SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

19ECE351 – IMAGE PROCESSING AND COMPUTER VISION

III B.E. ECE / V SEMESTER

UNIT 2 – IMAGE ENHANCEMENT AND RESTORATION

TOPIC – Image Enhancement

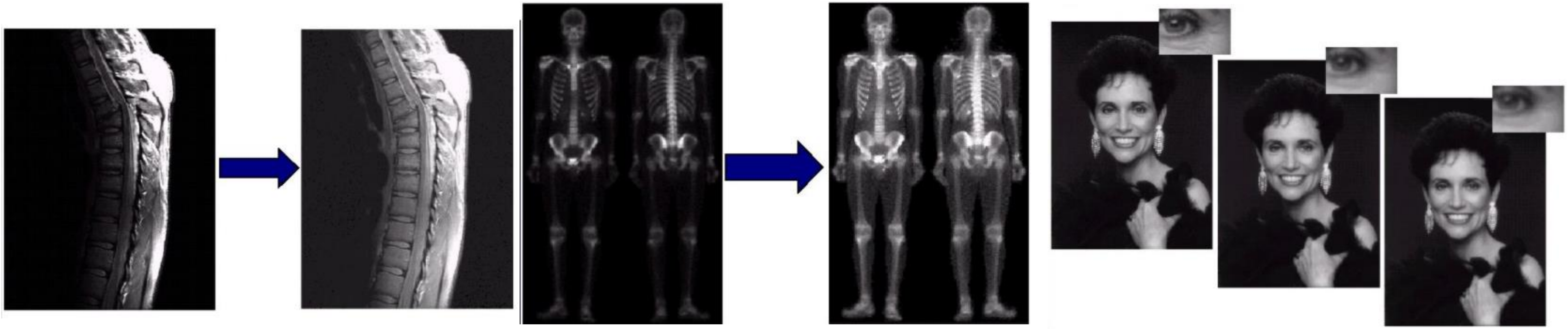
What Is Image Enhancement?

Image enhancement is the process of making images more useful.

The reasons for doing this include:

- Highlighting interesting detail in images,
- Removing noise from images,
- Making images more visually appealing.

Image Enhancement Examples

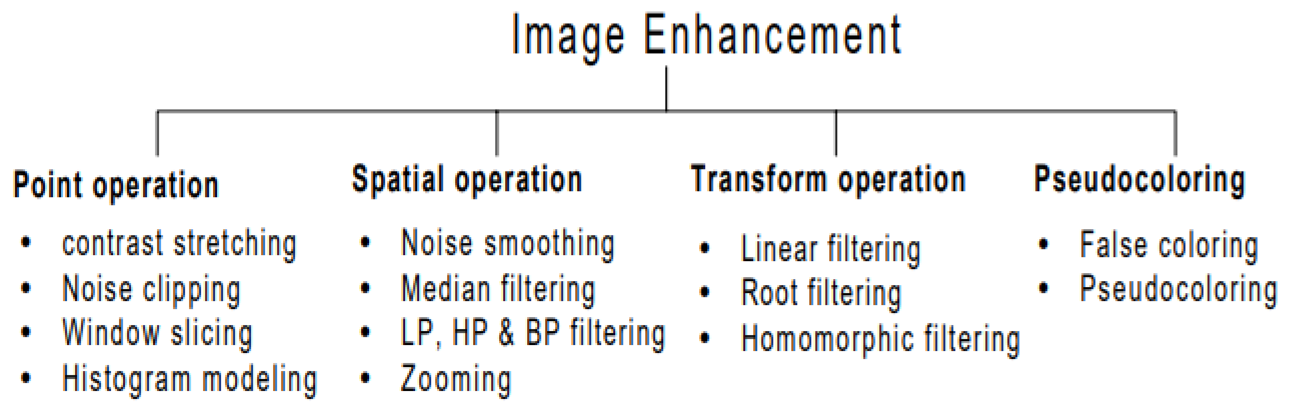


There are two broad categories of image enhancement techniques

- Spatial domain techniques
  - Direct manipulation of image pixels
- Frequency domain techniques
  - Manipulation of Fourier transform or wavelet transform of an image

1. <u>Introduction</u>

- The principal objective of image enhancement is to process a given image so that the result is more suitable than the original image for a specific application.
- It accentuates or sharpens image features such as edges, boundaries, or contrast to make a graphic display more helpful for display and analysis.
- The enhancement doesn't increase the inherent information content of the data, but it increases the dynamic range of the chosen features so that they can be detected easily.



- The greatest difficulty in image enhancement is quantifying the criterion for enhancement and, therefore, a large number of image enhancement techniques are empirical and require interactive procedures to obtain satisfactory results.
- Image enhancement methods can be based on either spatial or frequency domain techniques.

Spatial domain enhancement methods:

- Spatial domain techniques are performed to the image plane itself and they are based on direct manipulation of pixels in an image.
- The operation can be formulated as $g(x,y) = \mathbf{T}[f(x,y)]$, where g is the output, f is the input image and **T** is an operation on f defined over some neighborhood of (x,y).
- According to the operations on the image pixels, it can be further divided into 2 categories: *Point operations* and *spatial operations* (including linear and non-linear operations).

Frequency domain enhancement methods:

- These methods enhance an image $f(x,y)$ by convoluting the image with a linear, position invariant operator.
- The 2D convolution is performed in frequency domain with DFT.

Spatial domain: $g(x,y)=f(x,y)*h(x,y)$
Frequency domain: $G(w_1,w_2)=F(w_1,w_2)H(w_1,w_2)$

What is Spatial Filtering?

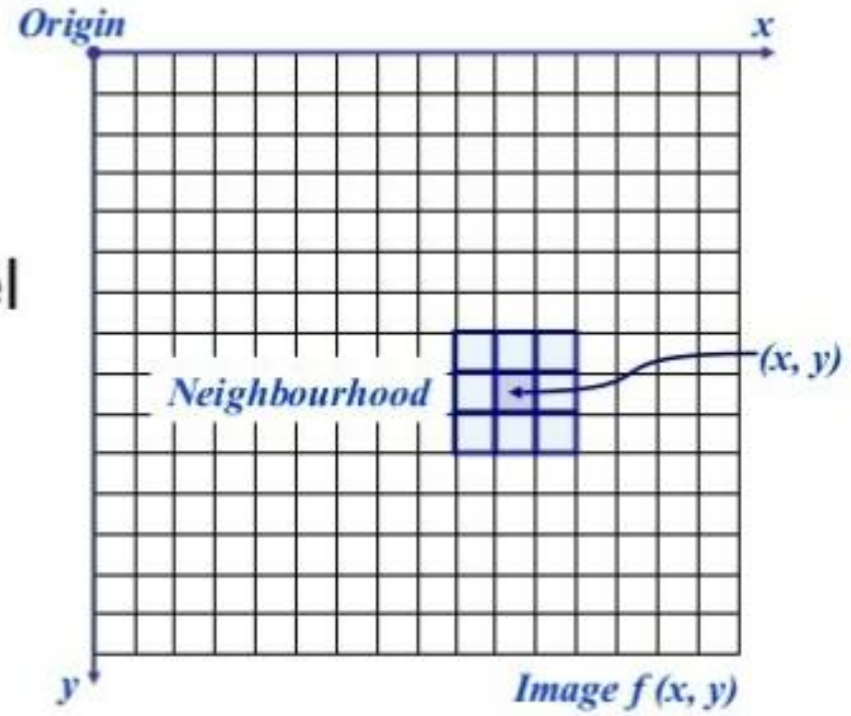
Spatial Filtering technique is used directly on pixels of an image. Mask is usually considered to be added in size so that it has specific center pixel. This mask is moved on the image such that the center of the mask traverses all image pixels.

Neighborhood Operations

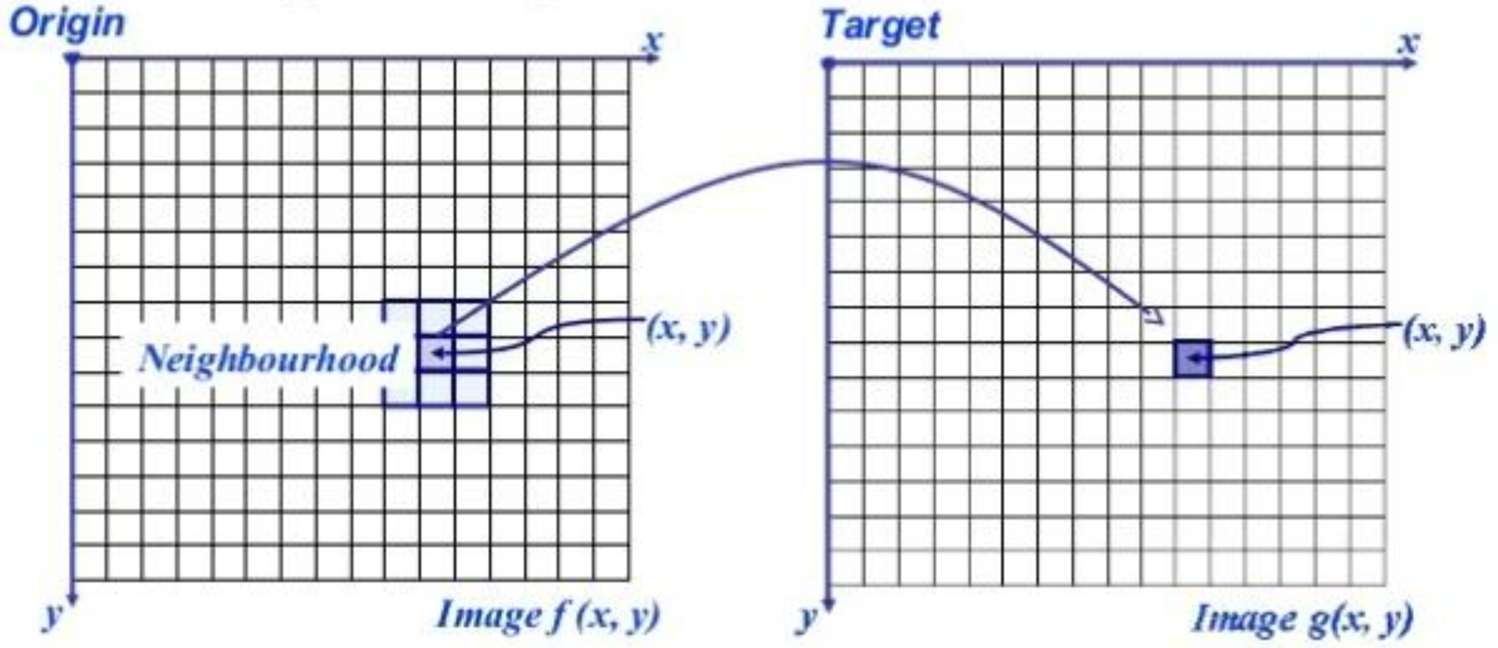
Neighbourhood operations simply operate on a larger neighbourhood of pixels than point operations

Neighbourhoods are mostly a rectangle around a central pixel

Any size rectangle and any shape filter are possible



For each pixel in the origin image, the outcome is written on the same location at the target image.



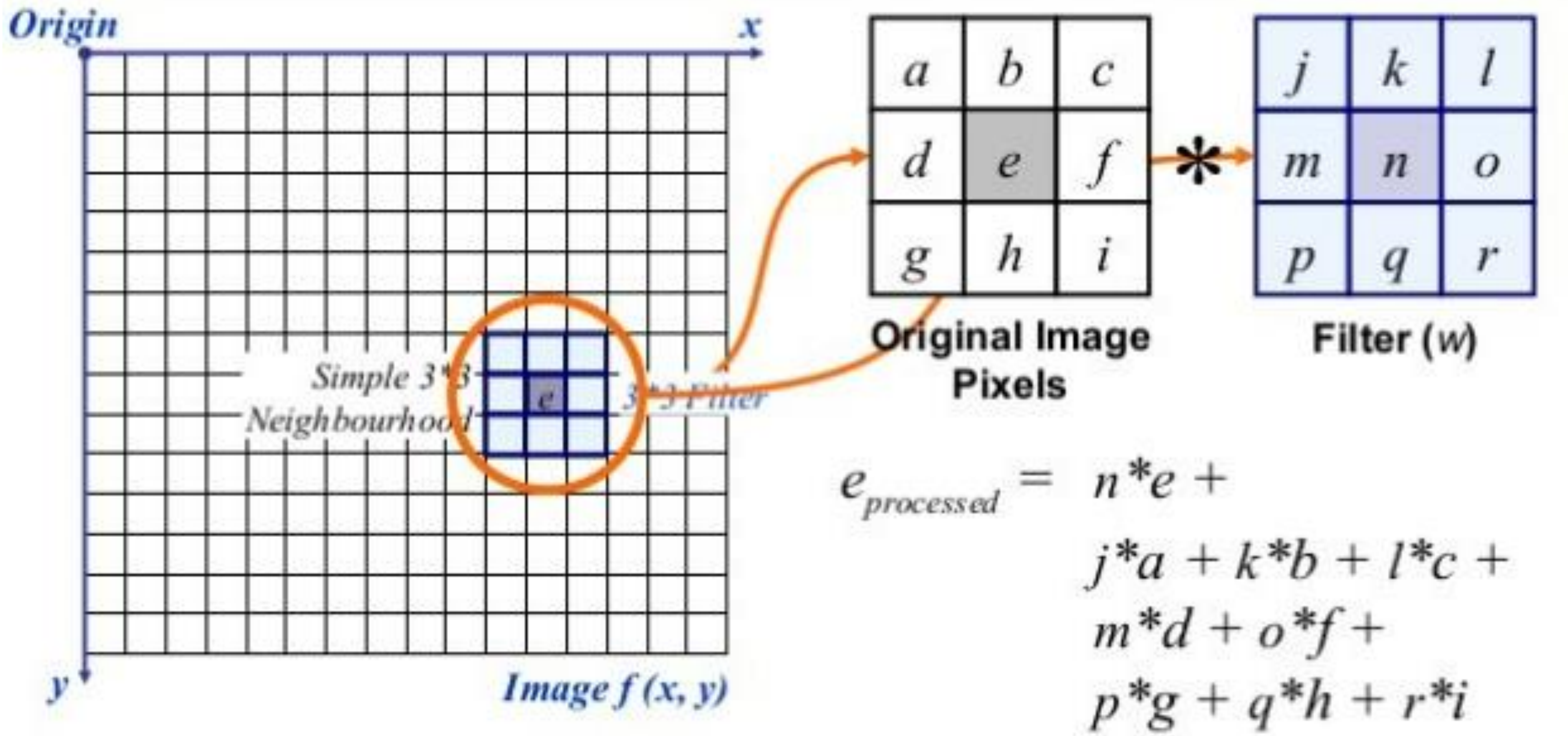
Simple Neighborhood Operations

Simple neighbourhood operations example:

– **Min:** Set the pixel value to the minimum in the neighbourhood

– **Max:** Set the pixel value to the maximum in the neighbourhood

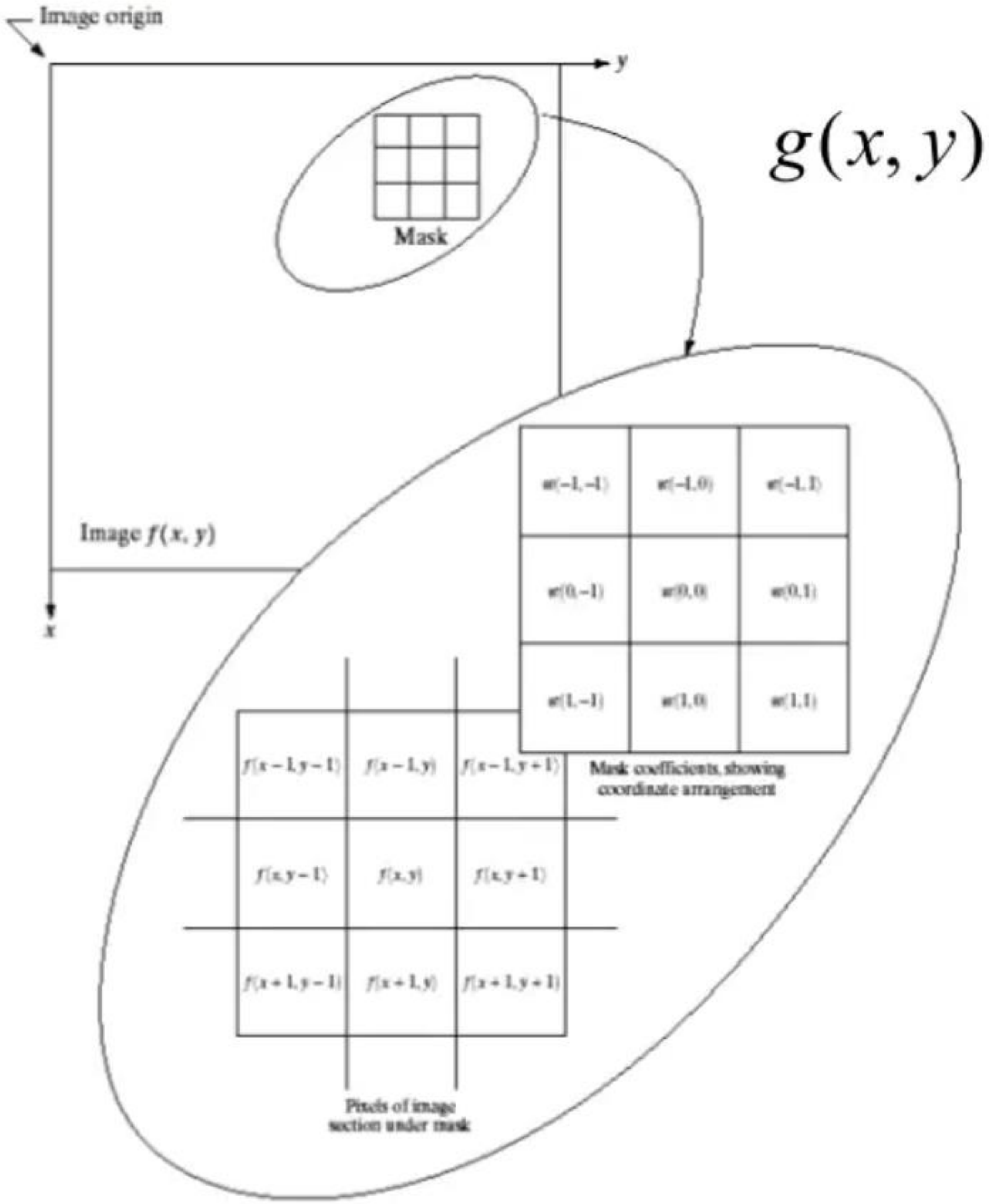
Spatial Filtering Process



$$e_{processed} = n*e + j*a + k*b + l*c + m*d + o*f + p*g + q*h + r*i$$

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

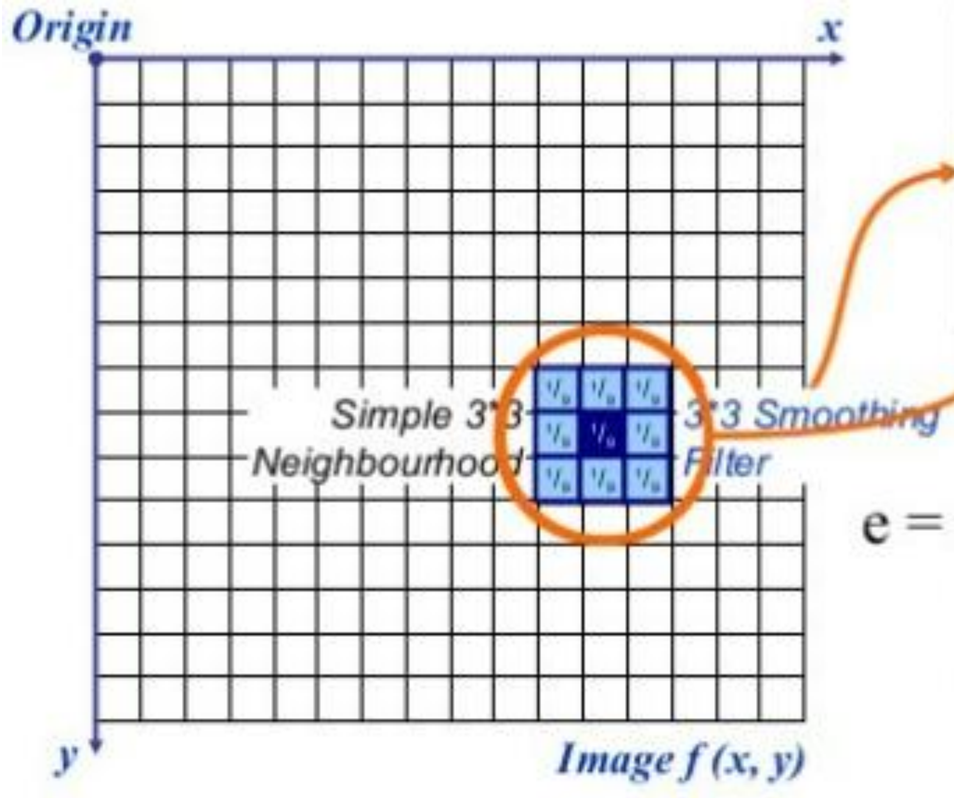
Smoothing Spatial Filters

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



Original Image Pixels:

104	100	108
99	106	98
95	90	85

*

Filter:

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

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

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

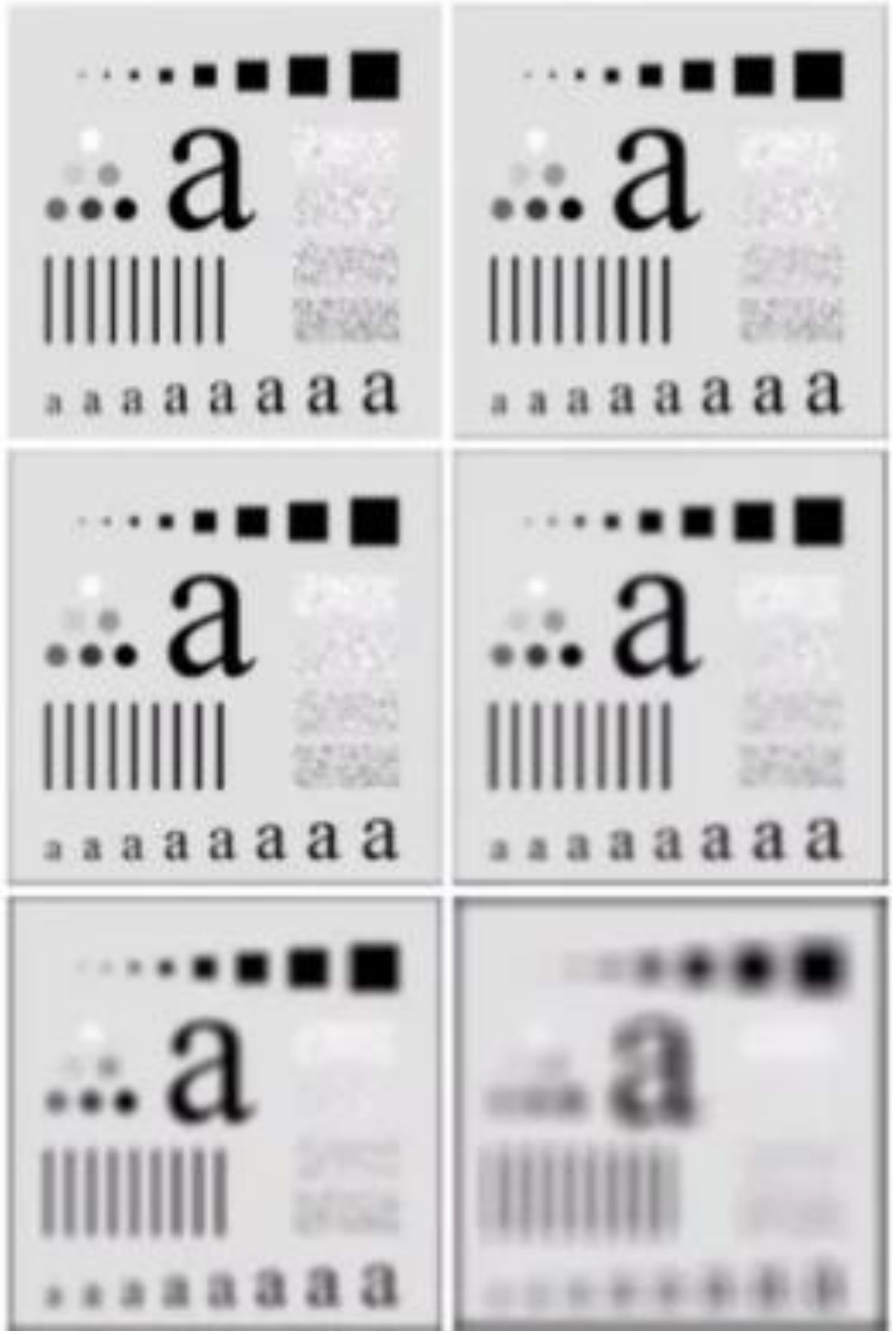
Image Smoothing Example

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



Weighted Smoothing Filters

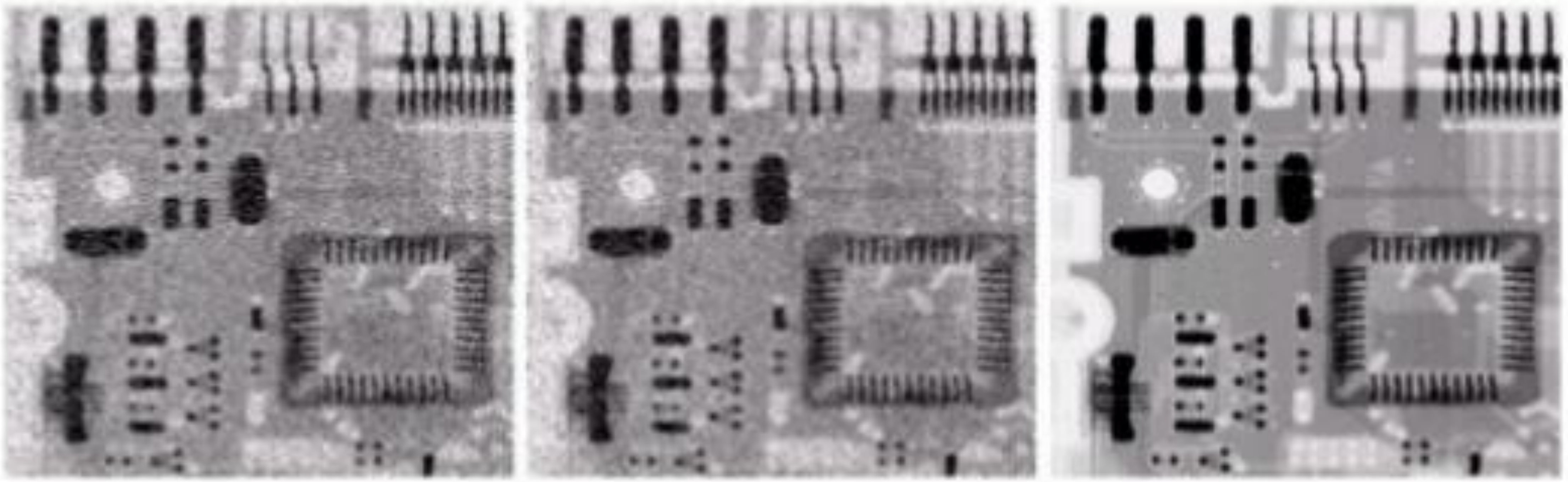
More effective smoothing filters can be generated by allowing different pixels in the neighbourhood different weights in the averaging function

- 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

Average Filter Vs. Median Filter



Original Image
With Noise

Image After
Averaging Filter

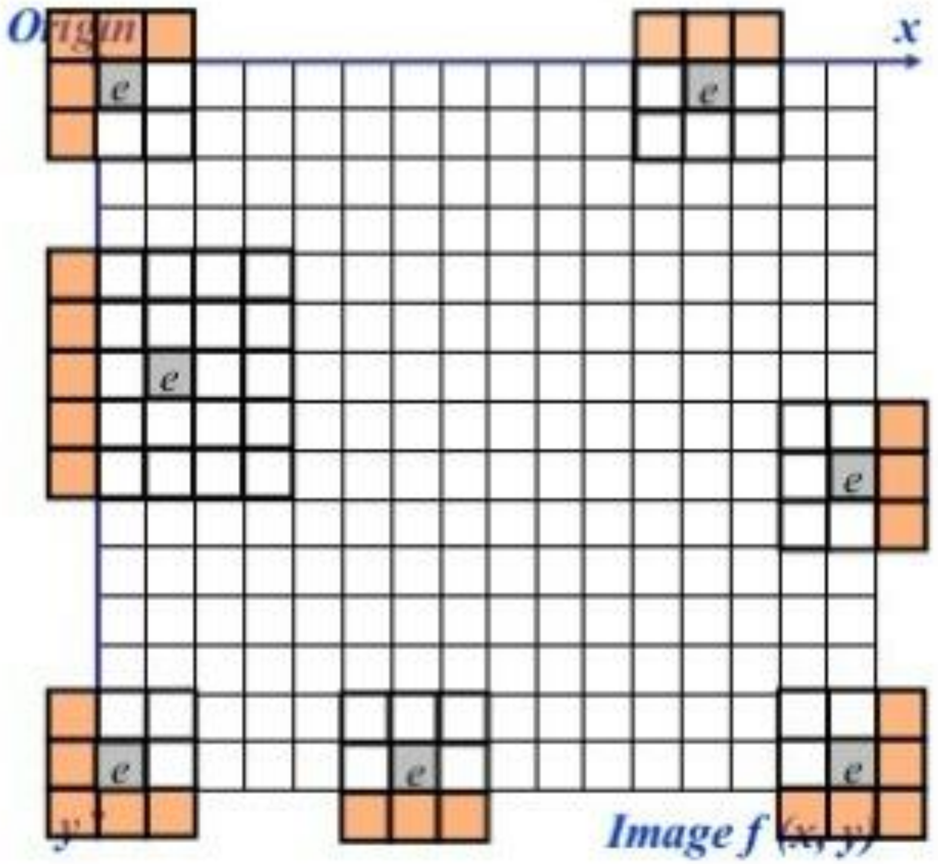
Image After
Median Filter

Filtering is often used to remove noise from images

Sometimes a median filter works better than an averaging filter

Strange things happen at the Edges

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



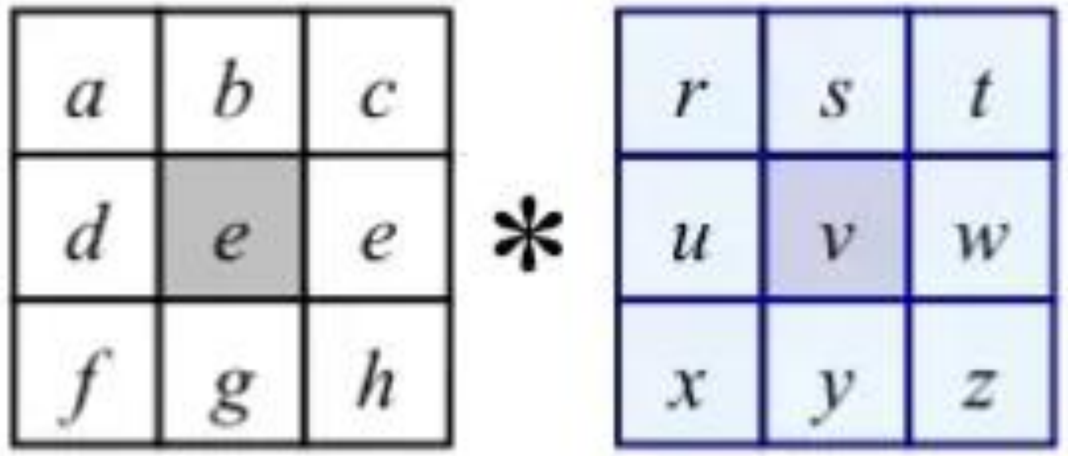
There are a few approaches to dealing with missing edge pixels:

- 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

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



Original Image Pixels

Filter

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

For symmetric filters it makes no difference

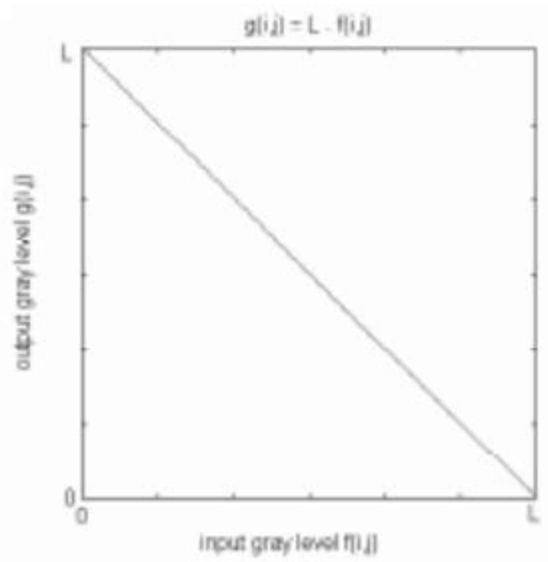
2. Enhancement by point processing

- These processing methods are based only on the intensity of single pixels.

2.1 Simple intensity transformation:

(a). *Image negatives*:

- Negatives of digital images are useful in numerous applications, such as displaying medical images and photographing a screen with monochrome positive film with the idea of using the resulting negatives as normal slides.
- Transform function T : g(x,y)=L-f(x,y), where L is the max. intensity.

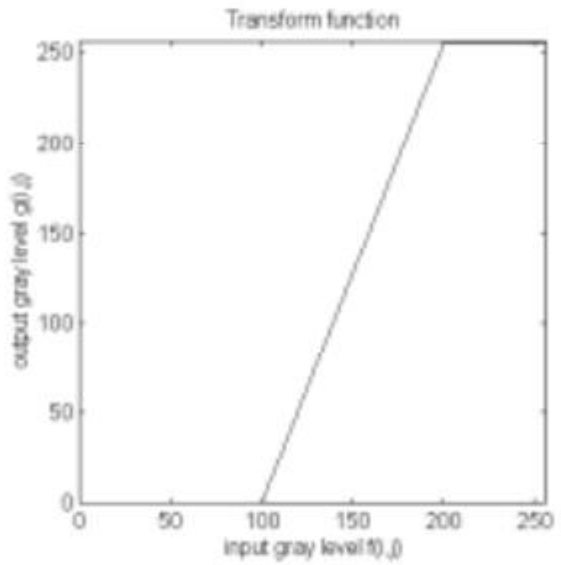


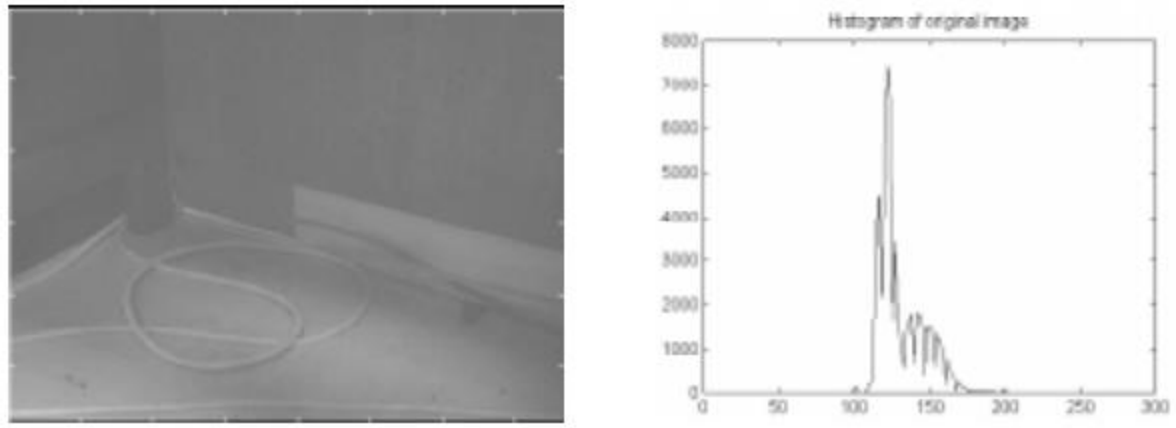
Original

Negative

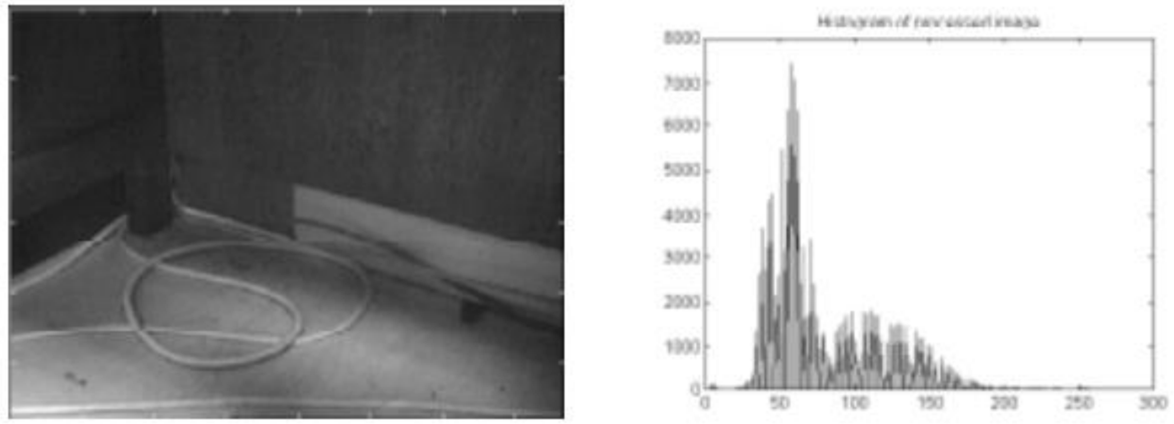
(b). *Contrast stretching*

- Low-contrast images can result from poor illumination, lack of dynamic range in the image sensor, or even wrong setting of a lens aperture during image acquisition.
- The idea behind contrast stretching is to increase the dynamic range of the gray levels in the image being processed.





Original



Processed image

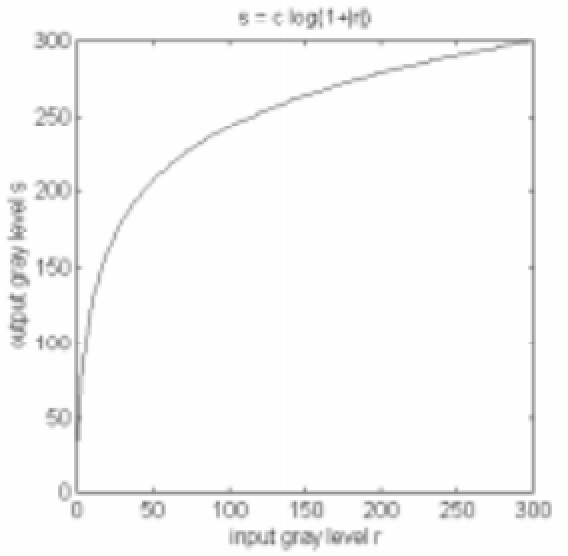
- Special case: If $r_1=r_2=0$, $s_1=0$ and $s_2=L-1$, then it is actually a thresholding that creates a binary images.

(c). *Compression of dynamic range*

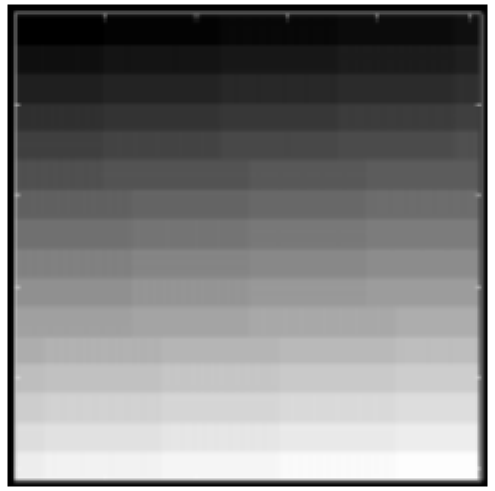
- Sometimes the dynamic range of a processed image far exceeds the capability of the display device, in which case only the brightest parts of the images are visible on the display screen.
- An effective way to compress the dynamic range of pixel values is to perform the following intensity transformation function:

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

where c is a scaling constant, and the logarithm function performs the desired compression.



Transform function



Original

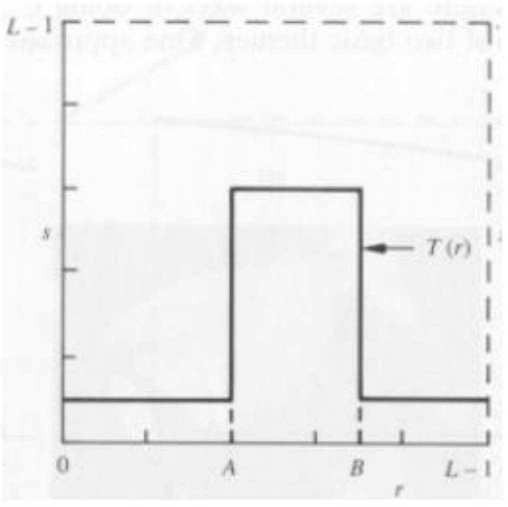
Processed output

(d) *Gray-level slicing*

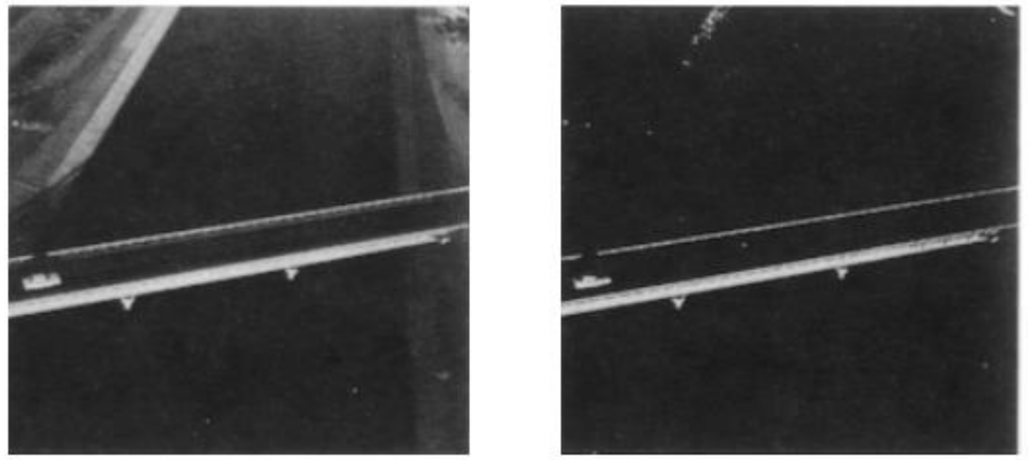
- Highlighting a specific range of gray levels in an image often is desired. Applications include enhancing features such as masses of water in satellite imagery and enhancing flaws in x-ray images.

- Example 1:

  A transformation function that highlights a range [A,B] of intensities while diminishing all others to a constant.



(a)

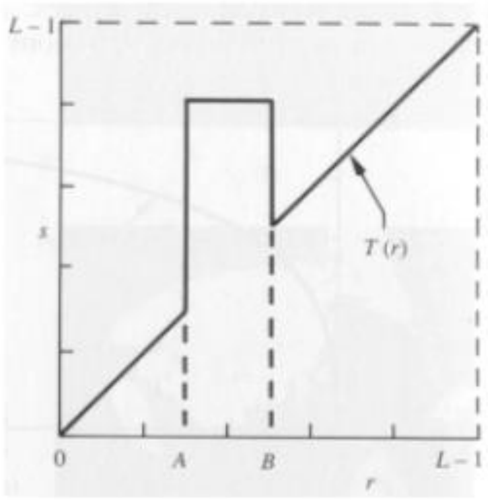


(b) (c)

Fig 1. (a) Transfer function, (b) Original image, (c) Processing output.

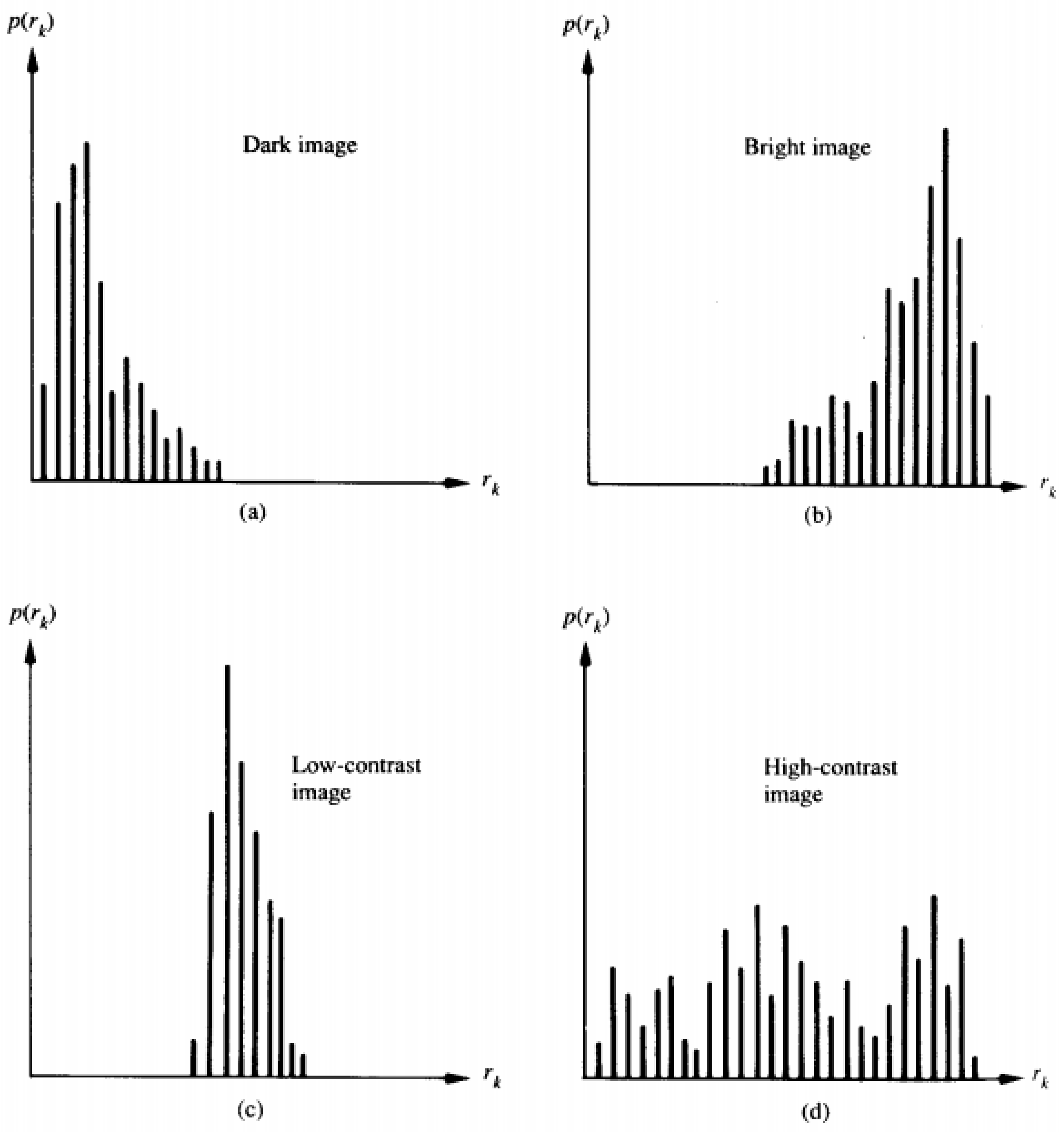
- Example 2:

  A transformation function that highlights a range [A,B] of intensities but preserves all others.



2.2 Histogram processing:

- The histogram of a digital image with gray levels in the range [0,L-1] is a discrete function $p(r_k)=n_k/n$, where r_k is the kth gray level, n_k is the number of pixels in the image with that gray level, n is the total number of pixels in the image, and k=0,1..L-1.
- $P(r_k)$ gives an estimate of the probability of occurrence of gray level r_k.
- The shape of the histogram of an image gives us useful information about the possibility for contrast enhancement.
- A histogram of a narrow shape indicates little dynamic range and thus corresponds to an image having low contrast.



(a) *Histogram equalization*

- The objective is to map an input image to an output image such that its histogram. is uniform after the mapping.
- Let r represent the gray levels in the image to be enhanced and s is the enhanced output with a transformation of the form $s=\mathbf{T}(r)$.
- Assumption:
  1. $\mathbf{T}(r)$ is single-valued and monotonically increasing in the interval [0,1], which preserves the order from black to white in the gray scale.
  2. $0 \le T(r) \le 1$ for $0 \le r \le 1$, which guarantees the mapping is consistent with the allowed range of pixel values.
- If $P_r(r)$ and $\mathbf{T}(r)$ are known and $\mathbf{T}^{-1}(s)$ satisfies condition (a), the pdf of the transformed gray levels is $P_s(s) = P_r(r)\left.\frac{dr}{ds}\right|_{r=T^{-1}(s)}$
- If $s = T(r) = \int_0^r P_r(w)dw$ for $0 \le r \le 1$, then we have $\frac{ds}{dr} = P_r(r)$ and hence $P_s(s) = 1$ for $0 \le s \le 1$.
- Using a transformation function equal to the cumulative distribution of r produces an image whose gray levels have a uniform density, which implies an increase in the dynamic range of the pixels.
- In order to be useful for digital image processing, eqns. should be formulated in discrete form:

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

- A plot of $P_r(r_k)$ versus r_k is actually a histogram, and the technique used for obtaining a uniform histogram is known as histogram equalization or histogram linearization.

sns
INSTITUTIONS

Thank you!