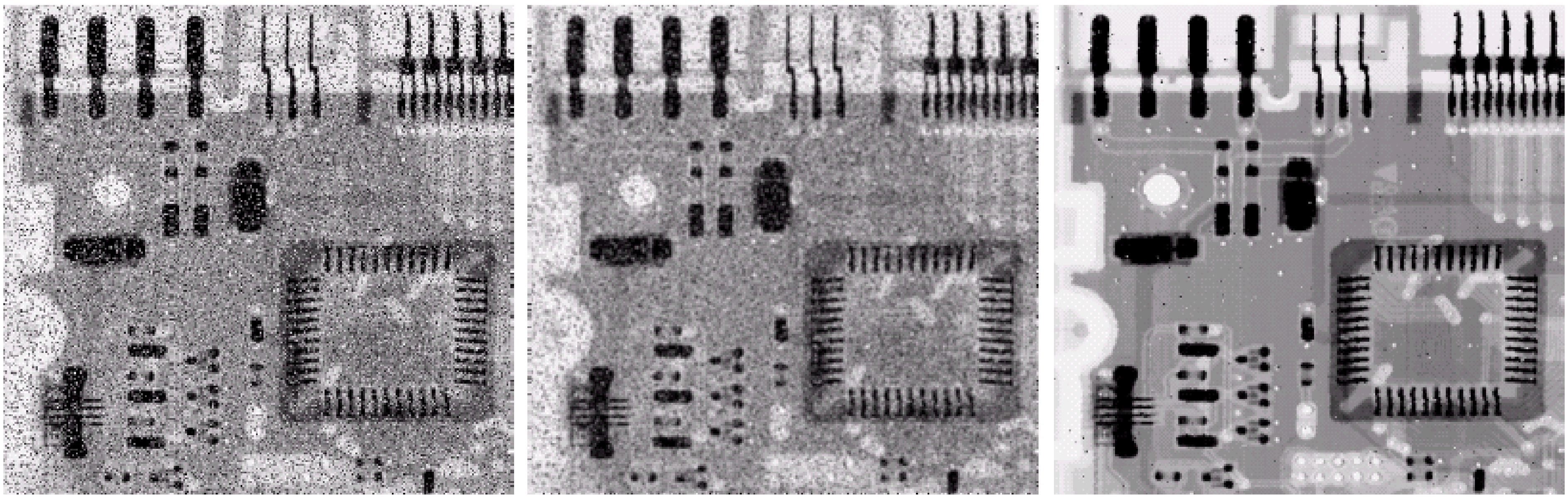
Median = ? 20

- Particularly effective when
  - The noise pattern consists of strong impulse noise (salt-and-pepper)

Salt and Pepper Noise



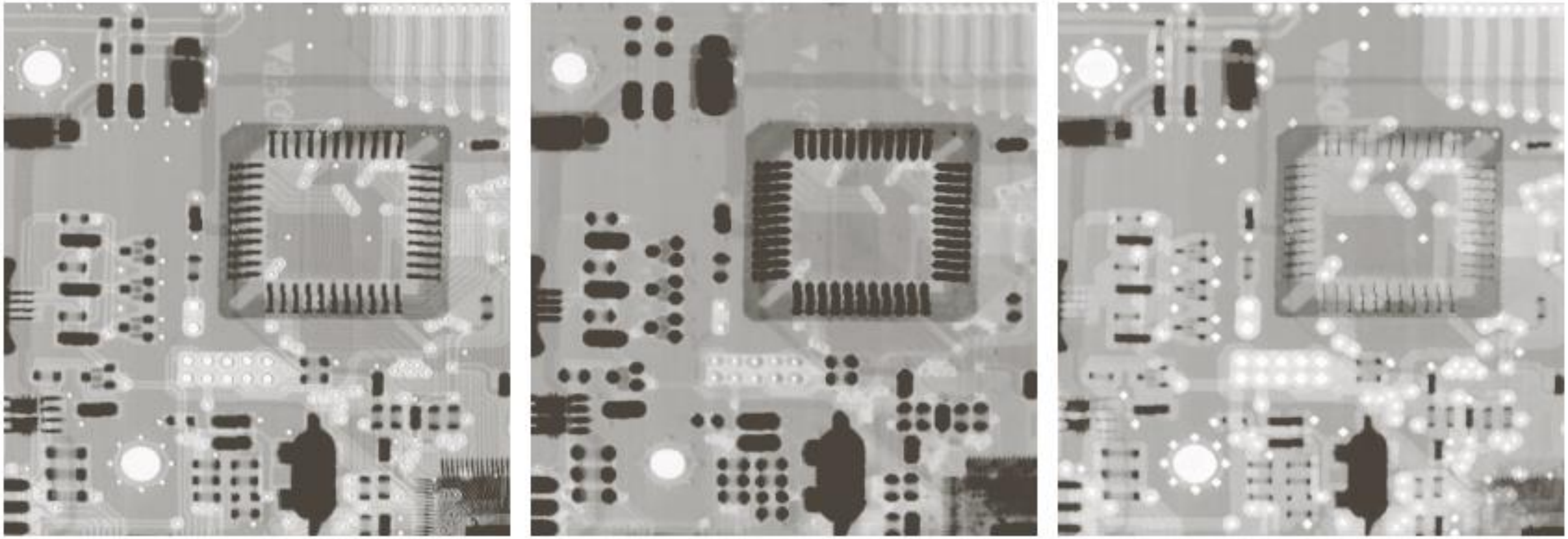
Median Filtering



a b c

FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3 × 3 averaging mask. (c) Noise reduction with a 3 × 3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Min/Max Filtering

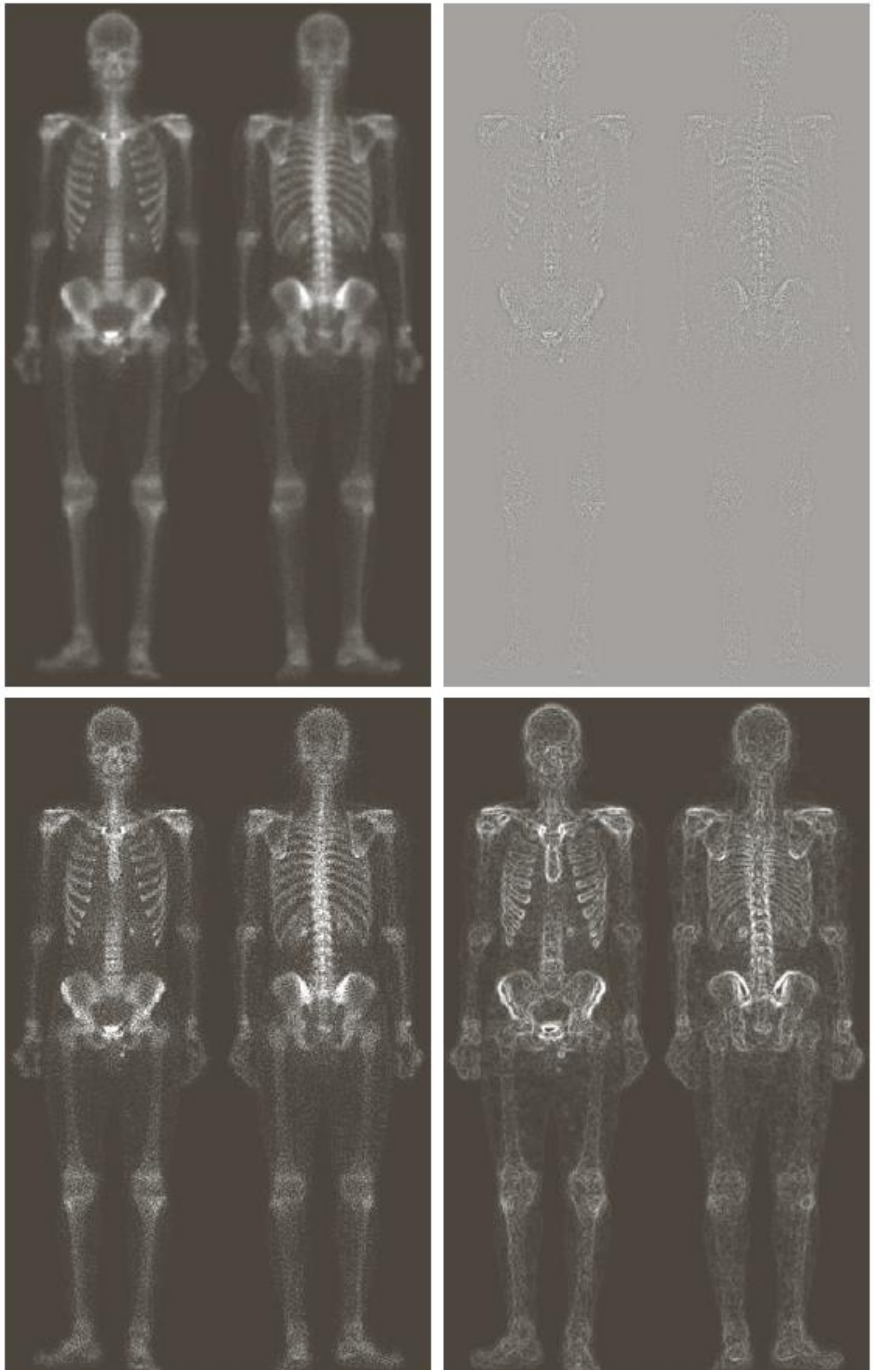


Combining Spatial Enhancement Methods

Successful image enhancement is typically not achieved using a single operation

Rather we combine a range of techniques in order to achieve a final result

This example will focus on enhancing the bone scan



a	b
c	d

FIGURE 3.43 (a) Image of whole body bone scan. (b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b). (d) Sobel gradient of (a).

Readings from Book (3rd Edn.)

- 3.5 Spatial filtering
- 3.5 Spatial filtering
- Sharpening Filters (Chapter – 3)
- Reading Assignment
- High Boost Filtering (Chap – 3.6.3)

Acknowledgements

Material in these slides has been taken from, the following resources

- Digital Image Processing", Rafael C. Gonzalez & Richard E. Woods, Addison-Wesley, 2002
- Peters, Richard Alan, II, Lectures on Image Processing, Vanderbilt University, Nashville, TN, April 2008
- Brian Mac Namee, Digitial Image Processing, School of Computing, Dublin Institute of Technology
- Computer Vision for Computer Graphics, Mark Borg