SPATIAL DOMAIN

Basics of Spatial Filtering

- Spatial filtering term is the filtering operations that are performed directly on the pixels of an image

Mechanics of spatial filtering

- The process consists simply of moving the filter mask from point to point in an image.

- At each point (x,y) the response of the filter at that point is calculated using a predefined relationship

Basics of Spatial Filtering

- In spatial filtering the output image is computed directly by simple calculations on the pixels of the input image.

- Spatial filtering can be either linear or non-linear.

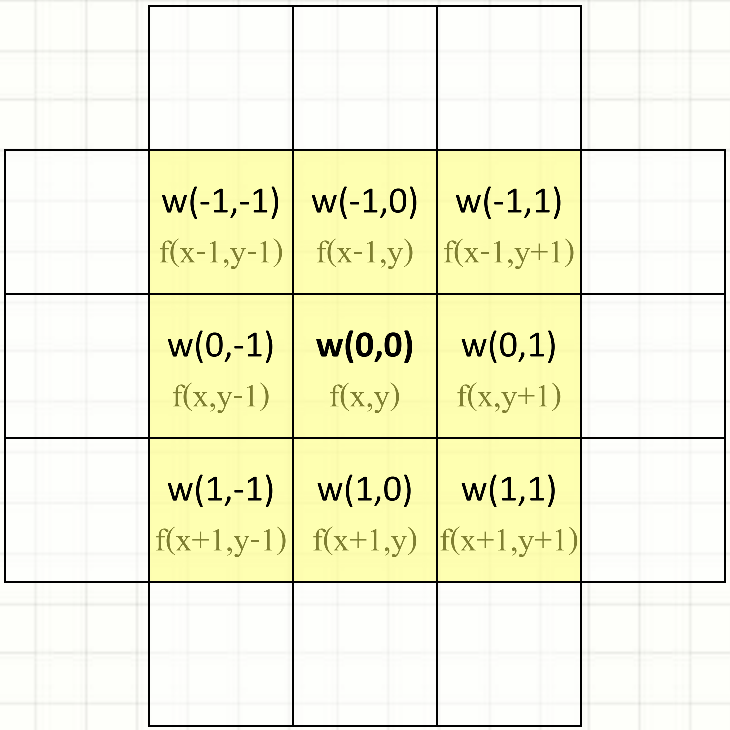
- For each output pixel, some neighborhood of input pixels is used in the computation.
- In general, linear filtering of an image *f* of size *MXN* with a filter mask of size *mxn* is given by

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

  where $a=(m-1)/2$ and $b=(n-1)/2$
- This concept called convolution. Filter masks are sometimes called convolution masks or convolution kernels.

Linear spatial filtering

Pixels of image



The result is ***the sum of products of the mask coefficients*** with the corresponding pixels directly under the mask

Mask coefficients

w(-1,-1)	w(-1,0)	w(-1,1)
w(0,-1)	w(0,0)	w(0,1)
w(1,-1)	w(1,0)	w(1,1)

$$
\begin{aligned}
f(x,y) = \; & w(-1,-1)\,f(x-1,y-1) + w(-1,0)\,f(x-1,y) + w(-1,1)\,f(x-1,y+1) + \\
& w(0,-1)\,f(x,y-1) + w(0,0)\,f(x,y) + w(0,1)\,f(x,y+1) + \\
& w(1,-1)\,f(x+1,y-1) + w(1,0)\,f(x+1,y) + w(1,1)\,f(x+1,y+1)
\end{aligned}
$$

Linear filtering *(Contid..)*

- The coefficient w(0,0) coincides with image value f(x,y), indicating that the mask is centered at (x,y) when the computation of *sum of products* takes place.

- In general, linear filtering of an image f of size MxN with a filter mask of size mxn 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 *x* and *y* are varied so that each pixel in ***w* visits every pixel in *f*.**

Neighborhood processing

- As opposed to point (pixel) processing

Input

1	2	0	1	3	
2	1	4	2	2	
1	0	1	0	1	
1	2	1	0	2	
2	5	3	1	2	

Output

Spatial Correlation and Convolution

- These are very closely related concepts :

- Correlation is the process of moving a filter mask over the image and computing the sum of products at each location.

- The mechanics of convolution are the same, except that the filter is first is *rotated by 180°*

Spatial Correlation and Convolution

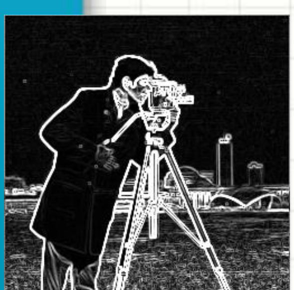
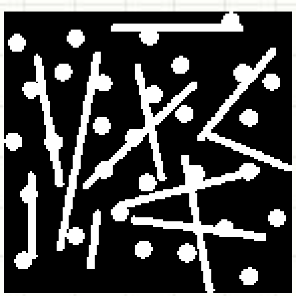
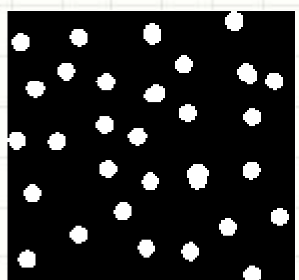
- Convolution (first 1D than 2D (images))
- Correlation

What can it be used for?

- Many many things defined by the programmer…. and some standard operations:
  - Blur image
  - Remove noise
  - Object detection
  - Morphology (later)
  - Edge detection (later)

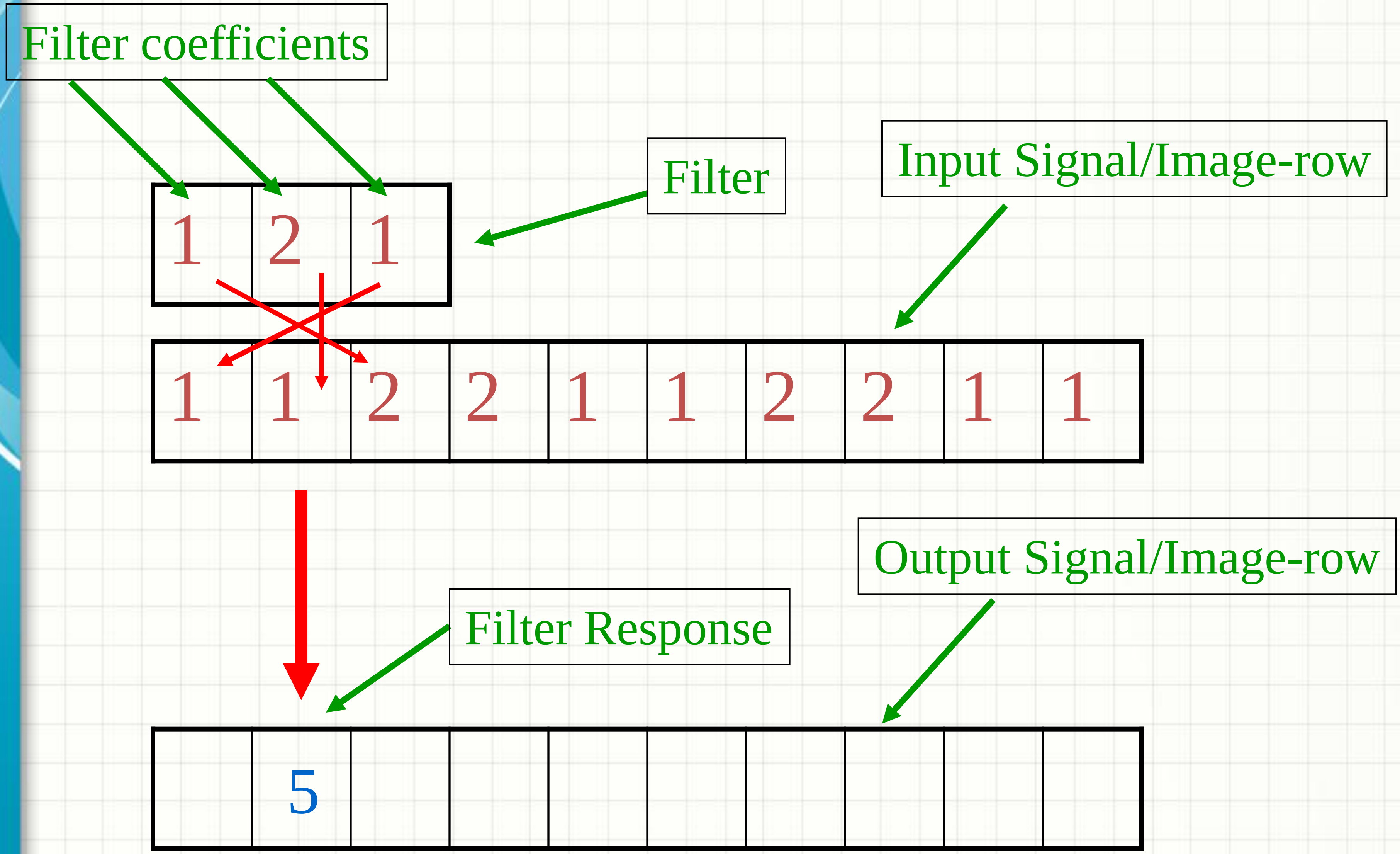





Convolution

Convolution (1D)

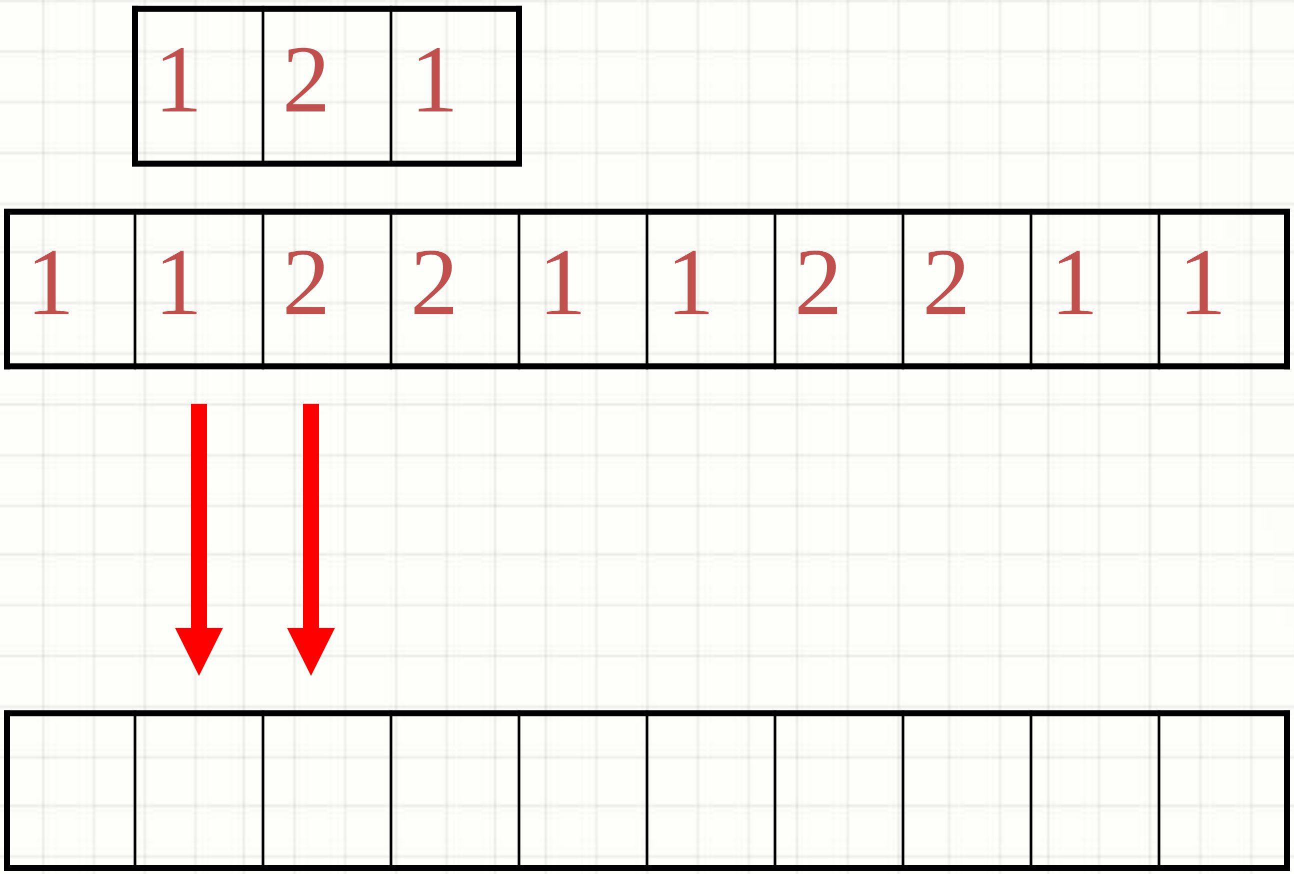


Convolution (1D)

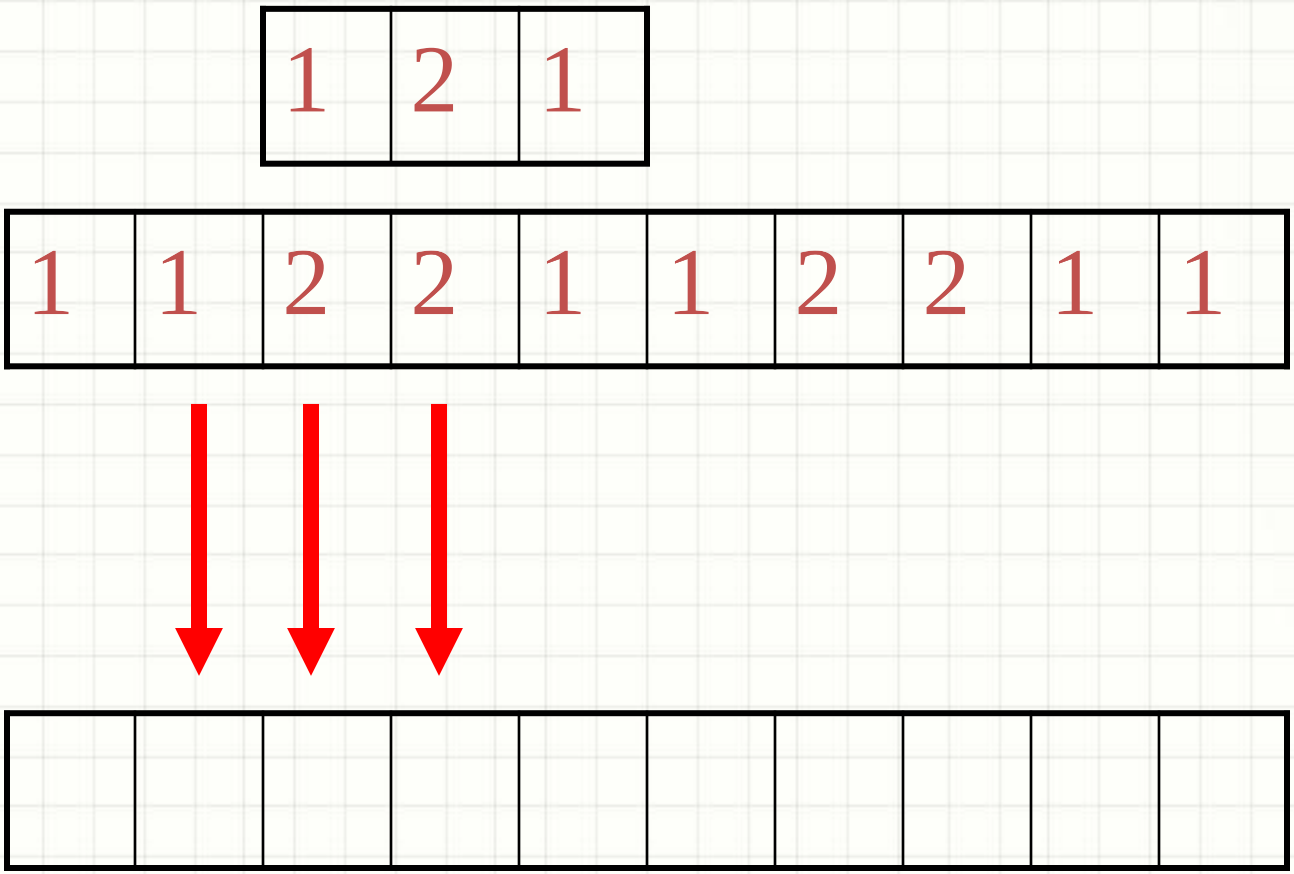
1	2	1

1	1	2	2	1	1	2	2	1	1

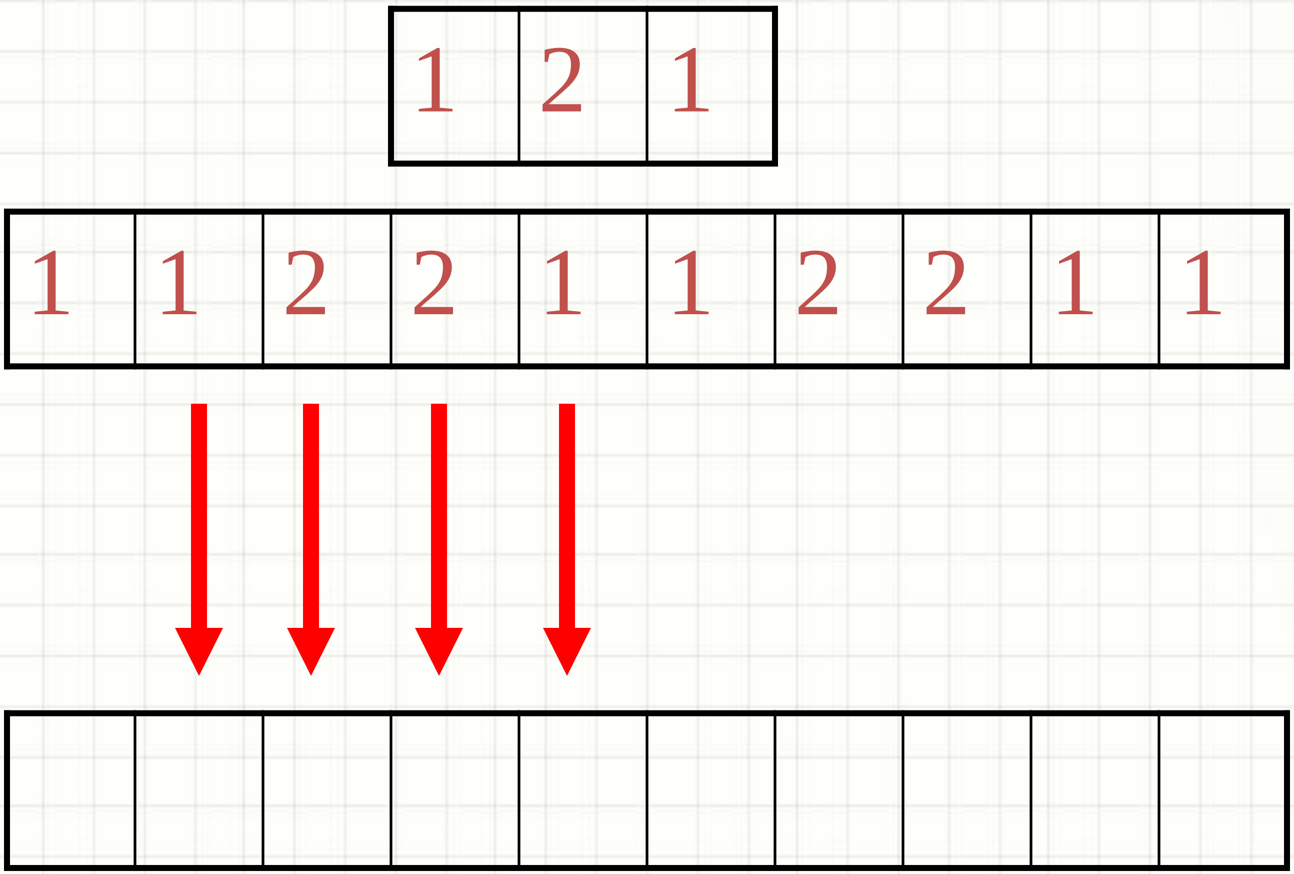
Convolution (1D)



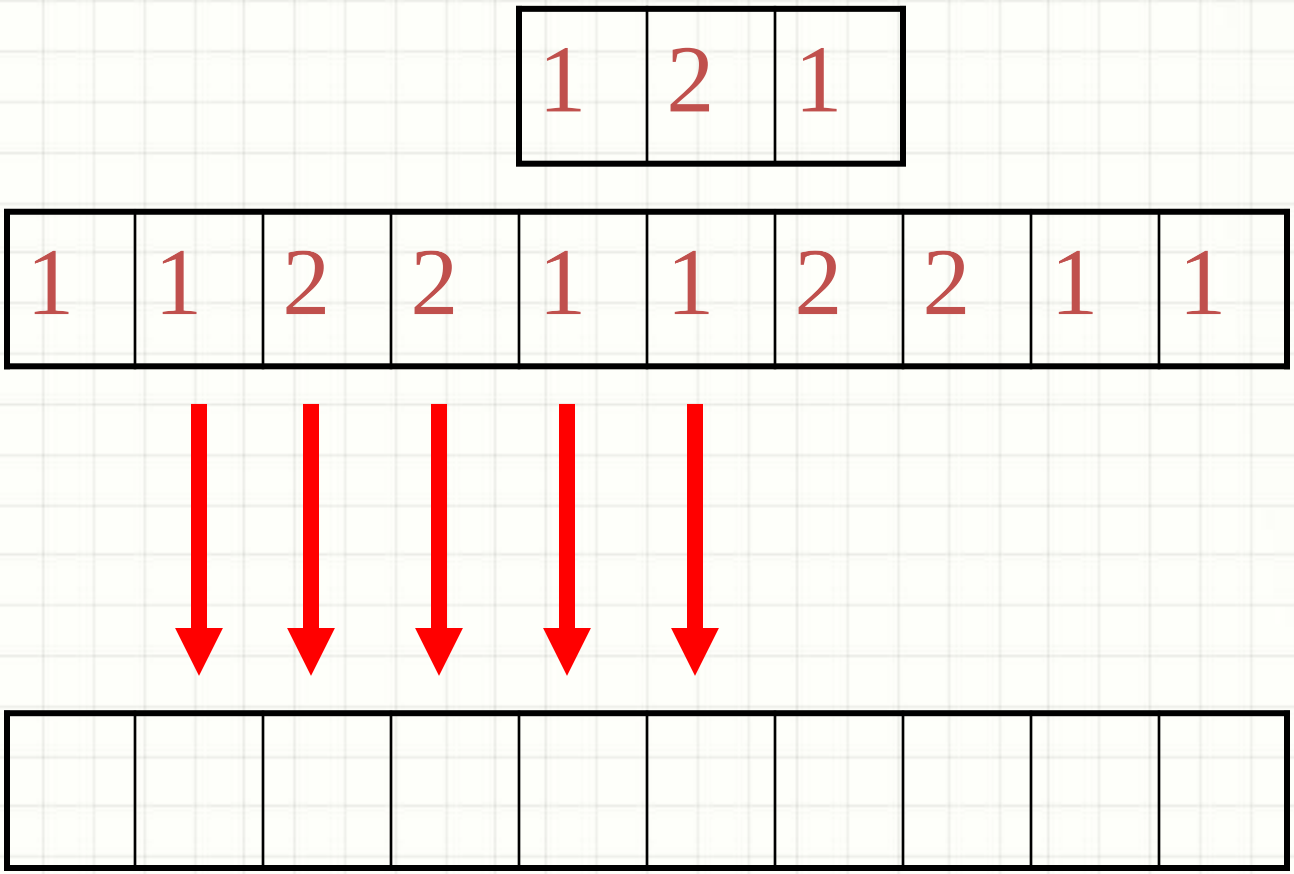
Convolution (1D)



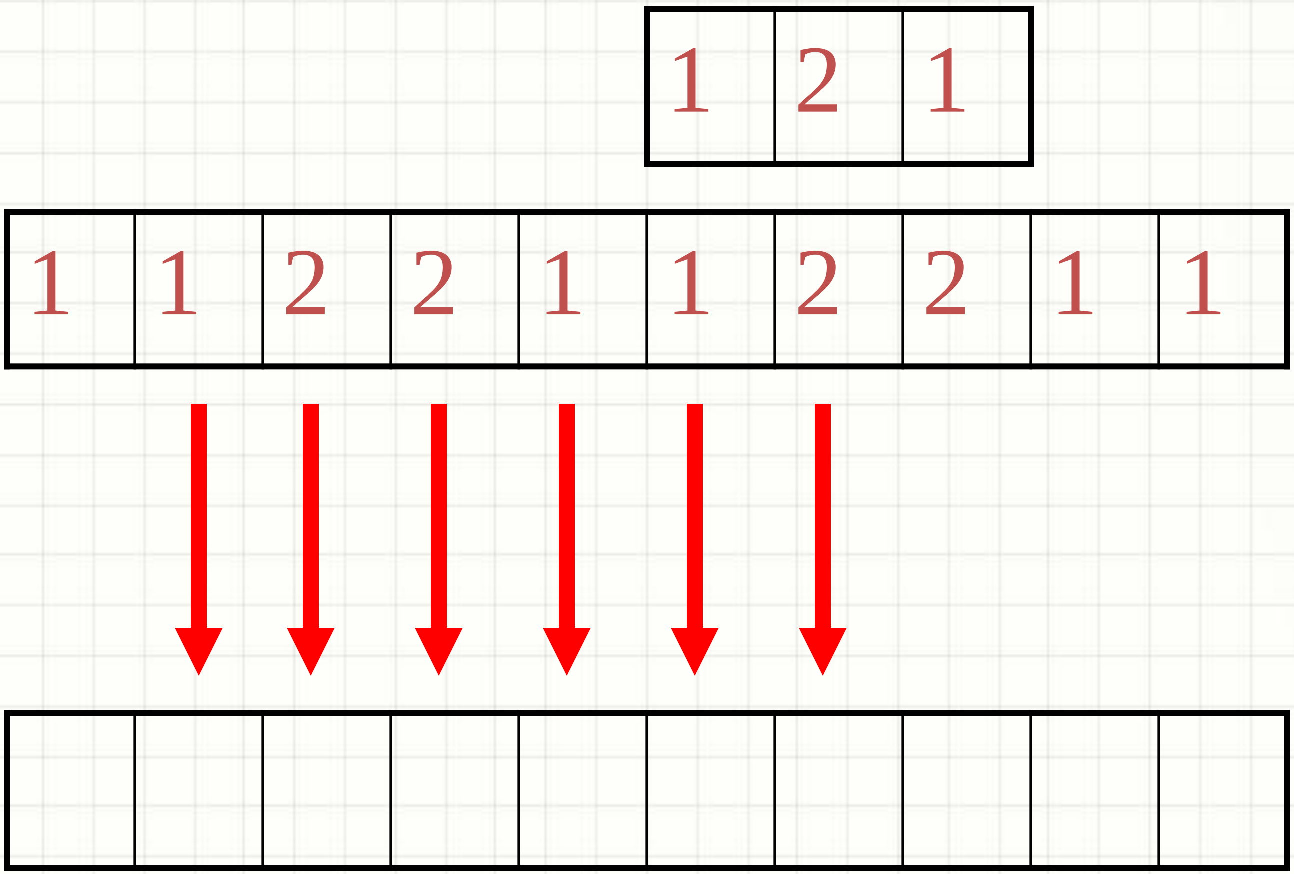
Convolution (1D)



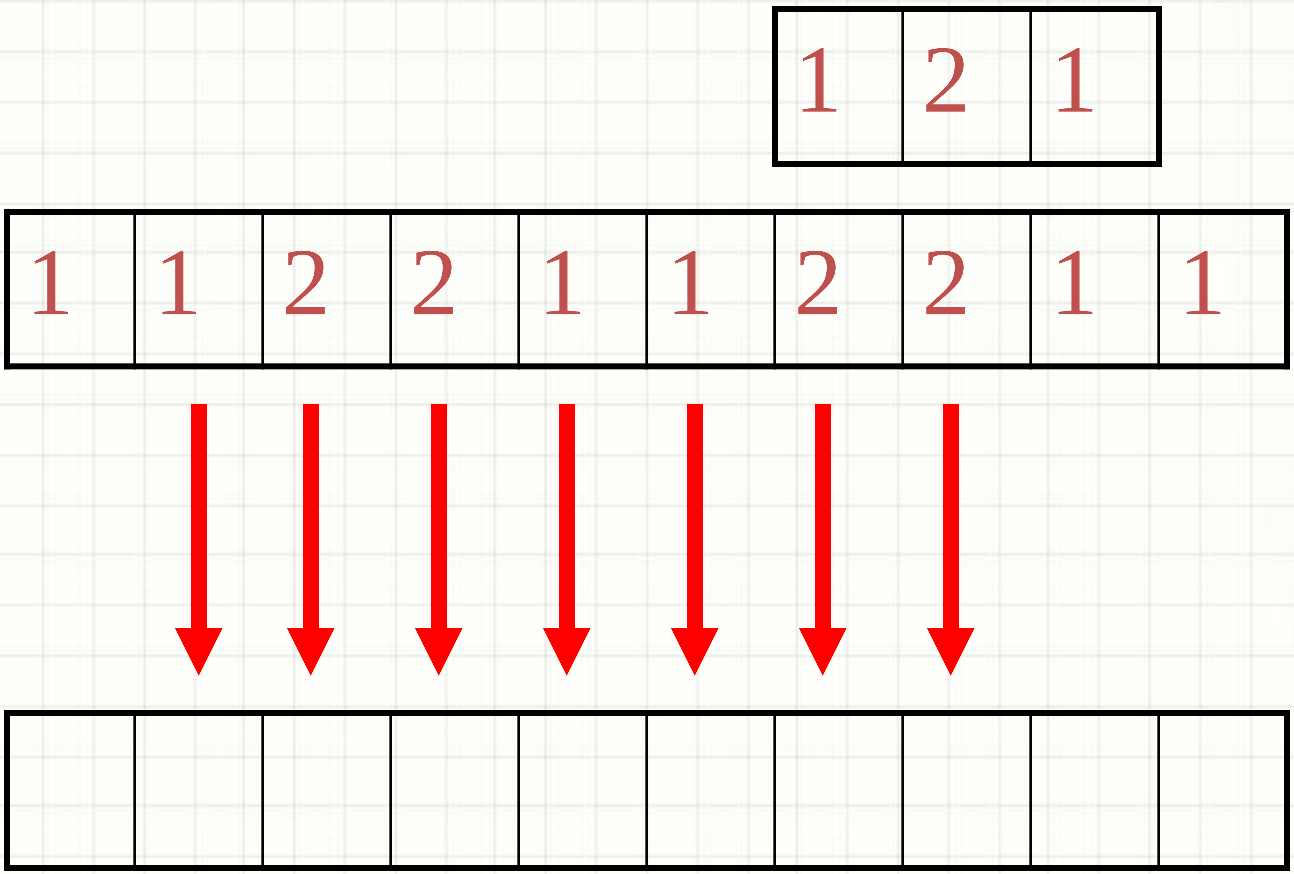
Convolution (1D)



Convolution (1D)

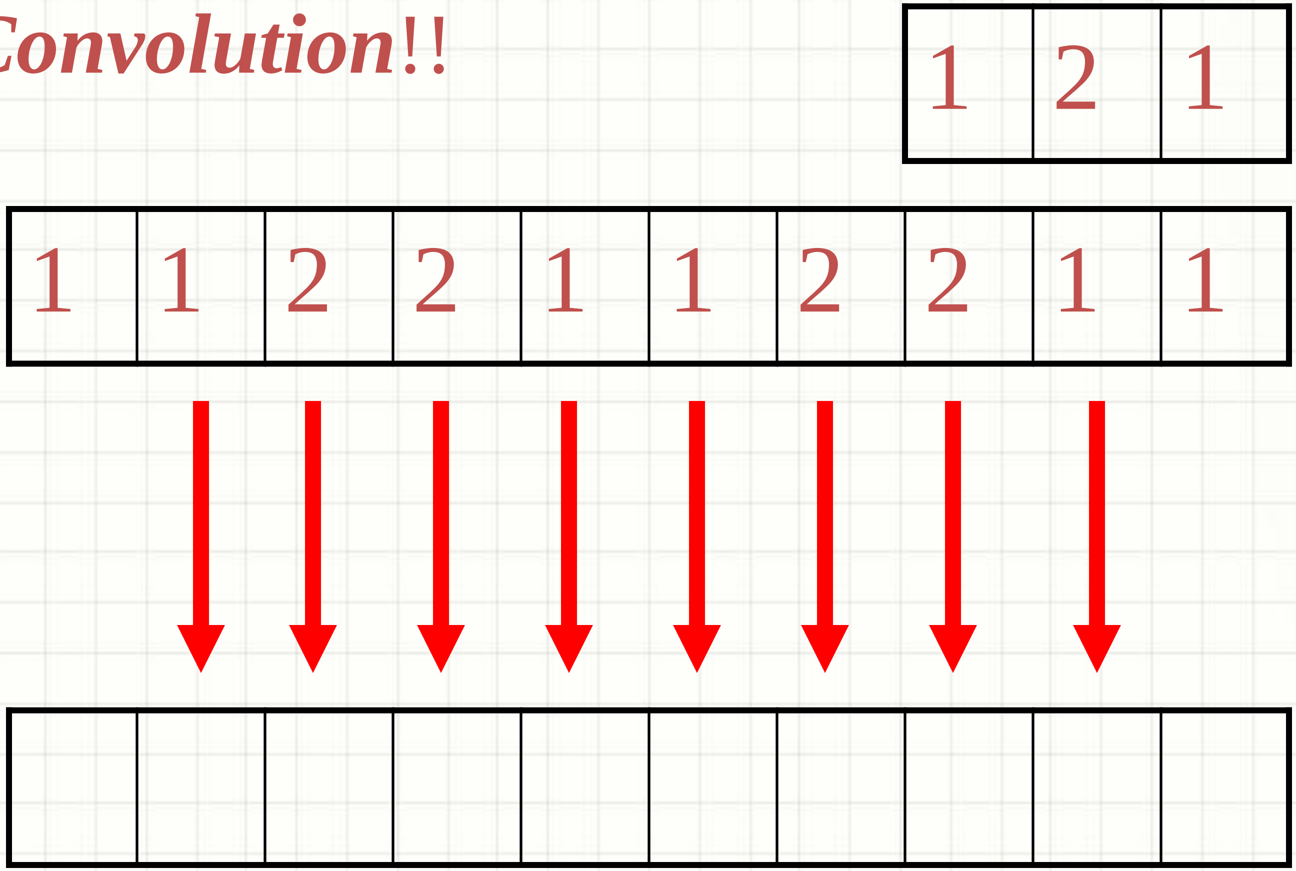


Convolution (1D)



Convolution (1D)

This process is called *Convolution*!!



Math of convolution

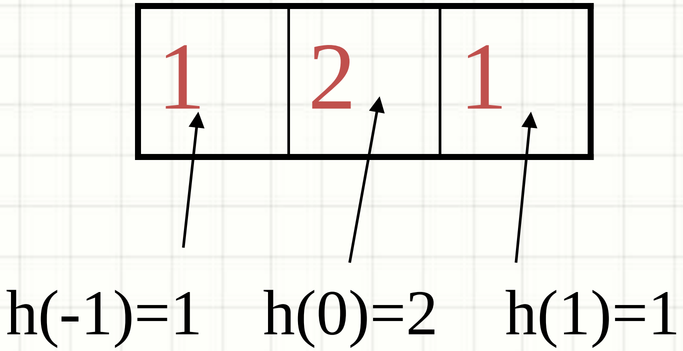
$$\boxed{g(x) = h * f(x) = \sum_{i=-n}^{n} h(i)\, f(x-i)}$$

g(x): output, h: filter, * means convolution,

f(x): input, n = width of filter

For example: Filter (h):

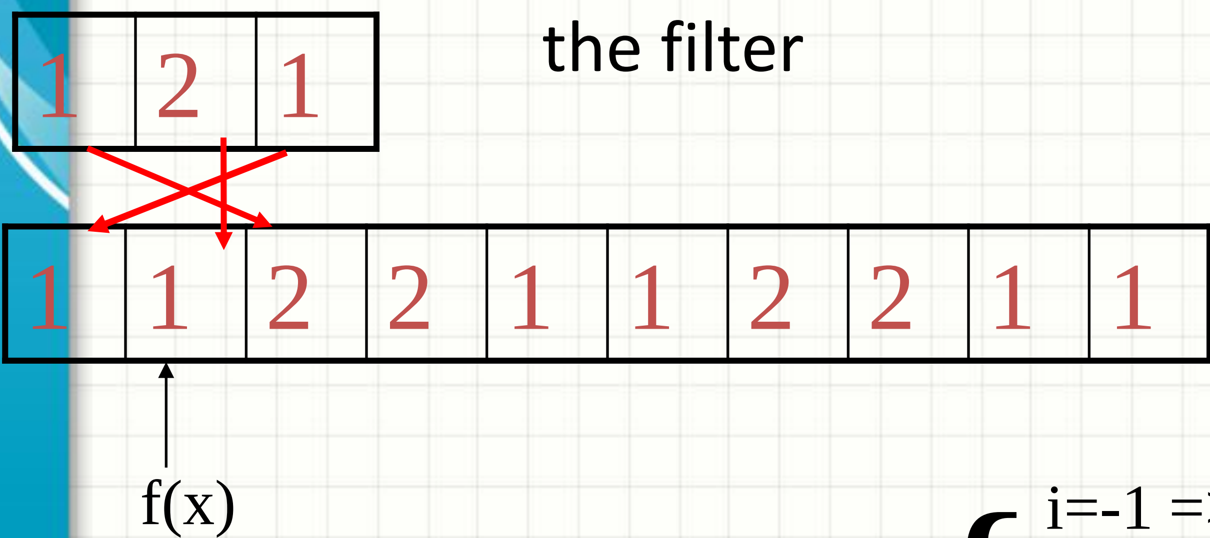
width = 3 => n=1



Math of convolution

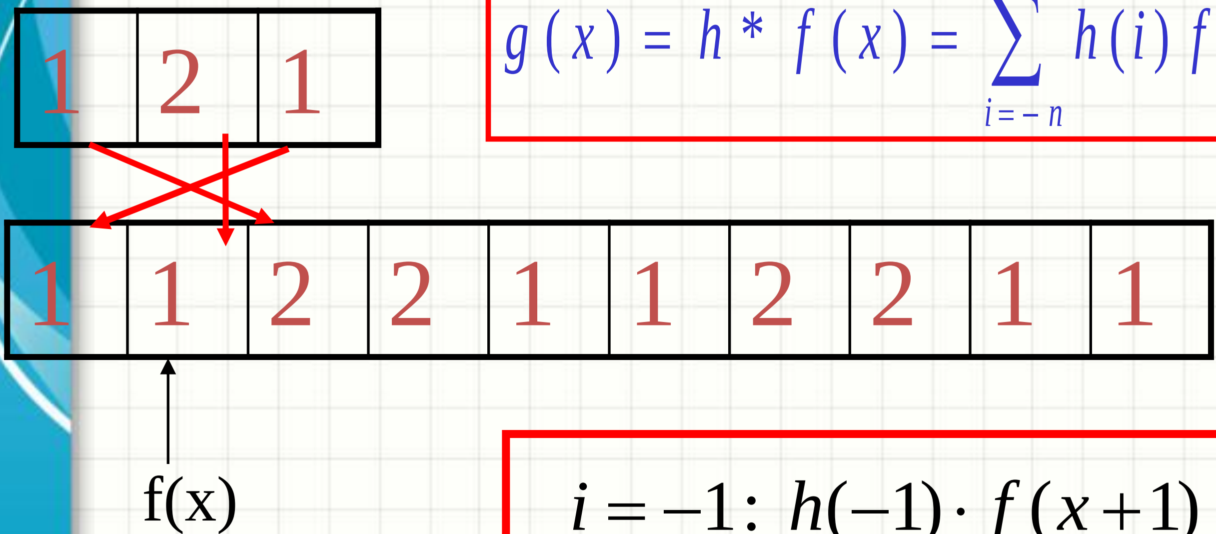
$$g(x) = h * f(x) = \sum_{i=-n}^{n} h(i)\, f(x-i)$$

- x is the pixel of interest, i.e., the position in the signal/image AND the center of the filter



$$n = 1 \Rightarrow i \in \{-1,0,1\} \in \begin{cases} \text{i=-1 => f(x-(-1))= f(x+1)=2} \\ \text{i=0 => f(x-0)= f(x)=1} \\ \text{i=1 => f(x-1)= f(x-1)=1} \end{cases}$$

Math of convolution



$$g(x) = h * f(x) = \sum_{i=-n}^{n} h(i) f(x-i)$$

$$i = -1: \; h(-1) \cdot f(x+1) = 1 \cdot 2 = 2$$

$$i = 0: \quad h(0) \cdot f(x) = 2 \cdot 1 = 2$$

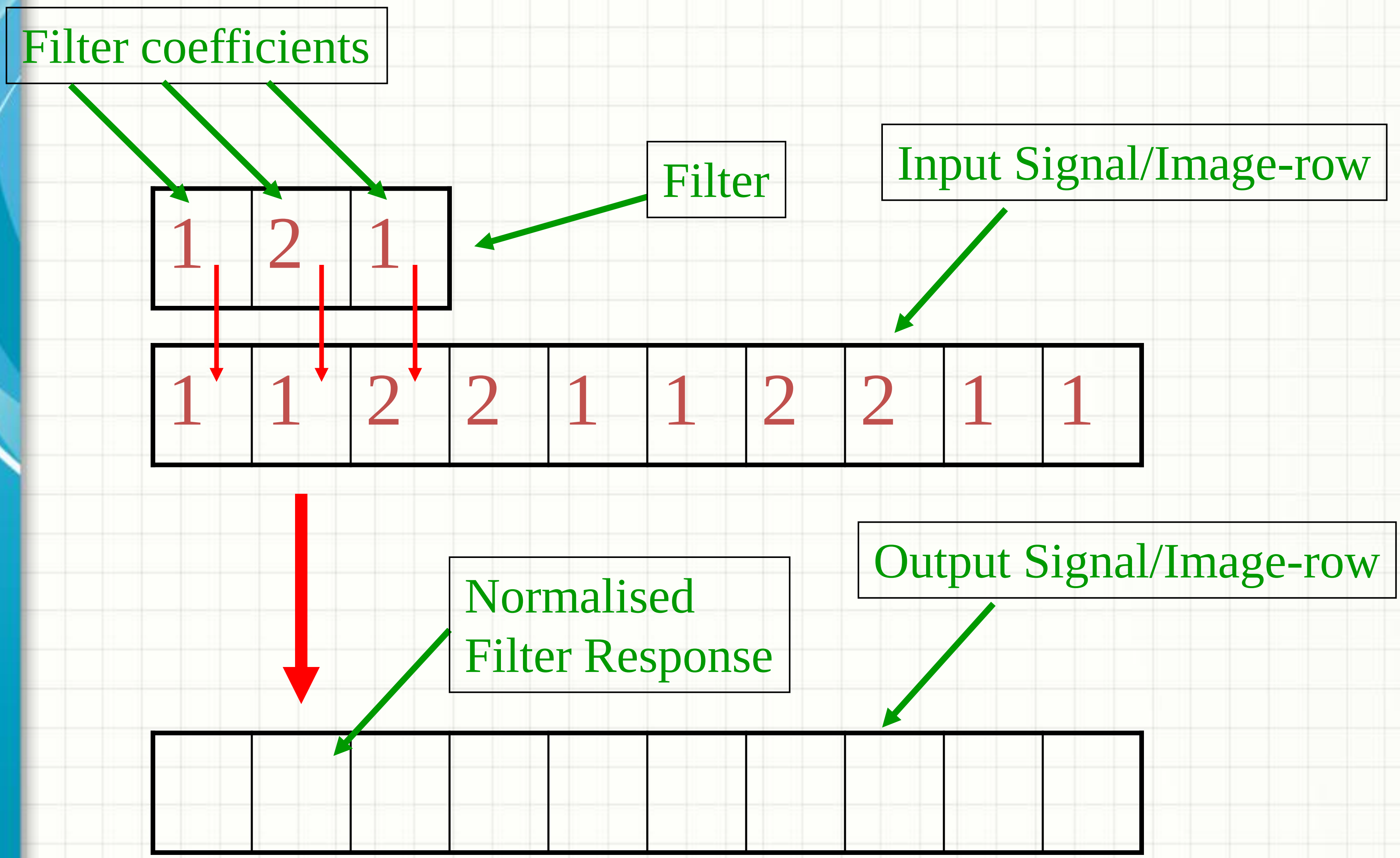
$$i = 1: \quad h(1) \cdot f(x-1) = 1 \cdot 1 = 1$$

$$g(x) = 2 + 2 + 1 = 5$$

$$\text{Normalise: } \; g(x) = 5/4$$

Correlation

Correlation (1D)



Correlation versus Convolution

Correlation

1	2	1

$$g(x) = h \circ f(x) = \sum_{i=-n}^{n} h(i) f(x+i)$$

1	1	2	2	1	1	2	2	1	1

Convolution

1	2	1

$$g(x) = h * f(x) = \sum_{i=-n}^{n} h(i) f(x-i)$$

1	1	2	2	1	1	2	2	1	1

In image processing we use CORRELATION
but (nearly) always call it CONVOLUTION!!!!!

Note: When the filter is symmetric: correlation = convolution!

Convolution/correlation on images(2D)

- The filter is now 2D
- Kernel (mask), kernel coefficients
- Size: **3x3**, 5x5, 7x7, ….

Normalisation

1	1	1
1	1	1
1	1	1

Input

1	2	0	1	3	
2	1	4	2	2	
1	0	1	0	1	
1	2	1	0	2	
2	5	3	1	2	

Output

	$\frac{1}{9}$				

Convolution/correlation on images

1	1	1
1	1	1
1	1	1

Input

1	2	0	1	3	
2	1	4	2	2	
1	0	1	0	1	
1	2	1	0	2	
2	5	3	1	2	

Output

	$\frac{12}{9}$	$\frac{1}{9}$			

Convolution/correlation on images

1	1	1
1	1	1
1	1	1

Input

1	2	0	1	3	
2	1	4	2	2	
1	0	1	0	1	
1	2	1	0	2	
2	5	3	1	2	

Output

	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$		

Convolution/correlation on images

1	1	1
1	1	1
1	1	1

Input

1	2	0	1	3	
2	1	4	2	2	
1	0	1	0	1	
1	2	1	0	2	
2	5	3	1	2	

Output

	[illegible]/9	[illegible]/9	[illegible]/9		
	[illegible]/9	This image cannot currently be displayed.	[illegible]/9		
	[illegible]/9	[illegible]/9	[illegible]/9		

Math. of 2D Convolution/Correlation

Convolution

Correlation

Note: When the filter is symmetric: correlation = convolution!

1	1	1
1	1	1
1	1	1

2	3	2
-1	0	-1
2	3	2

Rank Filters

Rank Filters

- Not based on convolution but still neighborhood processing
- Principle:
  - Define a mask, e.g., 3x3
  - Sort all pixel-values within the mask into ascending order
  - Select a pixel-value according to the filter type: Median, min., max., range, …

Median Filter

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

Sorted: 0,0,1,1,1,2,2,2,4

Input

1	2	0	1	3	
2	2	4	2	2	
1	0	1	0	1	
1	2	1	0	2	
2	5	3	1	2	

Output

	1				

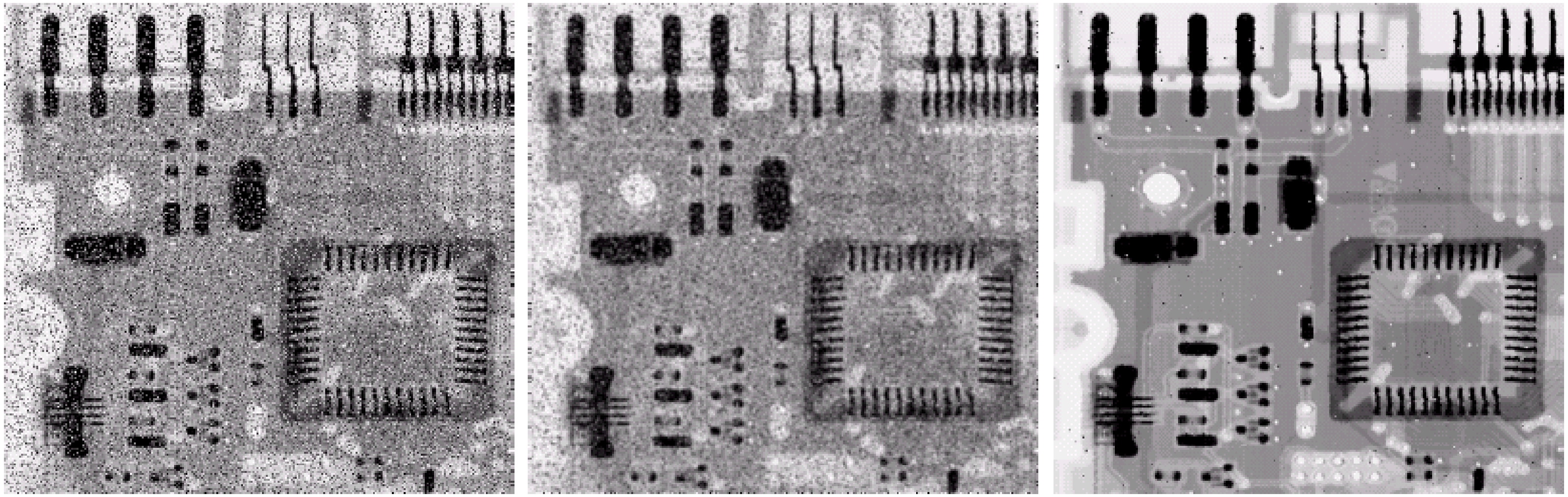
Median Filter

- Median Filter
  - Good for cleaning salt-and-pepper noise

(show: boats, add noise: salt/pepper, Median (size=1))

Median Filter

- Median Filter
  - Better than the mean filter as blurring is minimized and edges stay sharp



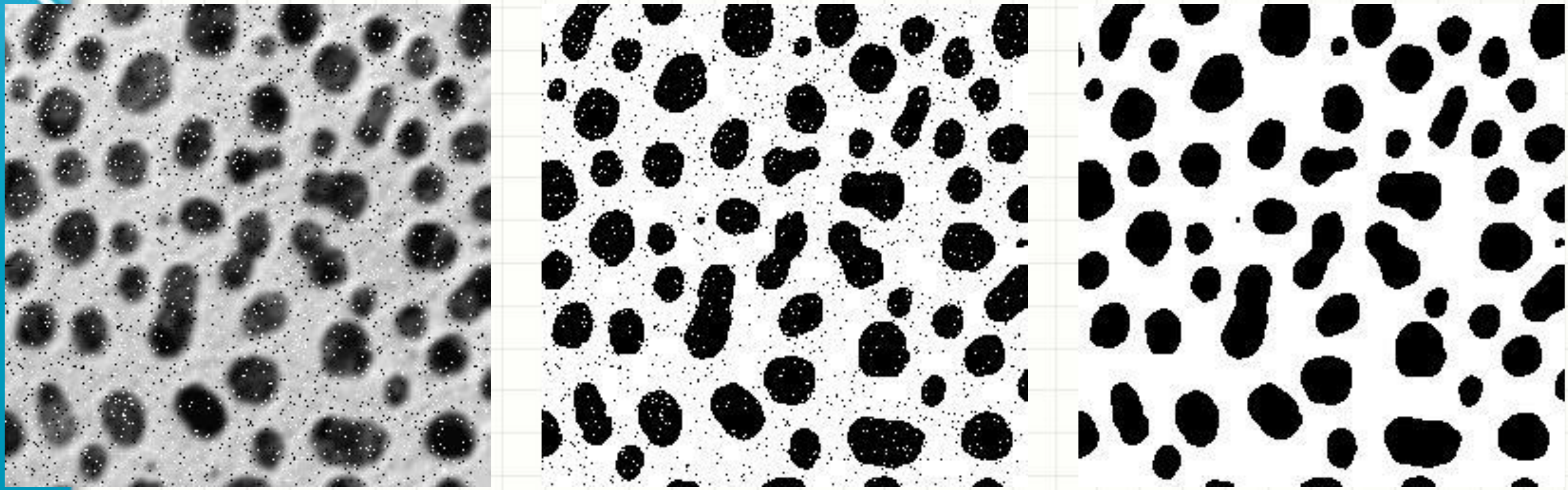
a b c

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.)

Median Filter

- Good at removing noise in binary images

Input Thresholded Median filtered



Nonlinear spatial filtering

- Nonlinear spatial filters also operate on neighborhoods, and the mechanics of sliding a mask past an image are the same as was just outlined.

- The filtering operation is based conditionally on the values of the pixels in the neighborhood under consideration

Blurring the image

- Also know as: Smoothing kernel, Mean filter, Low pass filter
- The simplest filter:
  - **Spatial low pass filter**

1	1	1

1	1	1
1	1	1
1	1	1

- Another mask:
  - **Gaussian filter:**

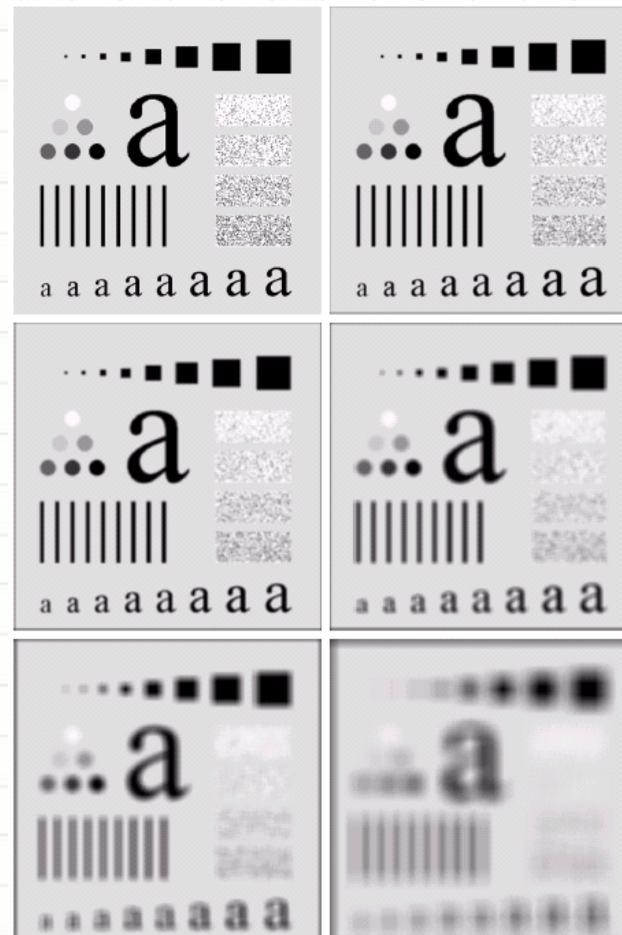
1	2	1

$\frac{1}{16}$

1	2	1
2	4	2
1	2	1

Applications of blurring

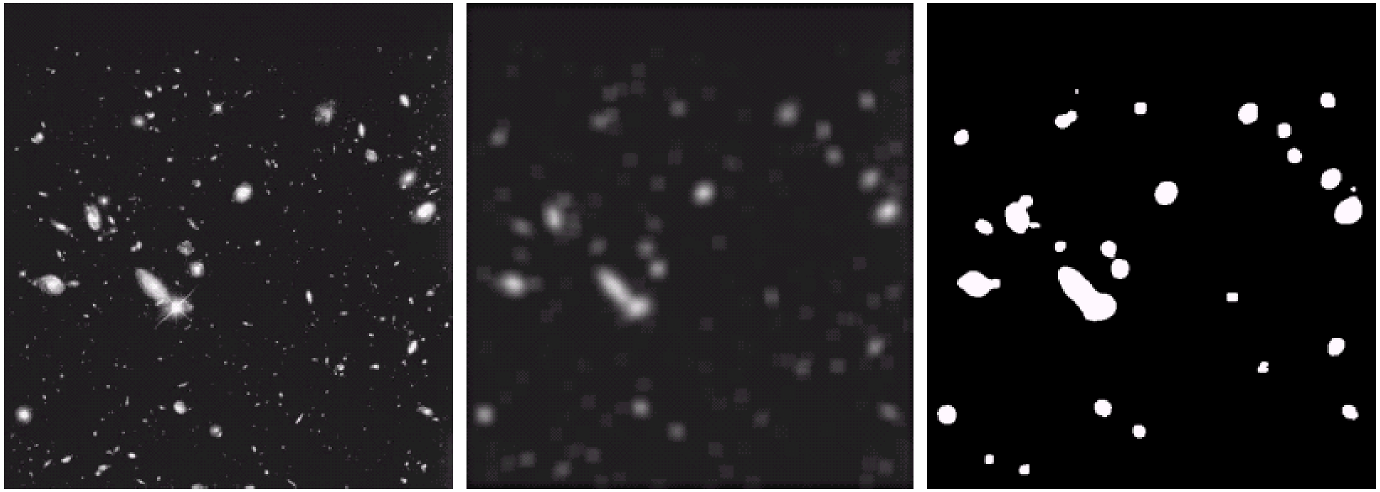
- Blurring to remove identity or other details
- Degree of blurring = kernel size



Show: camera, mean, convolution

Applications of blurring

- Preprocessing: enhance objects
- blurring + Thresholding



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.)

Smoothing Spatial Filters

- Smoothing filters are used for blurring and for noise reduction.
  - Blurring is used in preprocessing steps, 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

Type of smoothing filtering

- There are 2 way of smoothing spatial filters

  – Smoothing Linear Filters

  – Order-Statistics Filters

Smoothing Linear Filters

- Linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask.

- Sometimes called “**averaging filters**”.

- The idea is replacing the value of every pixel in an image by the **average of the gray levels** in the neighborhood defined by the filter mask.

Two 3x3 Smoothing Linear Filters

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

Standard average

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

Weighted average

5x5 Smoothing Linear Filters

$$\frac{1}{25}\times$$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Smoothing Linear Filters

- The general implementation for filtering an MxN image with a weighted averaging filter of size mxn is given by the expression

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

Result of Smoothing Linear Filters

Original Image

[3x3]

[5x5]

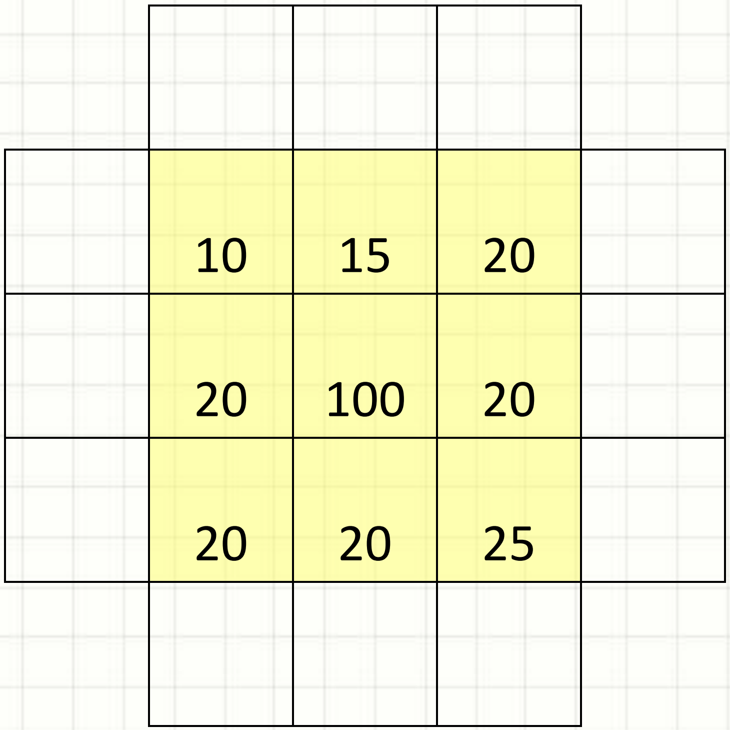
[7x7]

Order-Statistics Filters

- Order-statistics filters are nonlinear spatial filters whose response 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 determined by the ranking result.

- Best-known “median filter”

Process of Median filter

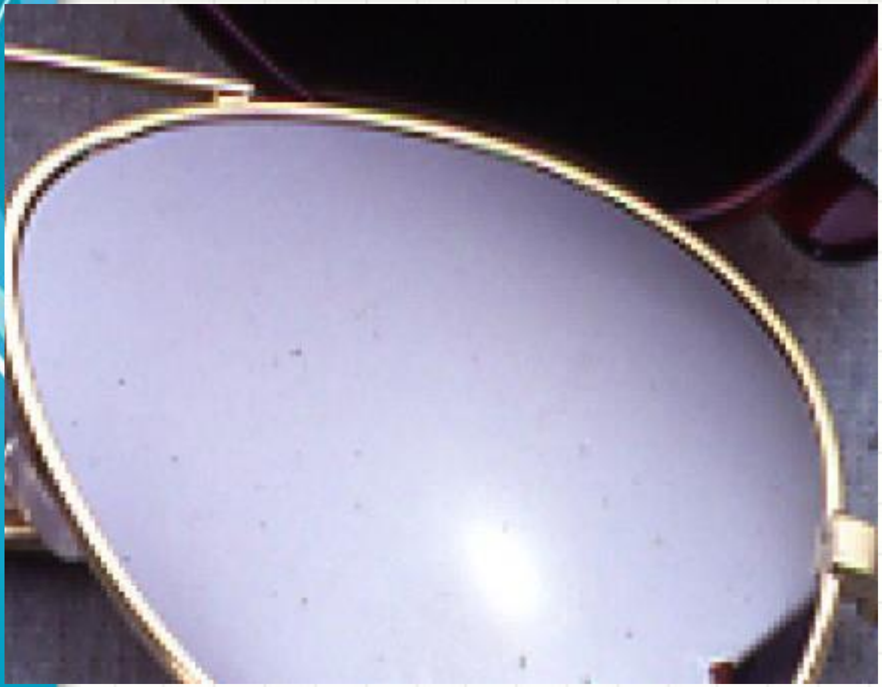


10, 15, 20, 20, 20, 20, 20, 25, 100

5^{th}

- Corp region of neighborhood
- Sort the values of the pixel in our region
- In the MxN mask the median is MxN div 2 +1

Result of median filter



Noise from Glass effect

Remove noise by median filter

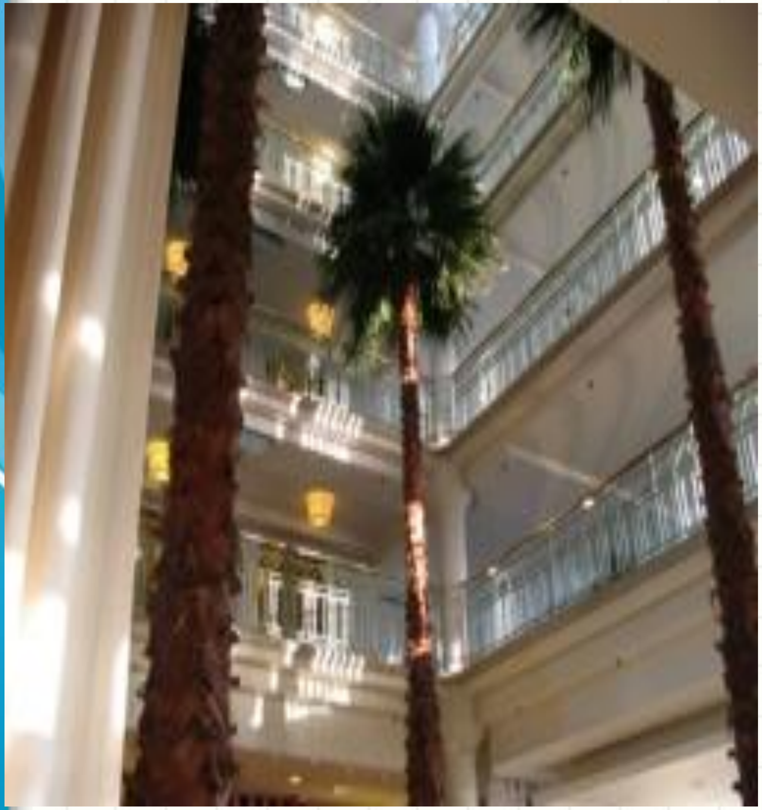
Other Filters

- The **Maximum filter** enhances bright values in the image by increasing its area. Similar to a dilate function each 3x3 (or other window size) is processed for the brightest surrounding pixel. That brightest pixel then becomes the new pixel value at the center of the window.
- For example, given the grayscale 3x3 pixel neighborhood;

22	77	48
150	77	158
0	77	219

- The center pixel would be changed from 77 to 219 as it is the brightest pixel within the current window.

SOURCE IMAGE



OUTPUT IMAGE

Minimum Filter:

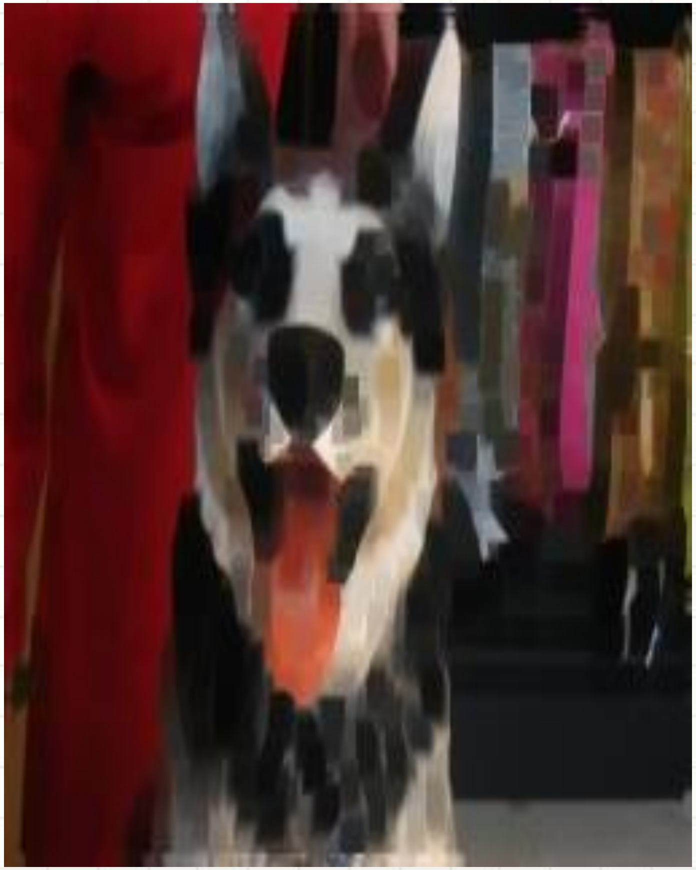
- The Minimum filter enhances dark values in the image by increasing its area. Similar to a dilate function each 3x3 (or other window size) is processed for the darkest surrounding pixel. That darkest pixel then becomes the new pixel value at the center of the window.
- For example, given the grayscale 3x3 pixel neighborhood;

22	77	48
150	77	158
0	77	219

- The center pixel would be changed from 77 to 0 as it is the darkest pixel within the current window.

SOURCE IMAGE

OUTPUT IMAGE



Midpoint Filter

- The Midpoint filter blurs the image by replacing each pixel with the average of the highest pixel and the lowest pixel (with respect to intensity) within the specified window size.
- For example, given the grayscale 3x3 pixel neighborhood;

22	77	48
150	77	158
0	77	219

- The center pixel would be changed from 77 to 109 as it is the midpoint between the brightest pixel 219 and the darkest pixel 0 within the current window.
- Midpoint = (darkest+lightest)/2

SOURCE IMAGE

OUTPUT IMAGE

Disadvantage of above filter

- Blur image Due to which the edges of the images were lost, edges in the image are the important information which need to be preserved while preprocessing.

Linear Spatial Filtering

```
g=imfilter(f,w,filtering_mode,...
                 boundary_options,size_
```

where f is the input image, w is the filter masl
filtered result, and the other parameters are s
the following table.