ICT2403 – Graphics and Image Processing

Intensity Transformation and Spatial Filtering – I
Spatial Filtering and Smoothing

K.A.S.H. Kulathilake

Ph.D., M.Phil., MCS, B.Sc.(Hons.)IT, SEDA(UK)

Senior Lecturer

Department of Computing

Rajarata University of Sri Lanka

kule@as.rjt.ac.lk

Learning Outcomes

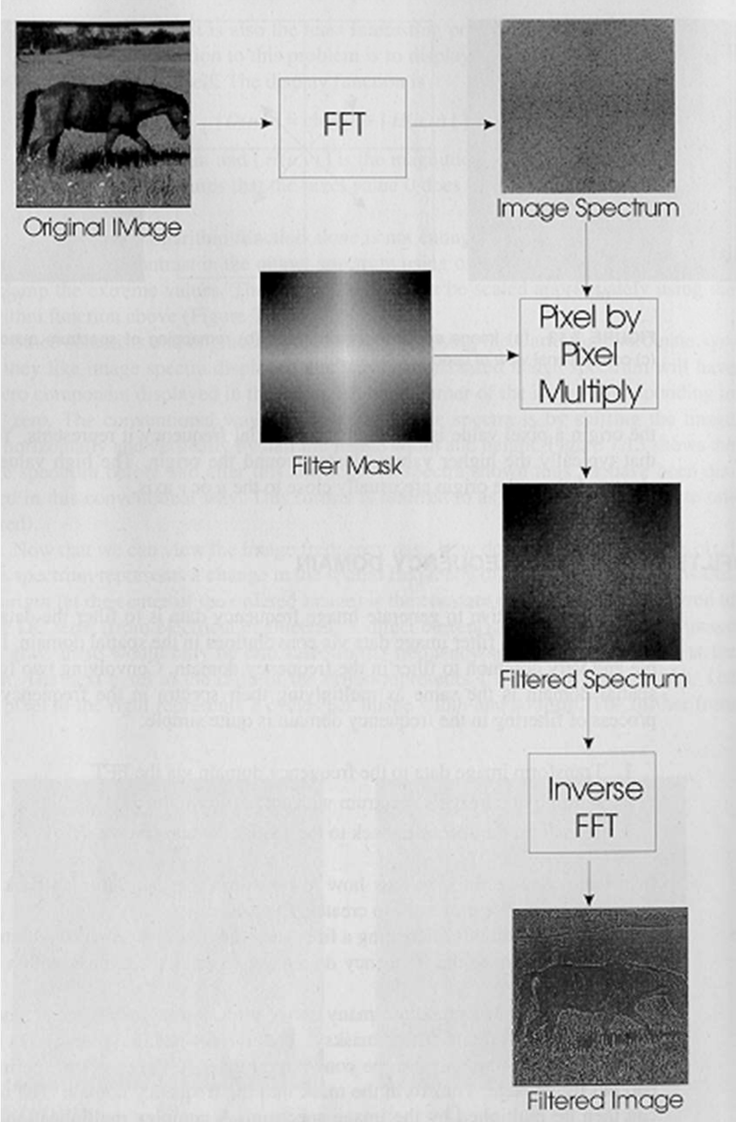
- At the end of this lecture, you should be able to;
  - describe the fundamentals of spatial filtering.
  - generating spatial filter masks.
  - identify smoothing via linear filters and non linear filters.
  - apply smoothing techniques for problem solving.

Filters in Image Processing

- Filter can accept or reject certain frequency components.
- In image processing filters are mainly used to suppress either the high frequencies in the image, i.e. smoothing the image, or the low frequencies, i.e. enhancing/ sharpening or detecting edges in the image.
- An image can be filtered either in the **frequency** or in the **spatial** domain.

Fundamentals of Frequency Domain Filtering

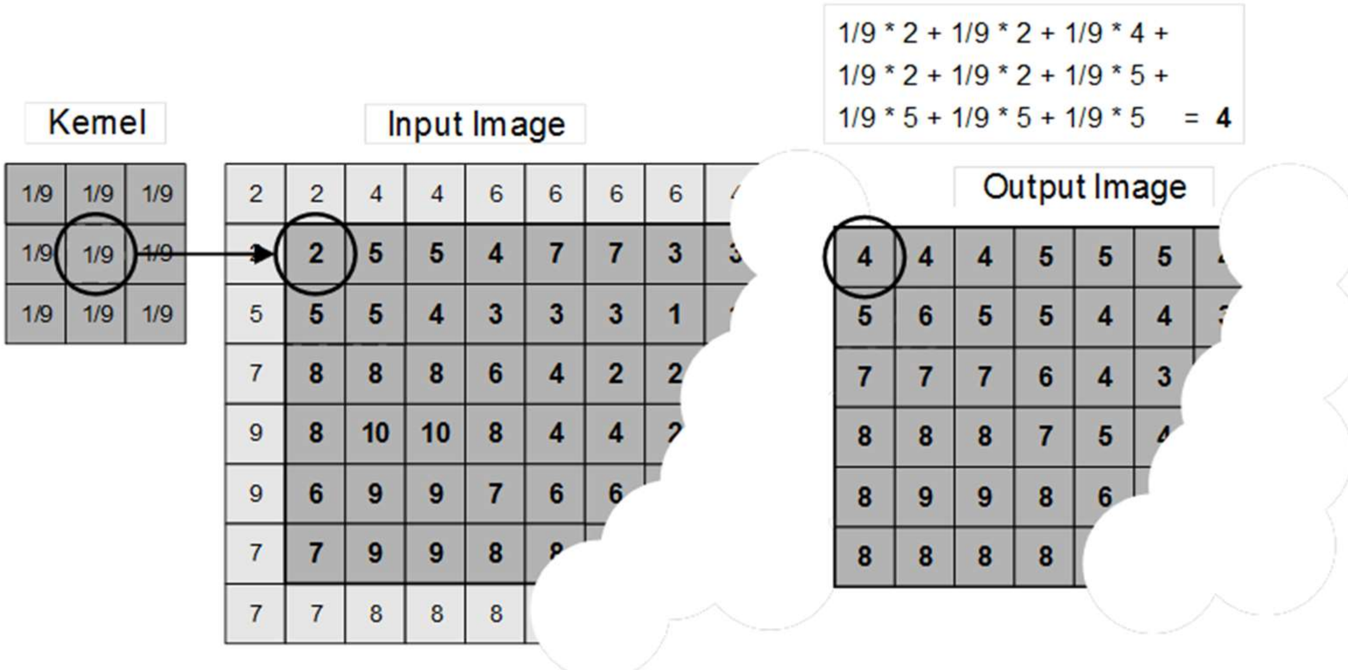
- Frequency domain filtering involves ;
  - transforming the image into the frequency domain,
  - multiplying it with the frequency filter function and
  - re-transforming the result into the spatial domain.
- The filter function is shaped so as to attenuate some frequencies and enhance others.
- For example, a simple lowpass function is 1 for frequencies smaller than the cut-off frequency and 0 for all others.



Fundamentals of Spatial Filtering

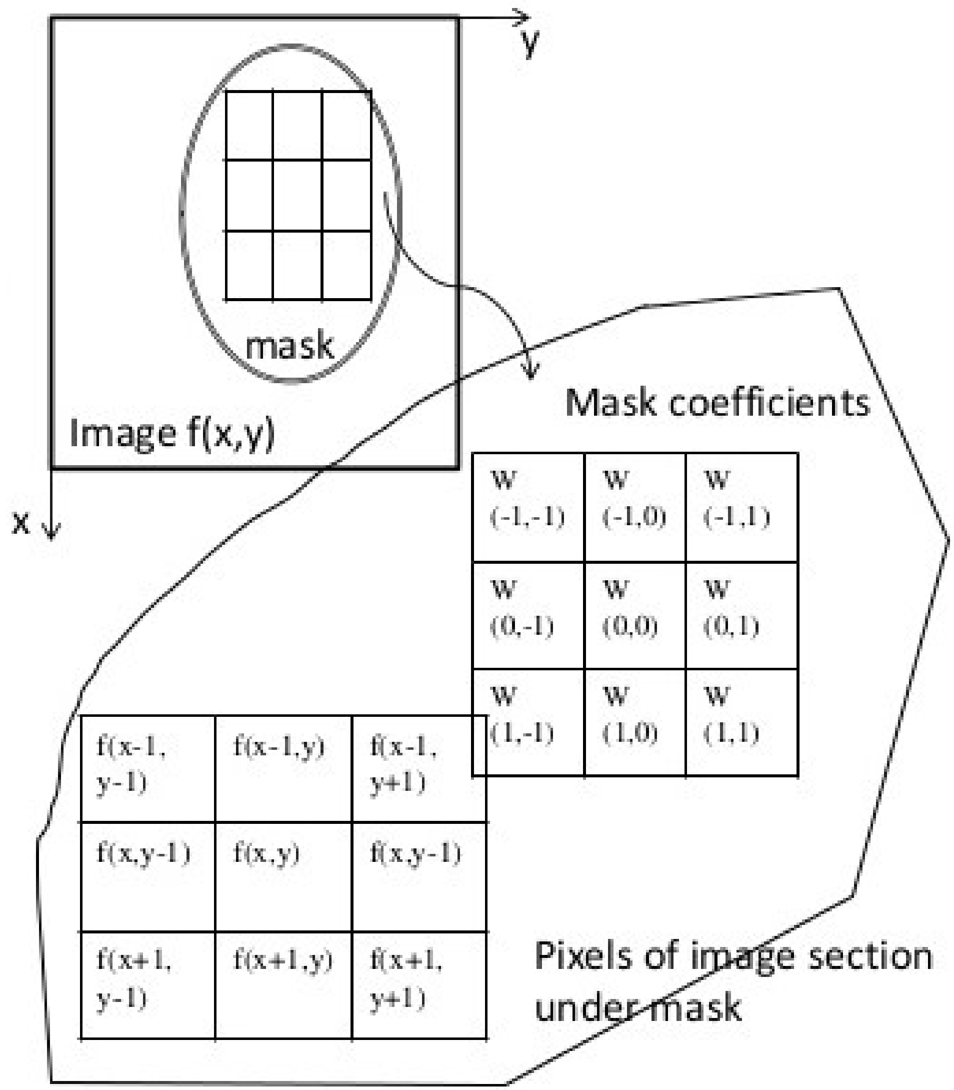
- Spatial filter contains two components;
  - MASK/ CONVOLUTION MATRIX/ KERNEL: A neighborhood (typically small rectangle:)
  - FILTER FUNCTION: A predefined operation that is performed on the image pixels encompassed by the neighborhood.
- Spatial filtering is the method of choice in situations when only additive random noise is present.
- Filtering creates a new pixel value with coordinates equal to the coordinates of the center of the neighborhood, and whose value is the result of the filtering operation.

Fundamentals of Spatial Filtering (Cont….)



Fundamentals of Spatial Filtering (Cont....)

If the operation performed on the image pixels is linear, then the filter is called a linear spatial filter, otherwise, the filter is non-linear.



Fundamentals of Spatial Filtering (Cont….)

- Spatial filtering of an image of size ***M×N*** with a filter of size ***m×n*** is given by the expression;

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

- Where ***w*** is the mask and ***f*** is the image function, ***x*** and ***y*** are varied so that each pixel in ***w*** visits every pixel in ***f***.
- ***m = 2a+1*** and ***n = 2b+1***, where ***a*** and ***b*** are positive integers.
- This concept is known as **correlation.**

Fundamentals of Spatial Filtering (Cont….)

- There are two closely related concepts that must be understood clearly when performing linear spatial filtering
  - **Correlation**
    - Correlation is the process of moving a filter mask over the image and computing the sum of products at each location exactly as explained previously.
  - **Convolution**
    - The mechanics of convolution are the same as correlation except that the filter is first rotated by 180 degrees.

Fundamentals of Spatial Filtering (Cont….)

Correlation

Filter

244	255	246
255	240	183
255	250	12

.*

1	2	3
4	5	6
7	8	9

=

244	510	738
1020	1200	1098
1785	2000	108

⇒ 8703

Convolution

Filter Rotated 180°

244	255	246
255	240	183
255	250	12

.*

9	8	7
6	5	4
3	2	1

=

2196	2040	1722
1530	1200	732
765	500	12

⇒ 10697

If the filter mask is symmetric, correlation and convolution yield the same result

Mask/Kernel/ Convolution Matrix

- In image processing, a kernel, convolution matrix, or mask is a small matrix.
- It is useful for blurring, sharpening, embossing, edge detection, and more.
- This is accomplished by means of convolution between a kernel and an image.

Mask/Kernel/ Convolution Matrix (Cont…)

- Important Features of Kernel
  - Size: 3×3, 5×5, 9×9, 21×21
  - Shape: rectangle, cross, strip, circular, diamond or user defined.
  - Coefficients / Values: set based on the operation.
  - Anchor point: mostly in middle

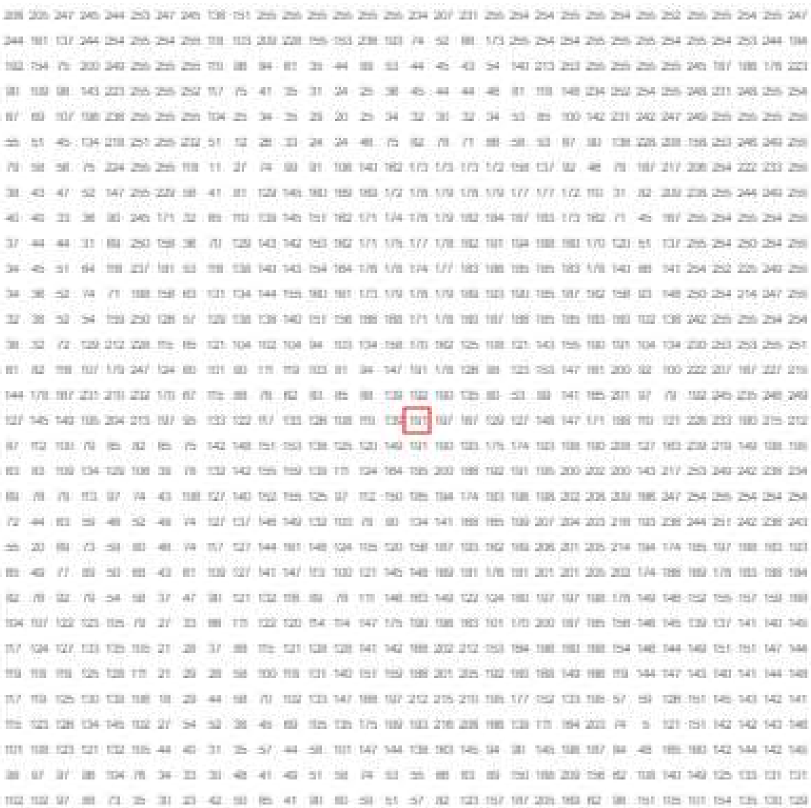
Values for sharpening operation

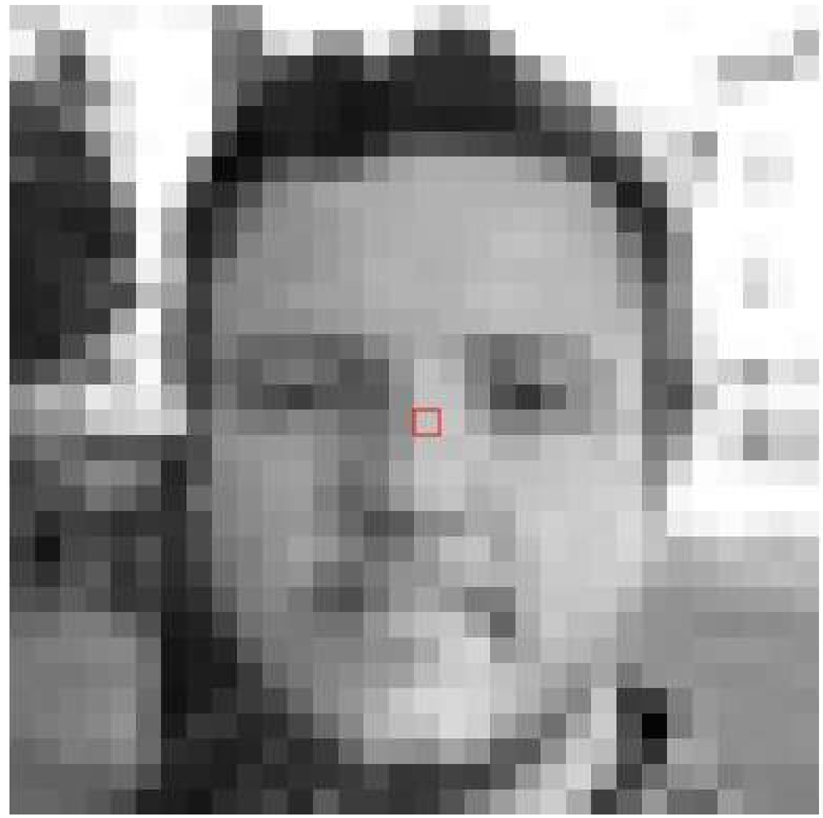
3×3 Kernel

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

Anchor

Mask/Kernel/ Convolution Matrix (Cont...)

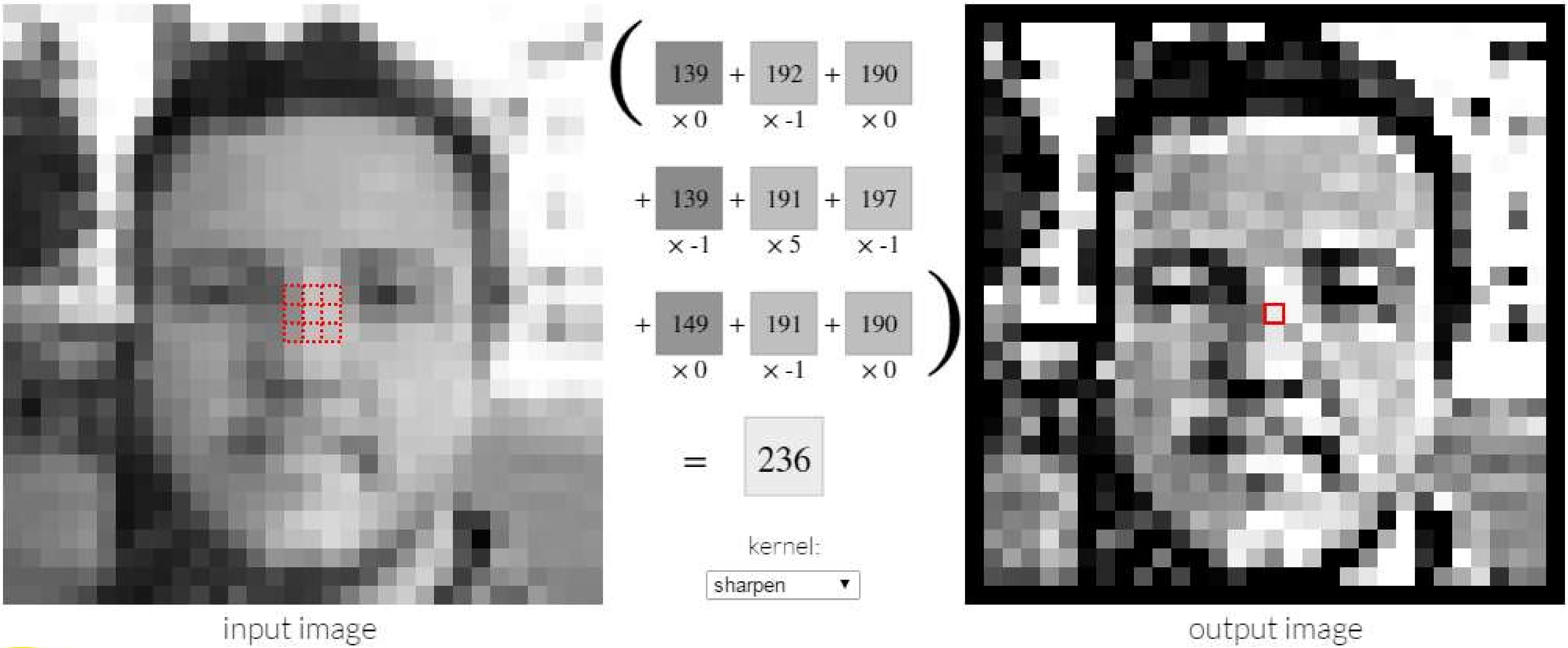




Kernel

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

Mask/Kernel/ Convolution Matrix (Cont…)



http://setosa.io/ev/image-kernels/
https://en.wikipedia.org/wiki/Kernel_(image_processing)

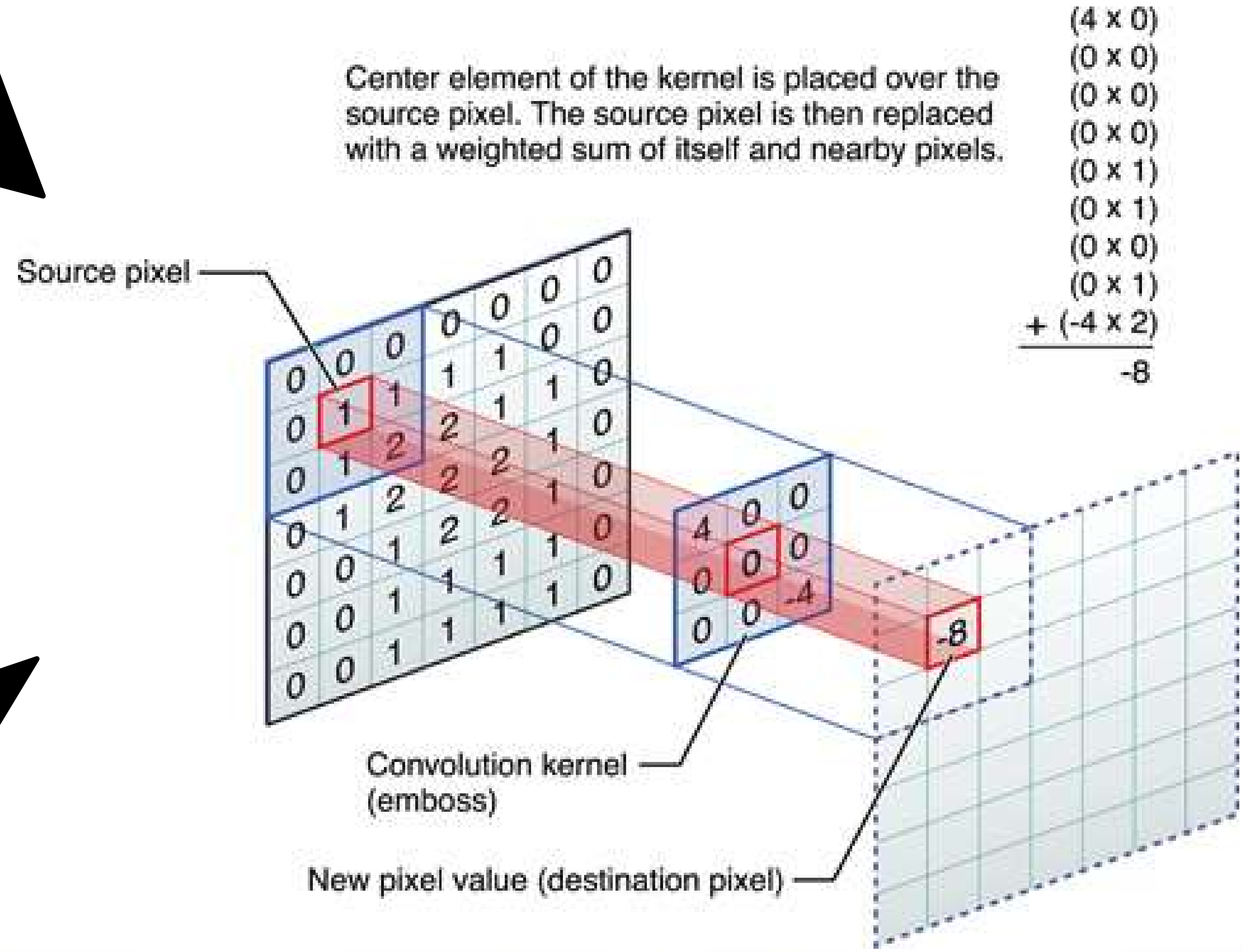
Convolution

- Convolution is a simple mathematical operation which is fundamental to many common image processing operators.
- This can be used in image processing to implement operators whose output pixel values are simple linear combinations of certain input pixel values.

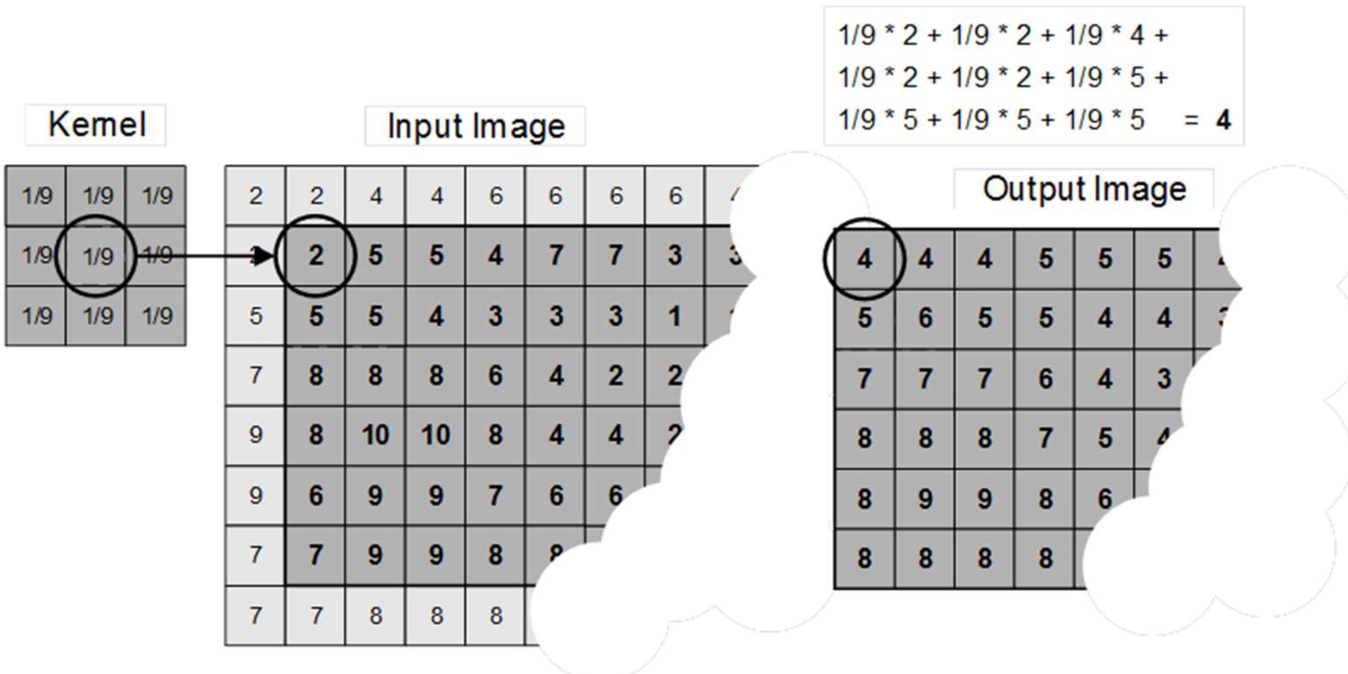
Convolution (Cont…)

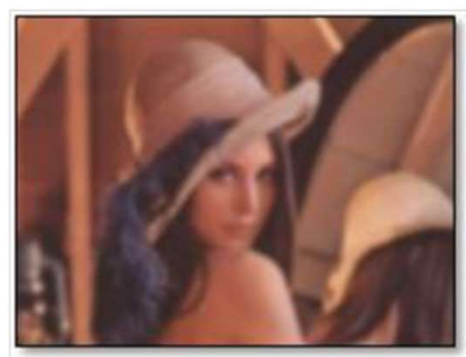
Each kernel position corresponds to a single output pixel, the value of which is calculated by multiplying together the kernel value and the underlying image pixel value for each of the cells in the kernel, and then adding all these numbers together.

The convolution is performed by sliding the kernel over the image, generally starting at the top left corner.



Convolution (Cont…)





Generating Spatial Filter Masks

- Generating an ***m×n*** linear spatial filter requires that we specify ***mn*** mask coefficients.
- These coefficients are selected based on what the filter is supposed to do.
- There are different types of filters available, some of them are;
  - Linear filters - In signal processing, a linear filter is a filter whose output is a linear function of its input.
  - Continuous function based filters – e.g. Gaussian smoothing (it's a kind of linear filter).
  - Non linear filters - In signal processing, a non-linear filter is a filter whose output is not a linear function of its input.

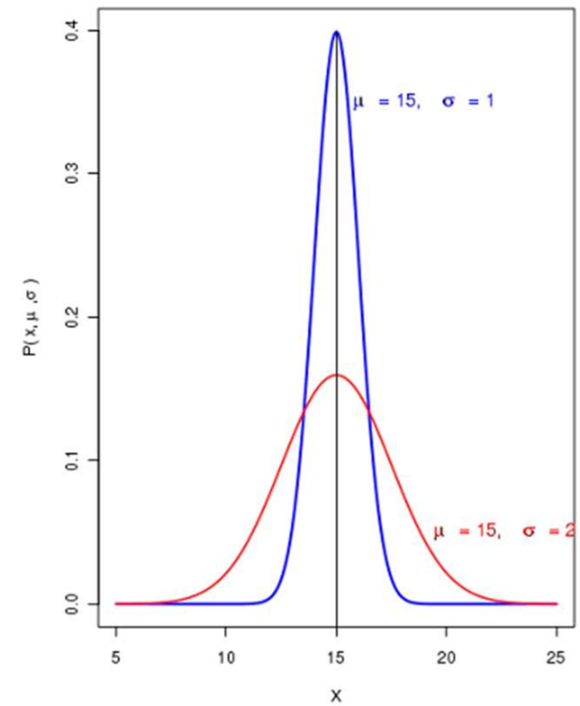
Generating Spatial Filter Masks (cont…)

- **Linear filters**
  - Implement sum of products
  - ex: filters used for smoothing

- **Continuous function based filters**
  - ex: Gaussian function of two variables as shown in the following equation;

$$g(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

  - Where σ is the standard deviation and coordinates x and y are the positive integers.
  - Recall that 2D Gaussian function has bell shape and the σ control the tightness of the bell.



Generating Spatial Filter Masks (cont...)

- **Nonlinear/ Order static filters**
  - Non linear filters require to specify the size of the filter and the operation(s) to be performed on the image pixel contained in the filter.
    - ex: filter 5×5 filter centered at an arbitrary point and perform max operation.
  - Non linear filters are quiet powerful in some applications, than the linear filters.

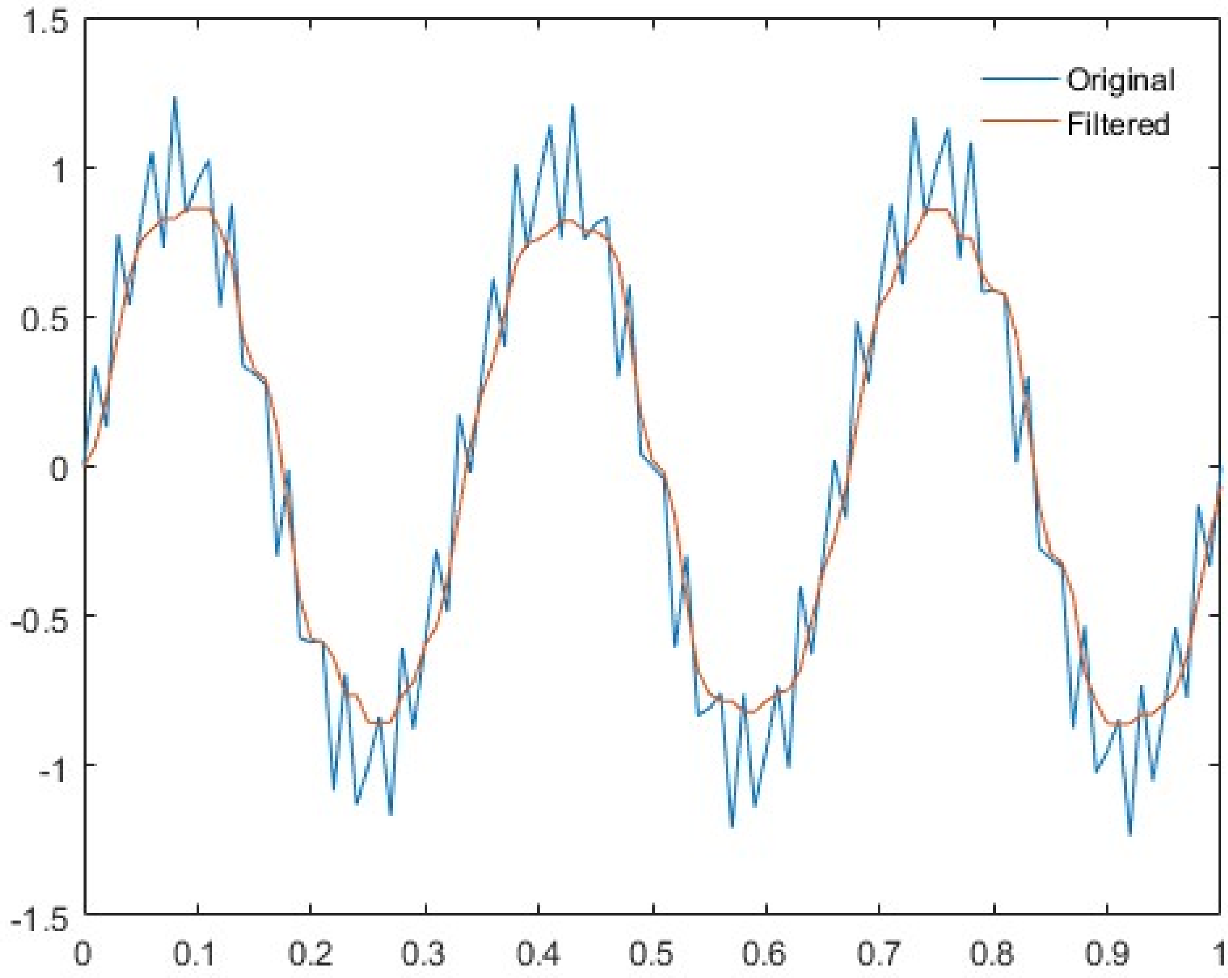
Smoothing Spatial Filters

- Smoothing filters are used for **blurring** and **noise reduction**.
- Blurring is used in preprocessing tasks, such as removal of small details from an image prior to 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.

Smoothing Linear Filters

- <u>**Average /Mean Filter**</u>
  - The output is simply the average of the pixels contained in the neighborhood of the filter mask.
  - In smoothing filter, it replaces the value of every pixel in an image by the average of the intensity levels in the neighborhood defined by the filter mask.
  - <u>**This process result in an image with reduced “sharp” transitions in intensities.**</u>
  - Most obvious application of smoothing is noise reduction because random noise typically consists of sharp transitions in intensity levels.

Smoothing Linear Filters (Cont…)



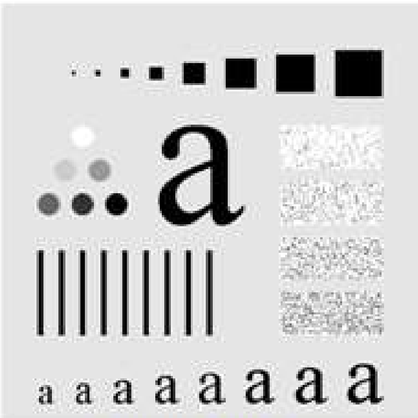
Smoothing Linear Filters (Cont…)

– Averaging filters blur edges, because edges represent the sharp intensity transition.

– Average blurring smoothes the **false contours** that result from using an insufficient number of intensity levels.

– A major use of averaging filters is in the reduction of “irrelevant” detail (pixel regions that are small with respect to the size of the filter mask) in an image.

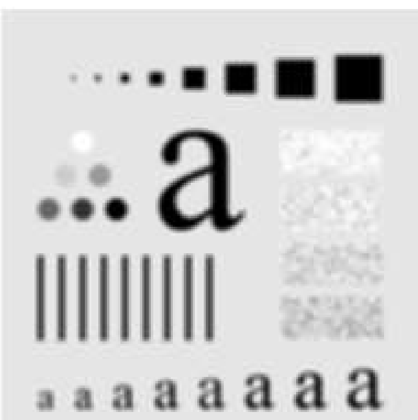
Smoothing Linear Filters (Cont…)



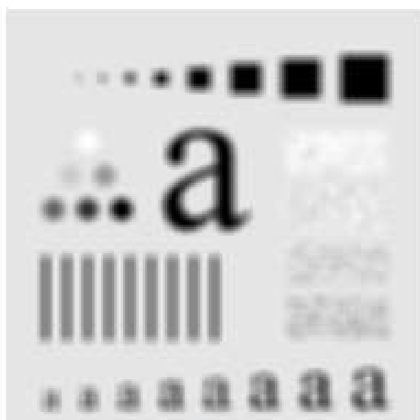
Original Image

3 by 3

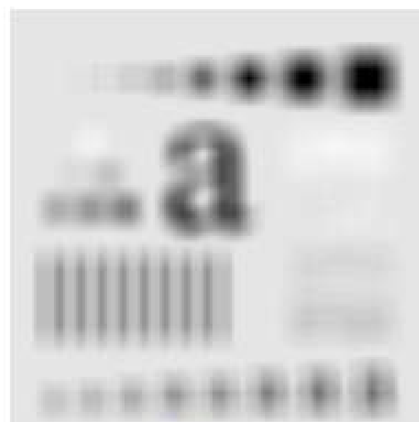
5 by 5



9 by 9



15 by 15



35 by 35

Observe blurring occur during smoothing operation

Smoothing Linear Filters (Cont…)

- Let $S_{x,y}$ represent the set of coordinates in neighbourhood of size $m×n$, centered at point (x, y).
- The arithmetic mean filter computes the average value of the corrupted image $g(x, y)$ in the area defined by $S_{x,y}$.
- The value of the restored image f' at point (x, y) is simply the arithmetic mean computed using the pixels in the region defined by $S_{x,y}$.

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s,t)$$

Smoothing Linear Filters (Cont…)

- There are different kinds of mean filters all of which exhibit slightly different behaviour:
- **Geometric Mean:**
  - Achieves similar smoothing to the arithmetic mean, but tends to lose less image detail

$$\hat{f}(x,y) = \left[\prod_{(s,t)\in S_{xy}} g(s,t) \right]^{\frac{1}{mn}}$$

Smoothing Linear Filters (Cont...)

- **<u>Harmonic Mean:</u>**
  - Works well for salt noise, but fails for pepper noise.
  - Also does well for other kinds of noise such as Gaussian noise

$$\hat{f}(x,y) = \frac{mn}{\displaystyle\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$$

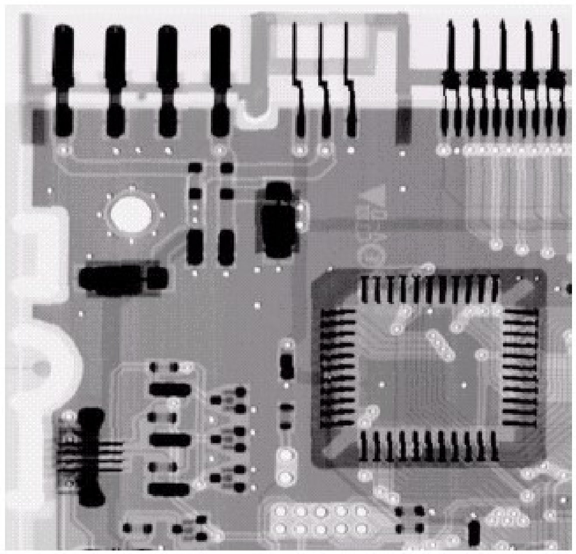
Smoothing Linear Filters (Cont…)

- **<u>Contraharmonic Mean:</u>**
  - Q is the *order* of the filter and adjusting its value changes the filter's behaviour.
  - Positive values of Q eliminate pepper noise.
  - Negative values of Q eliminate salt noise.
  - If Q =0 it represents the arithmetic filter.

$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

Smoothing Linear Filters (Cont…)

Original Image



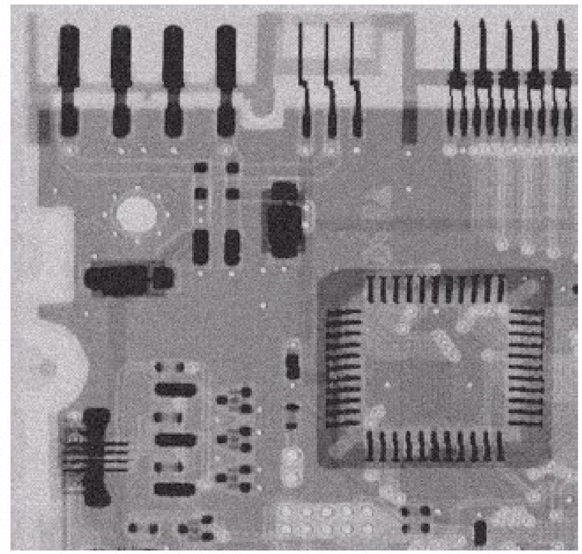
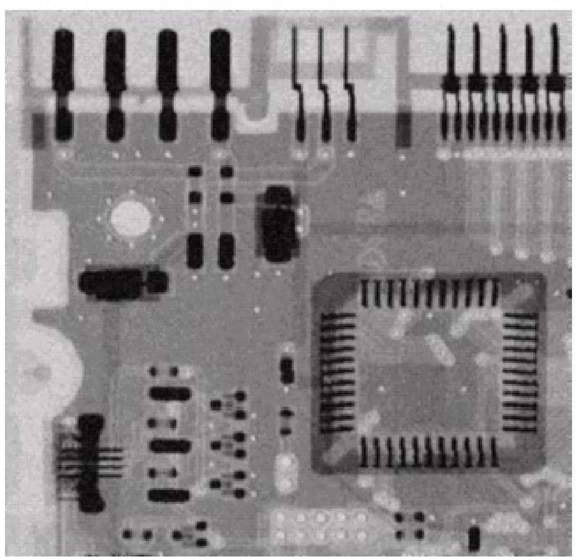
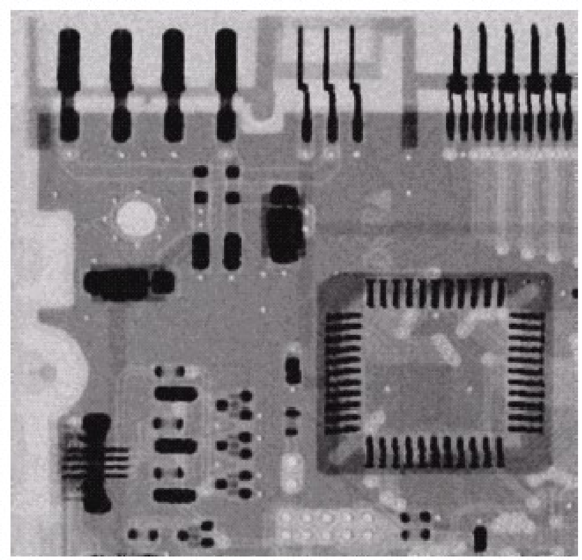


Image Corrupted By Gaussian Noise

After A 3*3 Arithmetic Mean Filter

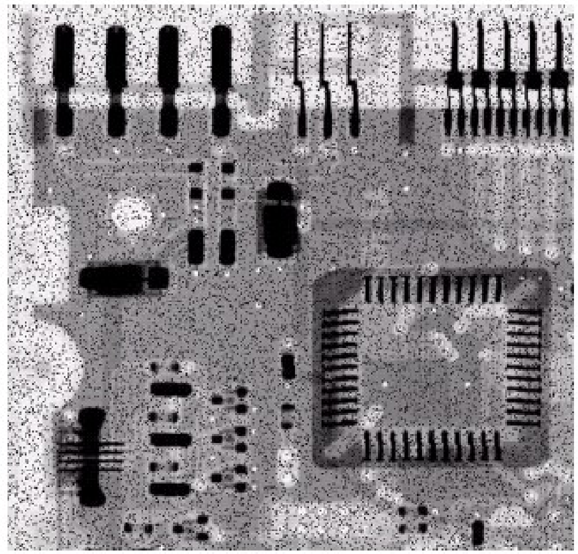




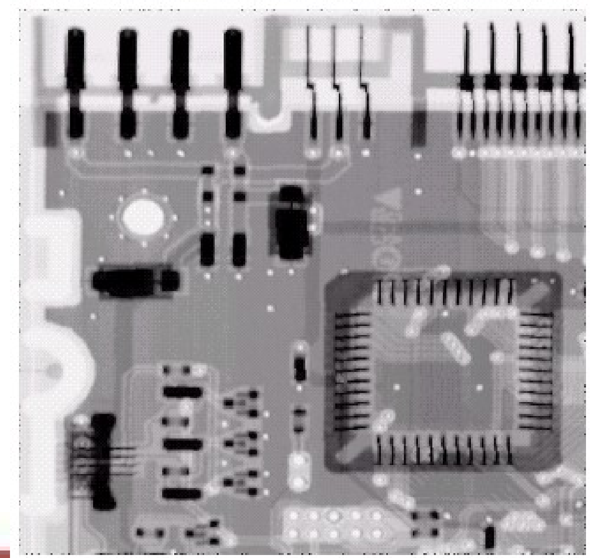
After A 3*3 Geometric Mean Filter

Smoothing Linear Filters (Cont…)

Image Corrupted By Pepper Noise



Result of Filtering Above With 3*3 Contraharmonic Q=1.5



Smoothing Linear Filters (Cont…)

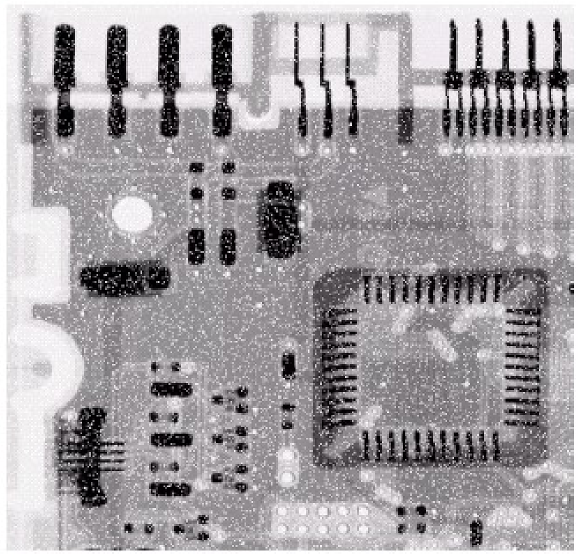
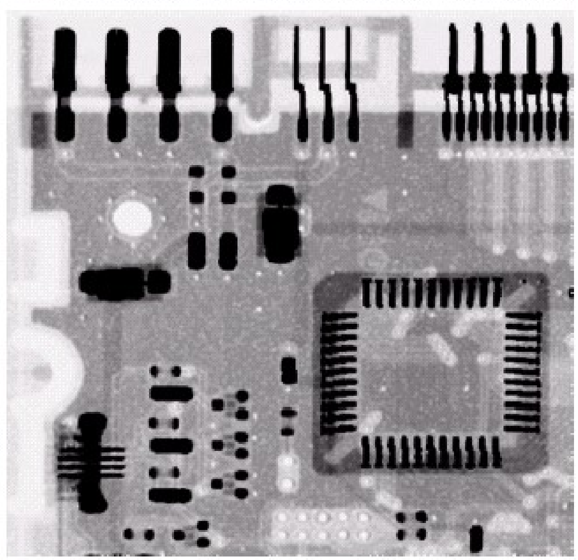


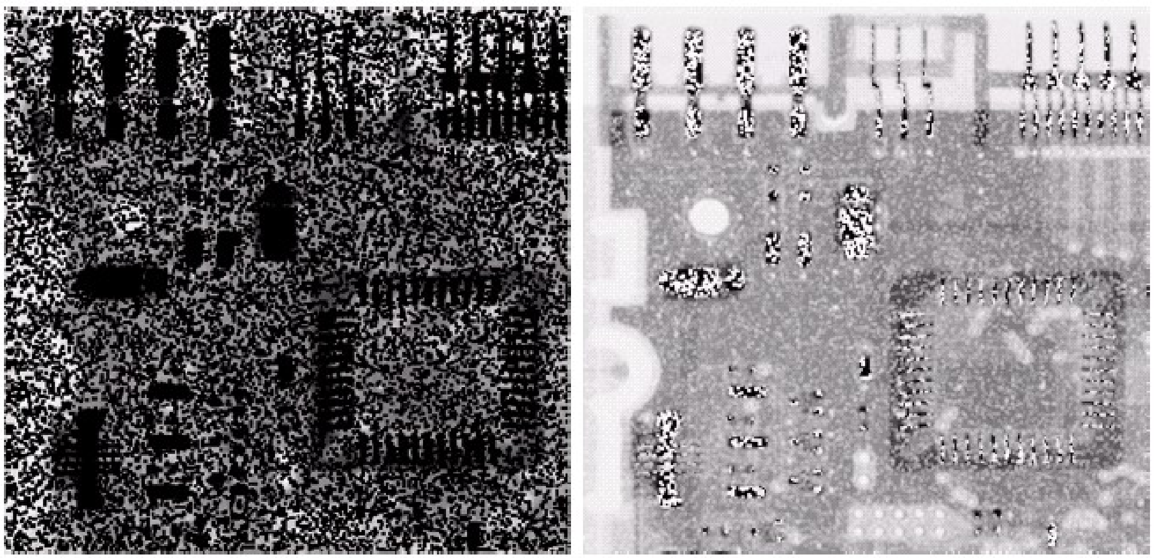
Image Corrupted By Salt Noise



Result of Filtering Above With 3*3 Contraharmonic Q=-1.5

Smoothing Linear Filters (Cont…)

Choosing the wrong value for Q when using the contraharmonic filter can have drastic results



Smoothing Linear Filters (Cont…)

- A spatial averaging filter in which all coefficients are equal is called a **Box Filter**.

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}$$

- 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 in spatial averaging filter is simply an attempt to reduce blurring in the smoothing process.

$$1/16 = \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline\end{array}$$

Smoothing Linear Filters (Cont…)

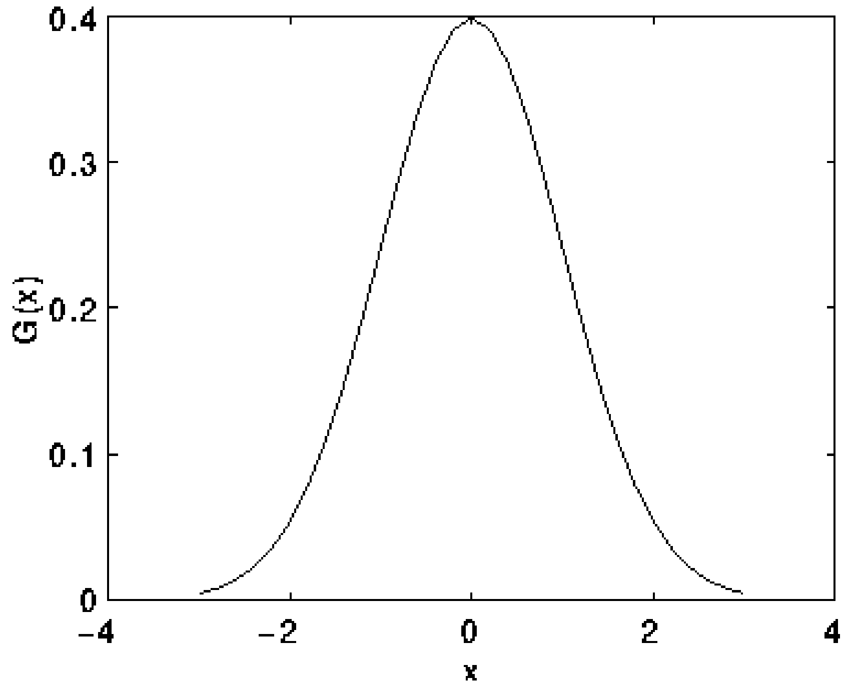
- **Gaussian Smoothing:**
  - The Gaussian filter is a non-uniform low pass filter.
  - The kernel coefficients diminish with increasing distance from the kernel's centre.
  - Central pixels have a higher weighting than those on the periphery.
  - Larger values of σ produce a wider peak (greater blurring).
  - Kernel size must increase with increasing σ to maintain the Gaussian nature of the filter (`bell-shaped') .
  - Gaussian kernel coefficients depend on the value of σ.
  - At the edge of the mask, coefficients must be close to 0.
  - The kernel is rotationally symmetric with no directional bias.
  - Gaussian filters might not preserve image brightness.

Smoothing Linear Filters (Cont…)

- The Gaussian distribution in 1-D has the form:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

- Where σ is the standard deviation of the distribution.
- We have also assumed that the distribution has a mean of zero (*i.e.* it is centered on the line *x*=0).

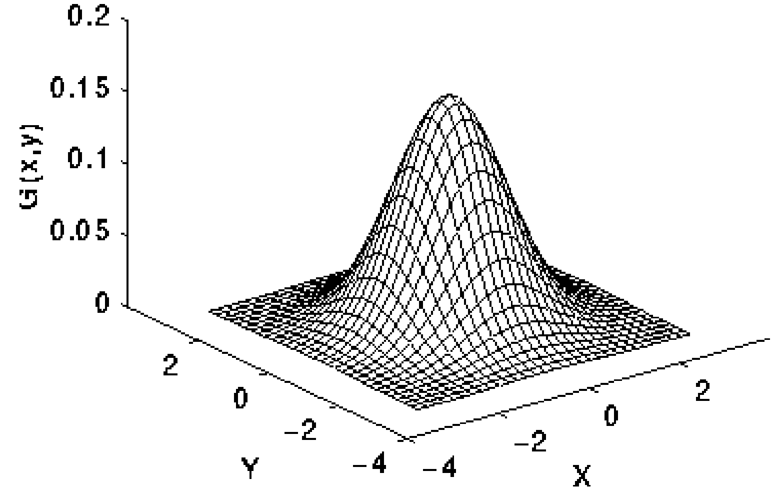


1-D Gaussian distribution with mean 0 and σ=1

Smoothing Linear Filters (Cont...)

- In 2-D, an isotropic (*i.e.* circularly symmetric) Gaussian has the form:

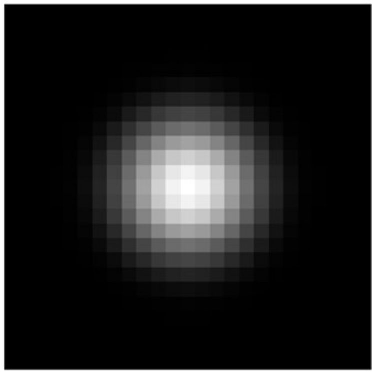
$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



2-D Gaussian distribution with mean (0,0) and σ=1

$\frac{1}{273}$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1



Discrete approximation to Gaussian function with σ=1

The std. dev σ of the Gaussian determines the amount of smoothing.

Smoothing Linear Filters (Cont…)

Apply the Gaussian filter to the image:
Borders: keep border values as they are

15	20	25	25	15	10
20	15	50	30	20	15
20	50	55	60	30	20
20	15	65	30	15	30
15	20	30	20	25	30
20	25	15	20	10	15

Original image

¼*

1	2	1

15	20	24	23	16	10
20	25	36	33	21	15
20	44	55	51	35	20
20	29	44	35	22	30
15	21	25	24	25	30
20	21	19	16	14	15

¼*

1
2
1

15	20	24	23	16	10
19	28	38	35	23	15
20	35	48	43	28	21
19	31	42	36	26	28
18	23	28	25	22	21
20	21	19	16	14	15

Or:

1	2	1
2	4	2
1	2	1

*1/16

Smoothing Order Static (Nonlinear) Filters

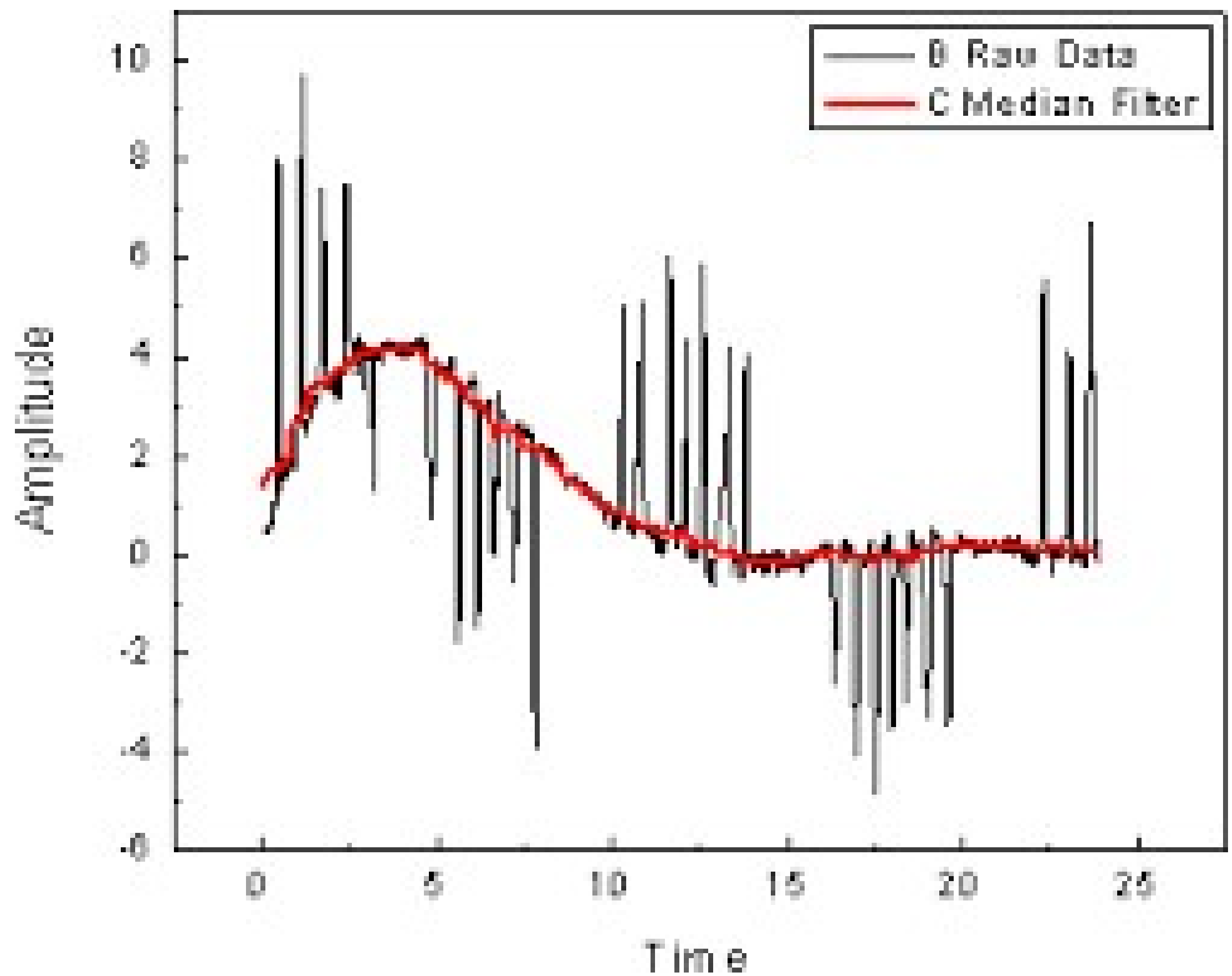
- The response of the order static or nonlinear filters is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determine by the ranking result.
- Useful nonlinear filters include;
  – Median filter
  – Max and min filter
  – Midpoint filter

Smoothing Order Static (Nonlinear) Filters (Cont…)

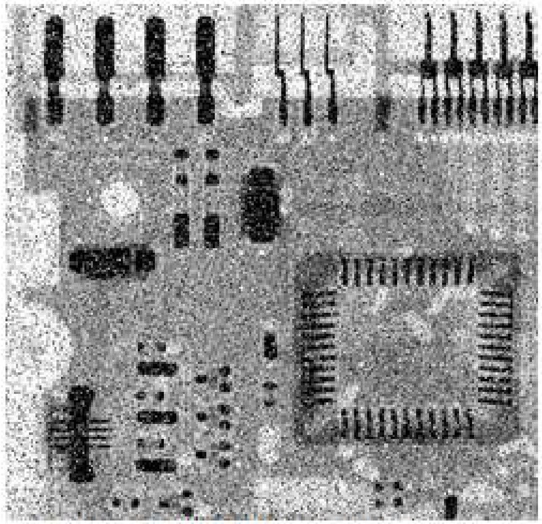
- **<u>Median Filter:</u>**
  - It can reduce certain typed of random noise with less blurring than the linear smoothing filters of similar size.
  - Provides excellent results when applying to reduce salt and pepper noise.

$$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{median}\{g(s,t)\}$$

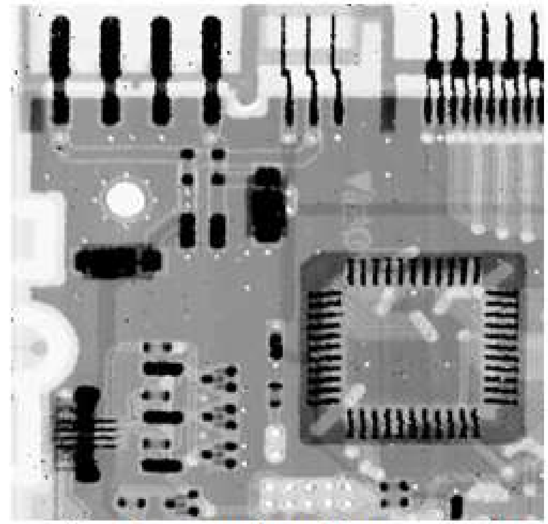
Smoothing Order Static (Nonlinear) Filters (Cont…)



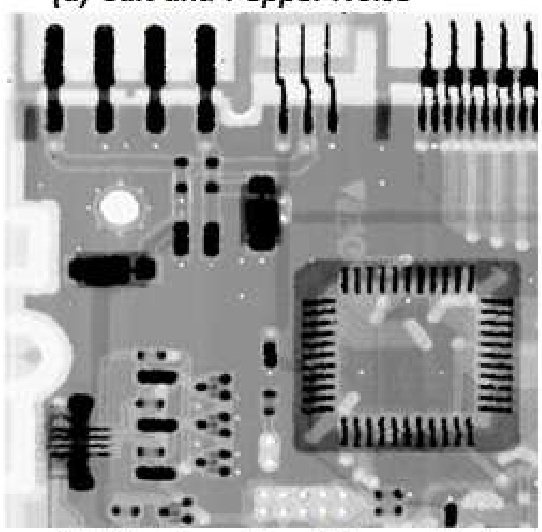
Smoothing Order Static (Nonlinear) Filters (Cont…)



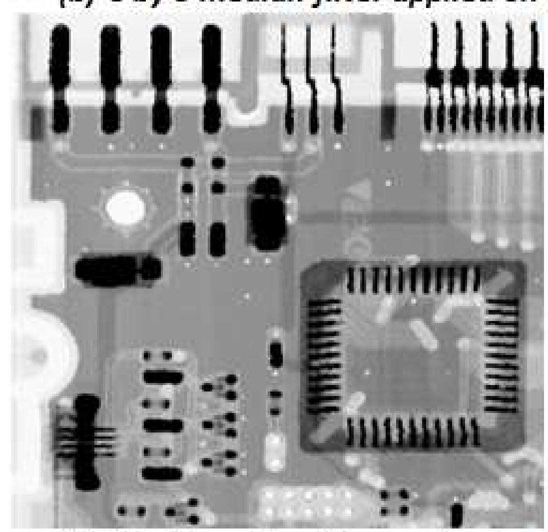
(a) Salt and Pepper Noise



(b) 3 by 3 median filter applied on (a)



(c) 3 by 3 median filter applied on (b)



(d) 3 by 3 median filter applied on (c)

Smoothing Order Static (Nonlinear) Filters (Cont…)

- Median filter smoothes ("washes") all edges and boundaries and may "erase" all details whose size is about $n/2\times m/2$, where $n \times m$ is a window size.
- As a result, an image becomes "fuzzy".
- Median filter is not so efficient for additive Gaussian noise removal, it yields to linear filters.

Smoothing Order Static (Nonlinear) Filters (Cont…)

- **Max Filter:**

$$\hat{f}(x,y) = \max_{(s,t)\in S_{xy}} \{g(s,t)\}$$

- **Min Filter:**

$$\hat{f}(x,y) = \min_{(s,t)\in S_{xy}} \{g(s,t)\}$$

– Max filter is useful for finding the brightest points in an image and min filter for finding the darkest points in an image.

– Max filter is good for pepper noise and min filter is good for salt noise

Smoothing Order Static (Nonlinear) Filters (Cont…)

- **Midpoint Filter:**

$$\hat{f}(x,y) = \frac{1}{2}\left[\max_{(s,t)\in S_{xy}}\{g(s,t)\} + \min_{(s,t)\in S_{xy}}\{g(s,t)\}\right]$$

  – Good for random Gaussian and uniform noise

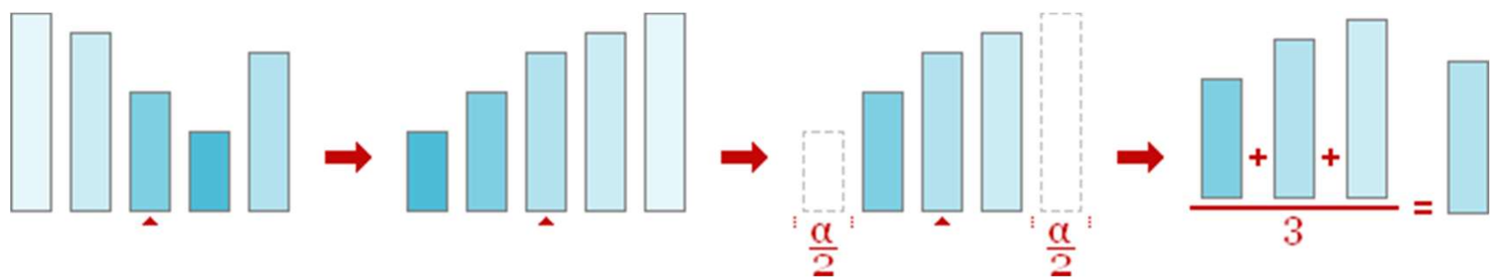
Smoothing Order Static (Nonlinear) Filters (Cont…)

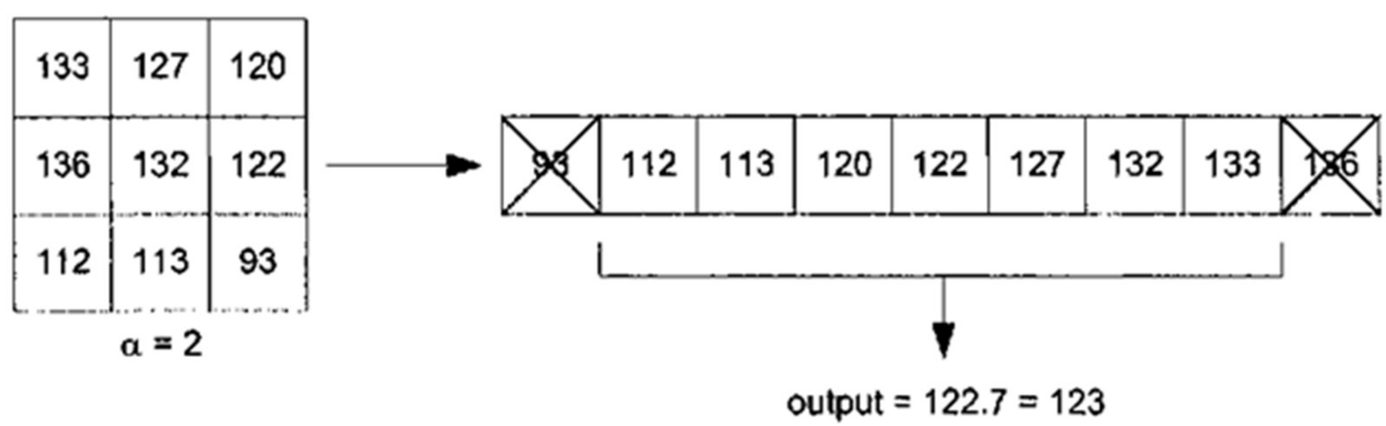
- **Alpha-trimmed Mean Filter:**

$$\hat{f}(x,y) = \frac{1}{mn-d} \sum_{(s,t)\in S_{xy}} g(s,t)$$

  - We can delete the $d/2$ lowest and $d/2$ highest grey levels.
  - So $g(s, t)$ represents the remaining $mn - d$ pixels.
  - When *d= 0* : arithmetic mean filter.
  - Good for combination of salt and pepper and Gaussian noise.

Smoothing Order Static (Nonlinear) Filters (Cont…)





Removing Periodic Noise

- Removing periodic noise form an image involves removing a particular range of frequencies from that image.
- *Band reject* filters can be used for this purpose.

Out of the scope of this course - ☹ ☹ ☹

Reference

- Chapter 03, Chapter 05 of Gonzalez, R.C., Woods, R.E., Digital Image Processing, 3rd ed. Addison-Wesley Pub.
- http://setosa.io/ev/image-kernels/
- https://en.wikipedia.org/wiki/Kernel_(image_processing)

Learning Outcomes Revisit

- Now, you should be able to;
  - describe the fundamentals of spatial filtering.
  - generating spatial filter masks.
  - identify smoothing via linear filters and non linear filters.
  - apply smoothing techniques for problem solving.

Next Lecture – Intensity Transformation and Spatial Filtering – III

QUESTIONS ?