Image : filtering in the spatial domain

Image and signal processing

Samia Bouchafa-Bruneau

M1 E3A - UEVE/Upsay

Introduction

- **Two kind of filters :**
  1. Linear Filters :
     - Consists of a convolution between the input image and a given filter (convolution kernel) in the spatial domain. The spatial filter can be considered as the impulse response of the frequency domain filter.
     - Filter frequency cut-off correspond to the filter dimension.
  2. Non linear Filters :
     - Non linear function of the neighborhood.

2D Discrete Convolution

- **2D Convolution**

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

$$f_e(x,y) * g_e(x,y) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f_e(m,n)g_e(x-m,y-n)$$

- Important property of convolution :

$$\Im\{f(x,y) * g(x,y)\} = F(u,v)\cdot G(u,v)$$

$$\Im\{f(x,y) \cdot g(x,y)\} = F(u,v) * G(u,v)$$

2D Discrete Convolution

- **Example in 1D :**

$A \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \end{bmatrix}$ $*$ $B \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$

1	2	3	4	5
3	2	1		
3	4	3		
	10			

1	2	3	4	5
	3	2	1	
	6	6	4	
	10	16	22	

2D Discrete Convolution

- **Managing image edges**
  Many possibilities (depending on the application) :

1. Zero padding
2. Reflection
3. Default

2D Discrete Convolution

- **Example in 2D :**

$$\begin{array}{|c|c|c|c|c|} \hline 10 & 5 & 20 & 20 & 20 \\ \hline 10 & 5 & 20 & 20 & 20 \\ \hline 10 & 5 & 20 & 20 & 20 \\ \hline 10 & 5 & 20 & 20 & 20 \\ \hline 10 & 5 & 20 & 20 & 20 \\ \hline \end{array} * \begin{array}{|c|c|} \hline -1 & 1 \\ \hline 0 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|c|c|} \hline -10 & 15 & -15 & 0 & 0 \\ \hline -10 & 15 & -10 & 20 & 20 \\ \hline -10 & 15 & -10 & 20 & 20 \\ \hline -10 & 15 & -10 & 20 & 20 \\ \hline -10 & 15 & -10 & 20 & 20 \\ \hline \end{array}$$

We have to consider the symmetric of $\begin{array}{|c|c|} \hline -1 & 1 \\ \hline 0 & 1 \\ \hline \end{array} = \begin{array}{|c|c|} \hline 1 & 0 \\ \hline 1 & -1 \\ \hline \end{array}$

- **Properties**
  - Symmetry : $\rightarrow$ for low pass filters generally.
  - Separability :
    $g(x,y) = [f(x,y) * h_x(x)] * h_y(x) = f(x,y) * [h_x(x) * h_y(x)]$ with $h_{xy}(x,y) = h_x(x) * h_y(x)$

    Example : $\frac{1}{3}\begin{bmatrix}1&1&1\end{bmatrix} * \frac{1}{3}\begin{bmatrix}1\\1\\1\end{bmatrix} = \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$
  - Cascade Filters :
    $g(x,y) = f(x,y) * h_1(x,y) * h_2(x,y) * h_3(x,y) * ... = f(x,y) * h(x,y)$
    with $h(x,y) = h_1(x,y) * h_2(x,y) * h_3(x,y) * ...$
  - Linear Combination of filters : $f(x,y) * h_1(x,y) + f(x,y) * h_2(x,y) = f(x,y) * [h_1(x,y) + h_2(x,y)] = f(x,y) * h(x,y)$ with $h(x,y) = h_1(x,y) + h_2(x,y)$.

    Example : $\begin{bmatrix}1&1&1\\-1&2&-1\\-1&-1&-1\end{bmatrix} = \begin{bmatrix}0&0&0\\-1&2&-1\\0&0&0\end{bmatrix} + \begin{bmatrix}1&1&1\\0&0&0\\-1&-1&-1\end{bmatrix}$

Low pass filtering

- **We use low pass filtering for :**
  - Smoothing
  - Reducing noise
  - Blurring the image

Low pass filtering

Average filtering

- Example of average filters of size 3×3 and 5×5 :

$$(1/9) \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad (1/25) \times \begin{array}{|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

- These filters are generally :
  - normalized
  - centered and
  - symmetric.

Low pass filtering

Average filtering

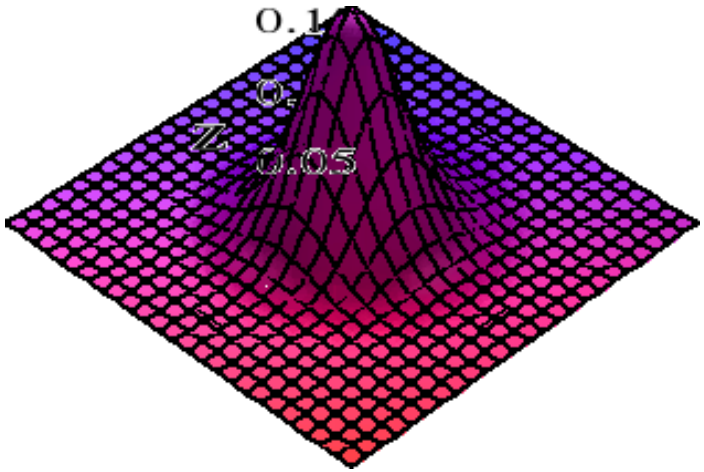
- Results with a 5×5 and 7×7 average filter :

- Reduce noise but also details $\rightsquigarrow$ blurring.

Low pass filtering

Gaussian Filter

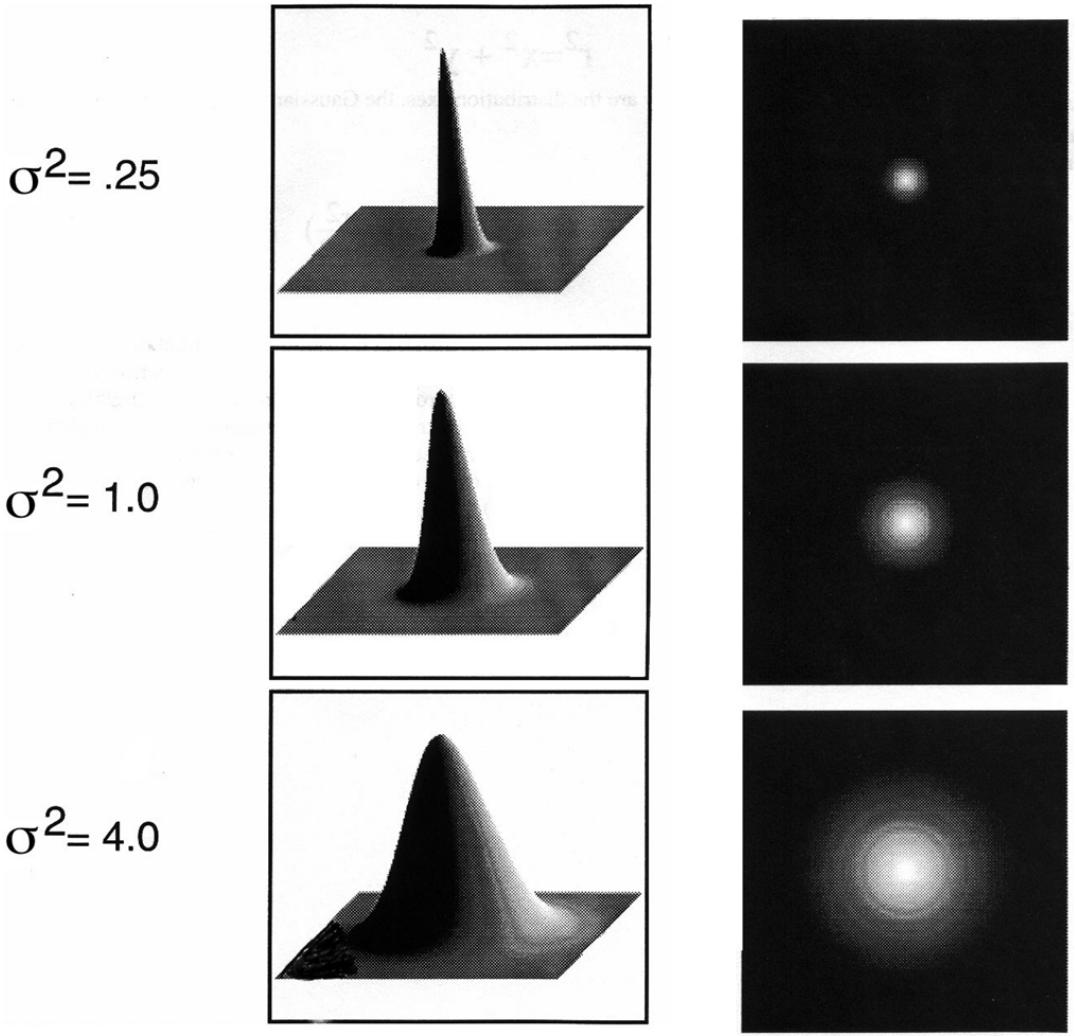
$$G(x,y) = \left[\frac{1}{2\pi\sigma^2}\exp\left(-\frac{(x^2+y^2)}{2\sigma^2}\right)\right]_{\sigma=1} = \frac{1}{2\pi}e^{-\frac{1}{2}x^2-\frac{1}{2}y^2}$$



Low pass filtering

Gaussian Filter

- **Varying the std :**



- Discrete Gaussian filter of size n : $W(i,j) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i^2+j^2)}{2\sigma^2}\right) = k \exp\left(-\frac{(i^2+j^2)}{2\sigma^2}\right)$ with $k = \frac{1}{2\pi\sigma^2}$. Then : $\frac{W(i,j)}{k} = \exp\left(-\frac{(i^2+j^2)}{2\sigma^2}\right)$
- Example for $n = 7$ and $\sigma^2 = 2$:

$\sigma^2 = 2$ n = 7

i \ j	-3	-2	-1	0	1	2	3
-3	.011	.039	.082	.105	.082	.039	.011
-2	.039	.135	.287	.368	.287	.135	.039
-1	.082	.287	.606	.779	.606	.287	.082
0	.105	.039	.779	1.000	.779	.368	.105
1	.082	.287	.606	.779	.606	.287	.082
2	.039	.135	.287	.368	.287	.135	.039
3	.011	.039	.082	.105	.082	.039	.011

$\frac{W(1,2)}{k} = \exp(-\frac{1^2+2^2}{2*2})$

Représentation entière, poser k=91 pour rendre ce coefficient unitaire

Low pass filtering

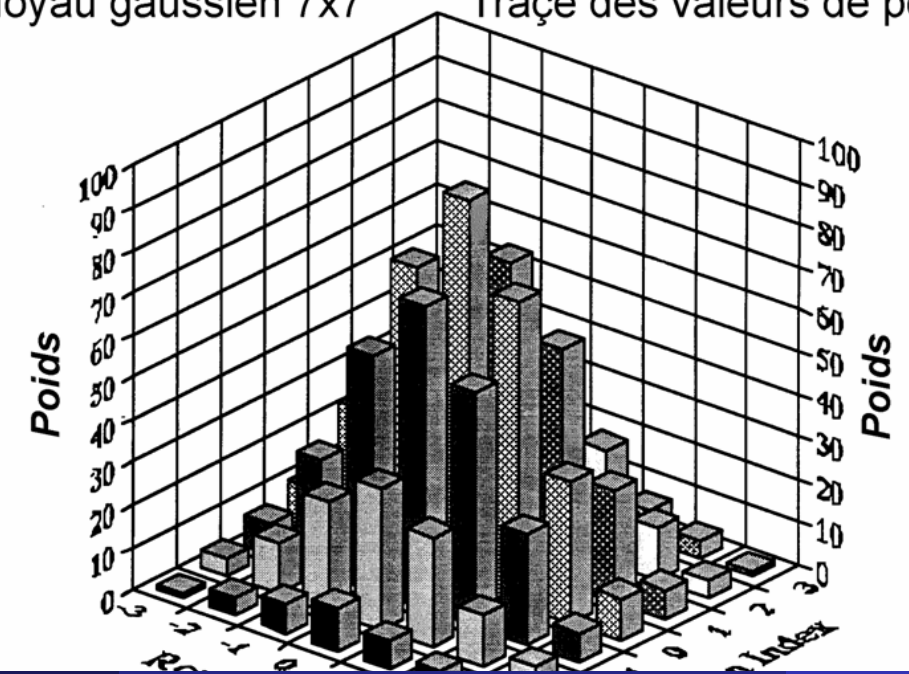
Discrete Gaussian Filter

1	4	7	10	7	4	1
4	12	26	33	26	12	4
7	26	55	71	55	26	7
10	33	71	91	71	33	10
7	26	55	71	55	26	7
4	12	26	33	26	12	4
1	4	7	10	7	4	1

$$\sum_{i=-3}^{3} \sum_{j=-3}^{3} W(i,j) = 1{,}115$$

Noyau gaussien 7x7

Traçé des valeurs de poids



Low pass filtering

Binomial Filters

- Discrete form of the Gaussian filter.
- Binomial filters and Gaussian filter are equivalent for high sizes of filters.
- Binomial Coefficients : $\begin{pmatrix} n \\ k \end{pmatrix} = C_n^k = \frac{n!}{k!(n-k)!}$
- Pascal's triangle : $\begin{pmatrix} n \\ k \end{pmatrix} + \begin{pmatrix} n \\ k+1 \end{pmatrix} = \begin{pmatrix} n+1 \\ k+1 \end{pmatrix}$
- The filter coefficients are those of the Pascal's triangle :

n/k	0	1	2	3	4	5
1	1	1				
2	1	2	1			
3	1	3	3	1		
4	1	4	6	4	1	
5	1	5	10	10	5	1

- **1D binomial filter :**
  1. $\boxed{1}\boxed{1} * \boxed{1}\boxed{1} = \boxed{1}\boxed{2}\boxed{1}$
  2. $\boxed{1}\boxed{1} * \boxed{1}\boxed{1} * \boxed{1}\boxed{1} = \boxed{1}\boxed{3}\boxed{3}\boxed{1}$
  3. etc.
- **For 2D binomial filters :** we use the separability property of filters
  1. $\boxed{1}\boxed{2}\boxed{1} * \begin{array}{|c|}\hline 1\\\hline 2\\\hline 1\\\hline\end{array} = \begin{array}{|c|c|c|}\hline 1&2&1\\\hline 2&4&2\\\hline 1&2&1\\\hline\end{array}$
  2. $\boxed{1}\boxed{3}\boxed{3}\boxed{1} * \begin{array}{|c|}\hline 1\\\hline 3\\\hline 3\\\hline 1\\\hline\end{array} = \begin{array}{|c|c|c|c|}\hline 1&3&3&1\\\hline 3&9&9&3\\\hline 3&9&9&3\\\hline 1&3&3&1\\\hline\end{array}$

Low pass filtering

2D Binomial Filters

- Comparison between binomial filter and average filter of size 5×5 :

Low pass filters

Comparison between Average and Gaussian filters

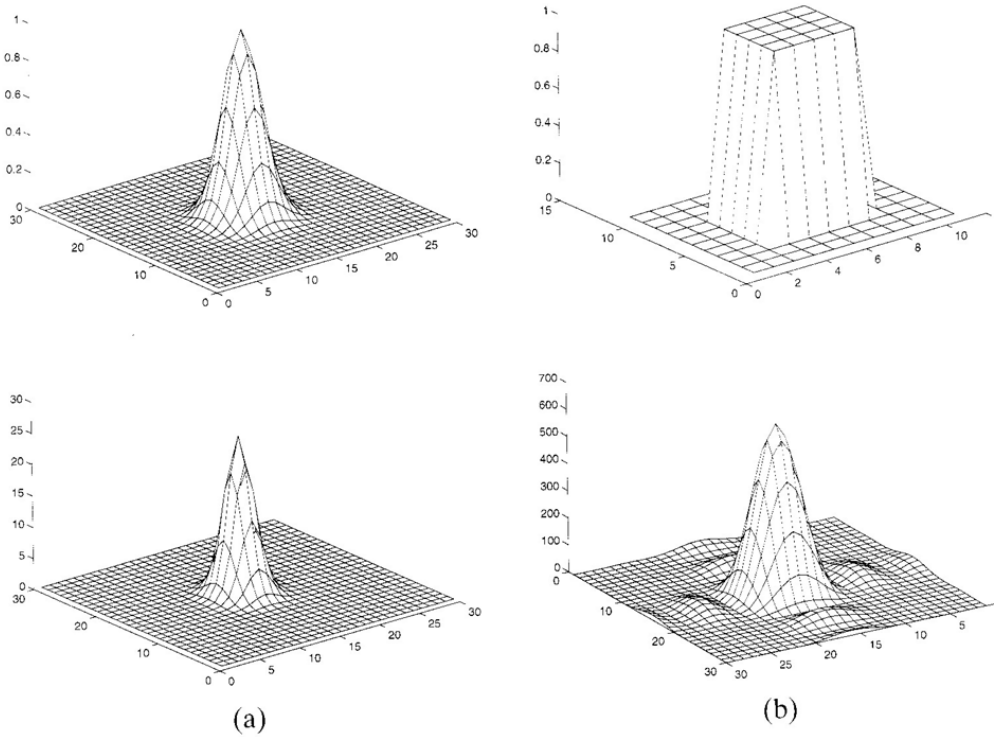


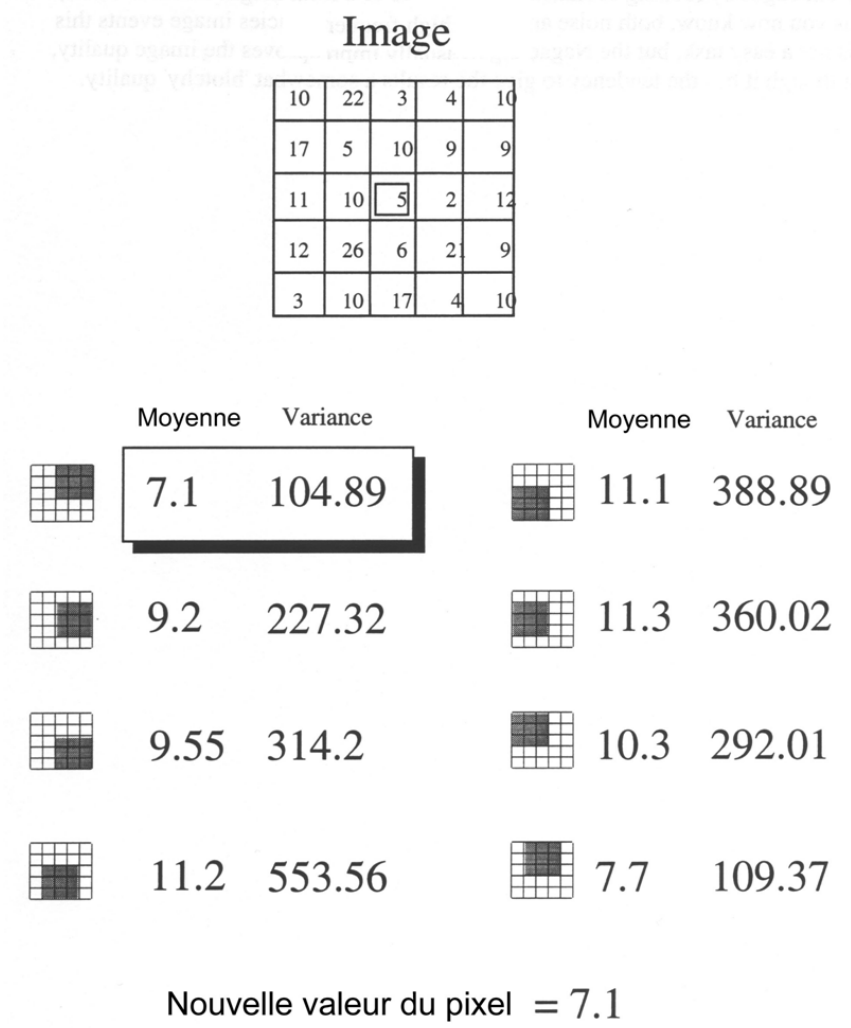
Figure 3.3 (a) The plot of a 5×5 Gaussian kernel of width 5 (top) and its Fourier transfor (bottom). (b) The same for a mean-filter kernel.

- In low frequency, both filters have the same shape. In high frequency, Gaussian filter tend to 0 while the average filter rewind.

Low pass filtering

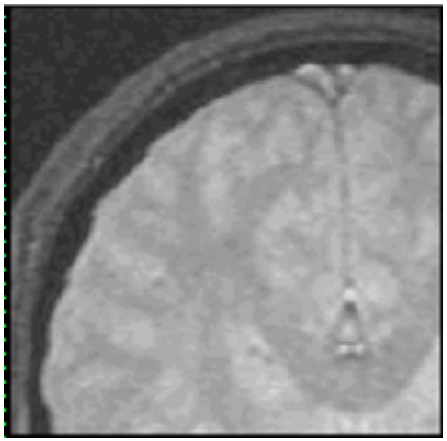
Nagao Filter

- **Principle :** replace each pixel value by the mean of the block with minimum std.

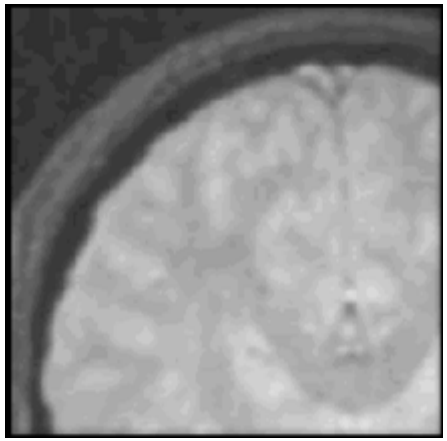


Low pass filtering

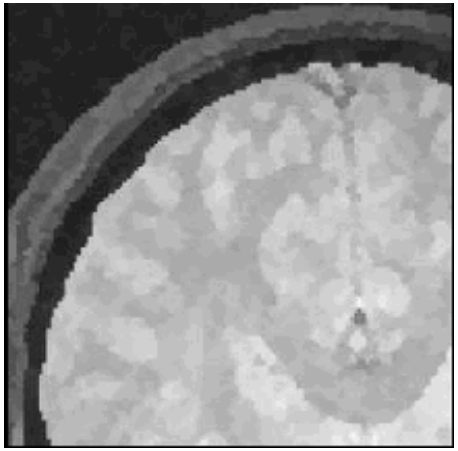
Nagao Filter - Example



Original image



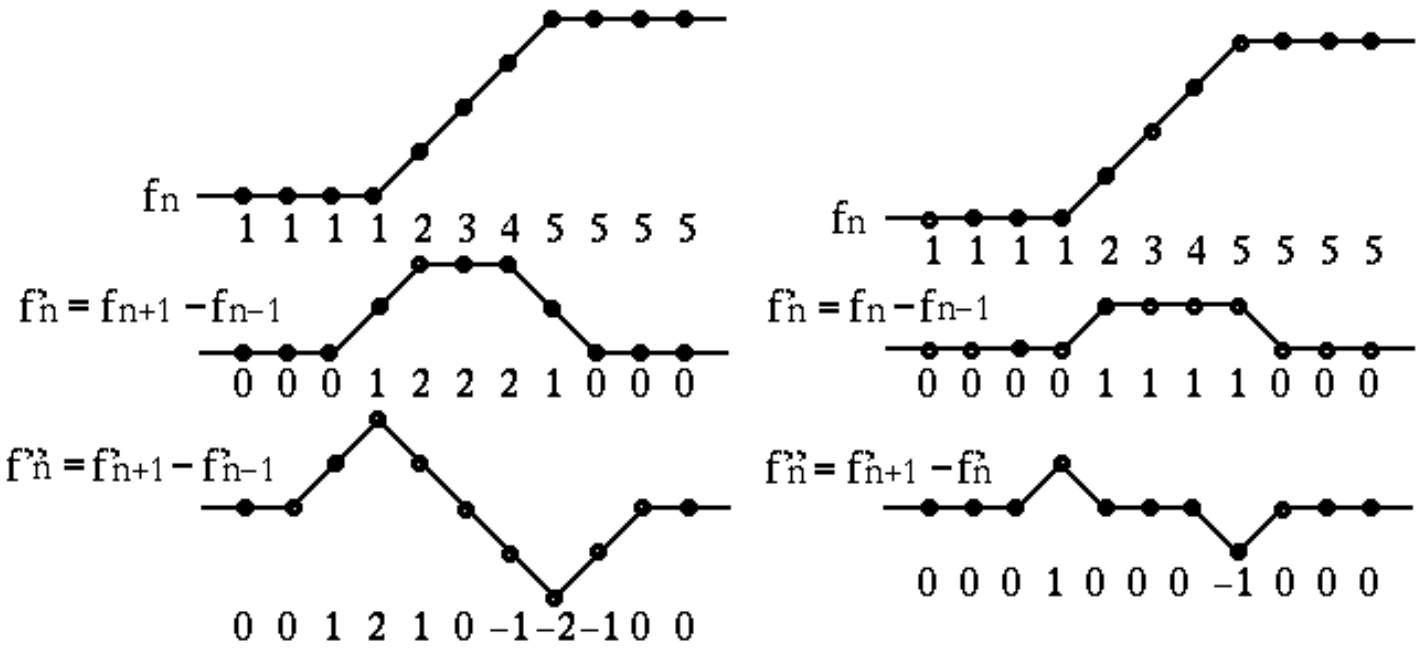
After average filtering



After Nagao filtering

High pass filtering

- **We use high pass filtering for :**
  - Transitions detection
  - Edge detection
  - Contrast increasing
- **Two approaches :**
  - First order $\rightarrow$ first derivative $\rightarrow$ gradients
  - Second order $\rightarrow$ second derivative $\rightarrow$ Laplacian



- **Definition :**
  - Consider $f(x, y)$, then : $\nabla f = \begin{pmatrix} G_x \\ G_y \end{pmatrix} = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix}$
  - Gradient amplitude : $|\nabla f| = \sqrt{G_x^2 + G_y^2}$ or $|\nabla f| = |G_x| + |G_y|$
  - Gradient orientation : $\theta = \arctan\left(\frac{G_y}{G_x}\right)$

High pass filtering

Gradients

- Gradient approximation :
  $\frac{\partial f}{\partial x} = \lim_{\Delta x \to 0} \frac{f(x+\Delta x,y)-f(x,y)}{\Delta x}$ or $\frac{\partial f}{\partial x} = \lim_{\Delta x \to 0} \frac{f(x,y)-f(x-\Delta x,y)}{\Delta x}$
  $\frac{\partial f}{\partial y} = \lim_{\Delta y \to 0} \frac{f(x,y+\Delta y)-f(x,y)}{\Delta y}$ or $\frac{\partial f}{\partial y} = \lim_{\Delta y \to 0} \frac{f(x,y)-f(x,y-\Delta y)}{\Delta y}$
  - For $\Delta x = 1, \Delta y = 1 \Longrightarrow$ this is equivalent to convolve the input image with $\begin{array}{|c|}\hline 1 \\ \hline -1 \\ \hline \end{array}$ and with $\begin{array}{|c|c|}\hline 1 & -1 \\ \hline \end{array}$
- In fact, no reason to consider the right first derivative instead of the left first derivative or to consider that they are equal.
  - We can chose the mean between them :
    $\frac{\partial f}{\partial x} = \frac{1}{2}\left[\frac{f(x+\Delta x,y)-f(x-\Delta x,y)}{\Delta x}\right]$ and $\frac{\partial f}{\partial y} = \frac{1}{2}\left[\frac{f(x,y+\Delta y)-f(x,y-\Delta y)}{\Delta y}\right]$
  - For $\Delta x = 1, \Delta y = 1 \Longrightarrow$ this is equivalent to convolve the input image with $\begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline \end{array}$ et avec $\begin{array}{|c|}\hline 1 \\ \hline 0 \\ \hline -1 \\ \hline \end{array}$.

High pass filtering

Gradients - Examples

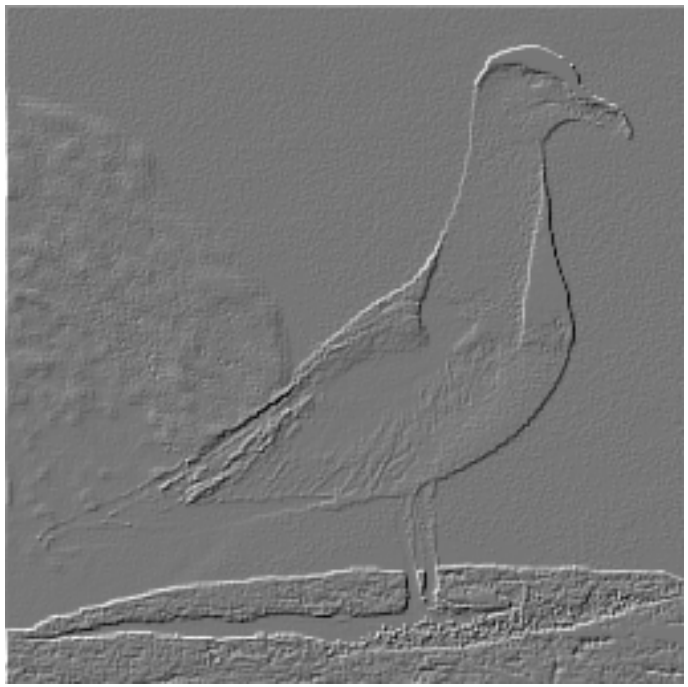
- Example of x and y first derivative :

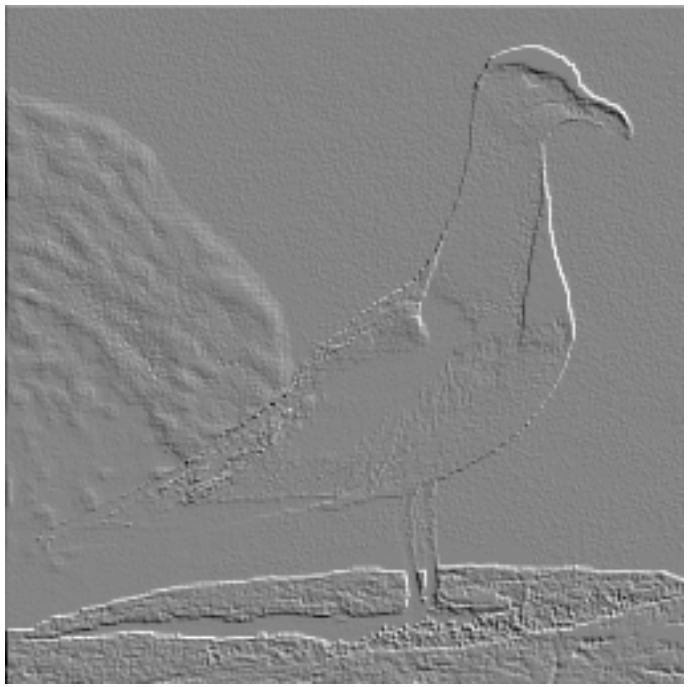
High pass filtering

Gradients - Roberts kernels

- **Roberts kernels :**

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \text{ and } \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$





High pass filtering

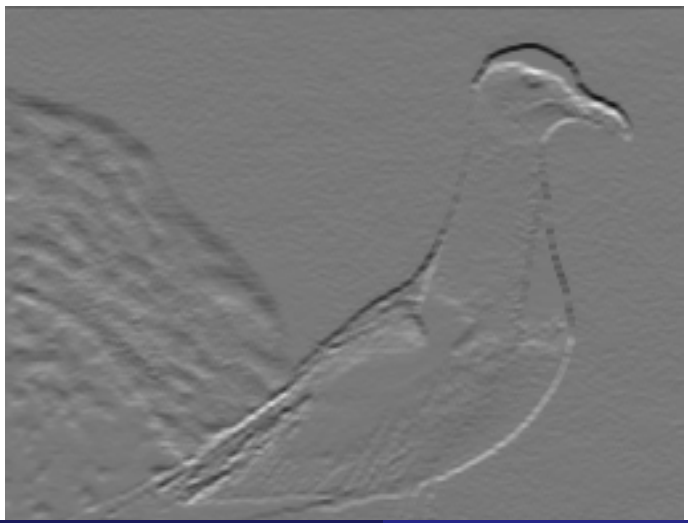
Gradients - Prewitt kernels

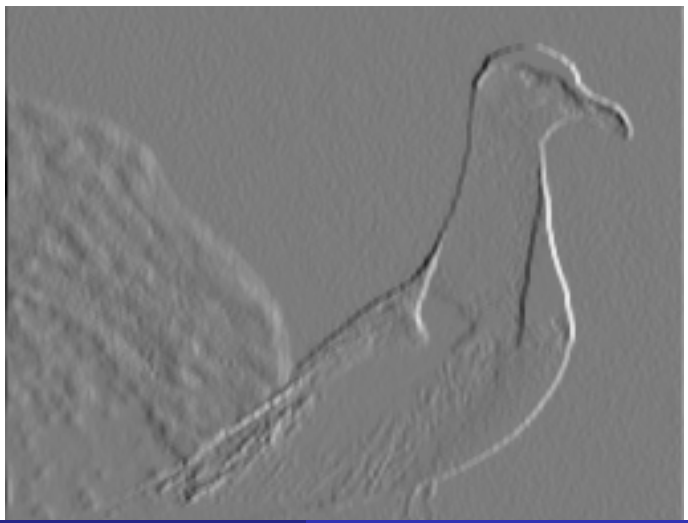
- **Prewitt kernels :**
  - Average filter + first derivative

$$\begin{bmatrix}-1\\0\\1\end{bmatrix} * \begin{bmatrix}1&1&1\end{bmatrix} = \begin{bmatrix}-1&-1&-1\\0&0&0\\1&1&1\end{bmatrix}$$

and

$$\begin{bmatrix}-1&0&1\end{bmatrix} * \begin{bmatrix}1\\1\\1\end{bmatrix} = \begin{bmatrix}-1&0&1\\-1&0&1\\-1&0&1\end{bmatrix}$$





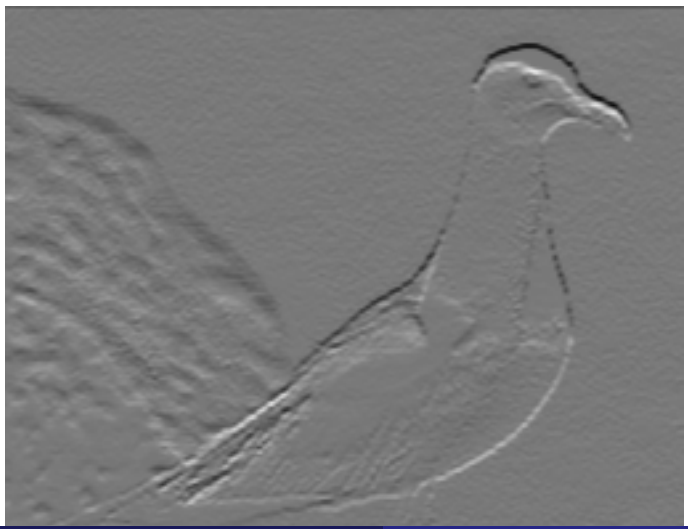
High pass filtering

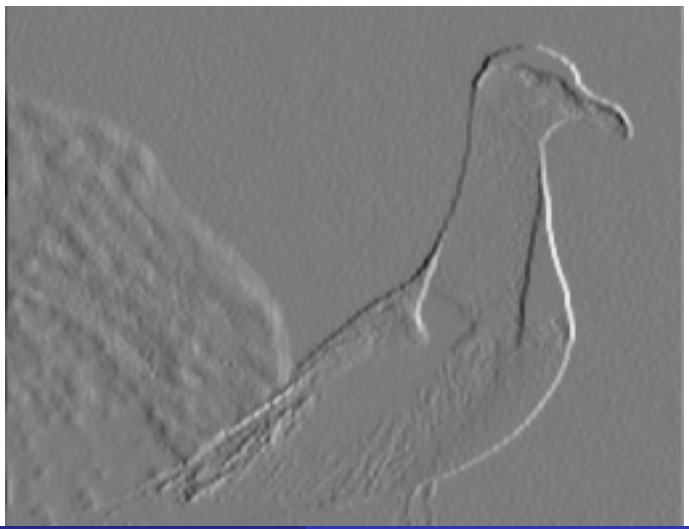
Gradients - Sobel kernels

- **Sobel Kernels :**
  - Gaussian filter + first derivative

$$\begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline \end{array} * \begin{array}{|c|} \hline -1 \\ \hline 0 \\ \hline 1 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$ et

$$\begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline \end{array} * \begin{array}{|c|} \hline 1 \\ \hline 2 \\ \hline 1 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$$





- **Kirsch Kernels**
  - Same principle than Prewitt but in all the possible edge's directions.
  - For a 3×3 neighborhood $\rightarrow$ 4 directions, 8 orientations.

-1	-1	-1
0	0	0
1	1	1

-1	-1	0
-1	0	1
0	1	1

-1	0	1
-1	0	1
-1	0	1

0	1	1
-1	0	1
-1	-1	0

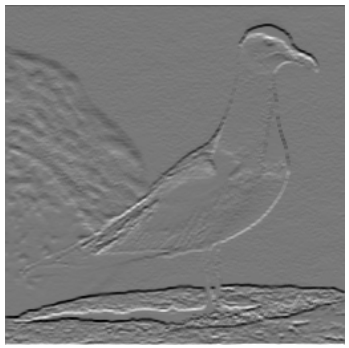
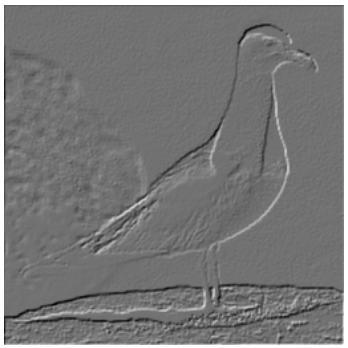
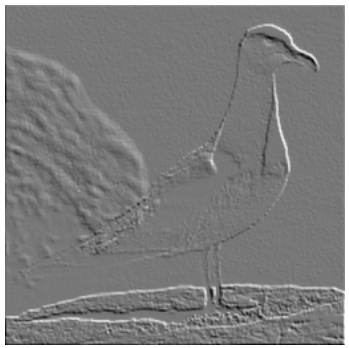
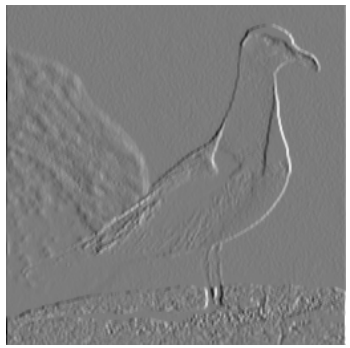
1	1	1
0	0	0
-1	-1	-1

1	1	0
1	0	-1
0	-1	-1

1	0	-1
1	0	-1
1	0	-1

0	-1	-1
1	0	-1
1	1	0

- **Kirsch kernels : example**

- **Kirsch kernels : example**

High pass filtering

Laplacian

- Second derivative :
  $\frac{\partial^2 f}{\partial x^2} = \frac{\partial((f(x+1,y)-f(x,y))}{\partial x} = [f(x+2,y) - f(x+1,y)] - [f(x+1,y) - f(x,y)] = f(x+2,y) - 2f(x+1,y) + f(x,y)$
  - Convolve with the kernel : $\begin{array}{|c|c|c|}\hline 1 & -2 & 1 \\ \hline \end{array}$
  - Remark : $\begin{array}{|c|c|c|}\hline 1 & -2 & 1 \\ \hline \end{array} = \begin{array}{|c|c|}\hline -1 & 1 \\ \hline \end{array} * \begin{array}{|c|c|}\hline -1 & 1 \\ \hline \end{array}$
- Laplacian operator :

$$\nabla^2 f = \left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}\right) \rightarrow \begin{array}{|c|c|c|}\hline 1 & -2 & 1 \\ \hline \end{array} + \begin{array}{|c|}\hline 1 \\ \hline -2 \\ \hline 1 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

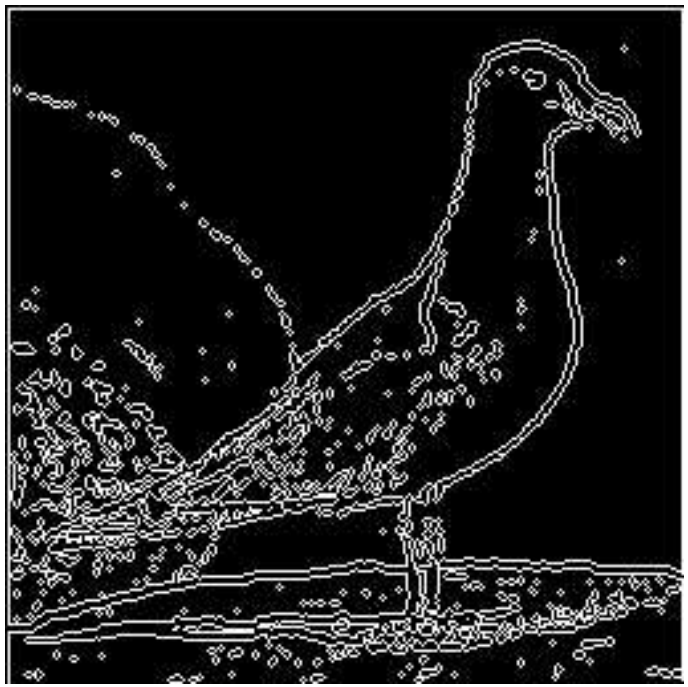
- Other forms : $\begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 4 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array}$ or

$$\begin{array}{|c|c|c|}\hline -1 & -1 & -1 \\ \hline -1 & 8 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline -1 & 0 & -1 \\ \hline 0 & 4 & 0 \\ \hline -1 & 0 & -1 \\ \hline \end{array} + \begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 4 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array}$$

High pass filtering

Laplacian - Example

- **Example :**



- **Advantages :** scalar$\Longrightarrow$ isotropic, just one convolution.

High pass filtering

Laplacian - other forms

- Average filter + second order derivative

$$\frac{1}{3}\begin{bmatrix}1\\1\\1\end{bmatrix} * \begin{bmatrix}1 & -2 & 1\end{bmatrix} = \frac{1}{3}\begin{bmatrix}1 & -2 & 1\\1 & -2 & 1\\1 & -2 & 1\end{bmatrix}$$

- Gaussian filter + second order derivative

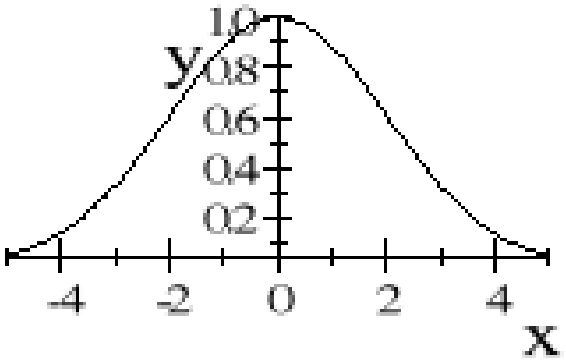
$$\frac{1}{4}\begin{bmatrix}1\\2\\1\end{bmatrix} * \begin{bmatrix}1 & -2 & 1\end{bmatrix} = \frac{1}{4}\begin{bmatrix}1 & -2 & 1\\2 & -4 & 2\\1 & -2 & 1\end{bmatrix}$$

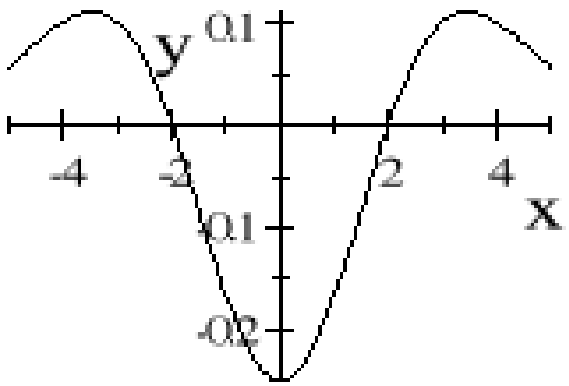
- Smoothing + Laplacian :
  $f(x,y) * G_\sigma * L = f * LoG$ (Laplacian of Gaussian)

- In 1D : $G_\sigma(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right) \rightarrow \frac{\partial^2 G}{\partial x^2} = \frac{-1}{\sigma^2}\left(1 - \frac{x^2}{\sigma^2}\right)\exp\left(-\frac{x^2}{2\sigma^2}\right)$

$$\left[G_\sigma(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right)\right]_{\sigma=2} = \qquad \left[\frac{\partial^2 G}{\partial x^2} = \frac{-1}{\sigma^2}\left(1 - \frac{x^2}{\sigma^2}\right)\exp\left(-\frac{x^2}{2\sigma^2}\right)\right]_{\sigma=}$$

$$xG_\sigma = e^{-\frac{1}{8}x^2} \qquad 0 = \frac{1}{4}e^{-\frac{1}{8}x^2}\left(\frac{1}{4}x^2 - 1\right)$$



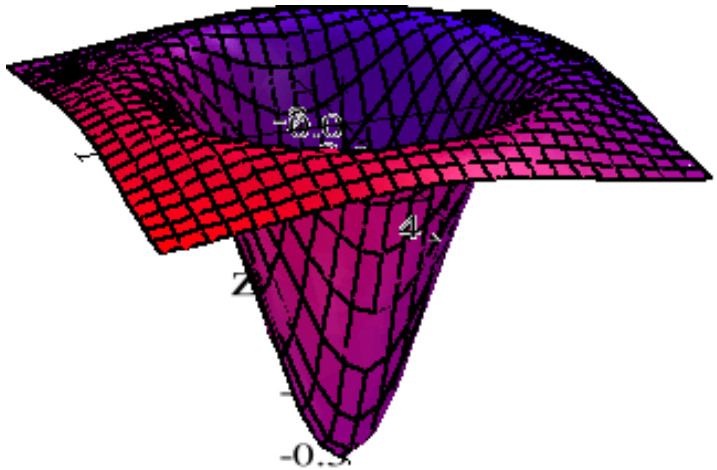


High pass filtering

Laplacian - Marr-Hildreth Filter(LOG)

- In 2D : $G_\sigma(x,y) = \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \rightarrow LoG(x,y) = \frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2} = \left(\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right) \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$

$$\left[\left(\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right) \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)\right]_{\sigma=2} = \frac{1}{16} e^{-\frac{1}{8}x^2-\frac{1}{8}y^2} \left(x^2+y^2-8\right)$$



- **Example of LOG filter**

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

pour $\sigma^2 = 4$

- **Difference of Gaussians (DoG) :**
  - The difference between two Gaussian filters is equivalent under some conditions to the LOG filter [Marr-Hildreth 80] :

$$DoG(x,y) = \frac{1}{2\pi\sigma_1^2}\exp\left(-\frac{x^2+y^2}{2\sigma_1^2}\right) - \frac{1}{2\pi\sigma_2^2}\exp\left(-\frac{x^2+y^2}{2\sigma_2^2}\right) \approx LoG(x,y) \text{ with } \sigma_1 > \sigma_2$$

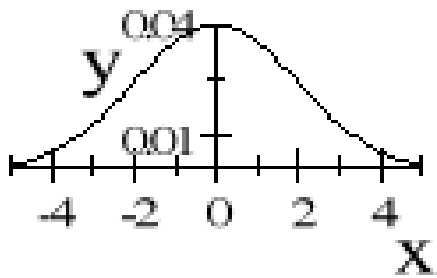
The two functions have the same zero-crossings if :

$$\sigma^2 = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 - \sigma_2^2}\log\left(\frac{\sigma_1^2}{\sigma_2^2}\right)$$

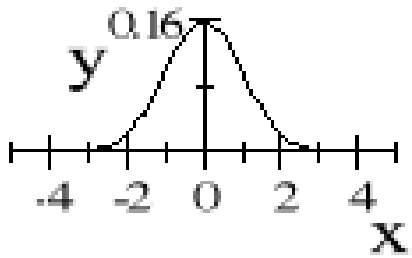
High pass filtering

Laplacian - Difference of Gaussians

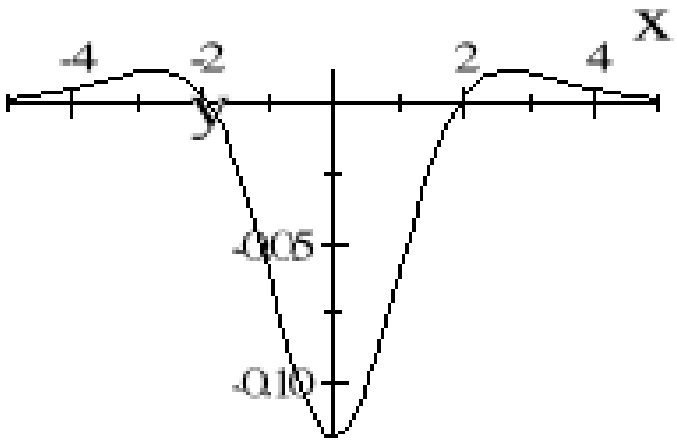
$$\left[G_\sigma(x) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2}{2\sigma^2}\right)\right]_{\sigma=2}$$



$$\left[G_\sigma(x) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2}{2\sigma^2}\right)\right]_{\sigma=1}$$



$$DoG(x) = \frac{1}{8\pi}e^{-\frac{1}{8}x^2} - \frac{1}{2\pi}e^{-\frac{1}{2}x^2}$$



Non-linear filtering

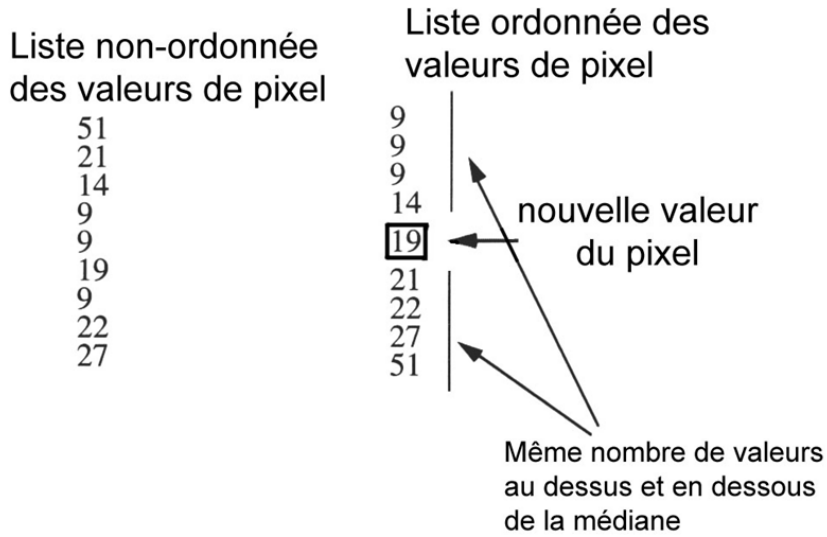
- **Non-linear filters :**
  - Non-linear function of the neighbors
  - Require defining the non-linear operator + neighborhood size.
- **Example of non-linear filters :**
  - Min $\rightarrow$ Each pixel value is replaced by the minimum value in the neighborhood.
  - Max $\rightarrow$ Each pixel value is replaced by the maximum value in the neighborhood.
  - Majority $\rightarrow$ Each pixel value is replaced by the most frequent value in the neighborhood.
  - Extreme $\rightarrow$ Each pixel value is replaced by the nearest between min and max value in the neighborhood.
  - etc.

    * Particular case : binary image $\rightarrow$ Min=AND Max=OR operators

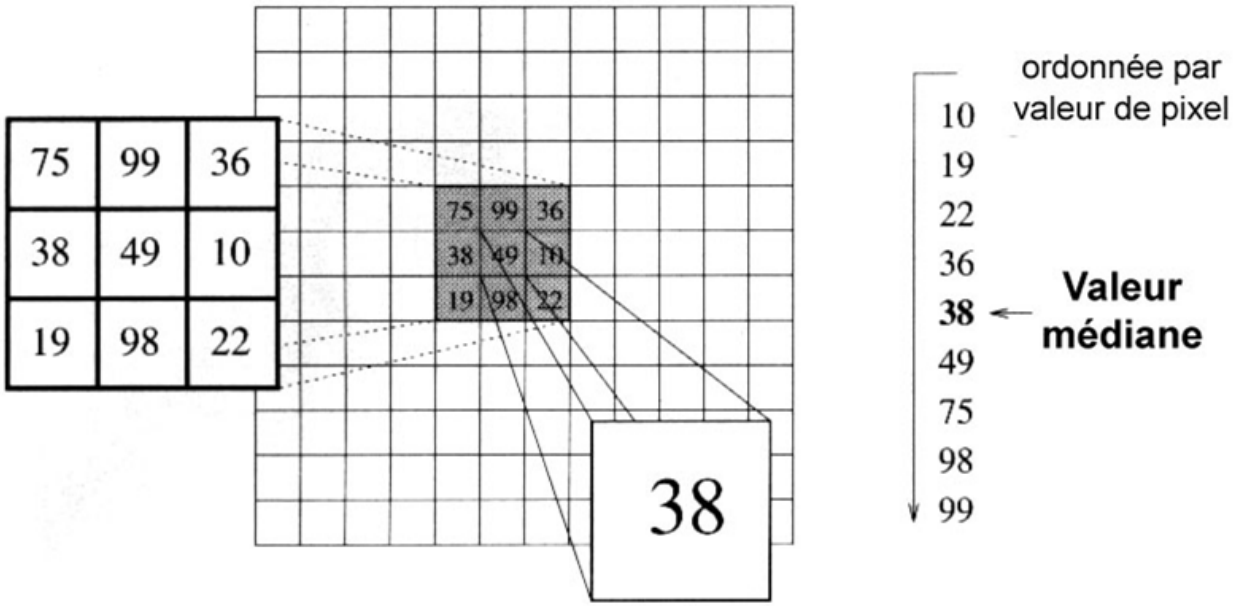
Median Filter

- Median filter : each pixel value is replaced by the median value in the neighborhood $\rightarrow$ remove impulse noise

10	12	40	16	19	10
14	22	52	10	55	41
10	14	51	21	14	10
32	22	9	9	19	14
41	18	9	22	27	11
10	7	8	8	4	5



Median Filter



Median Filter

- Many possibilities concerning the selection of pixels in the neighborhood :

1	1	1
1	1	1
1	1	1

0	1	0
1	1	1
0	1	0

0	0	1	0	0
0	0	1	0	0
1	1	1	1	1
0	0	1	0	0
0	0	1	0	0

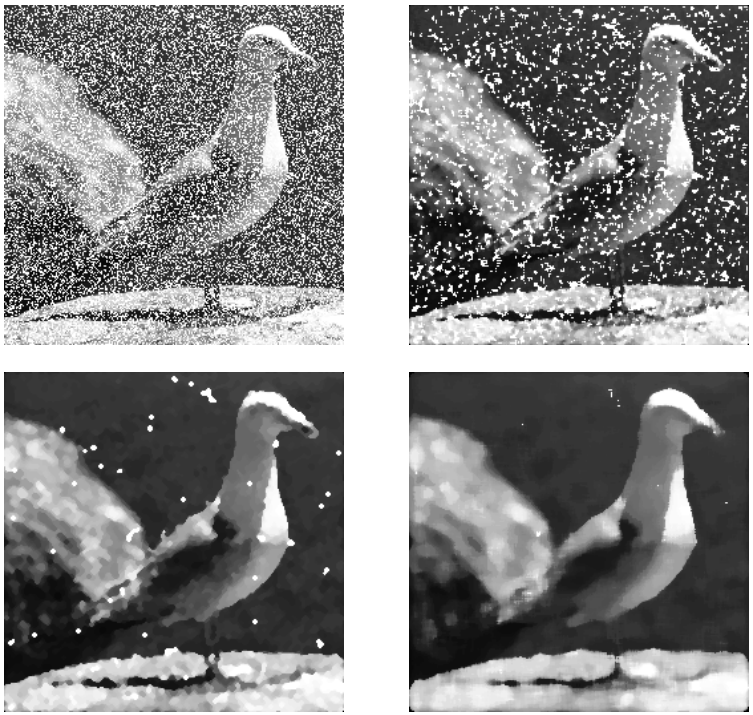
- Efficient for suppressing extreme values.
- If no optimization, the filter is time consuming.
- Details are suppressed .
- No new pixel values.

Median filter - Example 1

- Salt and Pepper noise of density 10%
- Noise could be suppressed if its density is less than half the filter size.

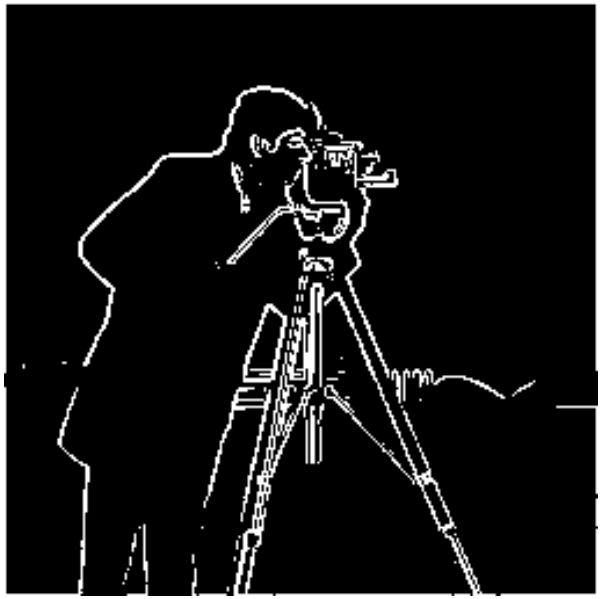
Median filter - Example 2

- Salt and Pepper noise of density 40%, 3×3 median filter, 3×3 median filter repeated 10 times and 9×9 median filter.



From contrast points to edges

Threshold =10



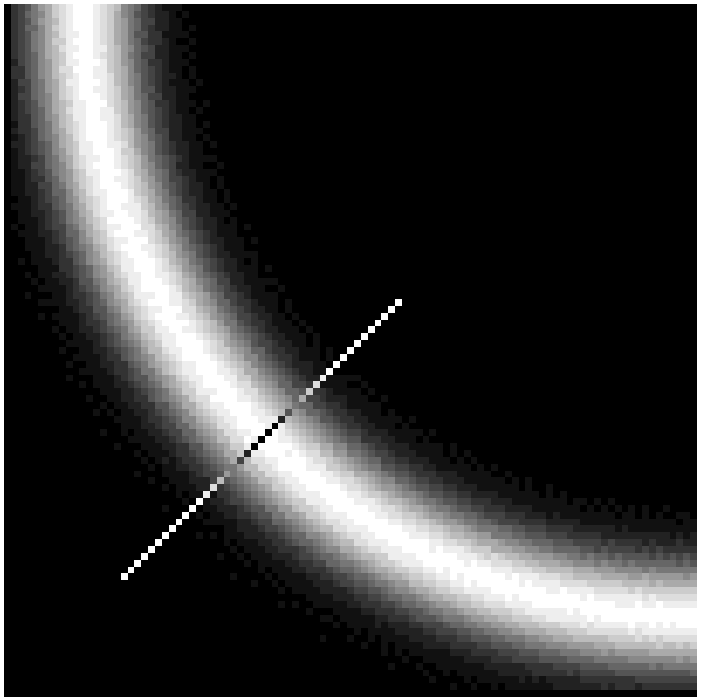
Threshold =100

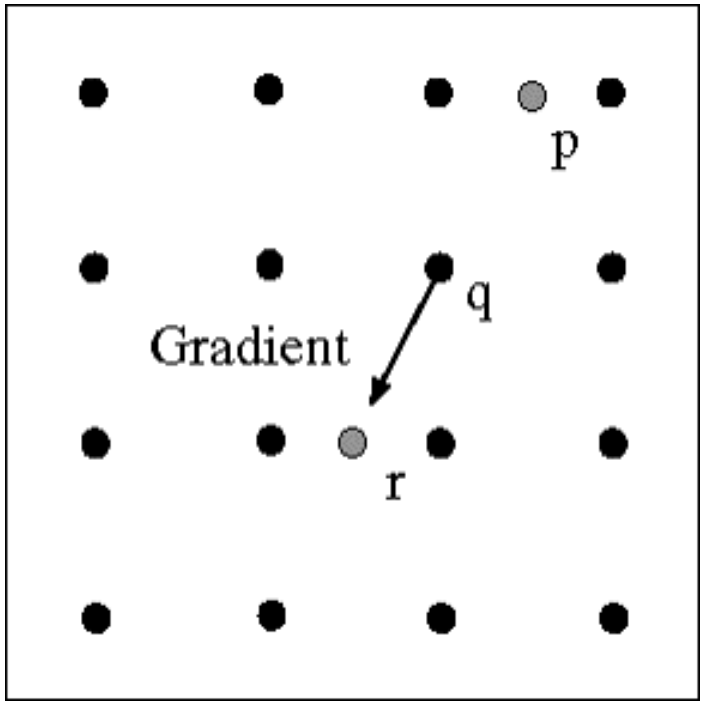
Threshold =200

The choice of a suitable threshold is delicate. Threshold too low : thick contours + noise. Threshold too high : discontinuities.

The Canny Edge detector

Non-maximum suppression



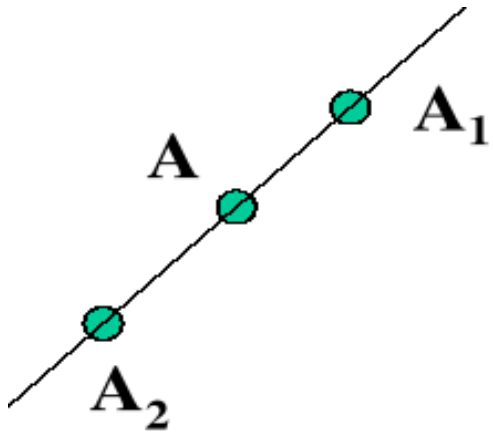


Check if pixel is local maximum along gradient direction (perpendicular to contour), select single max across width of the edge
$\rightarrow$ requires checking interpolated pixels p and r

The Canny Edge detector

Eliminate points that are not local *extrema* in gradient direction

- Starting from each edge point A, consider its direct neighbors A_1 and A_2 in gradient direction.
- A is a local maximum if : $G(A) > G(A_1)$ and $G(A) > G(A_2)$



Hysteresis thresholding

Use a high threshold to start edge curves, and a low threshold to continue them.

- High threshold $\rightarrow$ most reliable edges. Let f1 be the binary resulting image.
- Low threshold $\rightarrow$ Edges + noise. Let f2 be the binary resulting image.
- Keep only points in f2 that are connected to a non null point in f1.

The Canny Edge detector

Hysteresis thresholding

Original image

Low threshold (weak edges)

High threshold (strong edges)

Hysteresis threshold

The Canny Edge detector

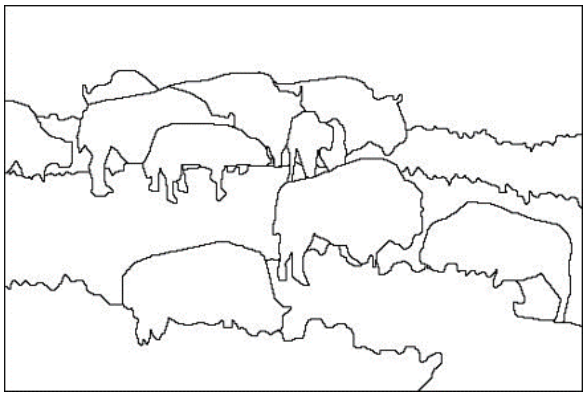
- Filter image with derivative of Gaussian
- Find magnitude and orientation of gradient
- Non-maximum suppression :
  - Thin wide "ridges" down to single pixel width
- Linking and thresholding (hysteresis) :
  - Define two thresholds : low and high
  - Use the high threshold to start edge curves and the low threshold to continue them

J. Canny, A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8 :679-714, 1986.

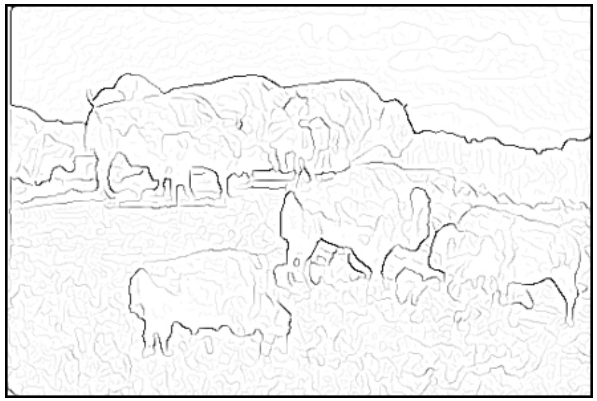
The Canny Edge detector

Edge detection is just the beginning...

Image



Human segmentation



Gradient magnitude