U3 Image Filtering: Spatial Domain

SJK002 Computer Vision

Master in Intelligent Sytems

UNIVERSITAT JAUME•I

Index

- Filtering in the the image spatial domain
  - Linear filters. Convolution
  - Noise types
  - Mean filter
  - Dealing with image boundaries
  - Gaussian filters
  - Median filter

Image filtering

- Image filtering cosists of eliminating certain elements of the image


- There are two image filtering types:
  - Filtering in the spatial domain
  - Filtering in the frequency domain

Linear filters

- Many image processing operations can be modelled as a linear system

Input
f(x,y)

Linear system
T

Output
h(x,y)

$$\text{Input } f(x,y) \Rightarrow \boxed{\text{Linear system } T} \Rightarrow \text{Output } h(x,y)$$

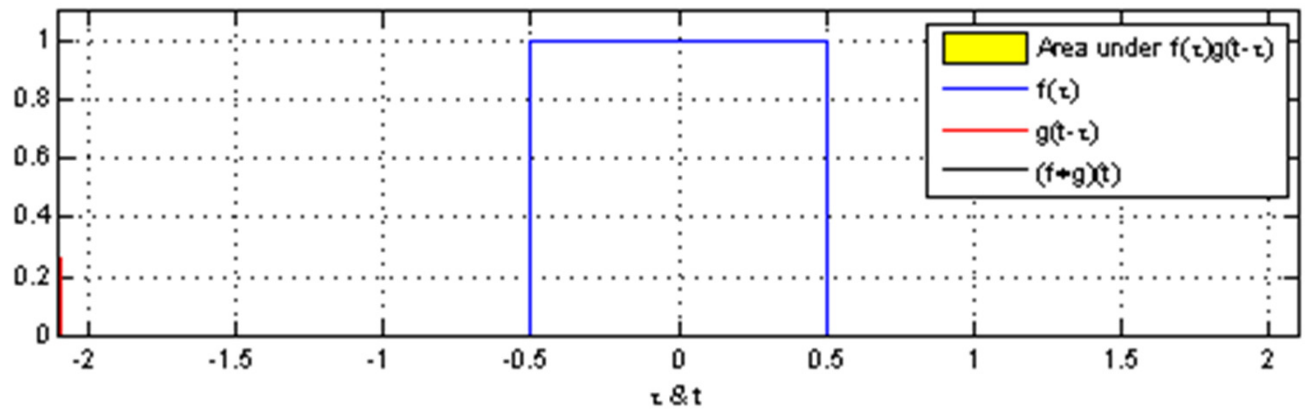
- Superposition principle:

$$T[a_1 f_1(x,y) + a_2 f_2(x,y)] = a_1 T[f_1(x,y)] + a_2 T[f_2(x,y)]$$

Convolution

- In a linear system, the convolution can be defined as:

$$(f * g)(x) = \int_{-\infty}^{+\infty} f(\alpha) g(x - \alpha) d\alpha$$



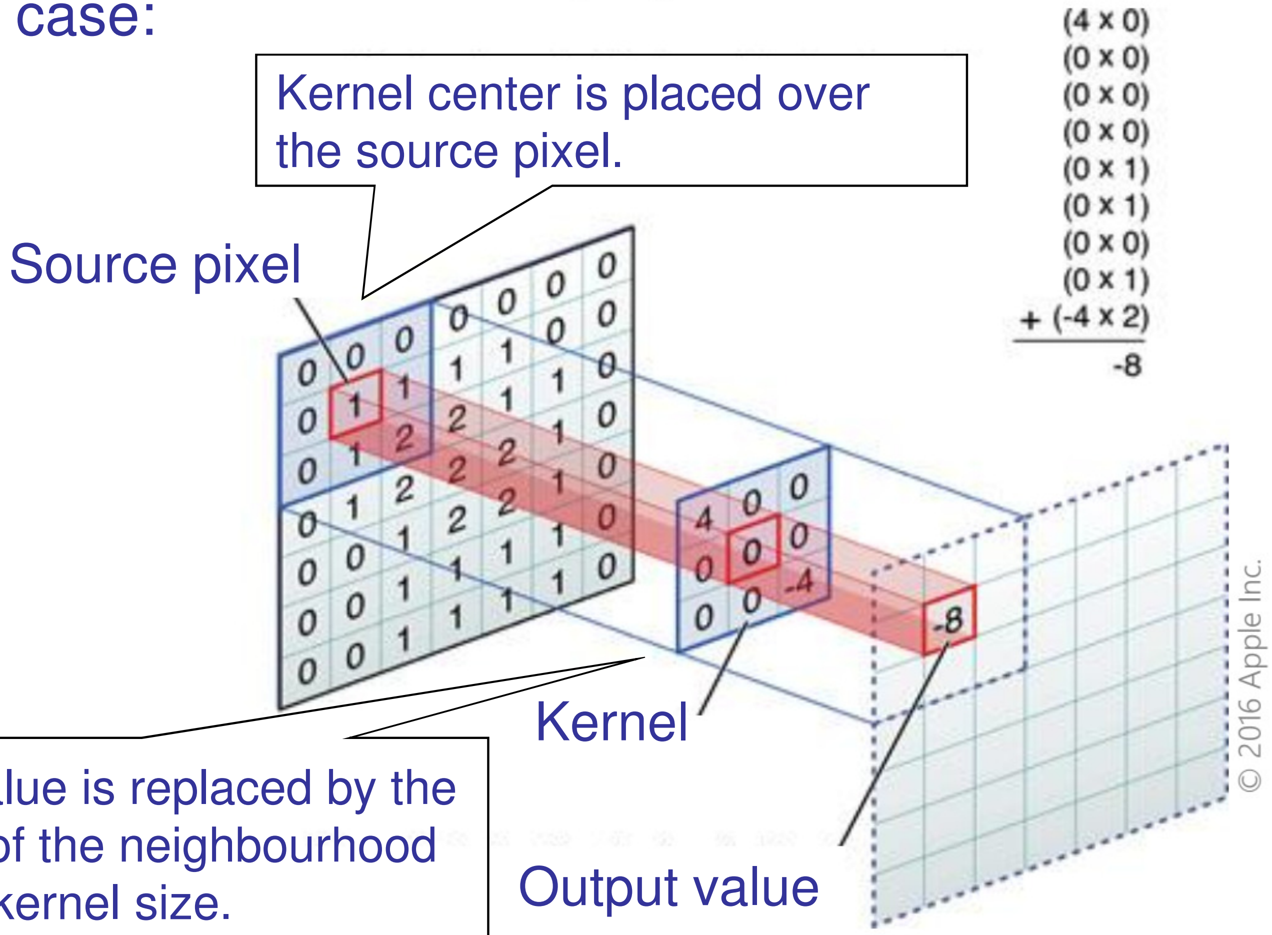
- In the discrete case:

$$(f * g)(i) = \sum_{k=-\infty}^{+\infty} f(k) g(i - k)$$

Convolution

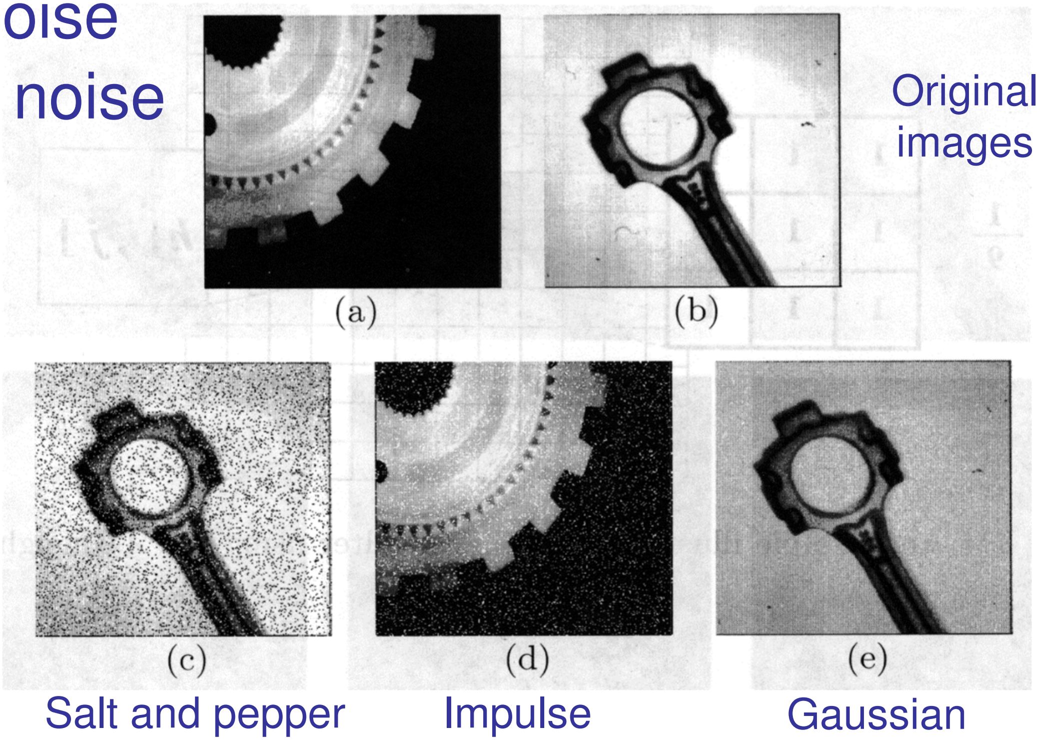
$$(K * I)(x, y) = \sum_i \sum_j K(i, j) I(x - i, y - j)$$

- 2D discrete case: images



Noise types

- **Noise**: Non desirable information that contaminates the image
- Some noise types:
  - Salt and pepper / Random
  - Impulse noise
  - Gaussian noise



Mean filter

- Linear filter
- The output value is the mean of the pixels in the neighbourhood

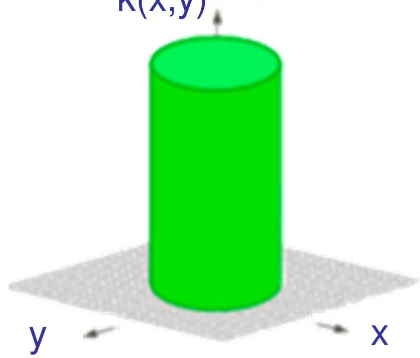
$$h[i,j] = \frac{1}{M} \sum_{(k,l) \in N} f[k,l]$$

(N: window size, M: nº pixels in N)

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- 3x3 neighbourhood:
- Normalize coefficients to 1 → Smoothing filter
- Output: reduce contrast
- Drawback: looses border information

Mean filter



Mean filter transfer function k(x,y)

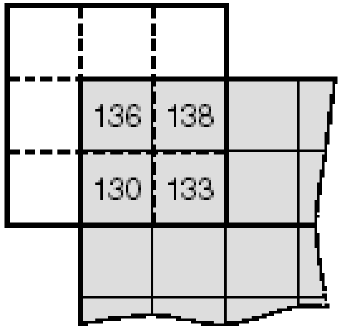
original

disk: radius=1

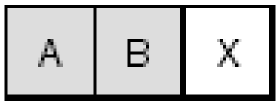
disk: radius=10

Dealing with image boundaries

- Extend the image



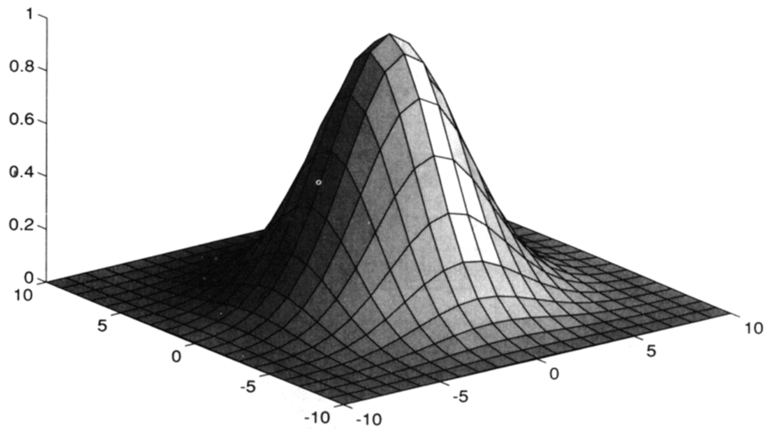
- Use the nearest neighbour value.
- Other complex models:
  - Interpolation.
  - Average:



$X = B + (B-A)$

Gaussian filters

- Linear filters with weights from a Gaussian function
- Very much used
- Good to eliminate Guassian noise
- Used in border detection as a pre-filtering step



Gaussian filters

- One-dimensional Gaussian function:

$$g(x) = e^{-\frac{x^2}{2\sigma^2}}$$

  zero mean and standard deviation σ.

- 2D Gaussian function for image processing:

$$g[x, y] = e^{-\frac{(x^2 + y^2)}{2\sigma^2}}$$

Gaussian filters

- Example:
  - Window size 5x5
  - Standard deviation σ=1 pixel
  - Window center represents (x,y)=(0,0)
  - Sample the Gaussian function at each kernel (window) element.
  - Normalize kernel to 1

Normalization constant

y \ x	-2	-1	0	1	2
-2	0.00	0.01	0.02	0.01	0.00
-1	0.01	0.06	0.10	0.06	0.01
0	0.02	0.10	0.16	0.10	0.02
1	0.01	0.06	0.10	0.06	0.01
2	0.00	0.01	0.02	0.01	0.00

$\times \frac{1}{0.96}$

Gaussian filters. Properties

- Rotational symmetry: same effect in any direction
- Single lobe (peak/maximum)
  - Pixel weights decrease with distance to the center
  - The farer the pixel, the less significant weight is
- Preserve low spatial frequencies and tend to eliminate high frequencies: low pass filter.
- The filtering “degree”/intensity is controlled by σ

Linear filters comparison

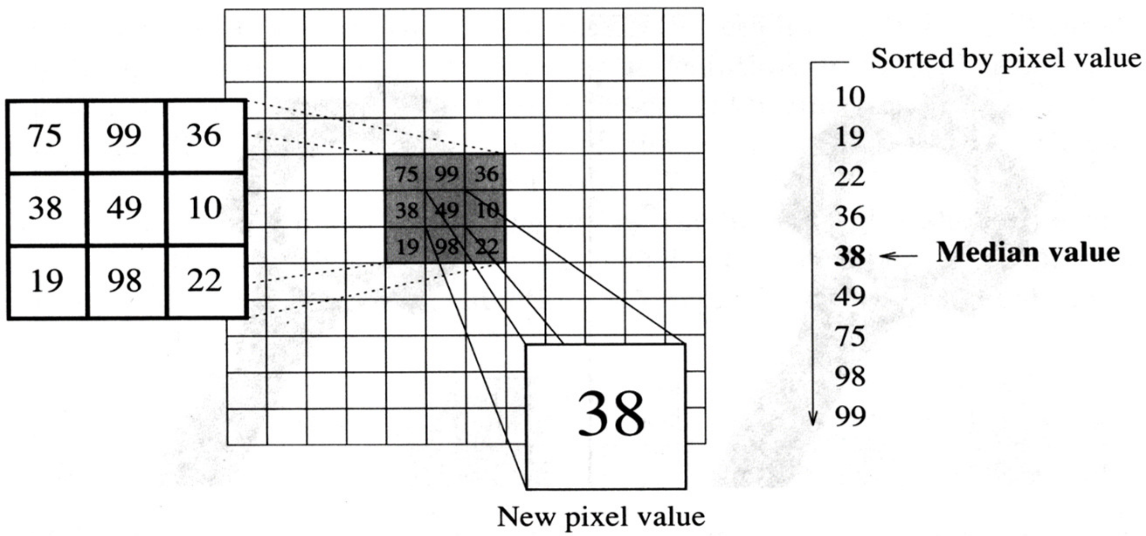
Mean filter | Gaussian filter

Image with Gaussian noise

Image with random noise

Median filter

- Pixel output value is calculated as:
  - Sort the N pixel values of the neighbourhood
  - Select the value which is in the middle of the sorted list
- It is not a linear filter.



Median filter

- Better option for salt-and-pepper and impulse noise
- Preserves better objects border information

Median filter | Mean filter

Image with Gaussian noise

Image with random noise

Filters comparison

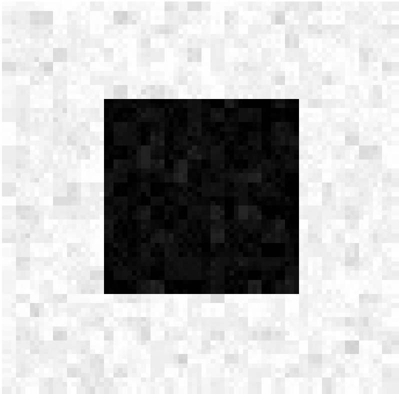
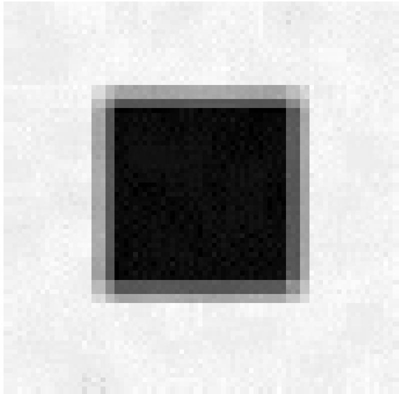
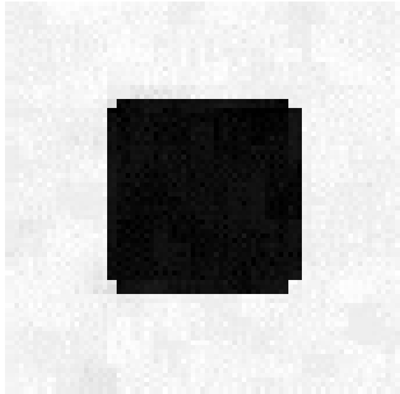


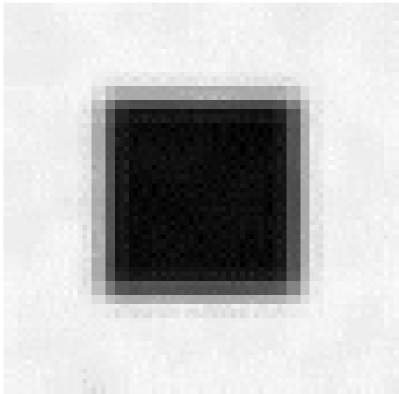
Image with noise



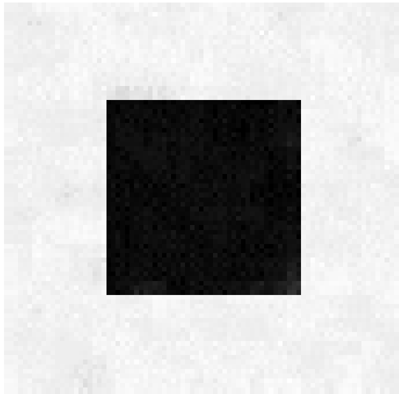
Mean



Median



Gaussian



Anisotropic