Digital Image Processing

Image Enhancement
(Spatial Filtering 1)

By

Dr. Hafeez Ur Rehman

Contents

In this session we will look at spatial filtering techniques:

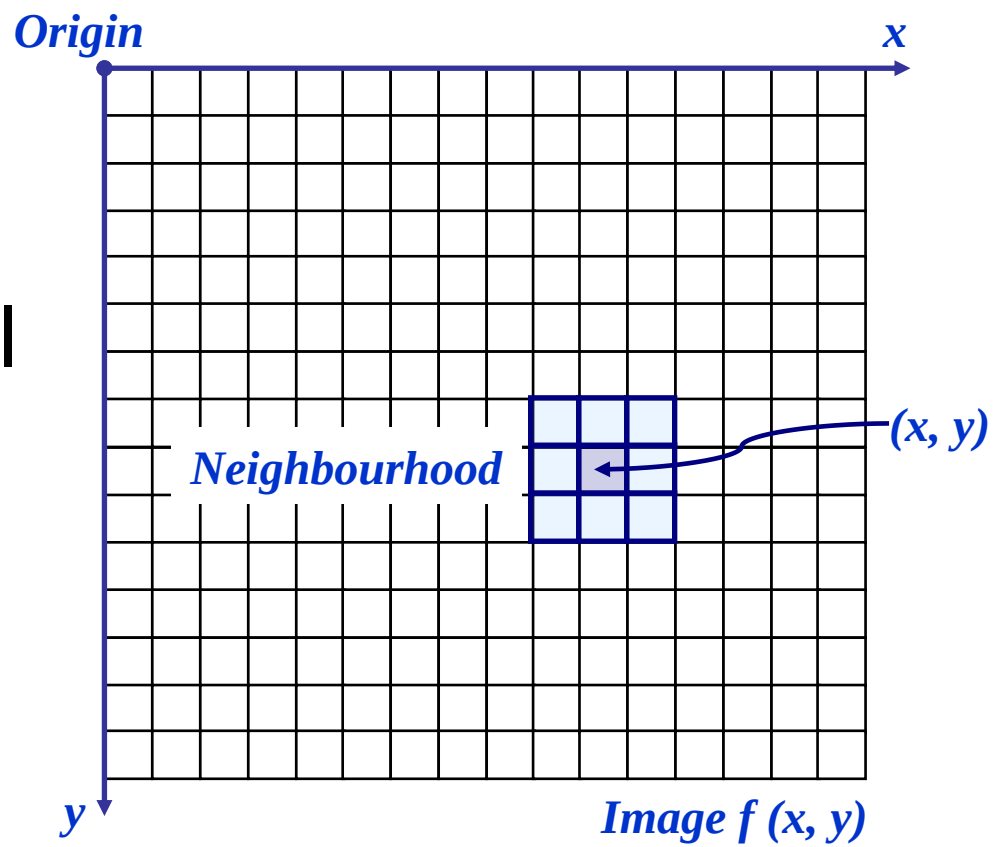
- Neighbourhood operations
- Linear vs Non-Linear Operations
- What is spatial filtering?
- What happens at the edges?
- Smoothing linear operations
- Simple non-linear operations
- Unsharp masking

Neighbourhood Operations

Neighbourhood operations simply **operate on a larger neighbourhood** of pixels than point operations

Neighbourhoods are mostly a rectangle around a central pixel

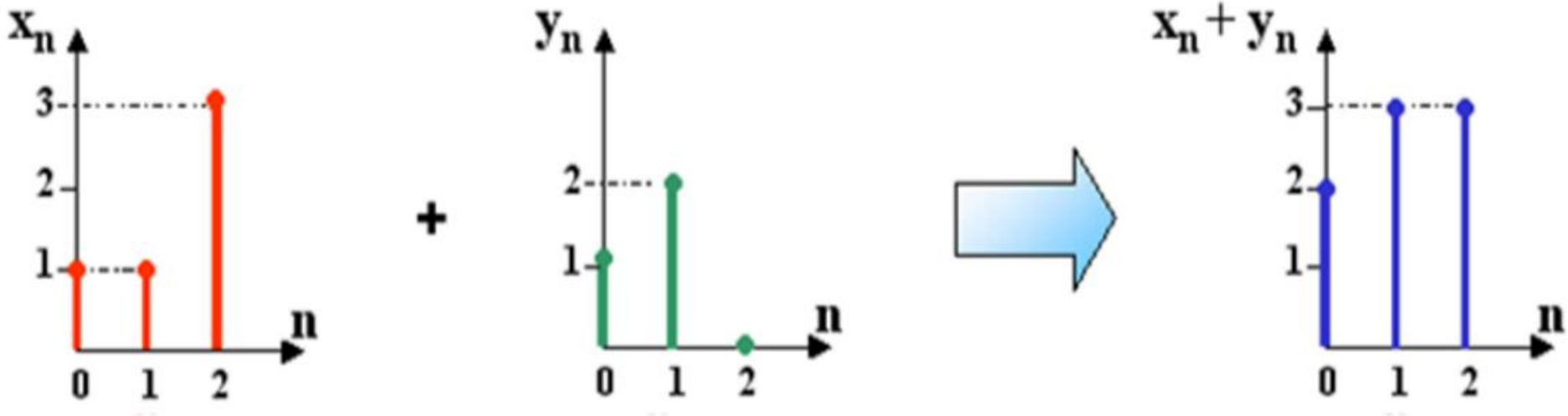
Any size rectangle and any shape filter are possible



Linear vs Non Linear Filters

In mathematics, a linear map or linear function $f(x)$ is a function that satisfies the following two properties:

- Additivity: $f(x + y) = f(x) + f(y)$.
- Homogeneity of degree 1: $f(\alpha x) = \alpha f(x)$ for all α.



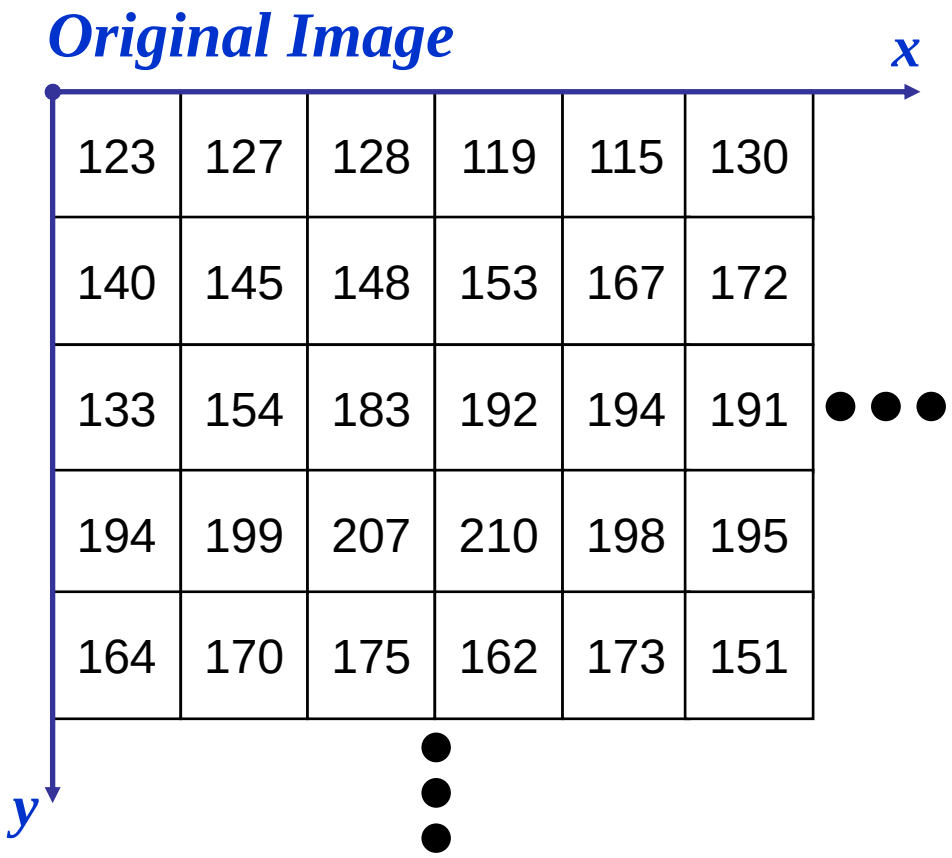
1. **mean(Xn)+mean(Yn)=mean(Xn+Yn)**
   8/3=1+5/3 **which is OK, so mean is linear function.**
2. **median(Xn)+median(Yn)=median(Xn+Yn),**
   1+1=3 **which is not true, median is non linear.**

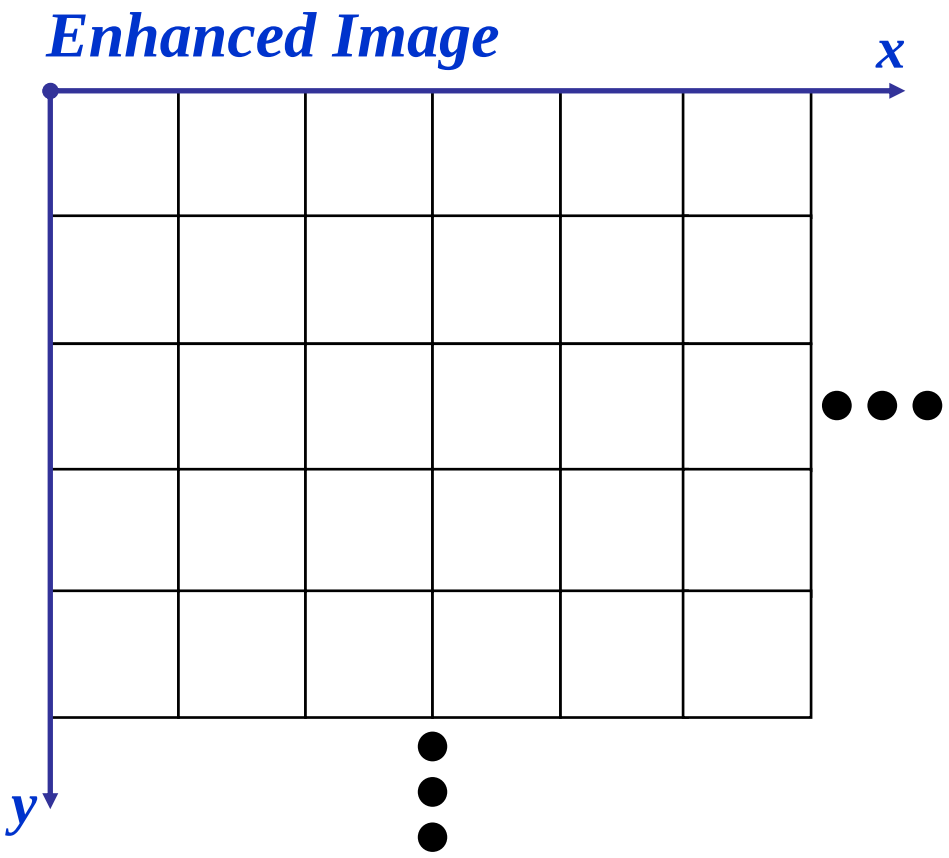
Linear vs Non Linear Filters

Knowing which type of filter to select depends on the goals and nature of the image data.

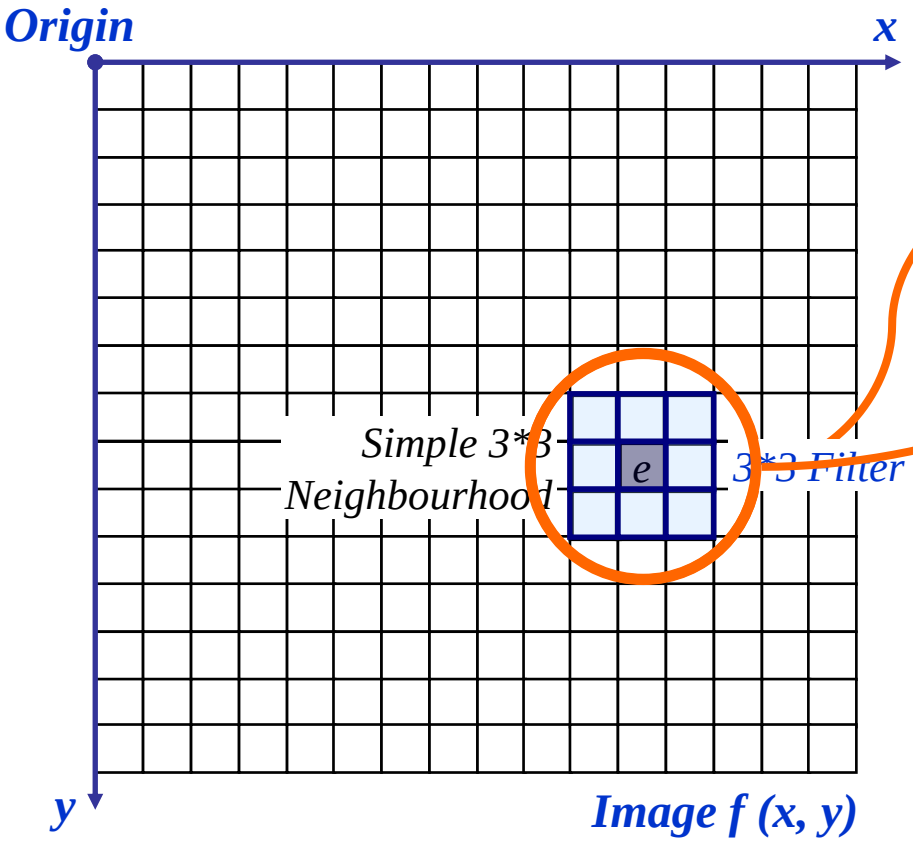
1. In cases where the input data contains a large amount of noise but the magnitude is low, a linear low-pass filter may suffice.
2. Conversely, if an image contains a low amount of noise but with relatively high magnitude, then a non linear (e.g., median filter) may be more appropriate.

Simple Neighbourhood Operations Example





The Spatial Filtering Process



a	*b*	*c*
d	*e*	*f*
g	*h*	*i*

Original Image Pixels

*

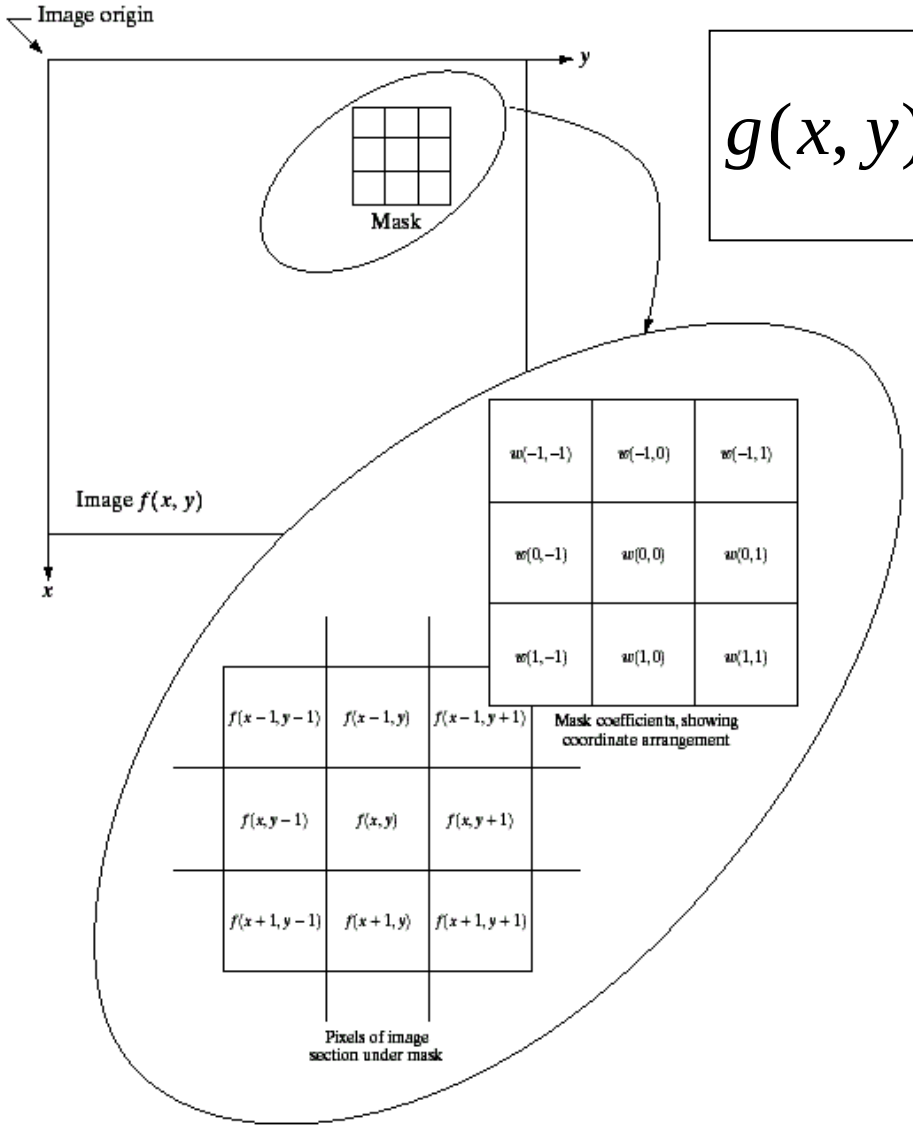
r	*s*	*t*
u	*v*	*w*
x	*y*	*z*

Filter

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

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

Spatial Filtering: Equation Form



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

Filtering can be given in equation form as shown above

Notations are based on the image shown to the left

Correlation & Convolution

The filtering we have been talking about so far is referred to as ***correlation*** with the filter itself referred to as the *correlation kernel*

Convolution is a similar operation, with just one subtle difference

a	*b*	*c*
d	*e*	*e*
f	*g*	*h*

Original Image Pixels

$*$

r	*s*	*t*
u	*v*	*w*
x	*y*	*z*

Filter

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

For symmetric filters it makes no difference

Correlation & Convolution

Properties of Convolution

There are a number of important mathematical properties associated with convolution.

1. **Convolution is commutative** .

$$c = a \otimes b = b \otimes a$$

2. **Convolution is associative**.

$$c = a \otimes (b \otimes c) = (a \otimes b) \otimes c = a \otimes b \otimes c$$

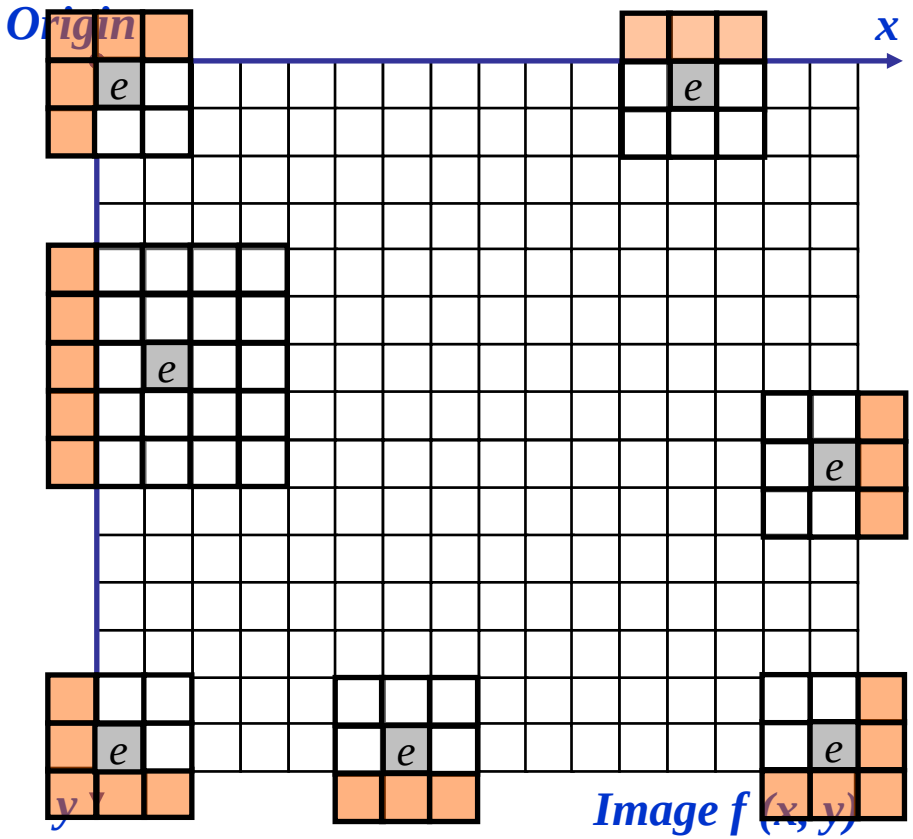
3. **Convolution is distributive**.

$$c = a \otimes (b + d) = (a \otimes b) + (a \otimes d)$$

where a, b, c, and d are all images, either continuous or discrete.

Strange Things Happen At The Edges!

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

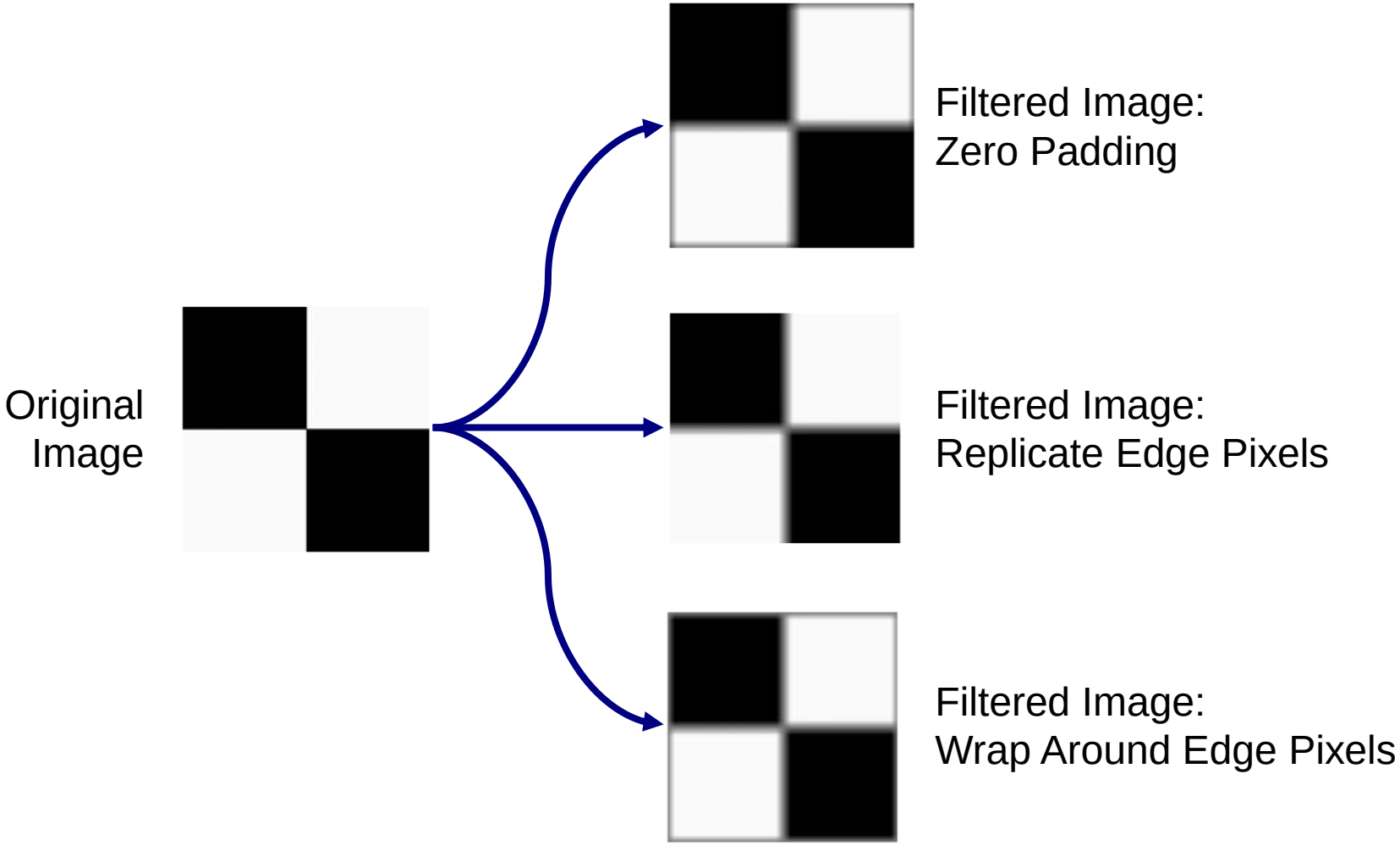


Strange Things Happen At The Edges! (cont…)

There are a few approaches to dealing with missing edge pixels:

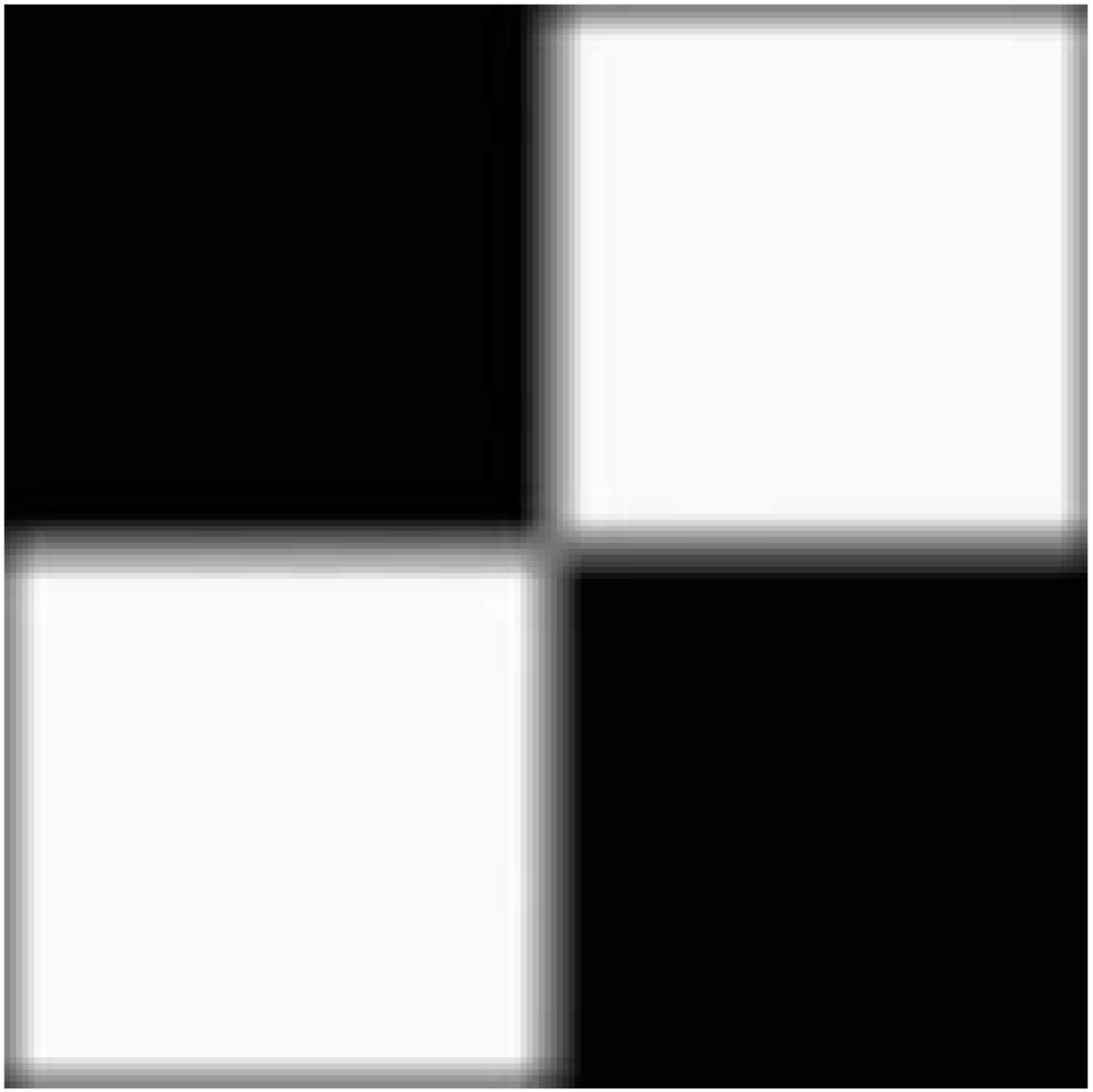
- Omit missing pixels
  - Only works with some filters
  - Can add extra code and slow down processing
- Pad the image
  - Typically with either all white or all black pixels
- Replicate border pixels
- Truncate the image, only if borders are not important.
- Allow pixels *wrap around* the image
  - Can cause some strange image artefacts

Strange Things Happen At The Edges! (cont…)



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

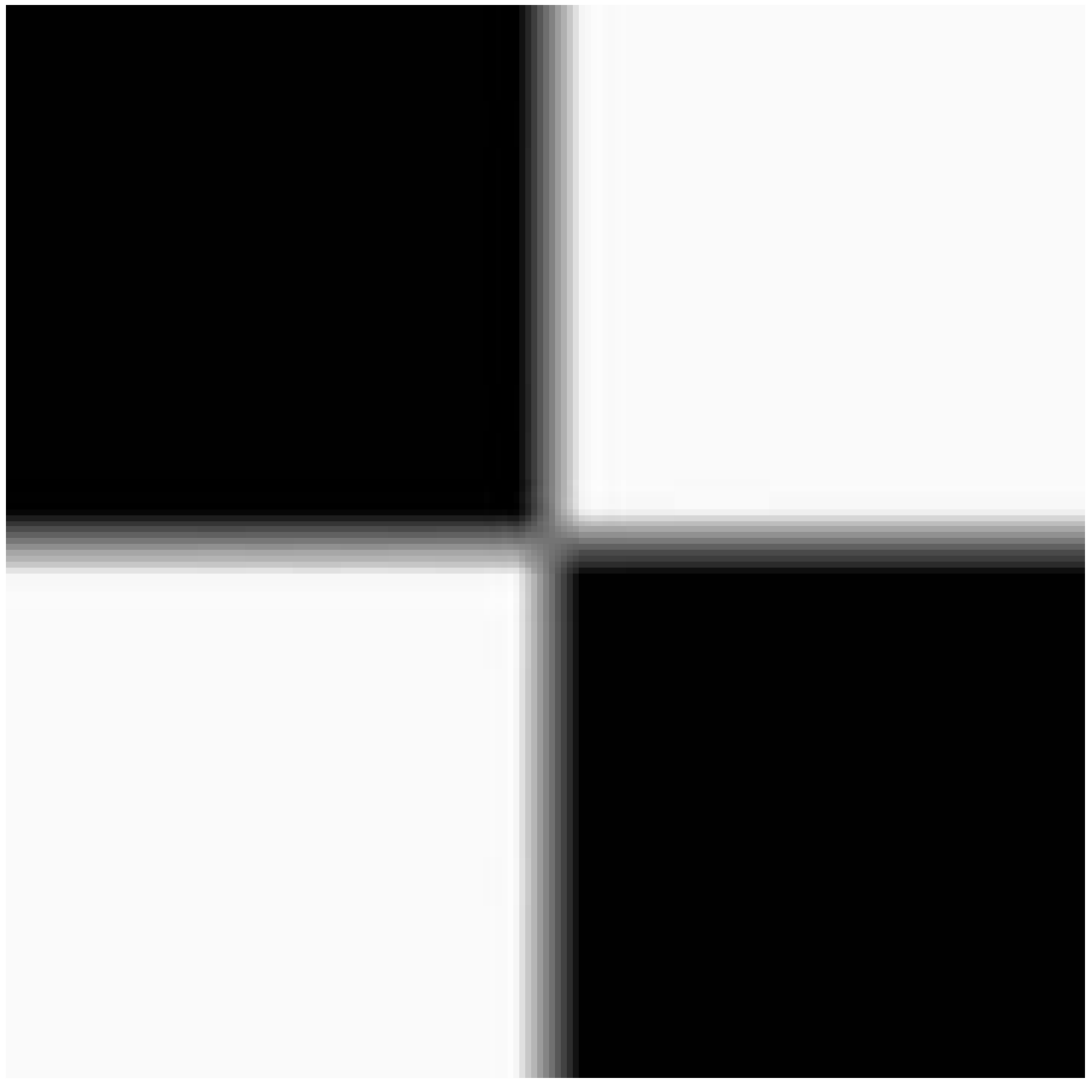
Strange Things Happen At The Edges! (cont…)



Zero Padding

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Strange Things Happen At The Edges! (cont…)



Replicate Edge Pixels

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Strange Things Happen At The Edges! (cont…)

Wrap Around Edge Pixels

Smoothing Spatial Filters

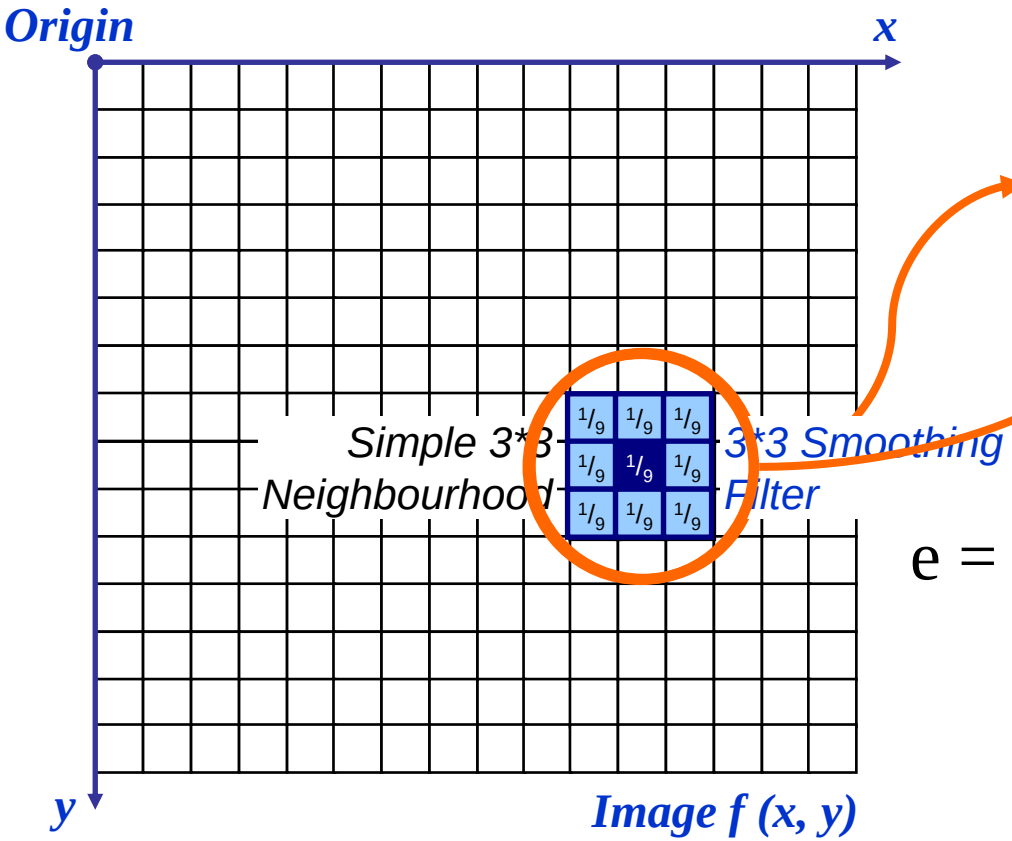
One of the simplest spatial filtering operations we can perform is a smoothing operation

- 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 detail

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

Simple averaging filter

Smoothing Spatial Filtering



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

$$e = \tfrac{1}{9}*106 + \tfrac{1}{9}*104 + \tfrac{1}{9}*100 + \tfrac{1}{9}*108 + \tfrac{1}{9}*99 + \tfrac{1}{9}*98 + \tfrac{1}{9}*95 + \tfrac{1}{9}*90 + \tfrac{1}{9}*85$$

$$= 98.3333$$

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

Image Smoothing Example

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 detail begins to disappear

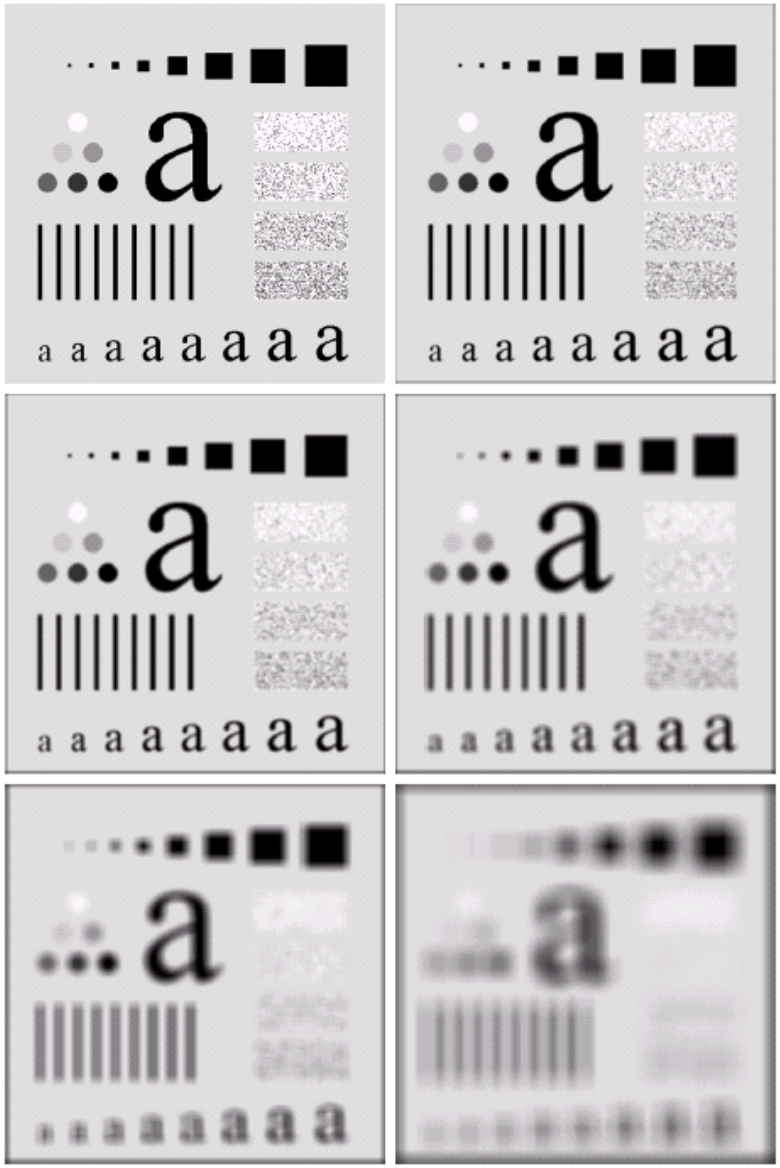


Image Smoothing Example

Original Image

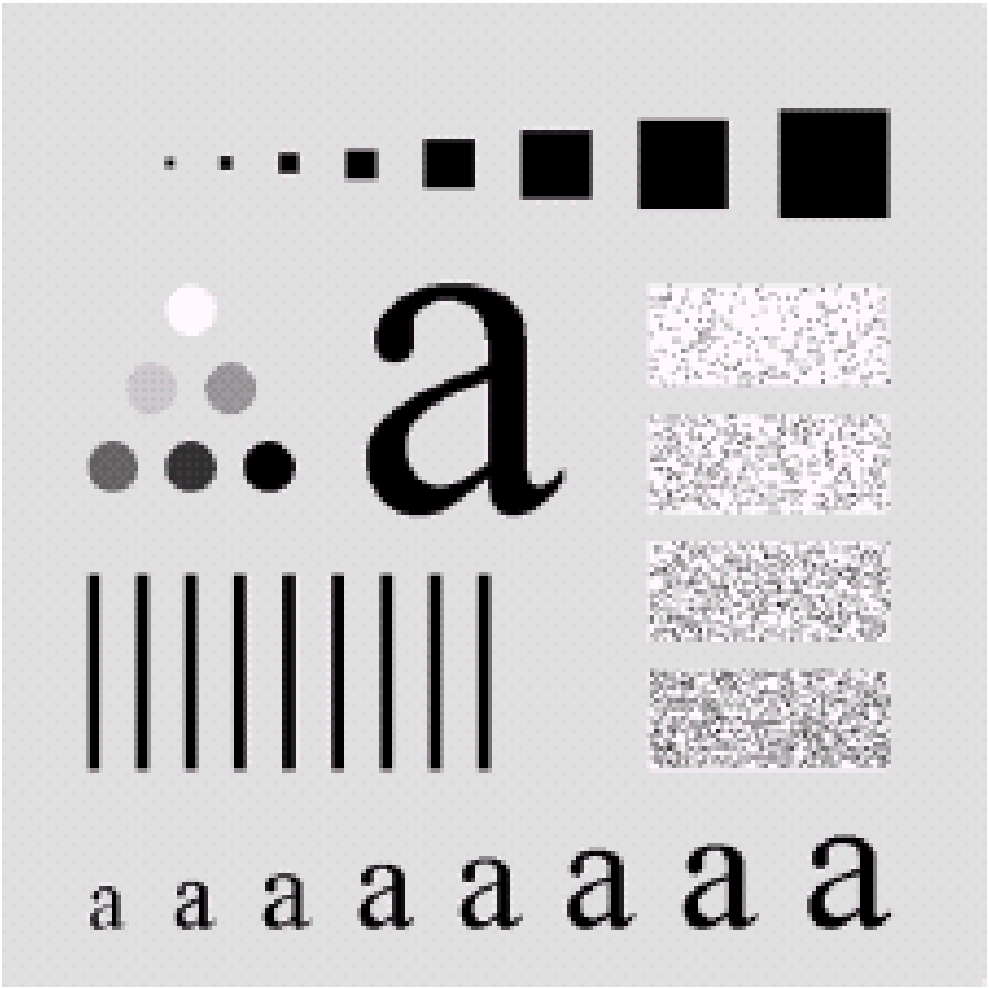


Image Smoothing Example

3x3 Average Filter

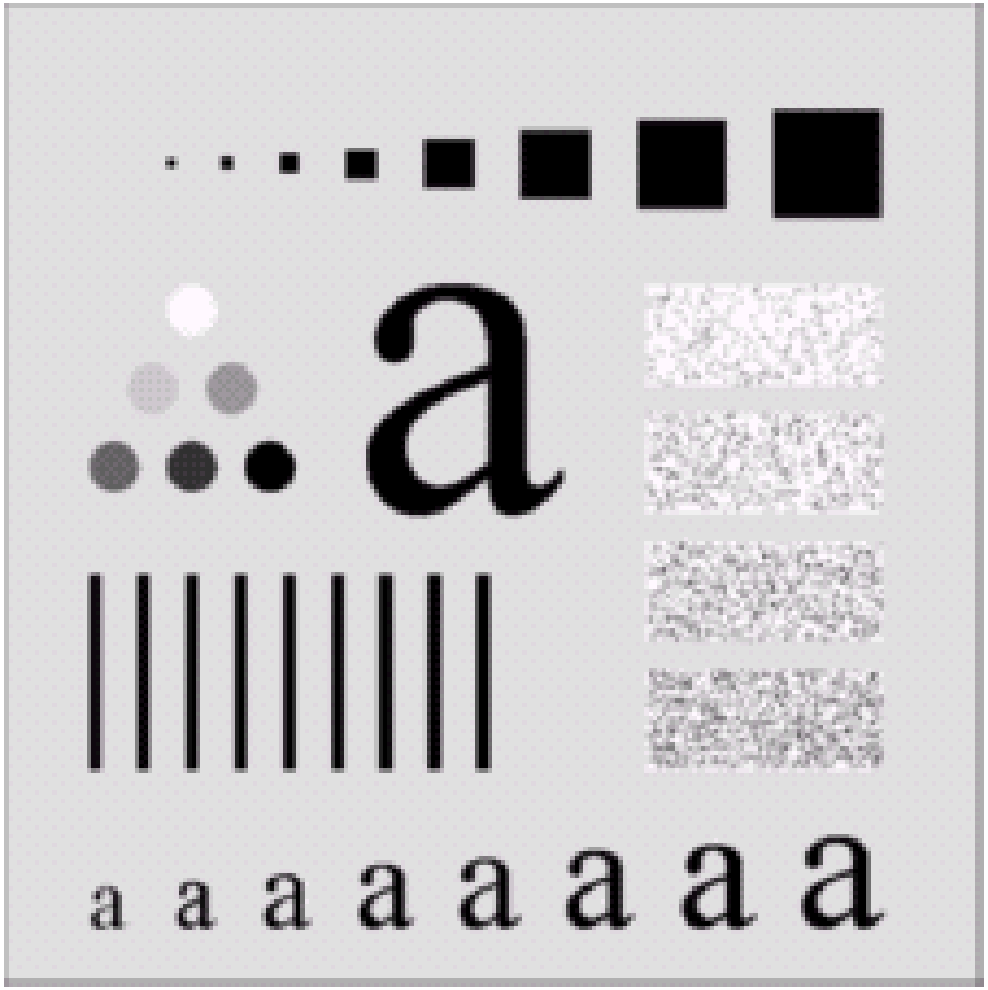


Image Smoothing Example

5x5 Average Filter

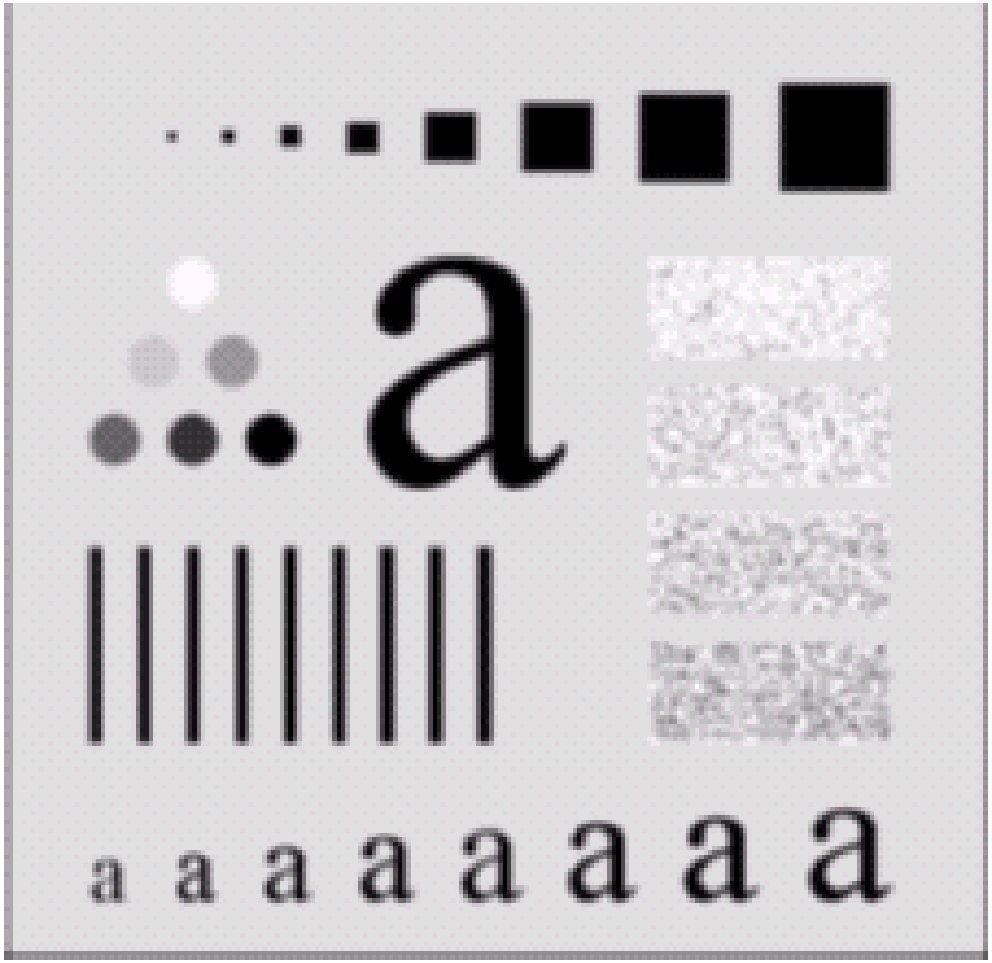


Image Smoothing Example

9x9 Average Filter

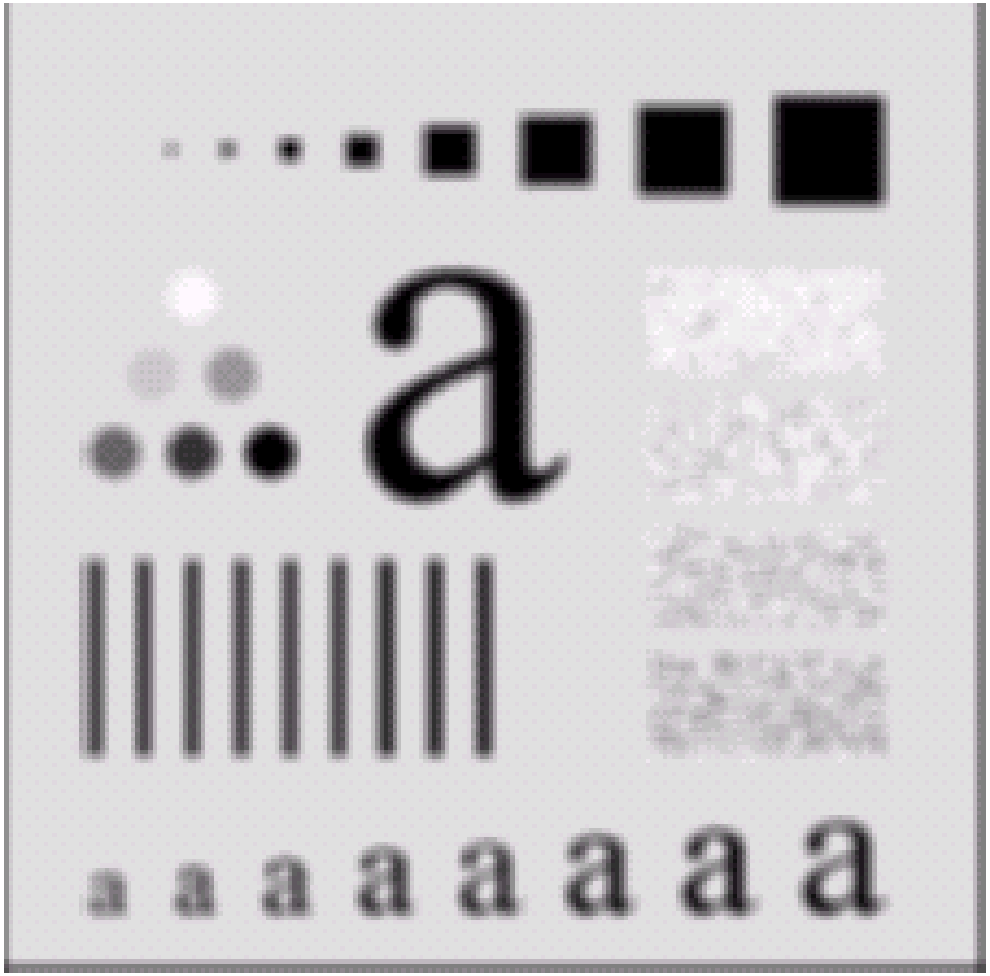


Image Smoothing Example

15x15 Average Filter

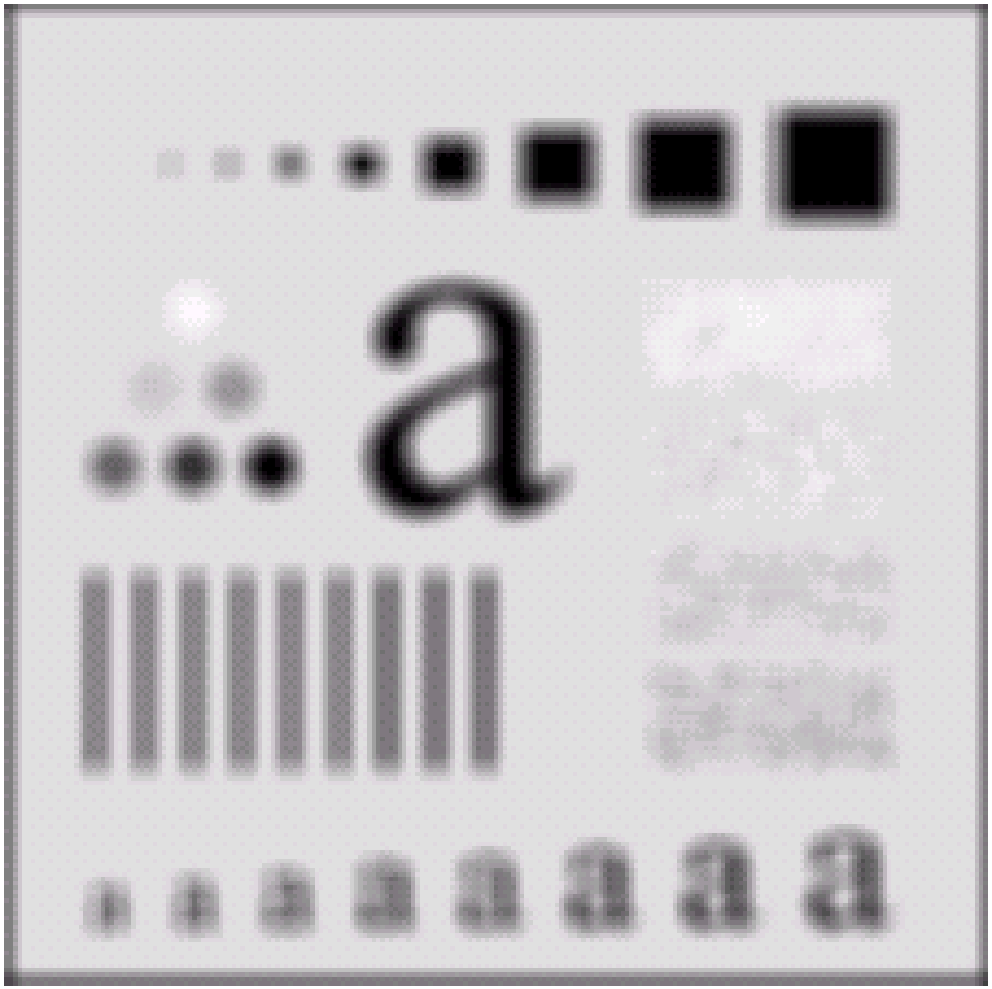
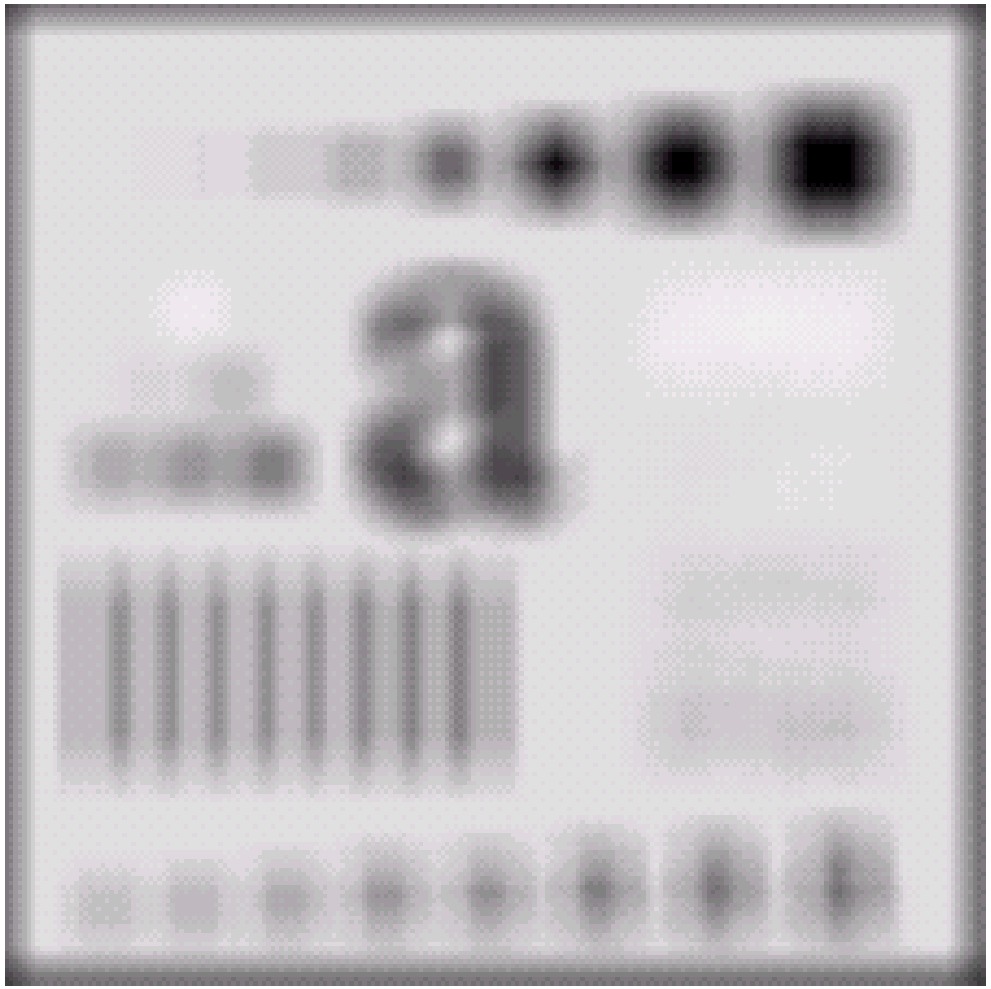


Image Smoothing Example

35x35 Average Filter



Weighted Smoothing Filters

More effective smoothing filters can be generated by allowing different pixels in the neighbourhood different weights in the averaging function

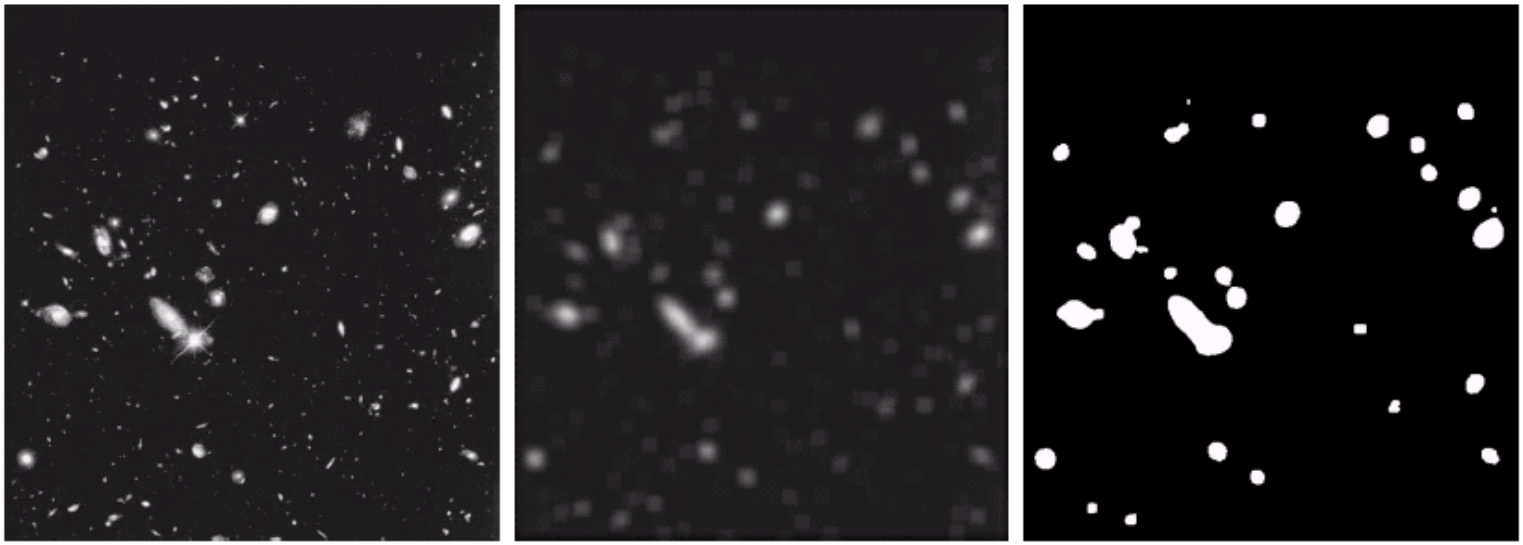
- Pixels closer to the central pixel are more important
- Often referred to as a *weighted averaging*

$^1/_{16}$	$^2/_{16}$	$^1/_{16}$
$^2/_{16}$	$^4/_{16}$	$^2/_{16}$
$^1/_{16}$	$^2/_{16}$	$^1/_{16}$

Weighted averaging filter

Another Smoothing Example

By smoothing the original image we get rid of lots of the finer detail which leaves only the gross features for thresholding



Original Image **Smoothed Image** **Thresholded Image**

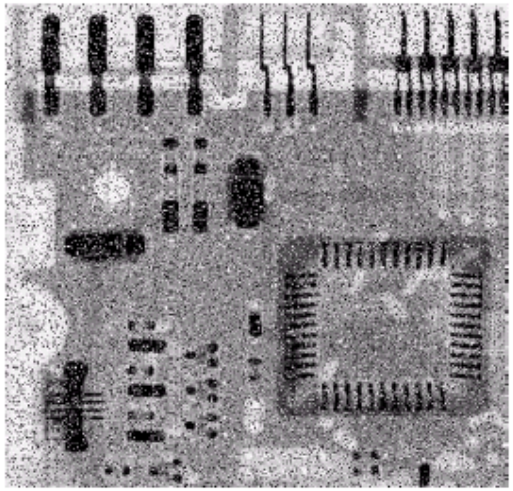
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Simple Non-Linear Operations

Some simple **non linear** neighbourhood operations include:

- **Min (erosion):** Set the pixel value to the minimum in the neighbourhood
- **Max (dilation):** 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 (e.g. from the set [1, 7, 15, 18, 24] 15 is the median). Sometimes the median works better than the average

Averaging Filter Vs. Median Filter Example



Original Image With Noise

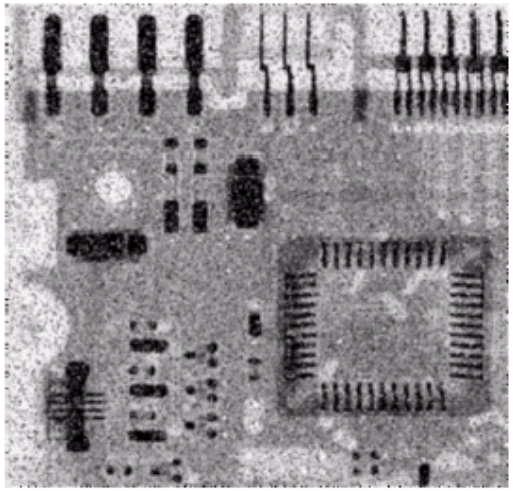


Image After Averaging Filter

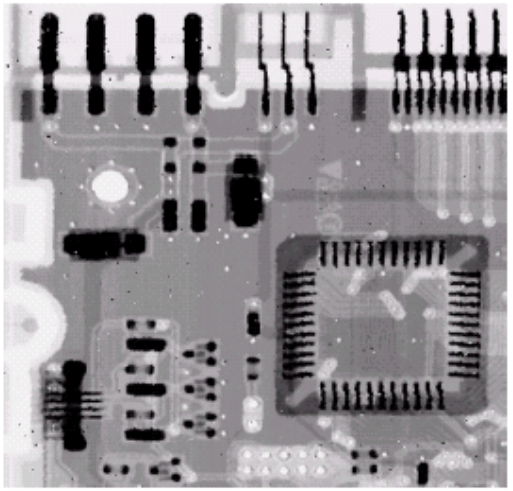


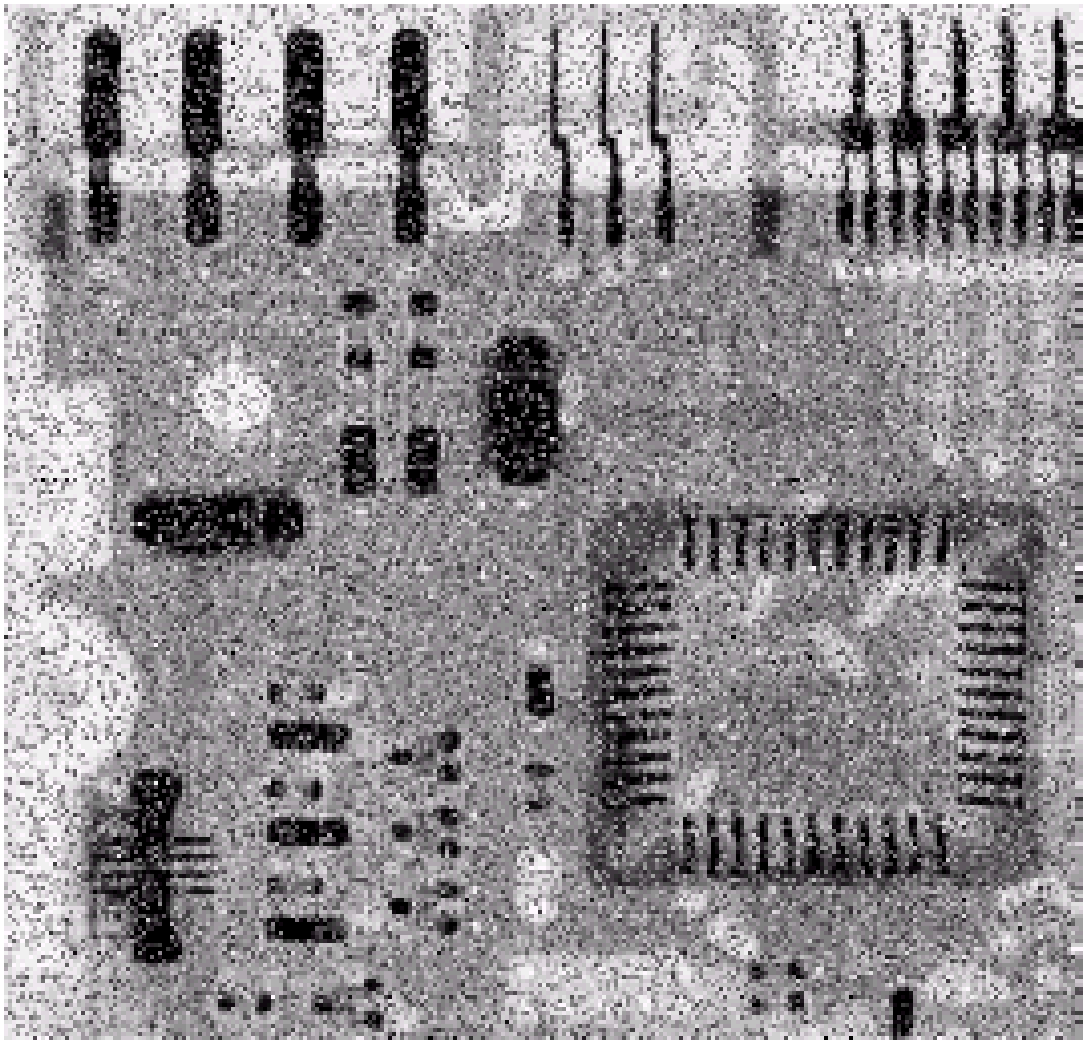
Image After Median Filter

Filtering is often used to remove noise from images

Sometimes a median filter works better than an averaging filter

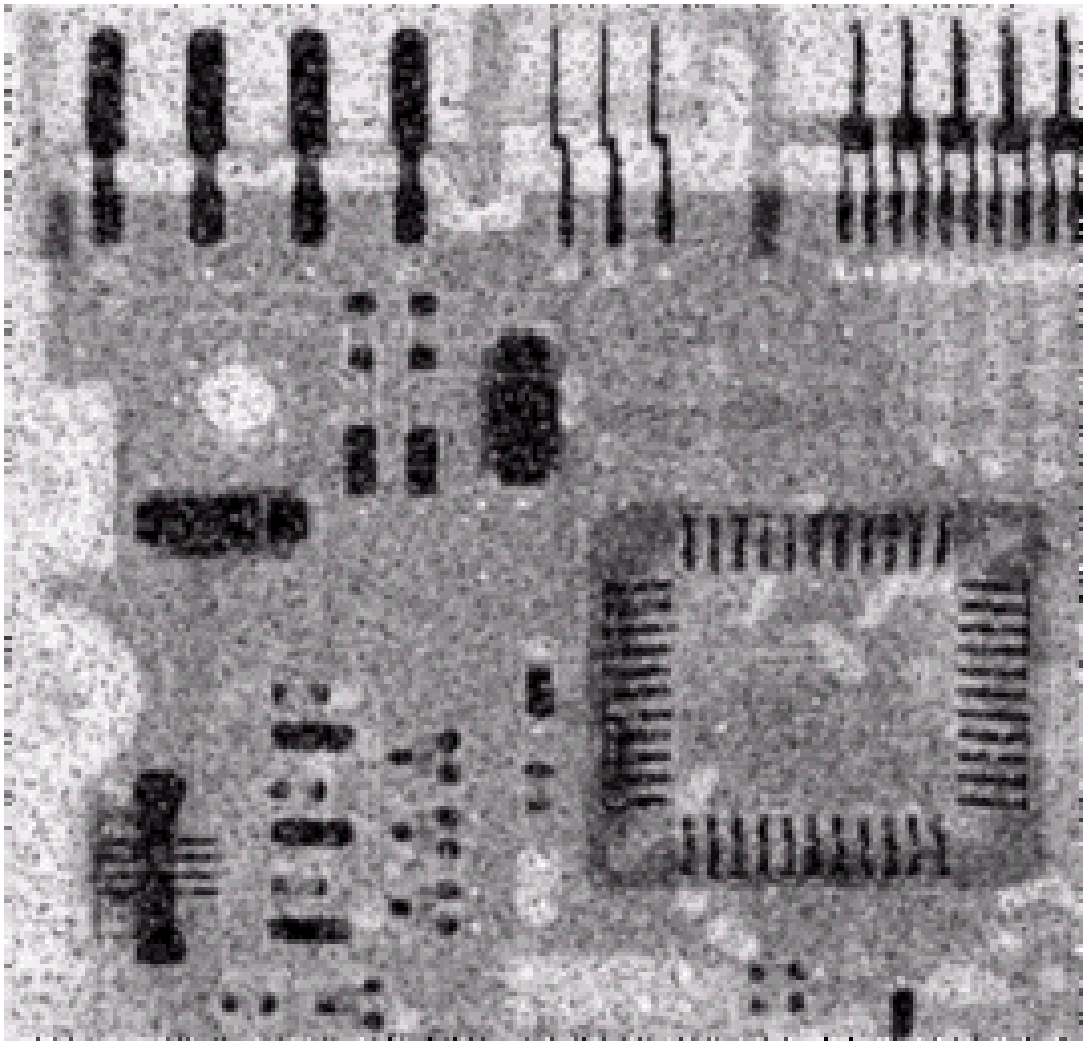
Averaging Filter Vs. Median Filter Example

Original Image



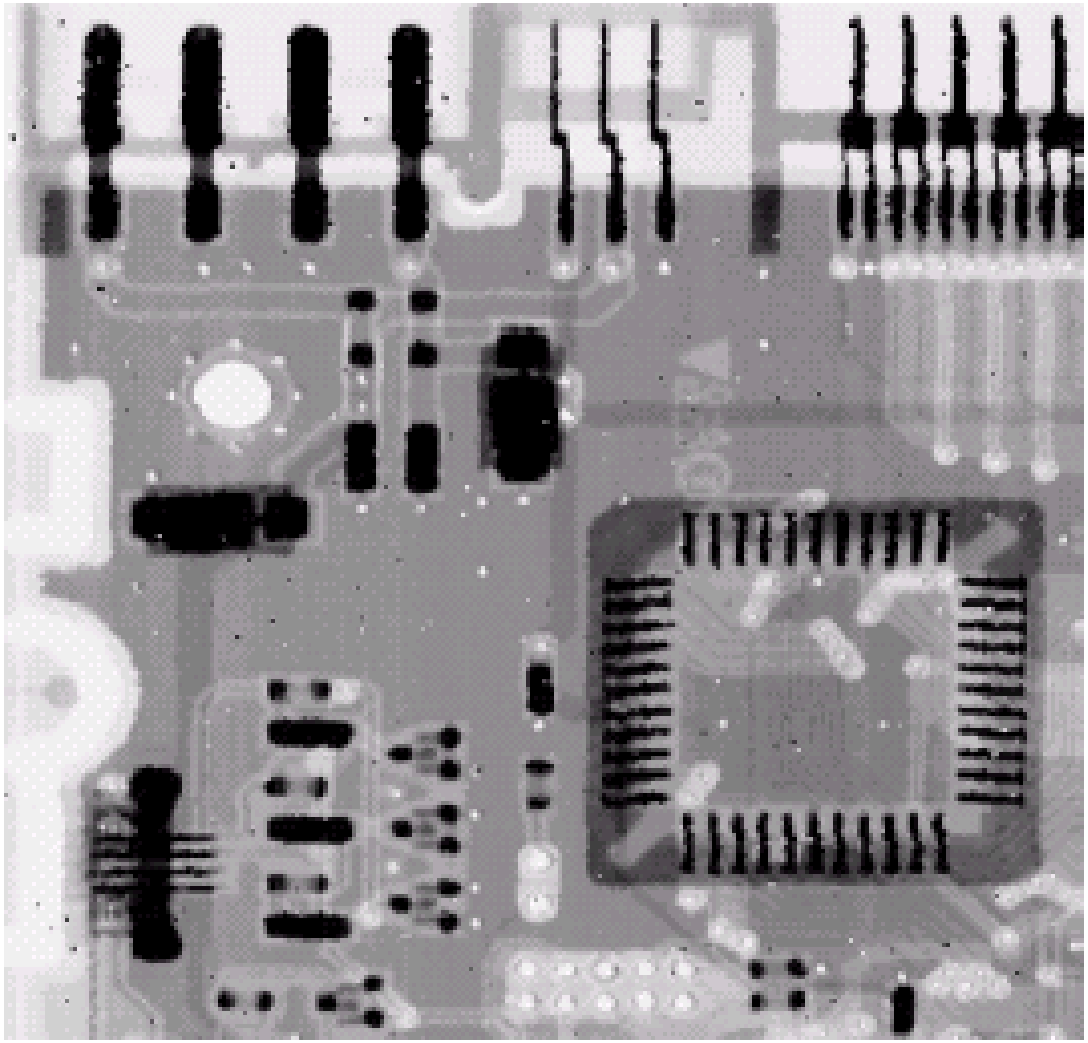
Averaging Filter Vs. Median Filter Example

Average Filter

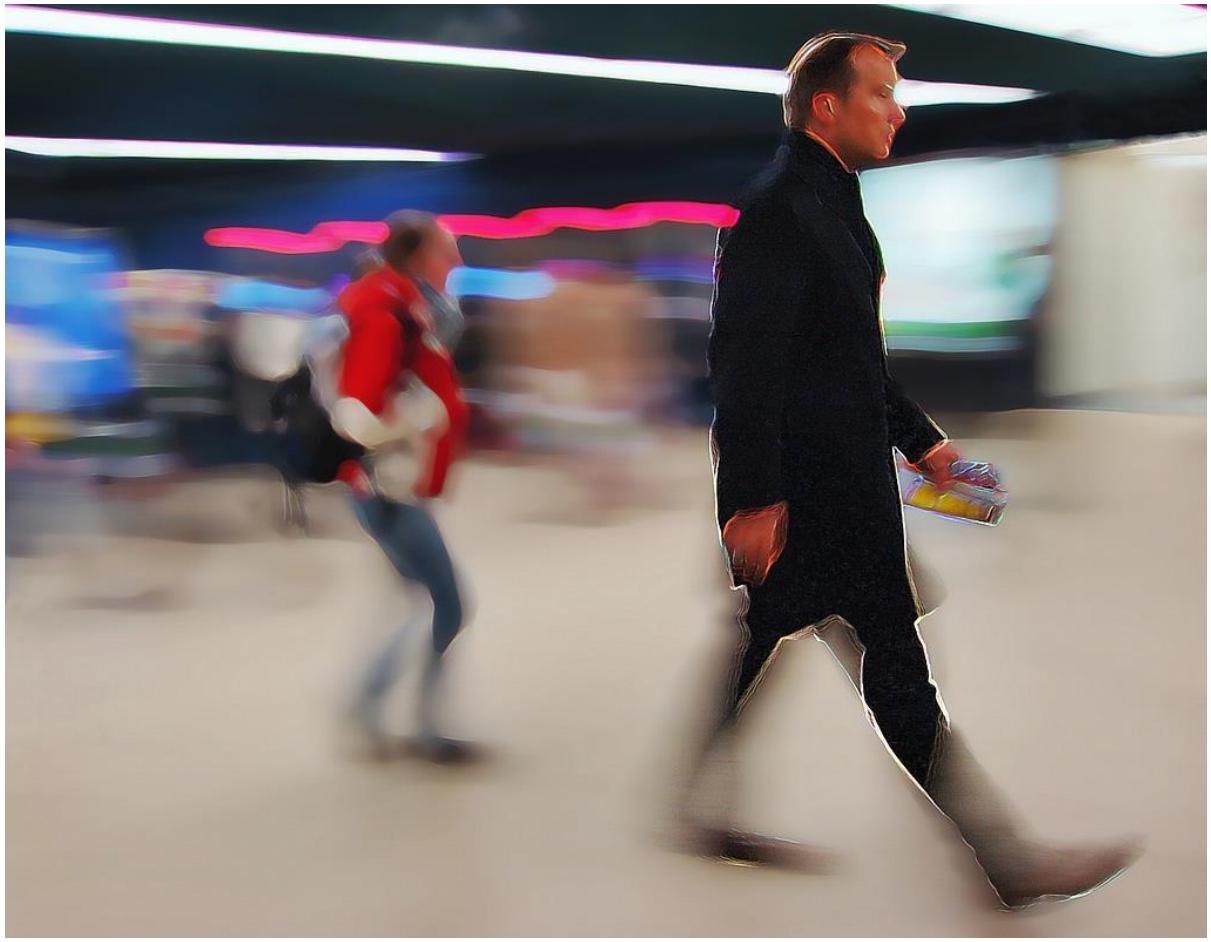


Averaging Filter Vs. Median Filter Example

Median Filter



sharpen !



Unlike smoothing the principal objective of **sharpening** is to **highlight transition in intensity**.

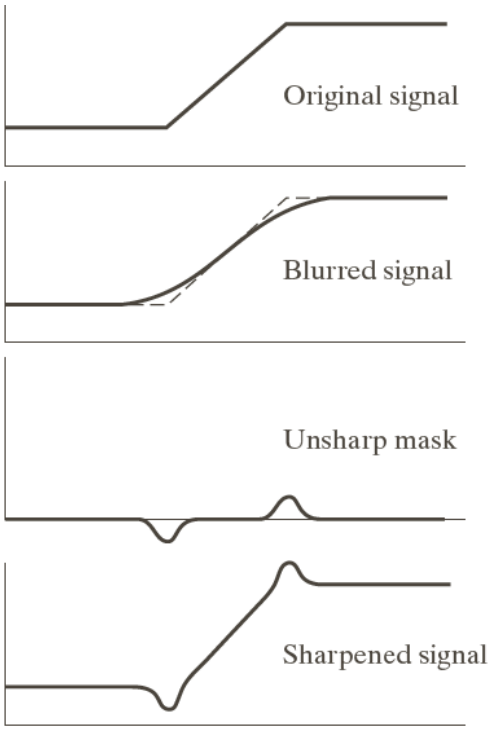
http://flickr.com/photos/t_schnitzlein/87607390/

unsharp masking

- **Unsharp masking** is an image manipulation technique for increasing the apparent sharpness of photographic images.
- The "unsharp" of the name derives from the fact that the technique uses a blurred, or "unsharp", positive to create a "mask" of the original image. The unsharped mask is then combined with the negative, creating a resulting image sharper than the original.

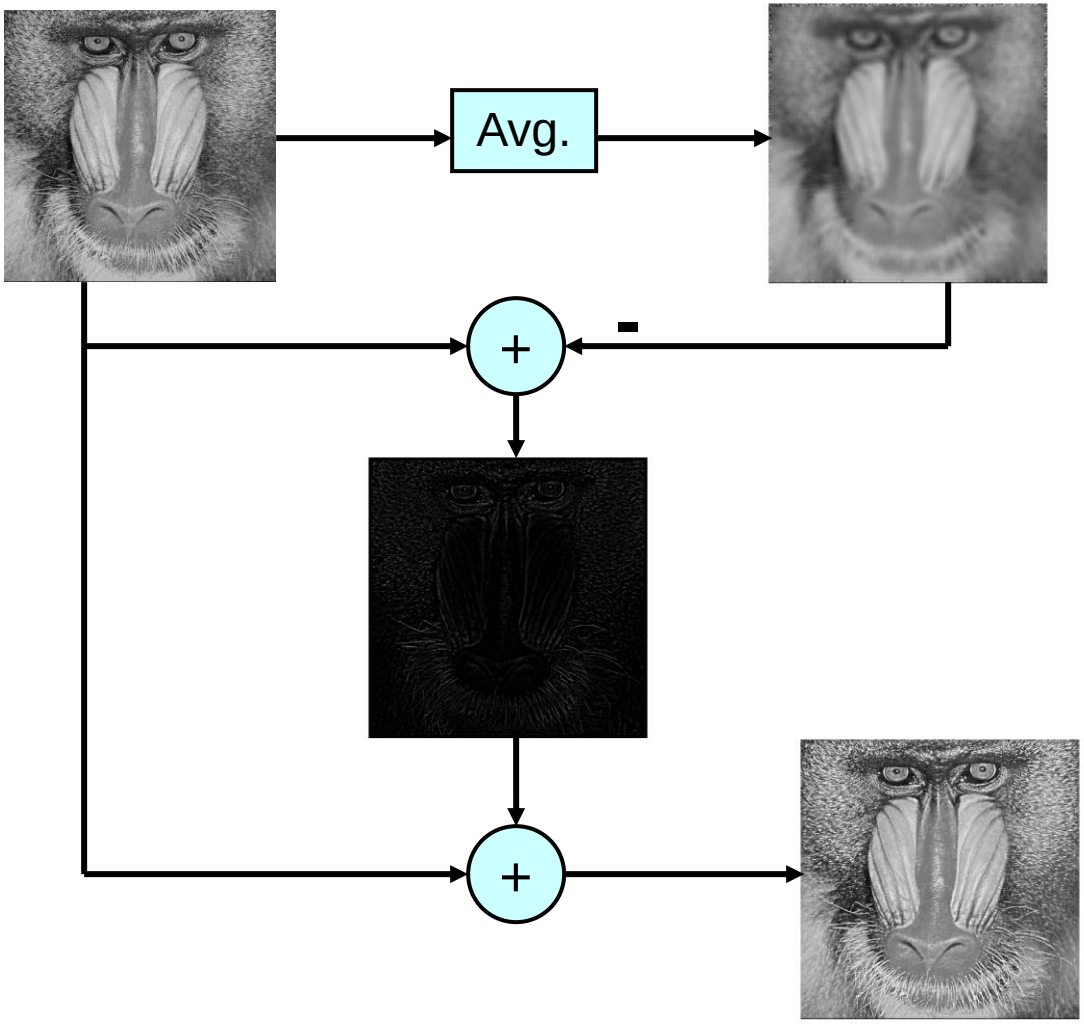
a
b
c
d

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).



- Steps
  1. Blur the image
  2. Subtract the blurred version from the original (this is called the *mask*)
  3. Add the "mask" to the original

high-boost filtering



Let $f'(x)$ is the blurred image.

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

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

Unsharp mask:

high-boost with A=1

unsharp mask example

The Moon 25th August 2005 1:19 GMT

ADD TO FAVES BLOG THIS ALL SIZES

Waning Gibbous Moon, 1:19 GMT Location Edinburgh, Scotland, UK
20.6 days old. Mirror Image!

The Moon with unsharp mask applied

ADD TO FAVES BLOG THIS ALL SIZES

Similar to last nights picture but with some unsharp masking and turned into a greyscale picture. Do you think it helps?

Waning Gibbous Moon, 1:19 GMT Location Edinburgh, Scotland, UK
20.6 days old. Mirror Image! 25th August 2005 1:19 GMT

Summary

In this lecture we have looked at the idea of spatial filtering and in particular:

- Neighbourhood operations
- The filtering process
- Smoothing filters
- Dealing with problems at image edges when using filtering
- Unsharp Masking

Next time we will look at sharpening filters using image derivatives as well as more on filtering and image enhancement