IMN259 - Analyse d'images

Chapitre 4

Filtrage

Adapté des notes de cours de
Marie-Flavie Auclair-Fortier et Pierre-Marc Jodoin

PLAN

1. Rehaussement du contraste
2. Réduction du bruit
   - 2.1. Bruit
   - 2.2. Filtre moyenneur
     - a) Non pondéré
     - b) Pondéré
   - 2.3. Filtre médian
   - 2.4. Filtre bilatéral
3. Amélioration de la netteté
4. Corrélation

Filtrer une image - rappel IMN359

Le filtrage d'une image est une opération ayant pour objet de réduire ou d'éliminer ou de rehausser certains éléments présents dans une image. De nombreux filtres peuvent s'opérer tant dans le domaine **spatial** que le domaine **spectral**. C'est le cas des filtres **linéaires**. Ces derniers sont directement liés à la théorie de la convolution.

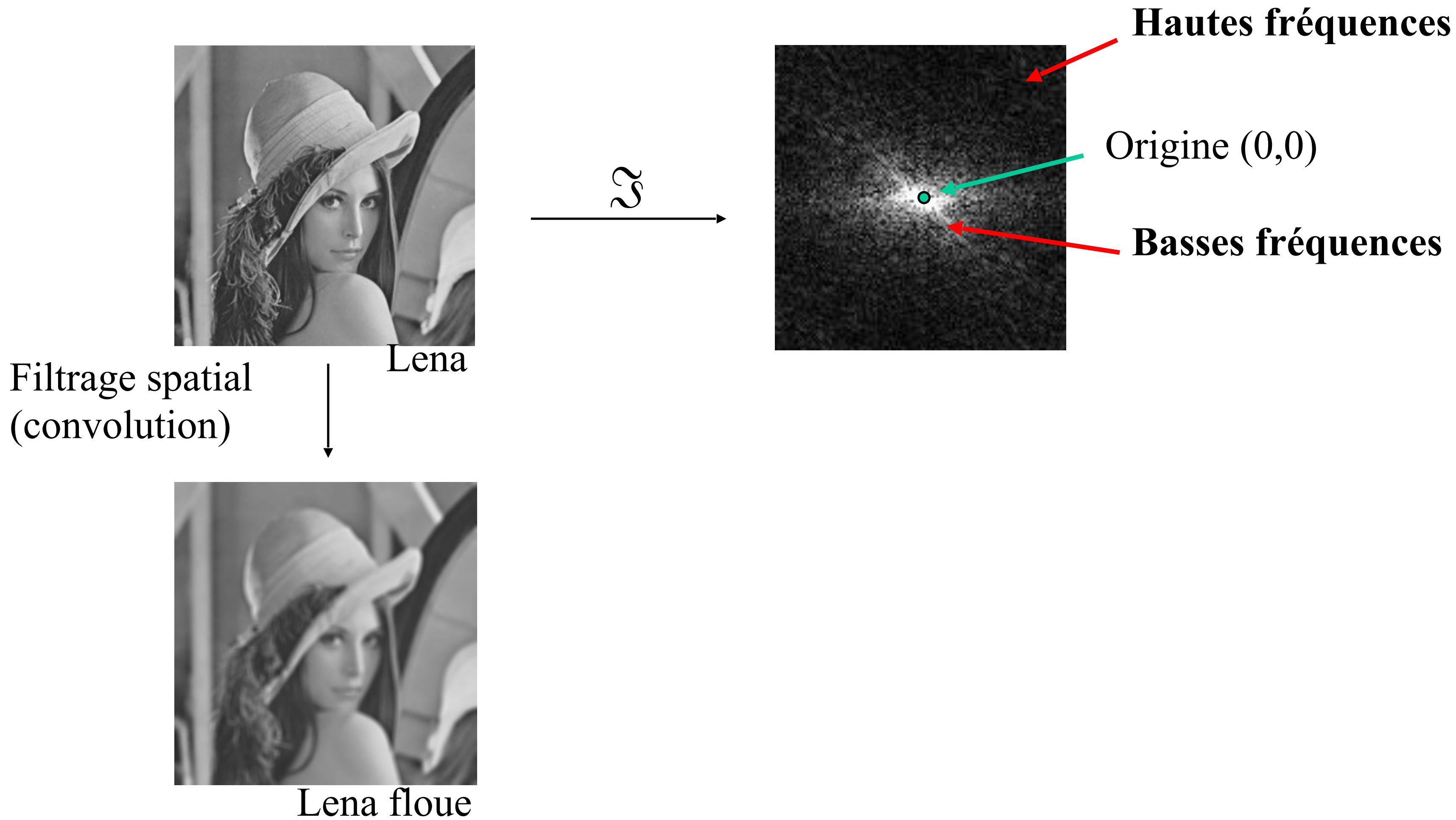
Lena

Filtrage spatial
(convolution)

Lena floue

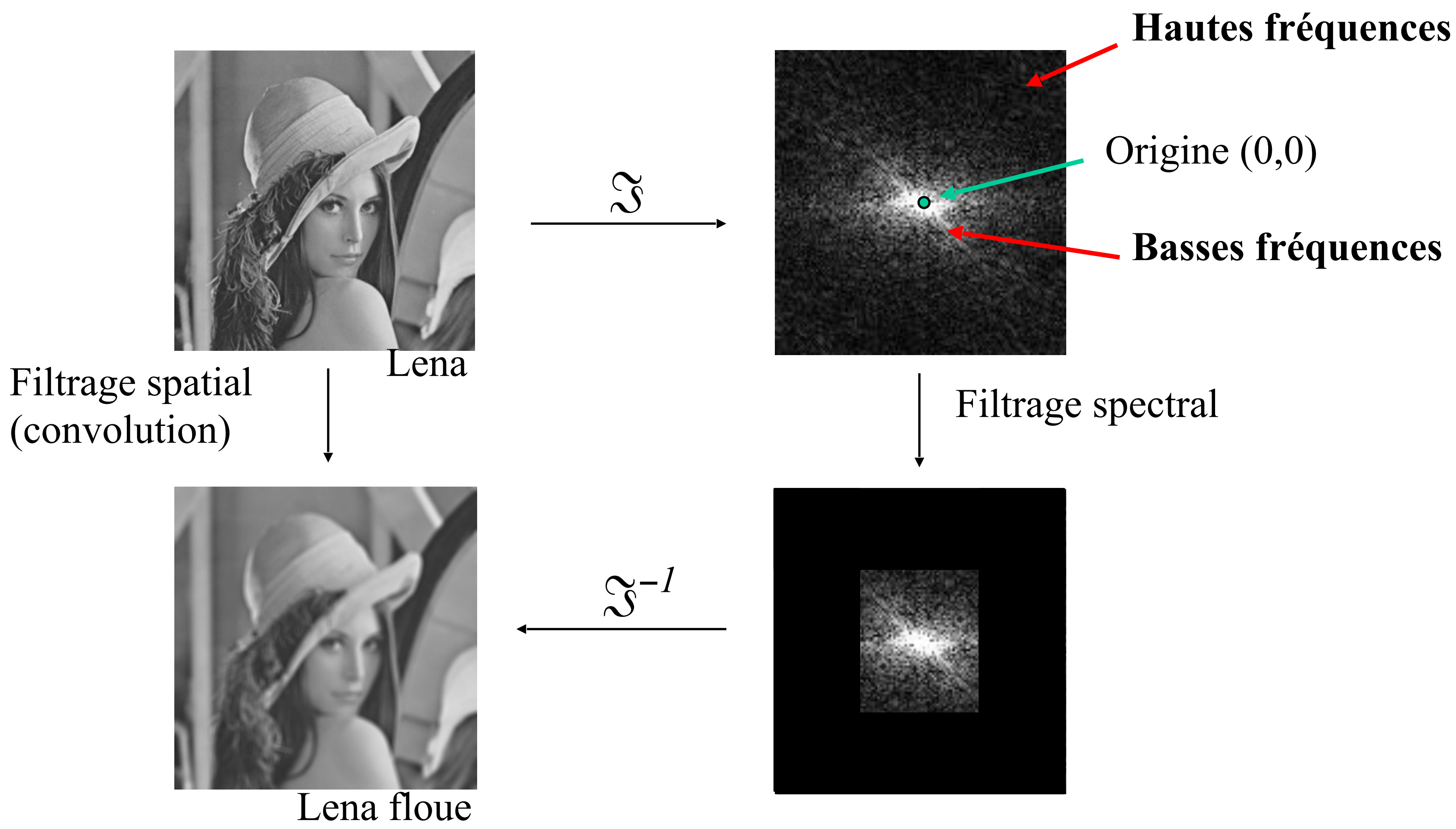
Filtrer une image - rappel IMN359

Le filtrage d’une image est une opération ayant pour objet de réduire ou d’éliminer ou de rehausser certains éléments présents dans une image. De nombreux filtres peuvent s’opérer tant dans le domaine **spatial** que le domaine **spectral**. C’est le cas des filtres **linéaires**. Ces derniers sont directement liés à la théorie de la convolution.



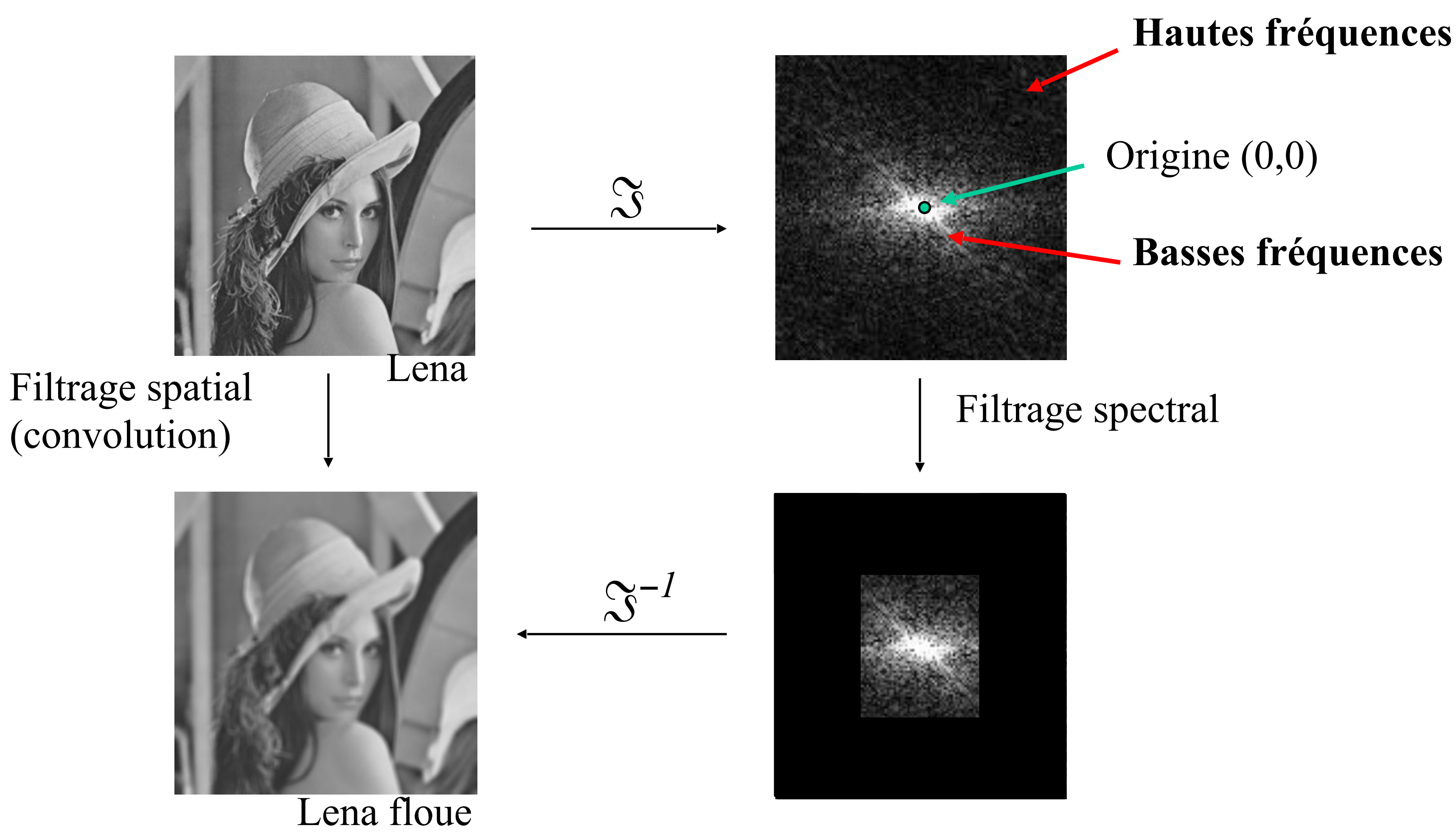
Filtrer une image - rappel IMN359

Le filtrage d'une image est une opération ayant pour objet de réduire ou d'éliminer ou de rehausser certains éléments présents dans une image. De nombreux filtres peuvent s'opérer tant dans le domaine **spatial** que le domaine **spectral**. C'est le cas des filtres **linéaires**. Ces derniers sont directement liés à la théorie de la convolution.



Filtrer une image - rappel IMN359

Le filtrage d'une image est une opération ayant pour objet de réduire ou d'éliminer ou de rehausser certains éléments présents dans une image. De nombreux filtres peuvent s'opérer tant dans le domaine **spatial** que le domaine **spectral**. C'est le cas des filtres **linéaires**. Ces derniers sont directement liés à la théorie de la convolution.



Rappel convolution

$$* \underset{\mathfrak{F}^{-1}}{\overset{\mathfrak{F}}{\rightleftarrows}} \times \qquad\qquad \times \underset{\mathfrak{F}^{-1}}{\overset{\mathfrak{F}}{\rightleftarrows}} *$$

Filtres passe-bas

Filtres utilisés lors de filtrages fréquentiels

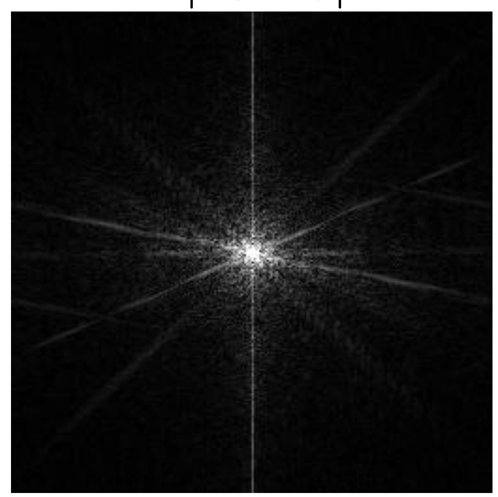
Filtre passe-bas rectangulaire: élimine toutes les fréquences situées au-delà d'un certain seuil. Le filtre passe-bas rectangulaire est la fonction « **porte** ».

Exemple :

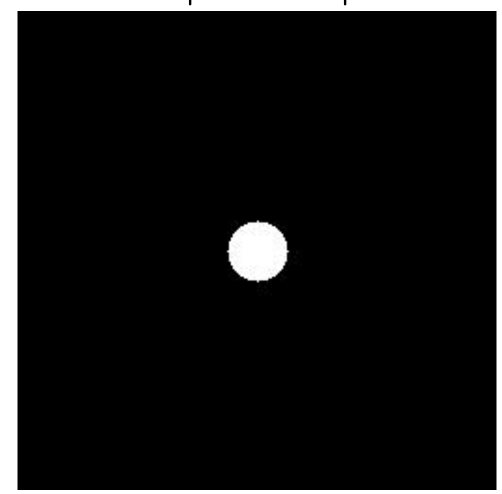
$f(x,y)$

« cameraman »

$|F(u,v)|$

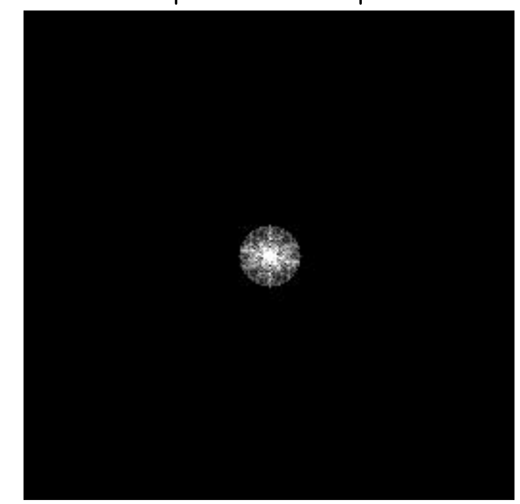


$|H(u,v)|$



rayon = *16* (/256)

$|FH(u,v)|$



Filtres passe-bas

Filtres utilisés lors de filtrages fréquentiels

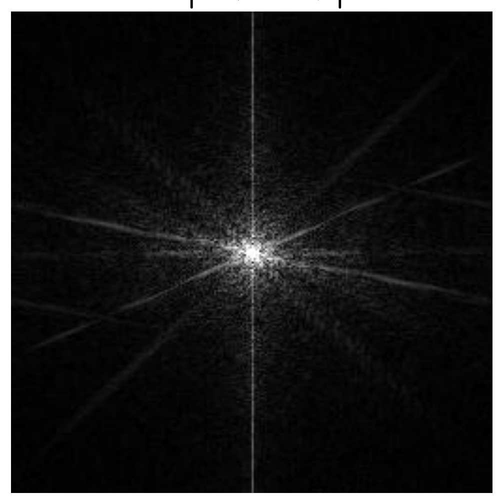
Filtre passe-bas rectangulaire: élimine toutes les fréquences situées au-delà d’un certain seuil. Le filtre passe-bas rectangulaire est la fonction « **porte** ».

Exemple :

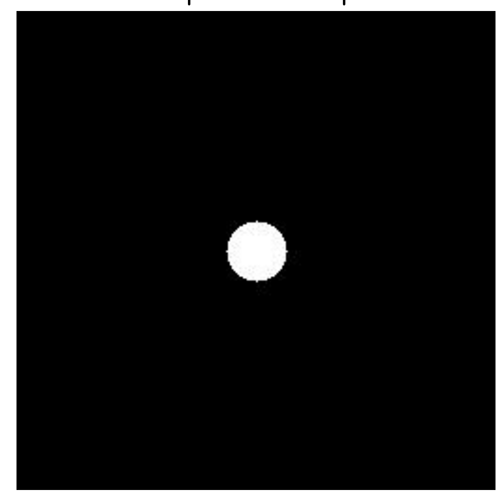
$f(x,y)$

« cameraman »

$|F(u,v)|$

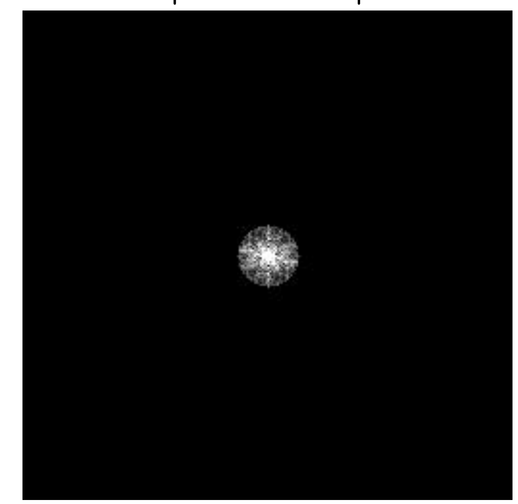


$|H(u,v)|$



rayon = *16* (/256)

$|FH(u,v)|$



rayon = *64* (/256)

rayon = *32* (/256)

rayon = *16* (/256)

rayon = *8* (/256)

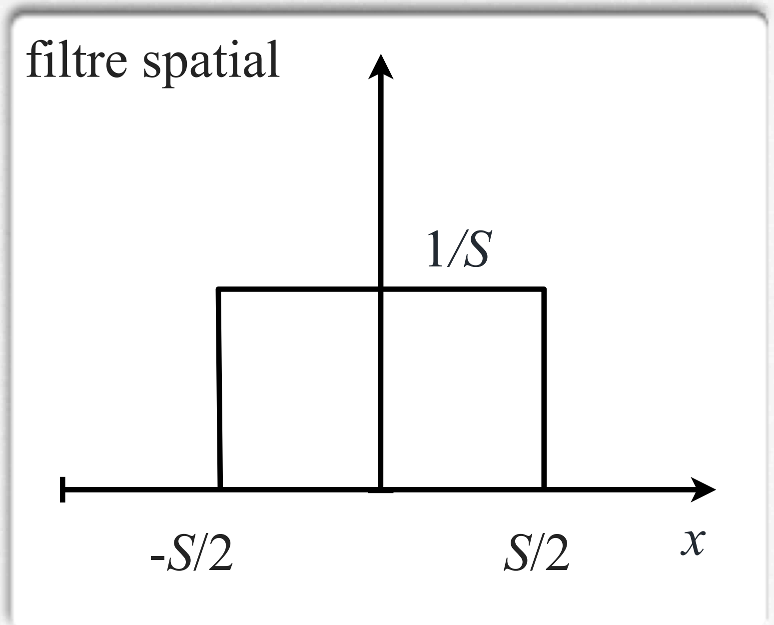
2. Réduction du bruit

2. Filtre moyenneur

a) Non pondéré

- 1D - rappel IMN359 (https://scil.usherbrooke.ca/courses/imn359/)

$$h_b(x) = \frac{\Pi\left(\frac{x}{S}\right)}{S}$$



2. RÉDUCTION DU BRUIT

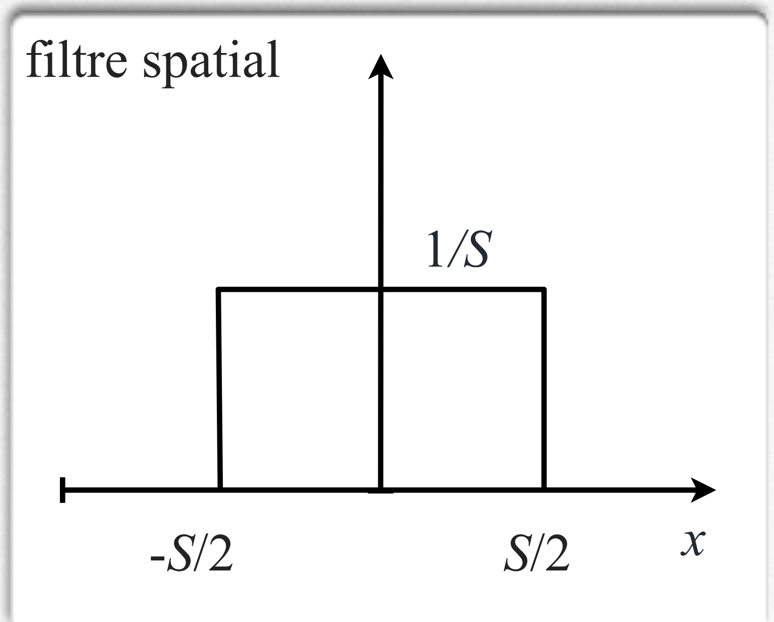
2. Filtre moyenneur

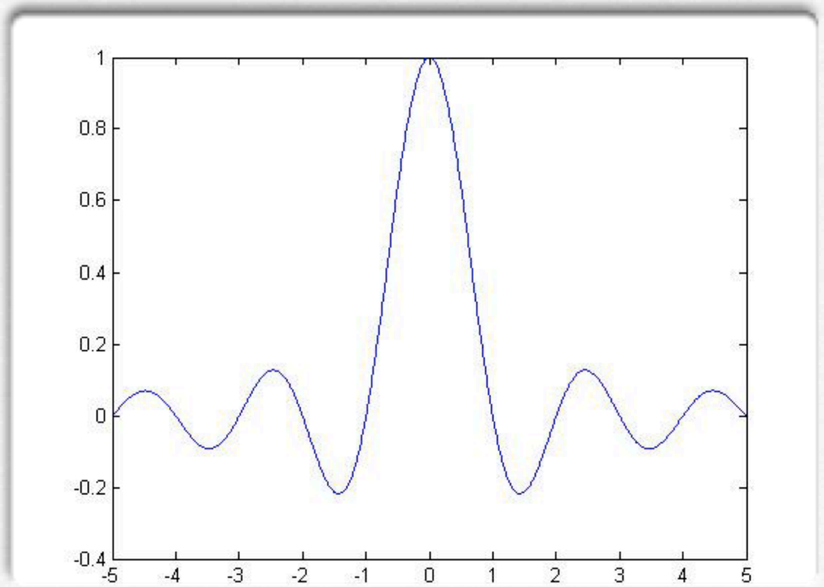
a) Non pondéré

- 1D - rappel IMN359 (https://scil.usherbrooke.ca/courses/imn359/)

$$h_b(x) = \frac{\Pi\left(x/S\right)}{S}$$

filtre spectral $H_b(u) = \mathrm{sinc}\left(Su\right)$



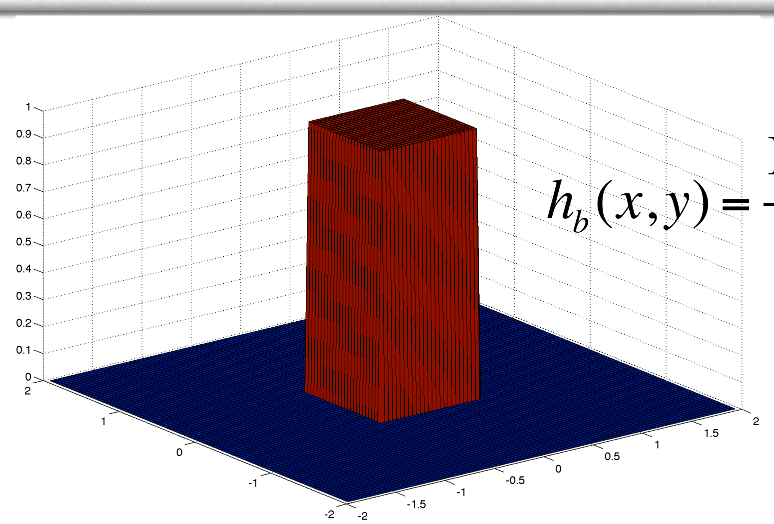


2. Réduction du bruit

2. Filtre moyenneur

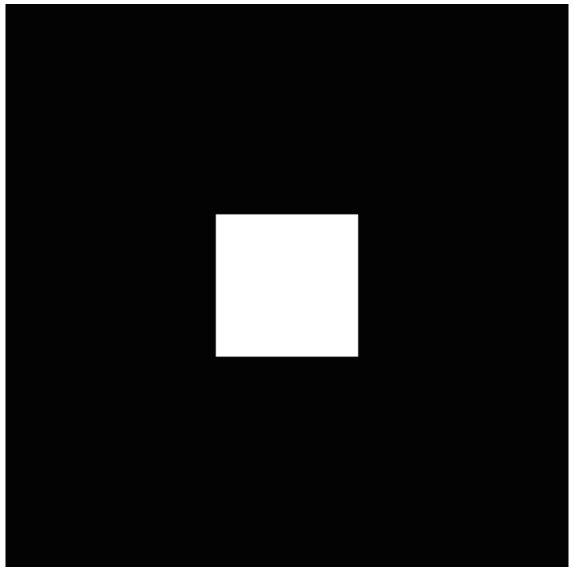
a) Non pondéré

- 2D - rappel IMN359 https://scil.usherbrooke.ca/courses/imn359/



$$h_b(x,y) = \frac{\Pi\left(x/S_x, y/S_y\right)}{S_x S_y}$$

filtre spatial



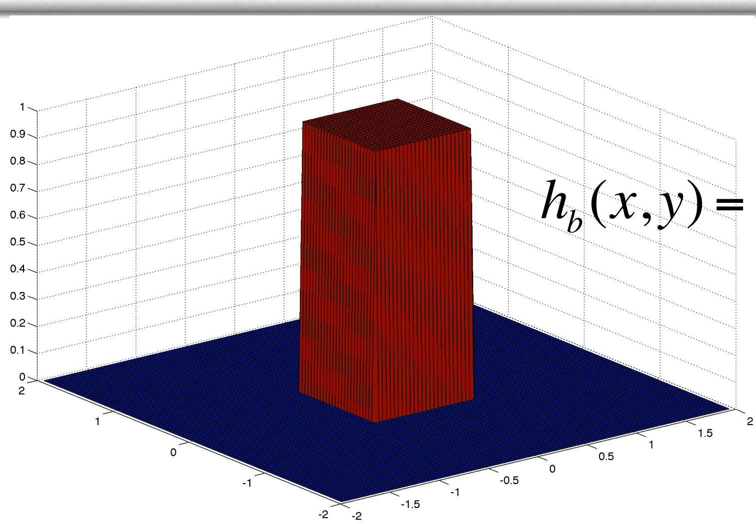
Source : PM Jodoin

2. RÉDUCTION DU BRUIT

2. Filtre moyenneur

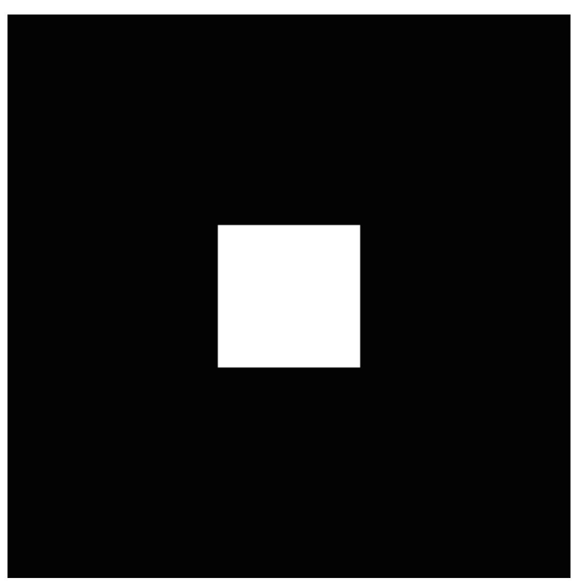
a) Non pondéré

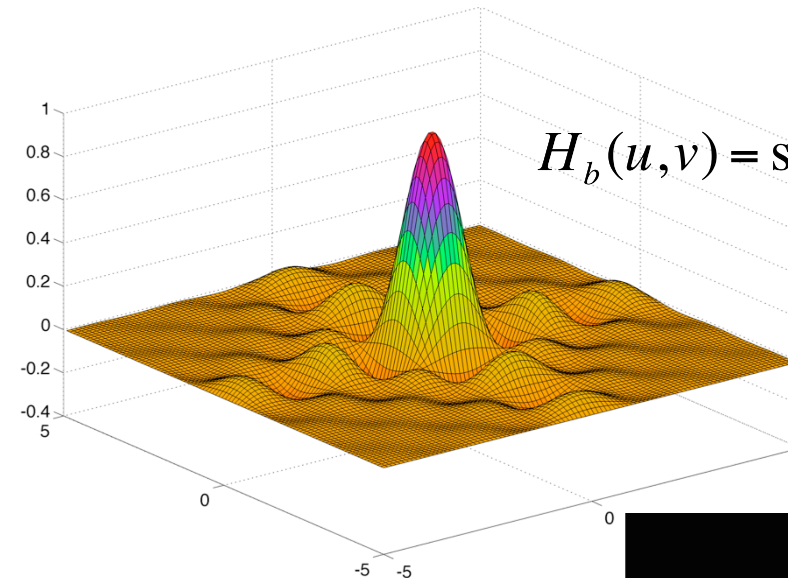
- 2D - rappel IMN359 https://scil.usherbrooke.ca/courses/imn359/



$$h_b(x,y) = \frac{\Pi\left(x/S_x, y/S_y\right)}{S_x S_y}$$

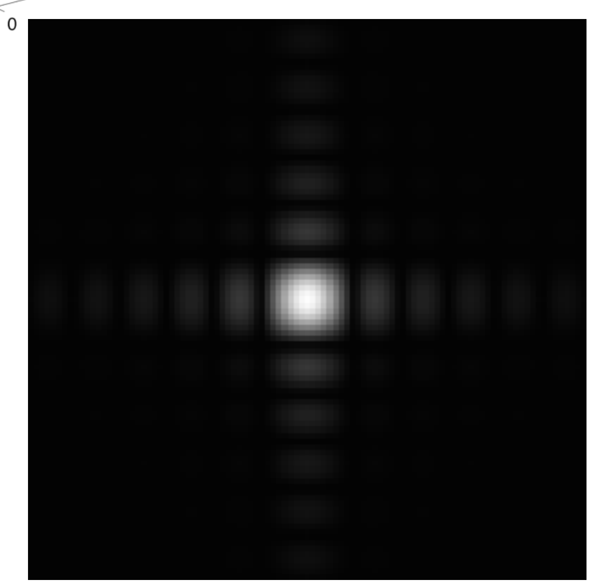
filtre spatial





$$H_b(u,v) = \mathrm{sinc}\left(S_x u\right)\mathrm{sinc}\left(S_y v\right)$$

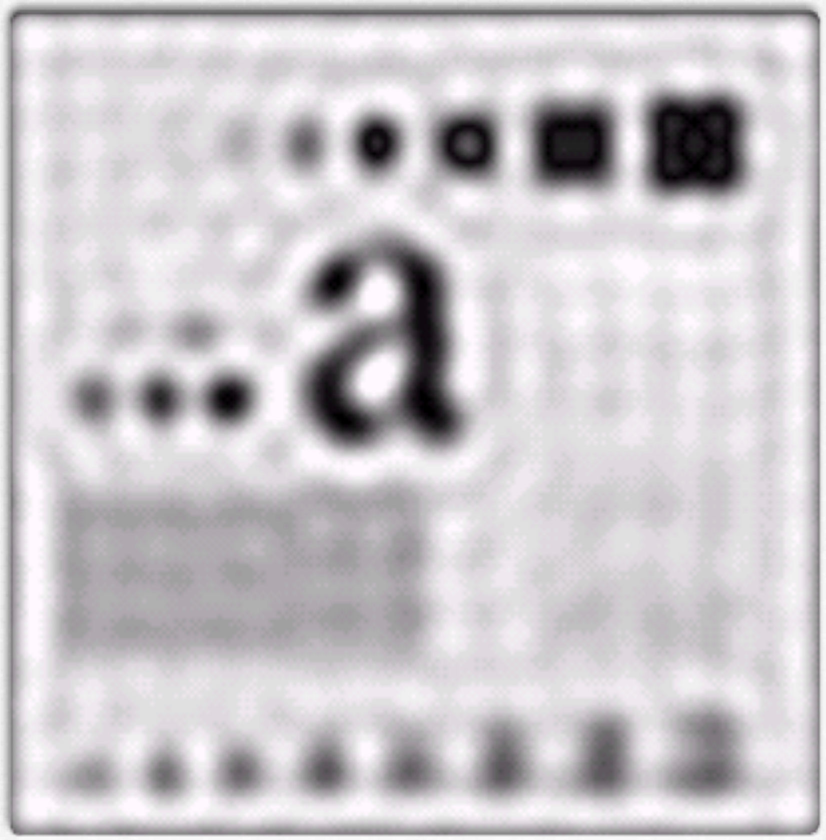
filtre spectral



Source : PM Jodoin

2. RÉDUCTION DU BRUIT

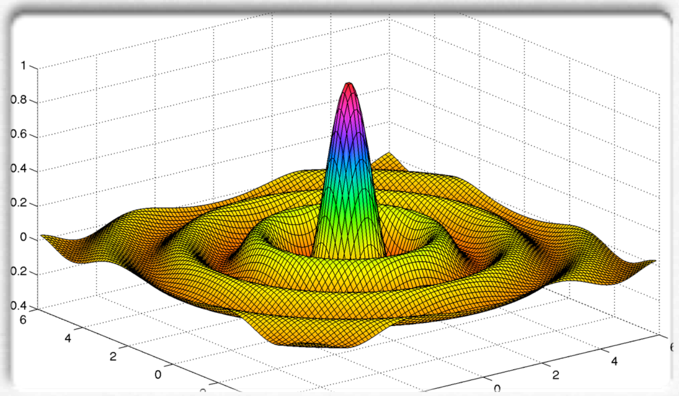
- Problème du filtre idéal passe-bas (rappel)

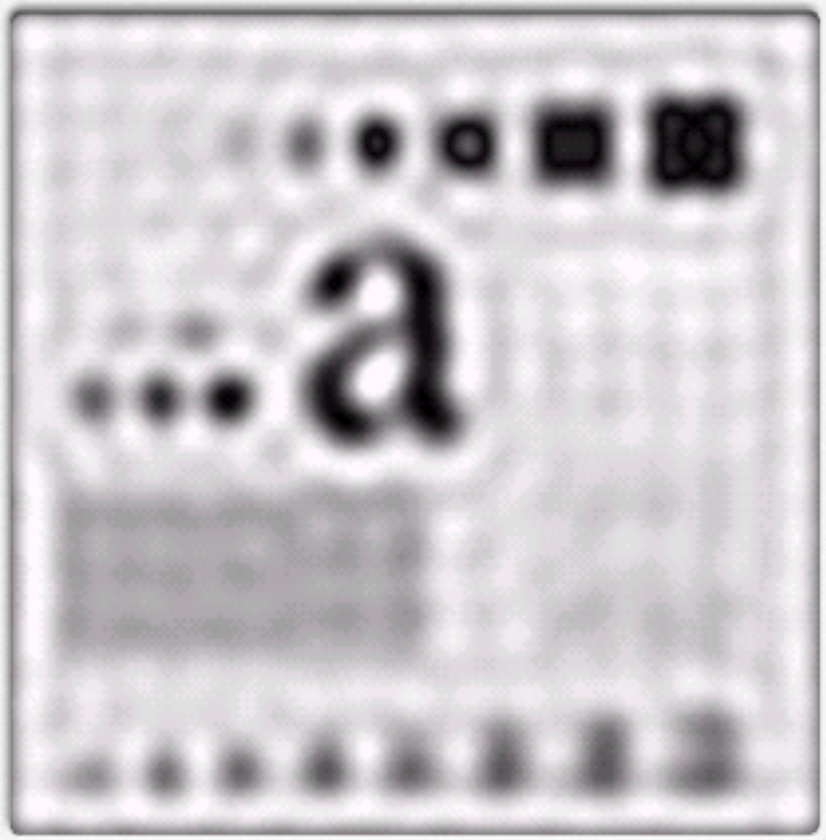


Source : DIP2

2. RÉDUCTION DU BRUIT

- Problème du filtre idéal passe-bas (rappel)
  - ✓ Présence d'ondulations autour des contours nets
  - ✓ La forme spatiale nous donne la puce à l'oeil

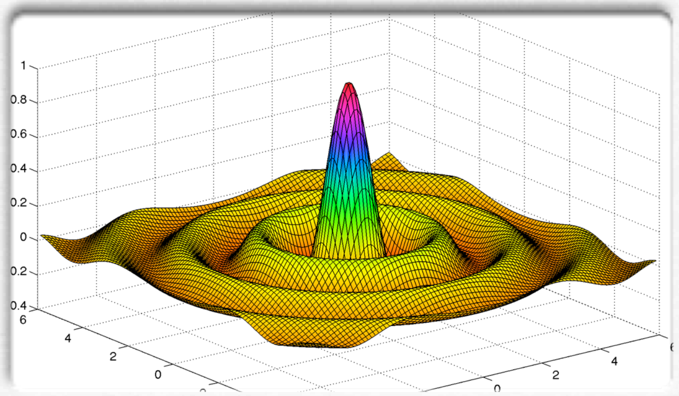




Source : DIP2

2. Réduction du bruit

- Problème du filtre idéal passe-bas (rappel)
  - ✓ Présence d'ondulations autour des contours nets
  - ✓ La forme spatiale nous donne la puce à l'oeil



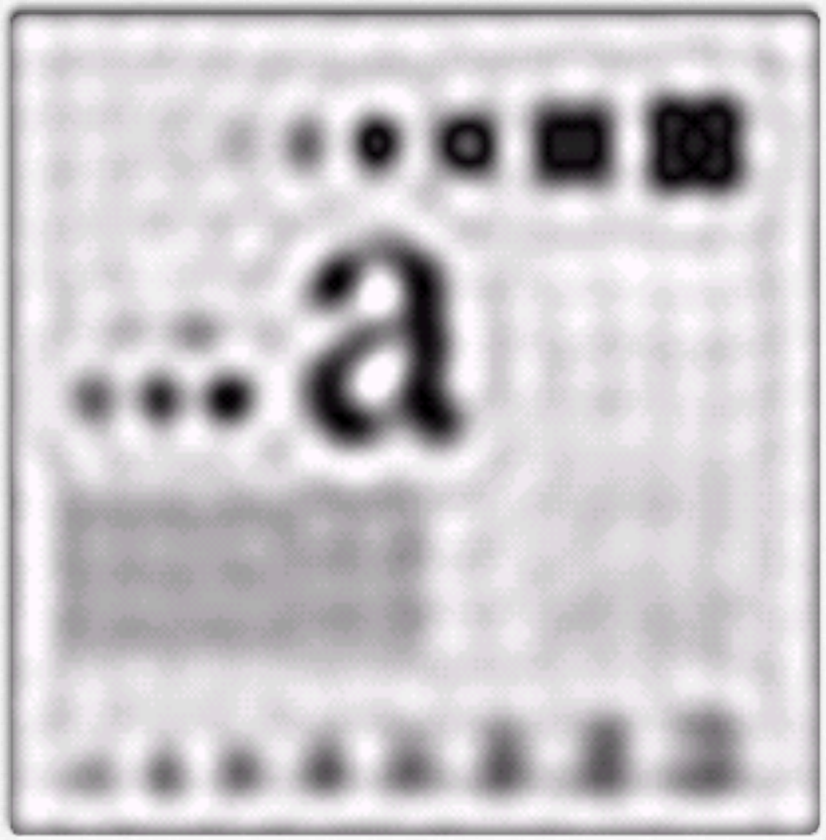
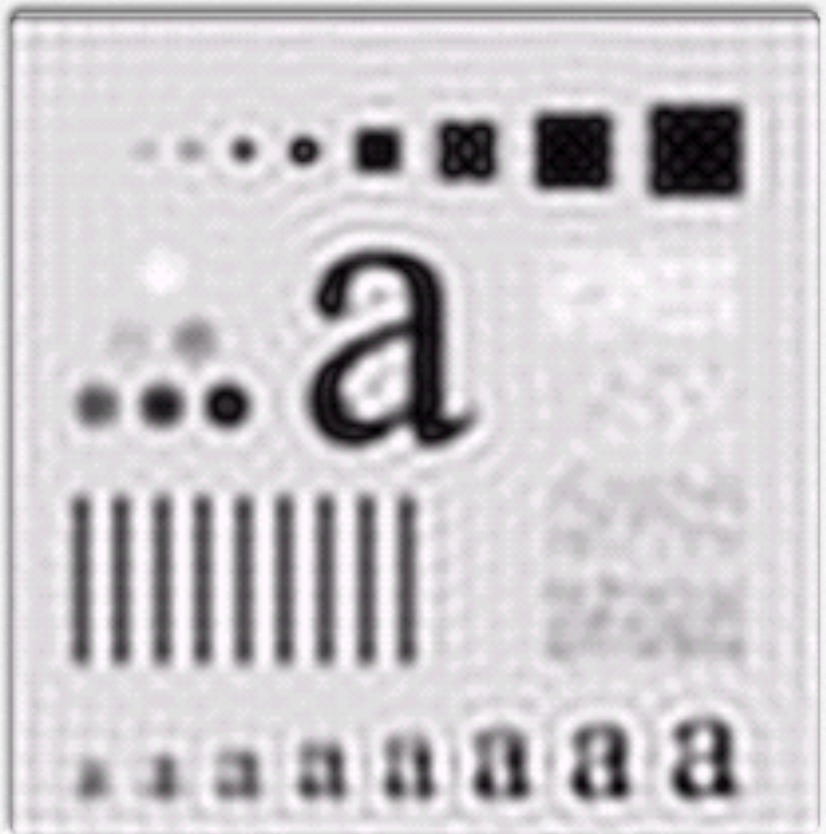


Image filtrée en augmentant le diamètre de la fonction porte



Source : DIP2

2. Réduction du bruit

2. Filtre moyenneur

a) Non pondéré

Image originale

Source :

2. Réduction du bruit

2. Filtre moyenneur

a) Non pondéré

Image originale

lissage moyen 3×3

Source :

2. RÉDUCTION DU BRUIT

2. Filtre moyenneur

a) Non pondéré

Image originale

lissage moyen 3 × 3

lissage moyen 5 × 5

Source :

2. RÉDUCTION DU BRUIT

2. Filtre moyenneur

a) Non pondéré

Image originale

lissage moyen 3 × 3

lissage moyen 5 × 5

lissage moyen 7 × 7

Source :

2. RÉDUCTION DU BRUIT

2. Filtre moyenneur

a) Non pondéré

Image originale

lissage moyen 3×3

lissage moyen 5×5

lissage moyen 7×7

Remarque :

- Quand N augmente, la variance du signal diminue
  - ✓ $N \to \infty$: on élimine complètement le bruit mais tout le reste aussi
    - ✓ variance $\to 0$, image constante

Source :

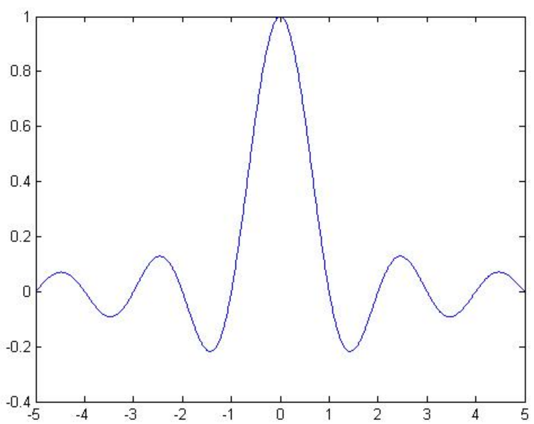
Filtres passe-bas

Filtre utilisé lors de filtrages spatiaux

Filtre moyenneur

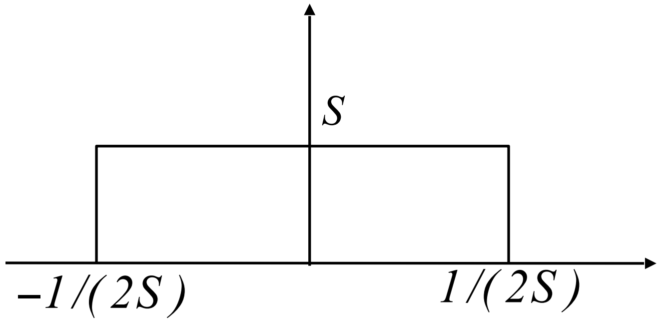
Forme spectrale du filtre

1D



$$H(u) = \frac{sinc(u/S)}{S}$$

Forme spatiale du filtre



$$h(x) = \Pi(Sx)$$

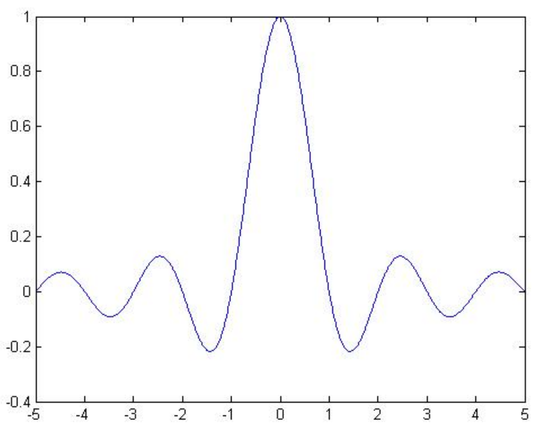
Filtres passe-bas

Filtre utilisé lors de filtrages spatiaux

Filtre moyenneur

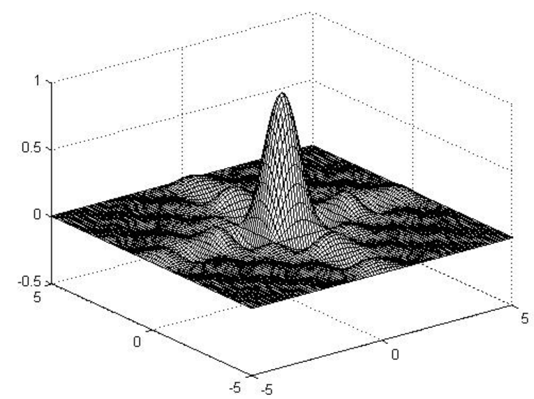
Forme spectrale du filtre

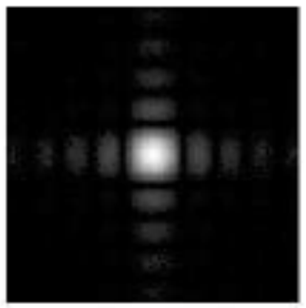
1D



$$H(u) = \frac{\operatorname{sinc}(u/S)}{S}$$

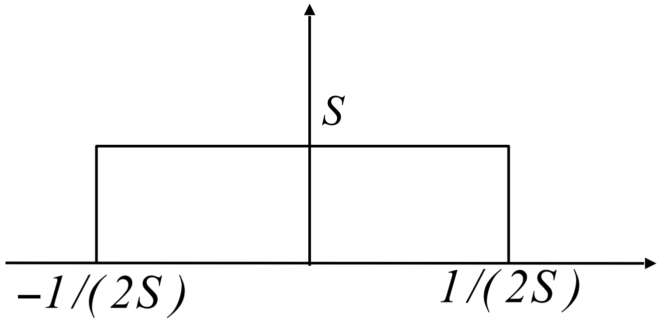
2D



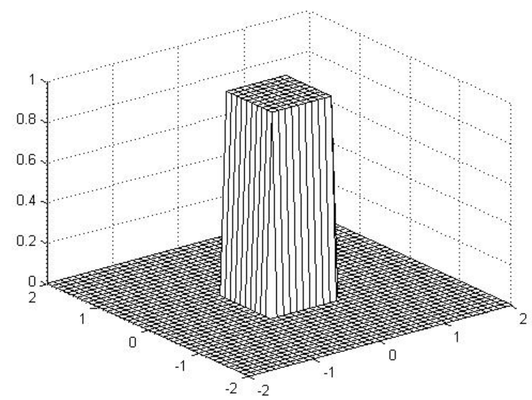


$$H(u,v) = \frac{\operatorname{sinc}(u/S_x)\operatorname{sinc}(v/S_y)}{S_x S_y}$$

Forme spatiale du filtre



$$h(x) = \Pi(Sx)$$



$$h(x,y) = \Pi(S_x x, S_y y)$$

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} / 9$$

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} / 25$$

Filtres passe-bas

Filtre utilisé lors de filtrages spatiaux

Filtre moyenneur

Exemple :

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} / 9, \quad 3 \times 3$$

5×5

11×11

15×15

Gauss, $\sigma = 7$

2. Réduction du bruit

2. Filtre moyenneur

b) Pondéré

lissage moyen 7 × 7

lissage gaussien $\sigma = 1.3$ (9 × 9)

Source :

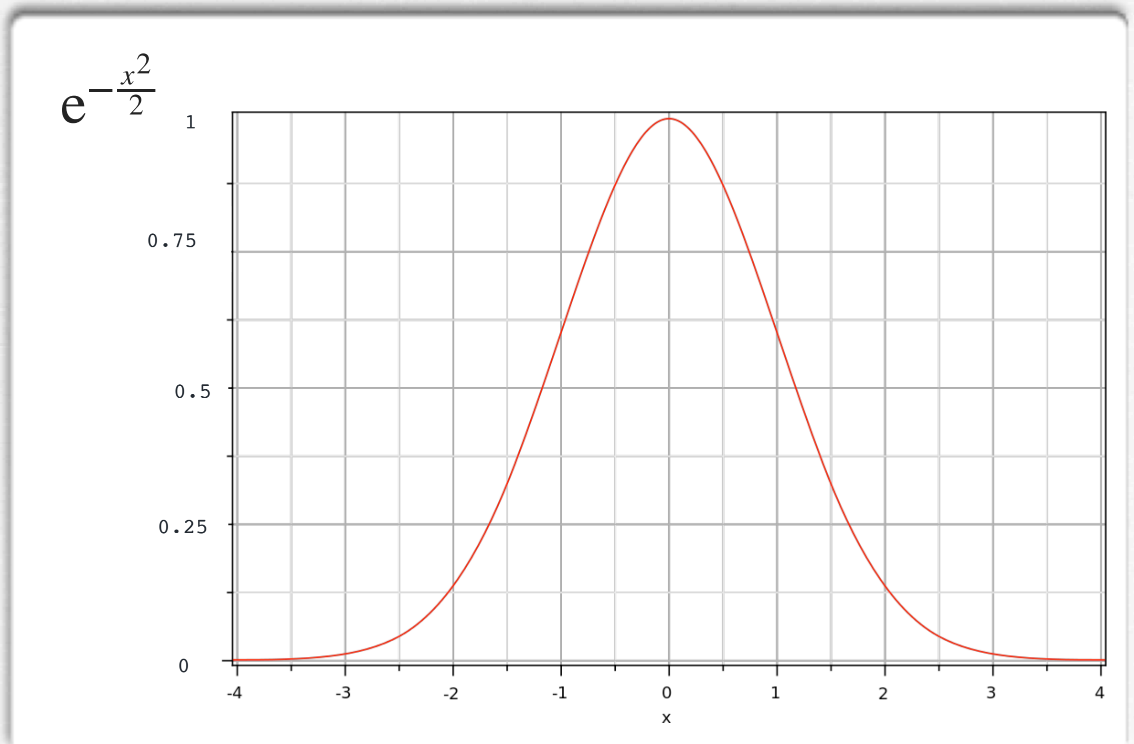
2. Réduction du bruit

2. Filtre moyenneur

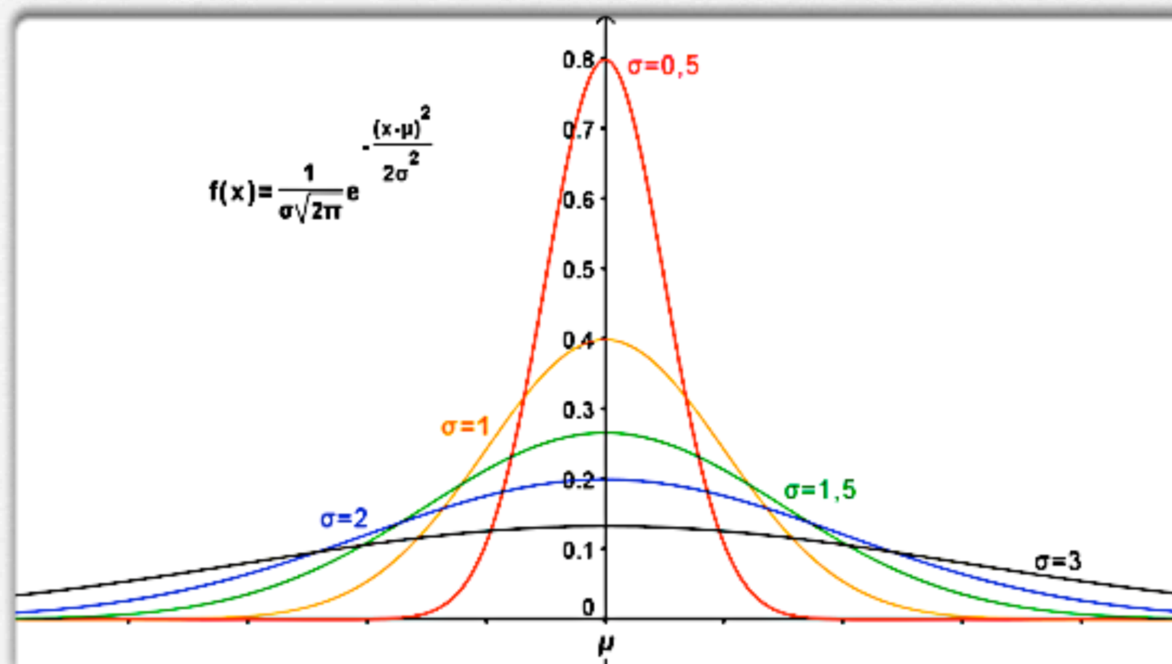
b) Pondéré

- Gaussien

$$g_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \rightleftharpoons e^{-2\pi^2\sigma^2u^2} = G_\sigma(u)$$



Le paramètre σ est la dispersion (variance) de la fonction



2. RÉDUCTION DU BRUIT

2. Filtre moyenneur

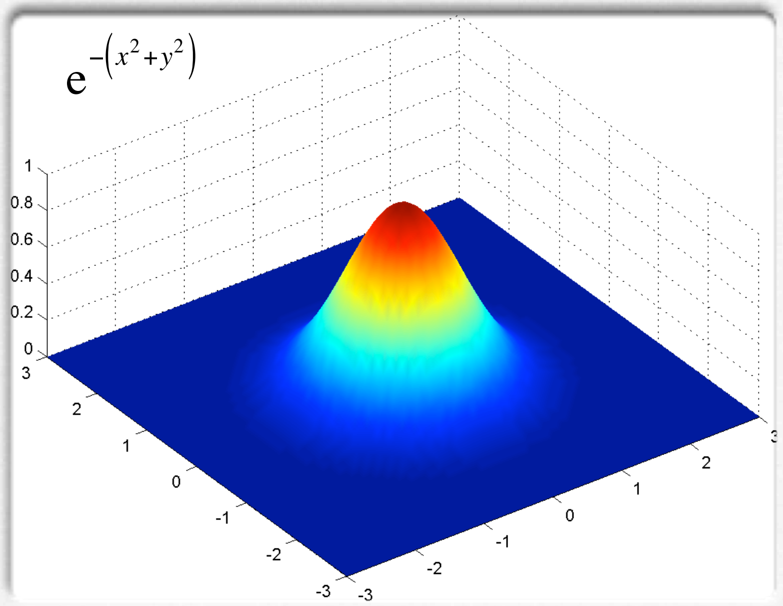
b) Pondéré

- Gaussien

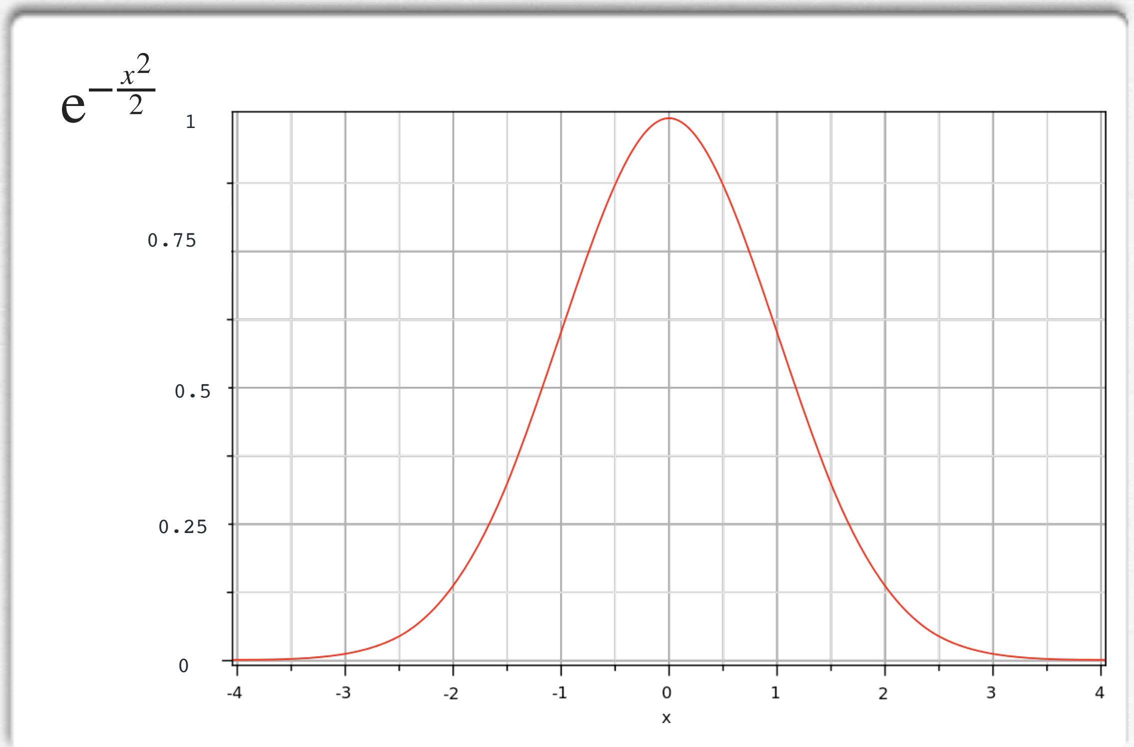
$$g_\sigma(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} = g_\sigma(x) g_\sigma(y)$$

$$\rightleftharpoons$$

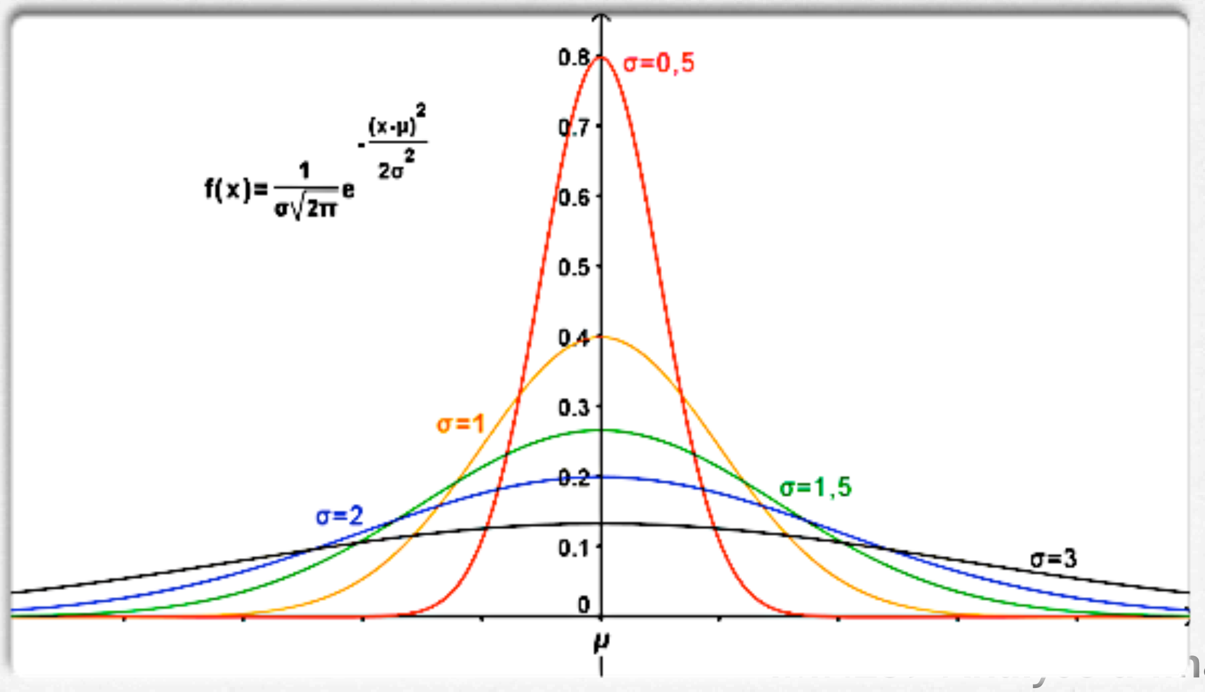
$$e^{-2\pi^2\sigma^2(u^2+v^2)} = G_\sigma(u,v)$$



$$g_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \rightleftharpoons e^{-2\pi^2\sigma^2u^2} = G_\sigma(u)$$



Le paramètre σ est la dispersion (variance) de la fonction



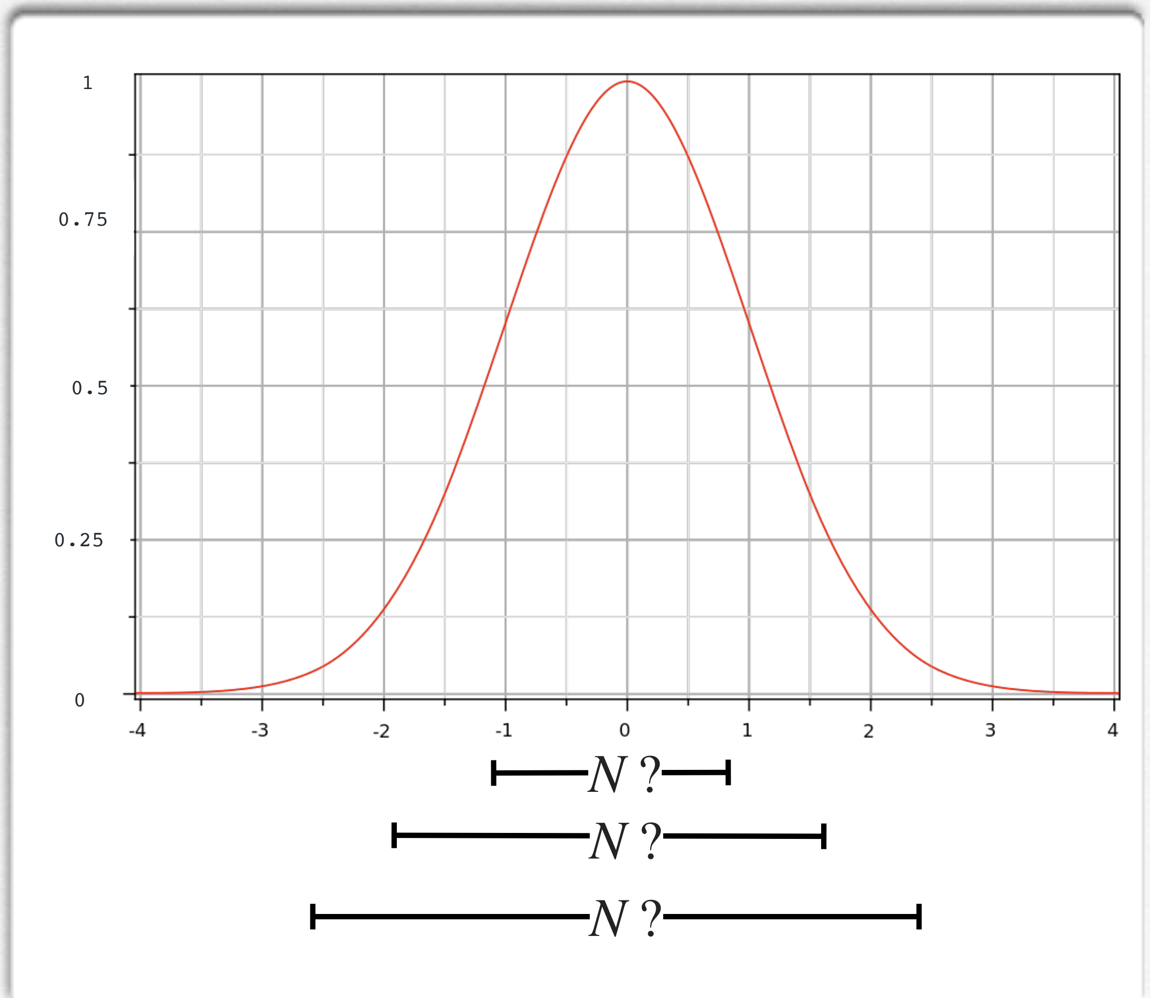
2. RÉDUCTION DU BRUIT

2. Filtre moyenneur

b) Pondéré

- Gaussien

Le paramètre σ est la dispersion (variance) de la fonction



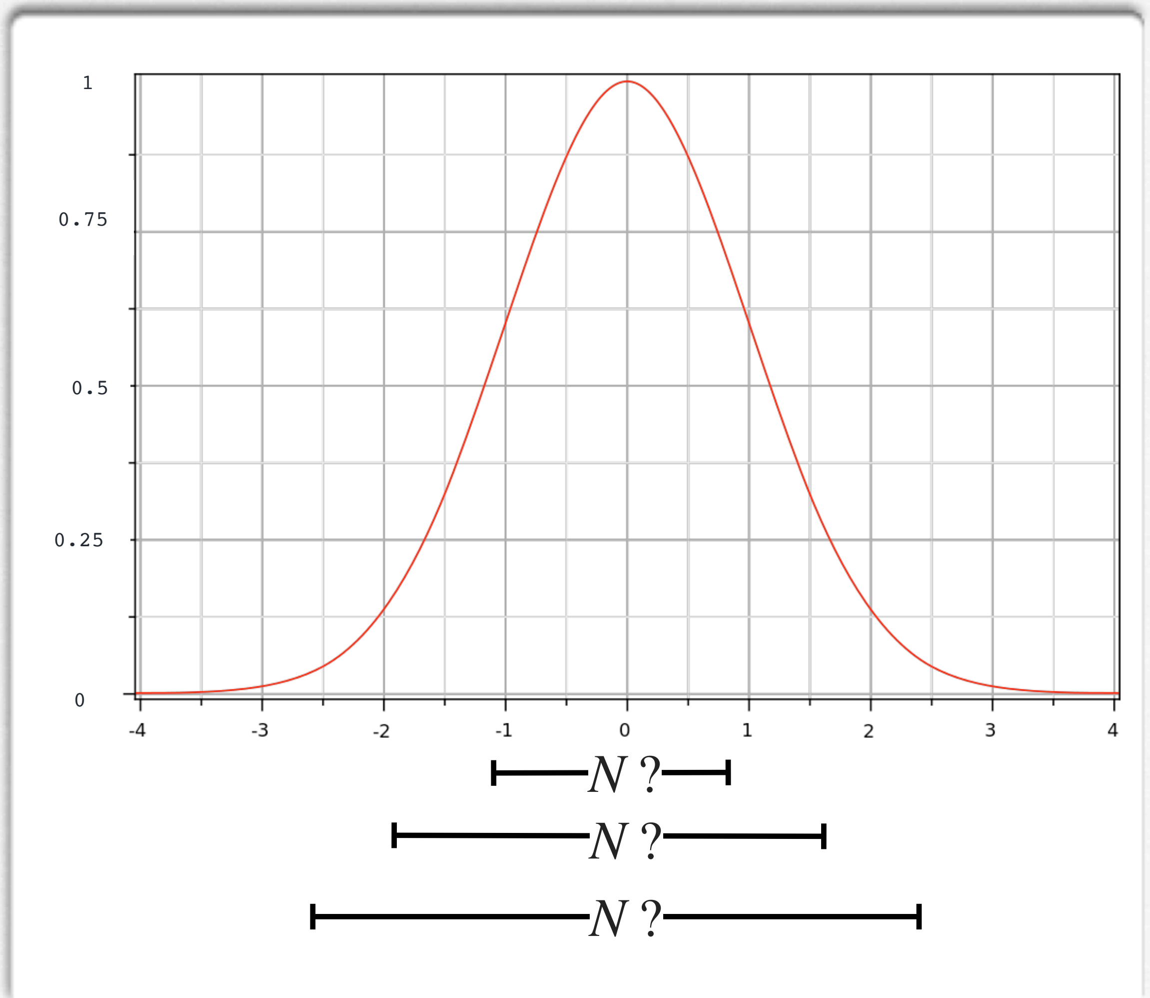
2. Réduction du bruit

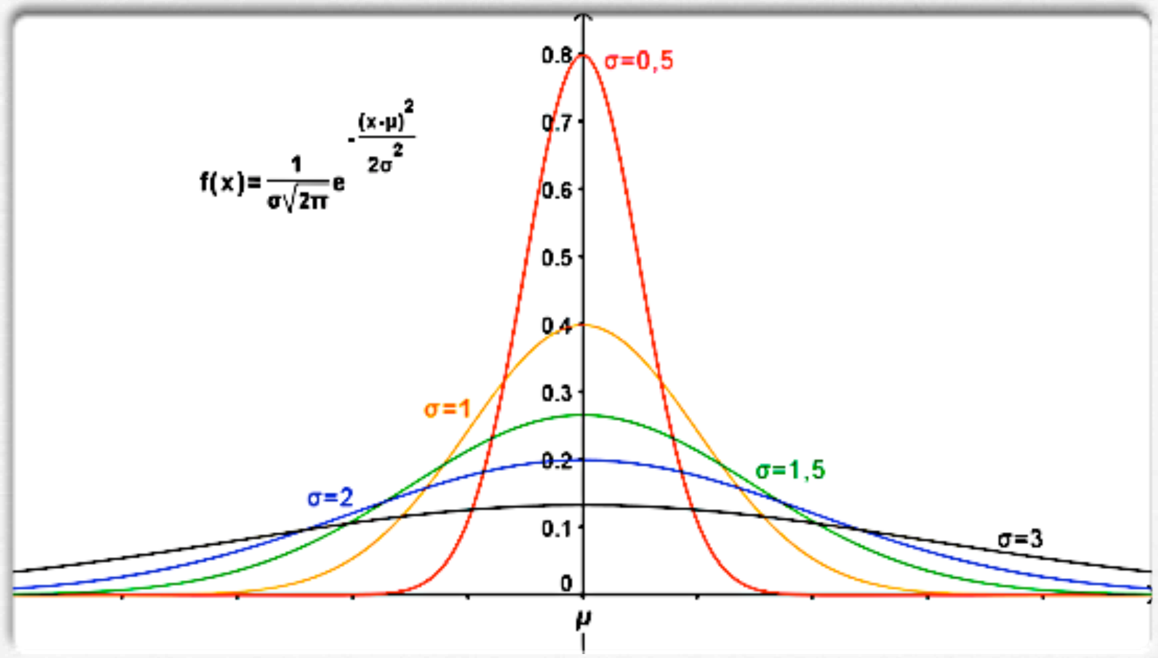
2. Filtre moyenneur

b) Pondéré

- Gaussien

Le paramètre σ est la dispersion (variance) de la fonction





Si σ augmente, N devrait aussi augmenter

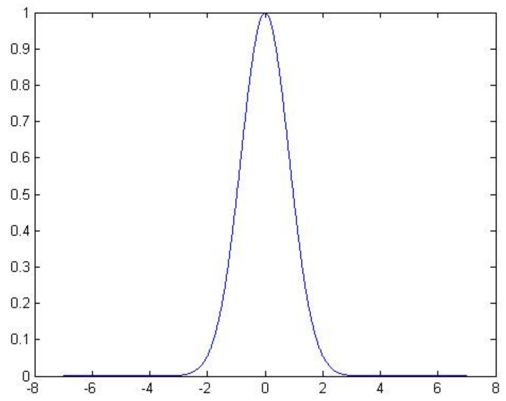
Filtres passe-bas

Filtre utilisé lors de filtrages fréquentiels ou spatiaux

Filtre gaussien

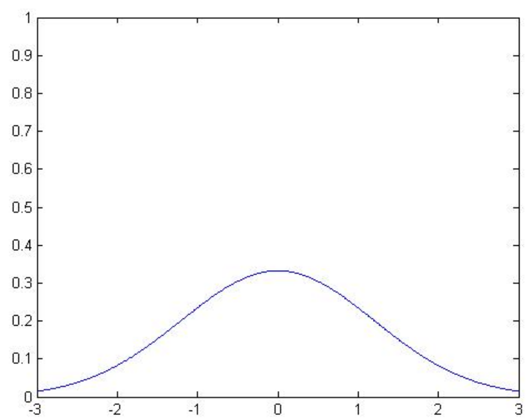
Forme spectrale du filtre

1D



$$H(u) = e^{-2\pi^2\sigma^2 u^2}$$

Forme spatiale du filtre



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

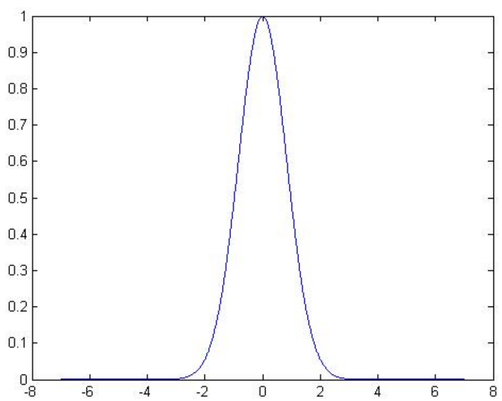
Filtres passe-bas

Filtre utilisé lors de filtrages fréquentiels ou spatiaux

Filtre gaussien

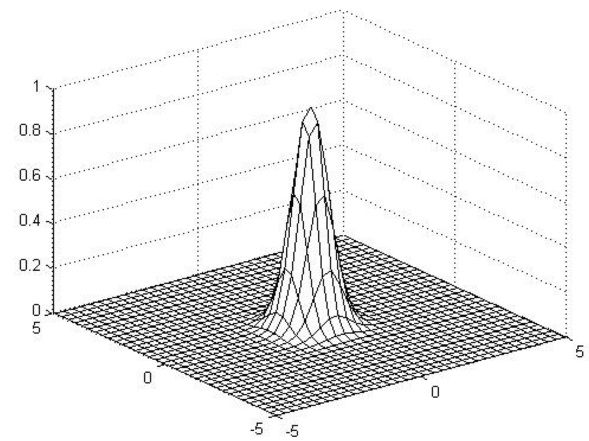
Forme spectrale du filtre

1D



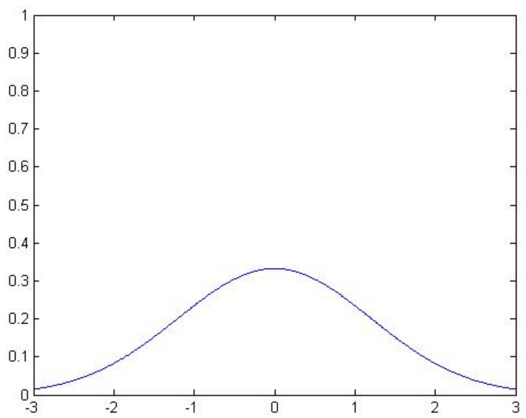
$$H(u) = e^{-2\pi^2\sigma^2u^2}$$

2D

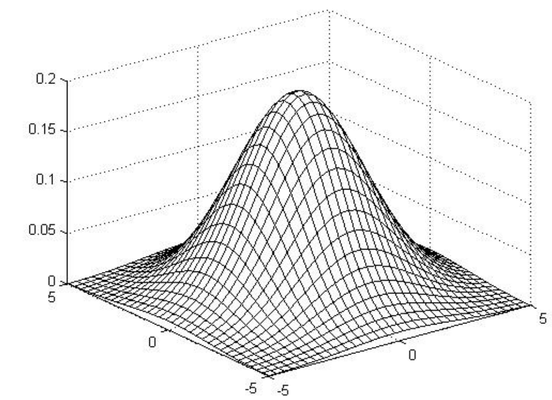


$$H(u,v) = e^{-2\pi^2(u^2+v^2)\sigma^2}$$

Forme spatiale du filtre



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



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

$\frac{1}{273}$

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

$\sigma = 1$

Note: normalement, un filtre spatial gaussien doit avoir une taille de $(6\sigma + 1) \times (6\sigma + 1)$

2. Réduction du bruit

2. Filtre moyenneur

b) Pondéré

- Gaussien

$\sigma = 1$ (7×7)

2. RÉDUCTION DU BRUIT

2. Filtre moyenneur

b) Pondéré

- Gaussien

$\sigma = 1$ (7×7)

$\sigma = 3$ (19×19)

2. Réduction du bruit

2. Filtre moyenneur

b) Pondéré

- Gaussien

$\sigma = 1$ (7×7)

$\sigma = 3$ (19×19)

$\sigma = 5$ (31×31)

2. Réduction du bruit

2. Filtre moyenneur

b) Pondéré

- Gaussien

$\sigma = 1$ (7×7)

$\sigma = 3$ (19×19)

$\sigma = 5$ (31×31)

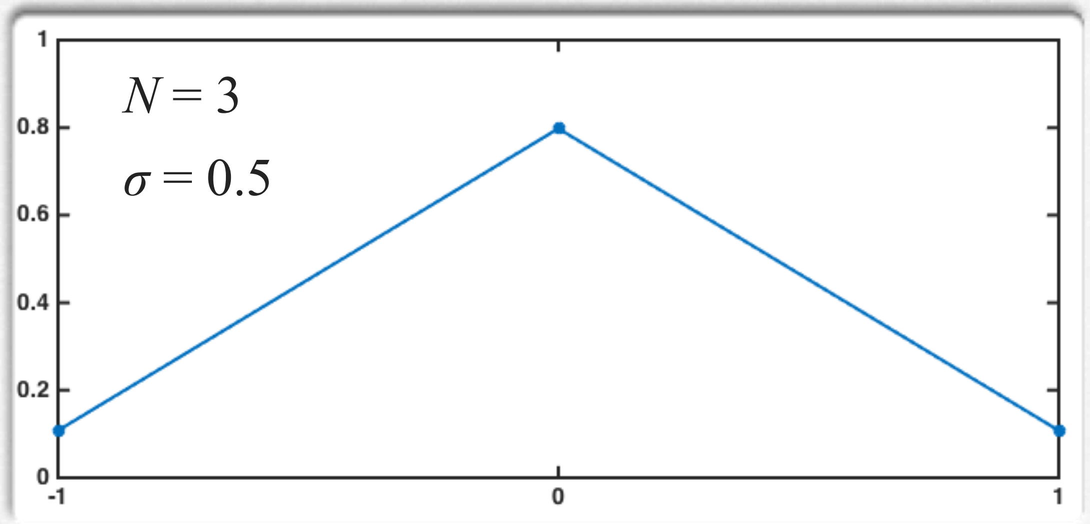
$\sigma = 7$ (43×43)

2. RÉDUCTION DU BRUIT

2. Filtre moyenneur

b) Pondéré

- Gaussien



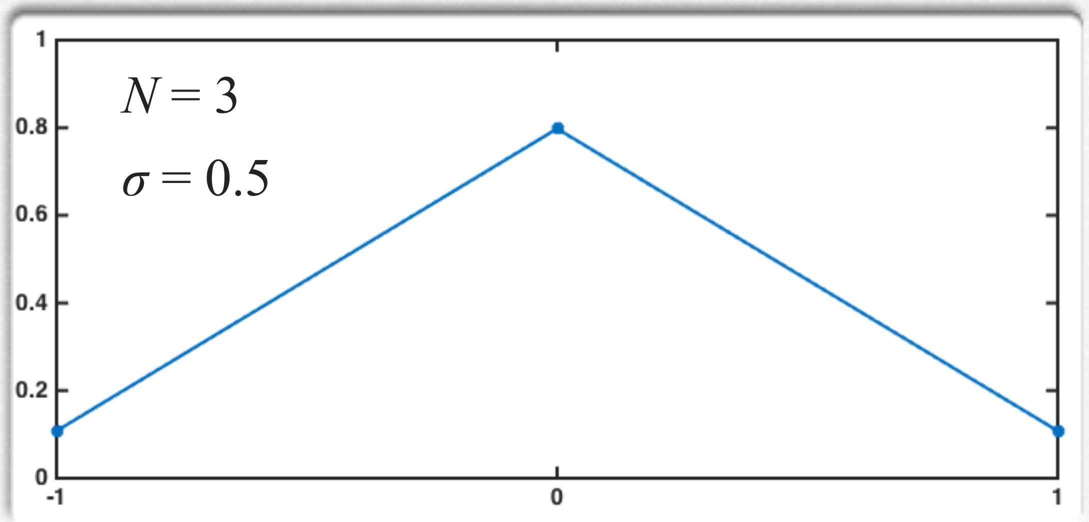
Source :

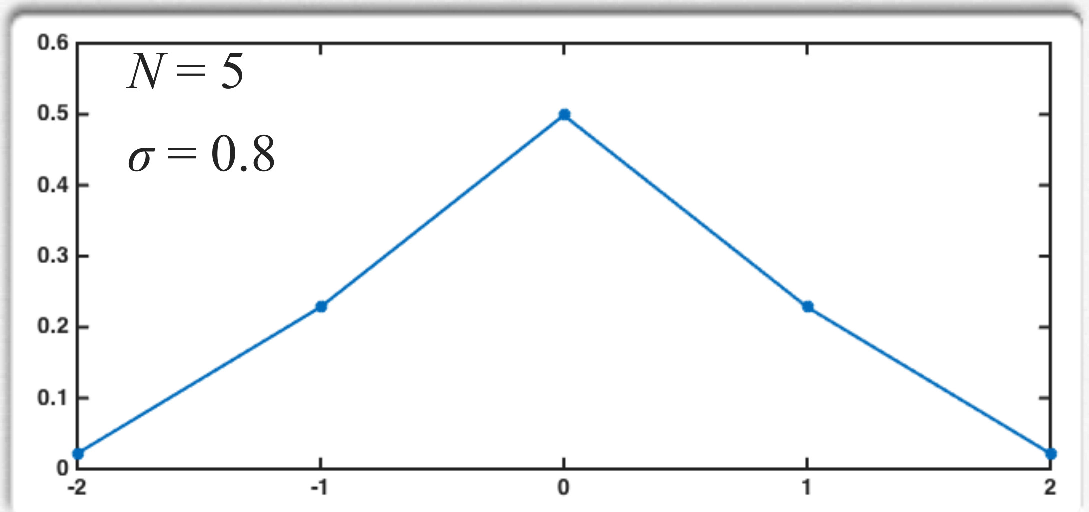
2. Réduction du bruit

2. Filtre moyenneur

b) Pondéré

- Gaussien





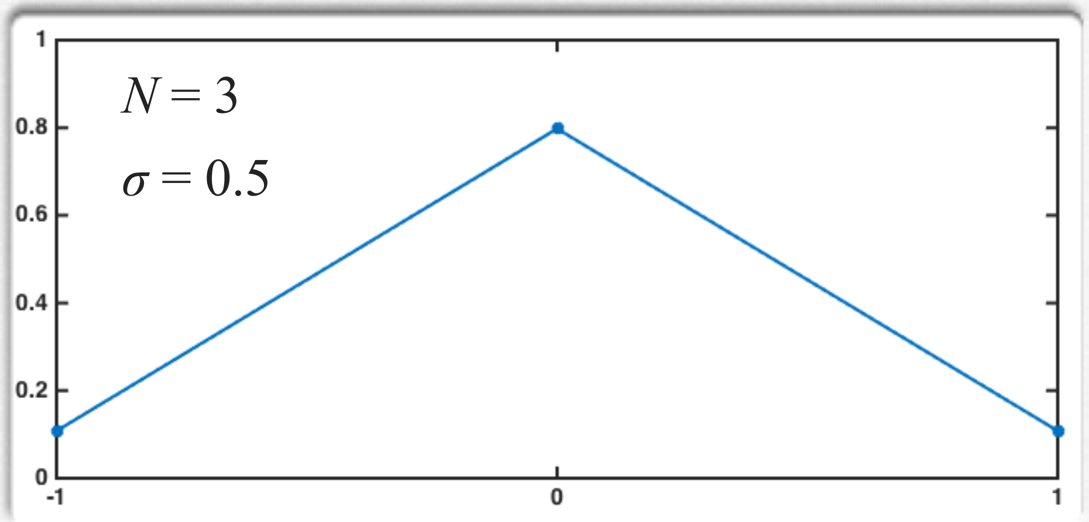
Source :

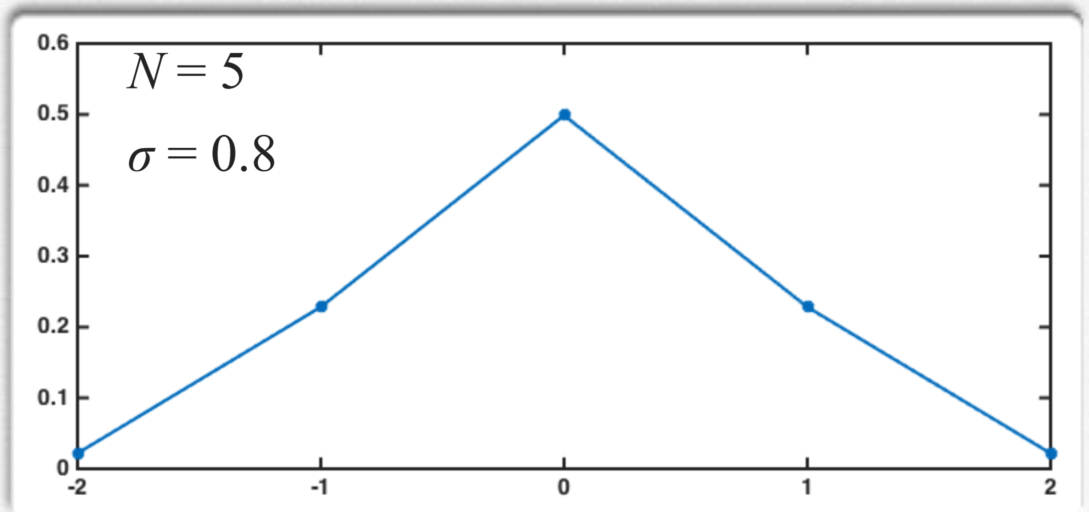
2. Réduction du bruit

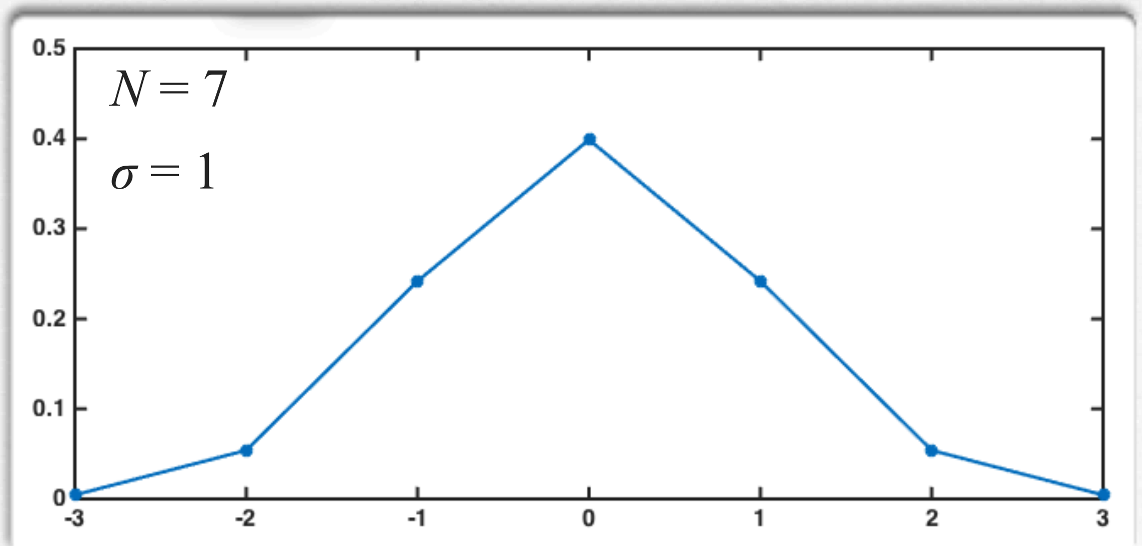
2. Filtre moyenneur

b) Pondéré

- Gaussien







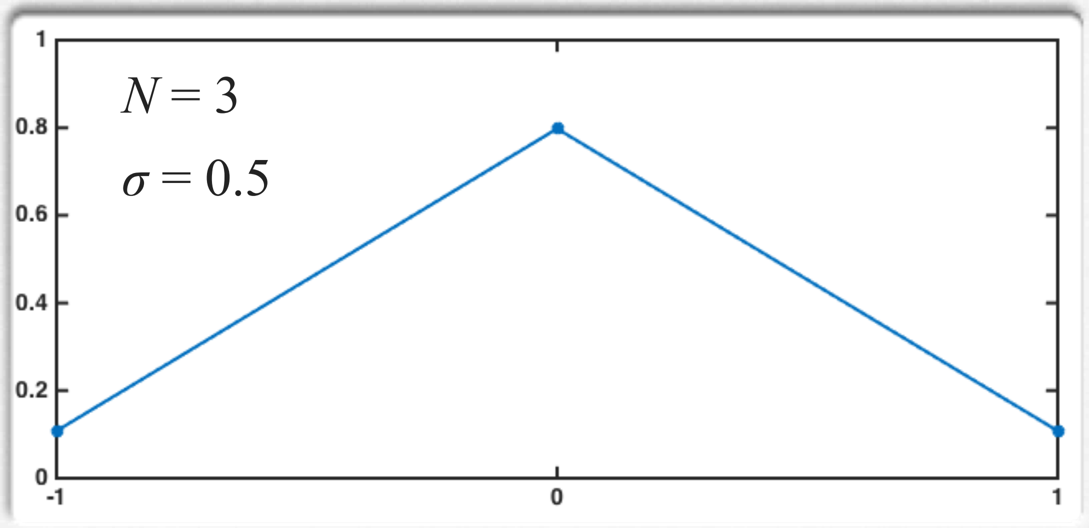
Source :

2. RÉDUCTION DU BRUIT

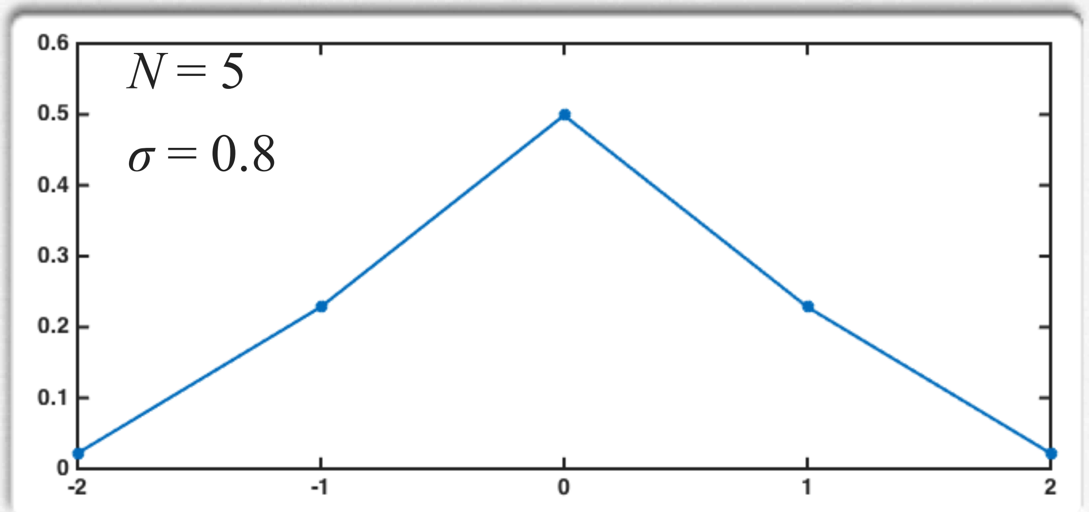
2. Filtre moyenneur

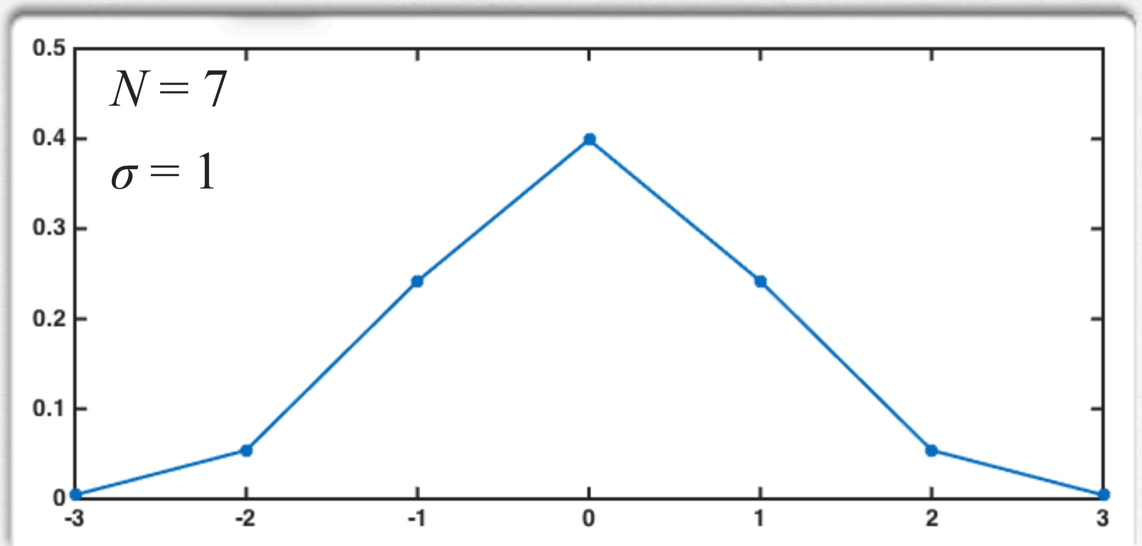
b) Pondéré

- Gaussien



> *Remarque* :
>
> - Les cas 3×3 et 5×5 ne contiennent pas assez de chiffres pour considérer h_b comme vraiment gaussien





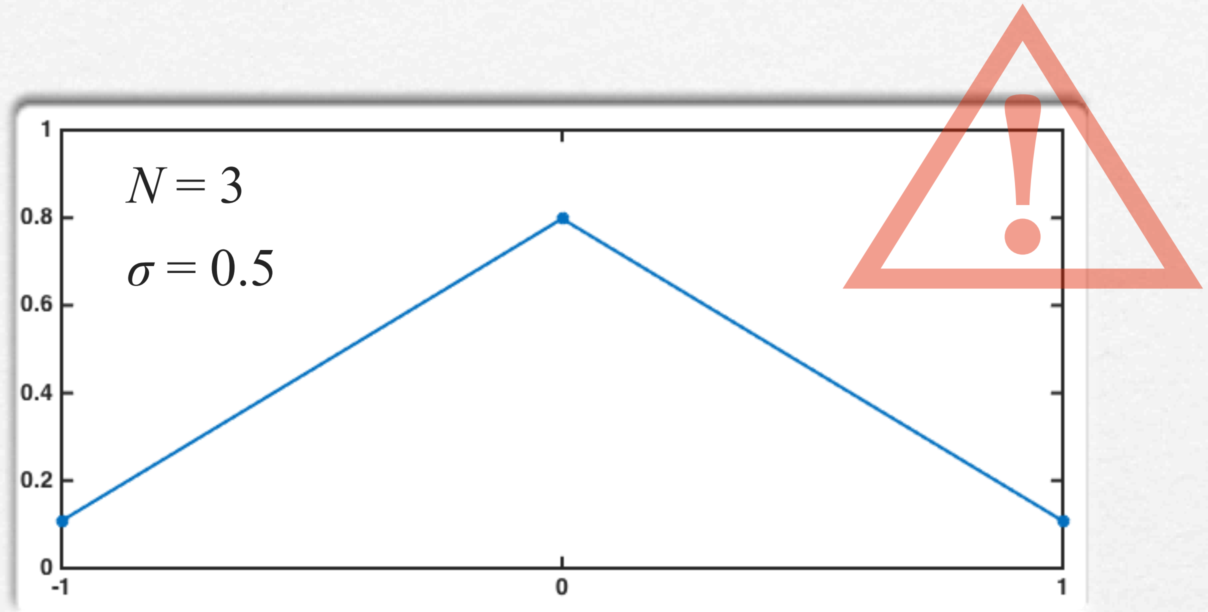
Source :

2. Réduction du bruit

2. Filtre moyenneur

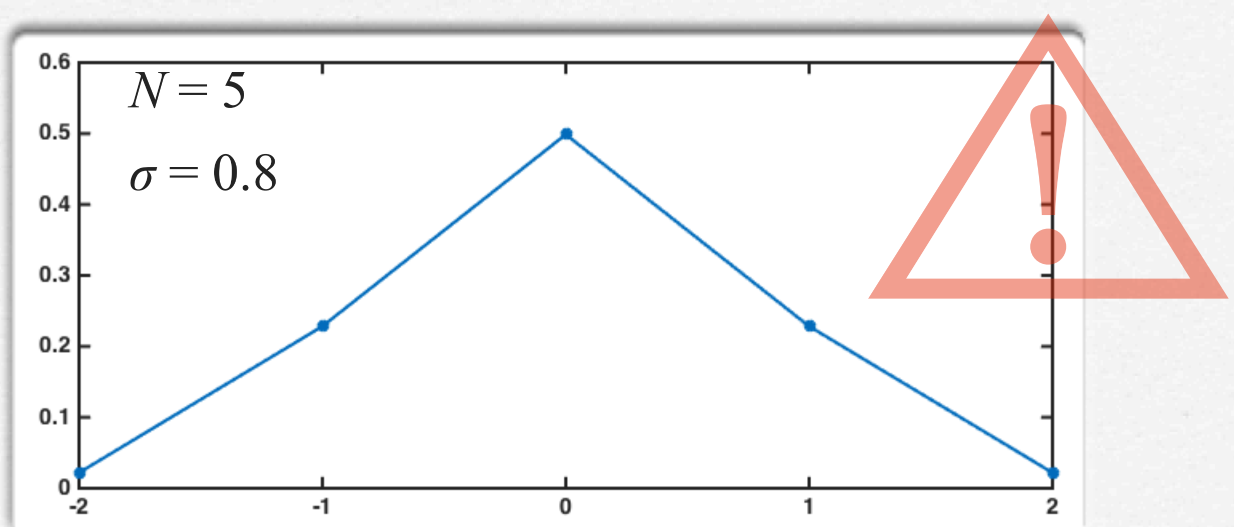
b) Pondéré

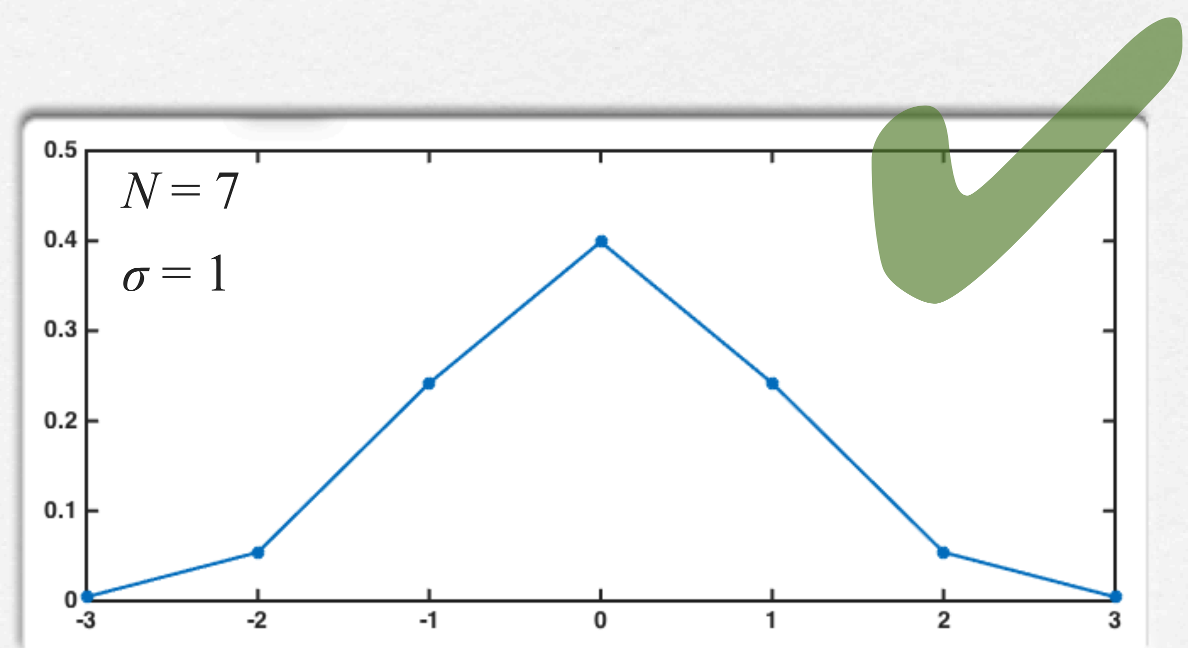
- Gaussien



Remarque :

- Les cas 3×3 et 5×5 ne contiennent pas assez de chiffres pour considérer h_b comme vraiment gaussien





Source :

2. Réduction du bruit

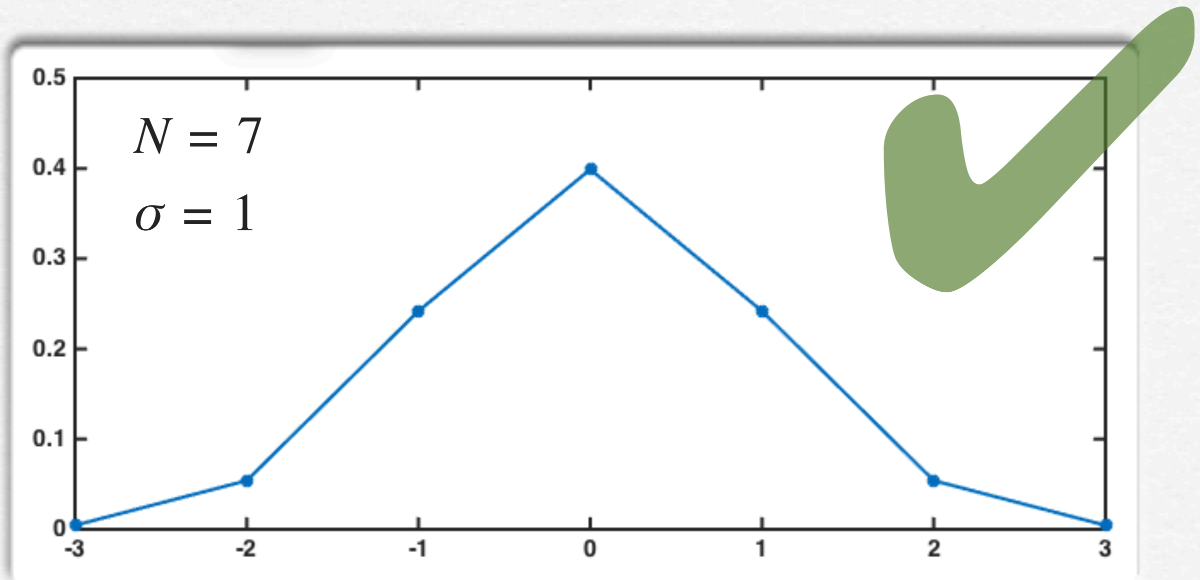
2. Filtre moyenneur

b) Pondéré

- Gaussien

Remarque :

- si on pose $\sigma < 1$, l'échantillonnage fait que les coefficients assez grands ne sont pas assez nombreux pour considérer h_b comme vraiment gaussien
  - ✓ revient à utiliser un $N < 7$



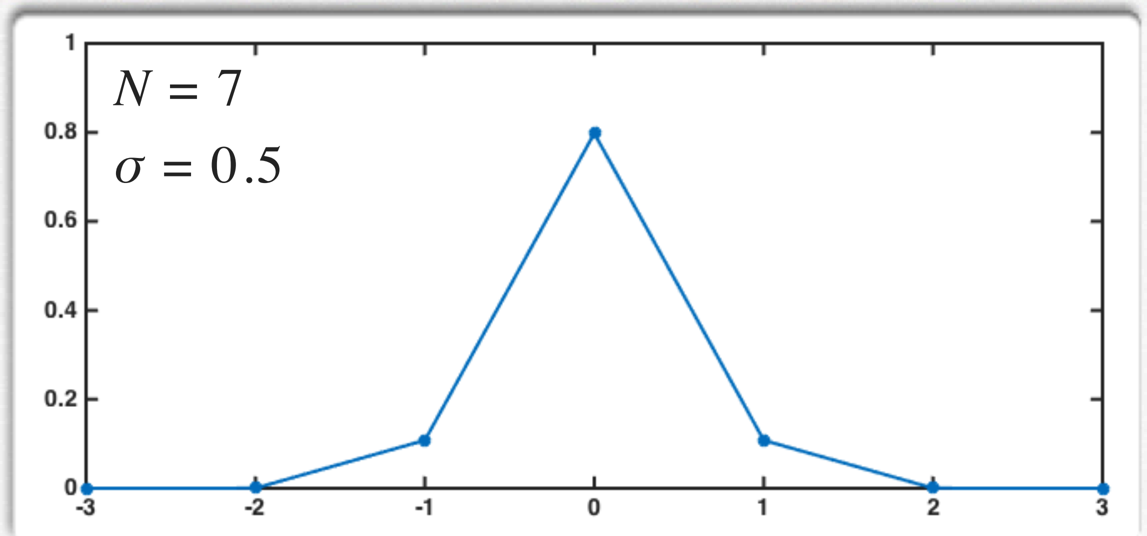
Source :

2. RÉDUCTION DU BRUIT

2. Filtre moyenneur

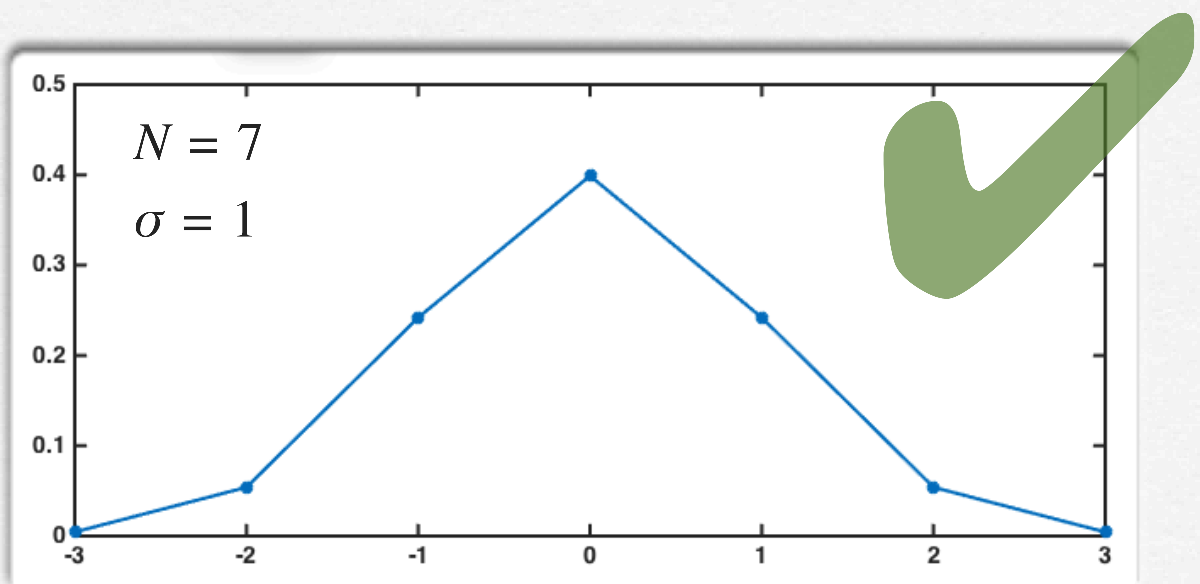
b) Pondéré

- Gaussien



Remarque :

- si on pose $\sigma < 1$, l'échantillonnage fait que les coefficients assez grands ne sont pas assez nombreux pour considérer h_b comme vraiment gaussien
  - ✓ revient à utiliser un $N < 7$



Source :

2. Réduction du bruit

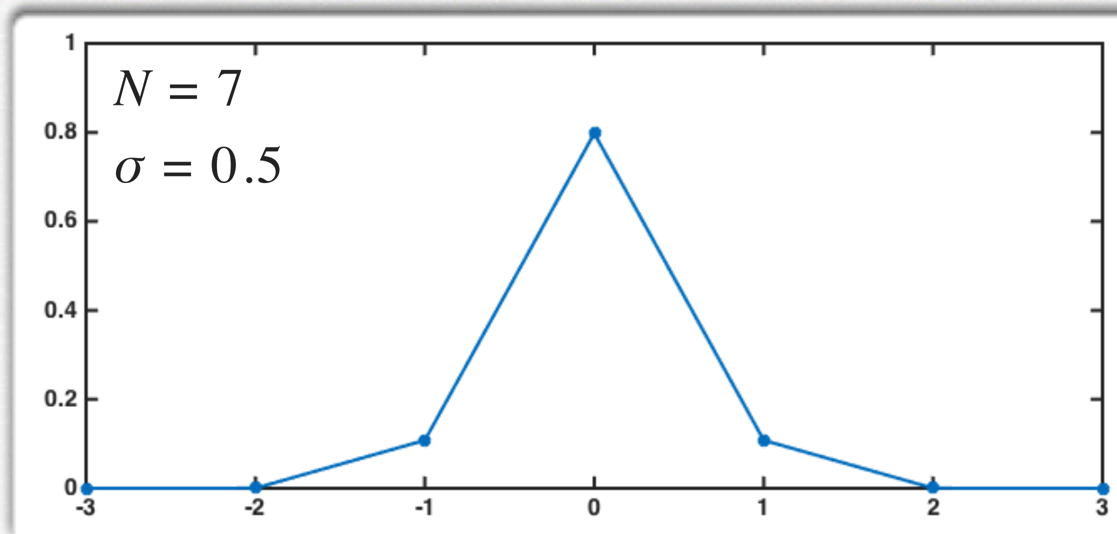
2. Filtre moyenneur

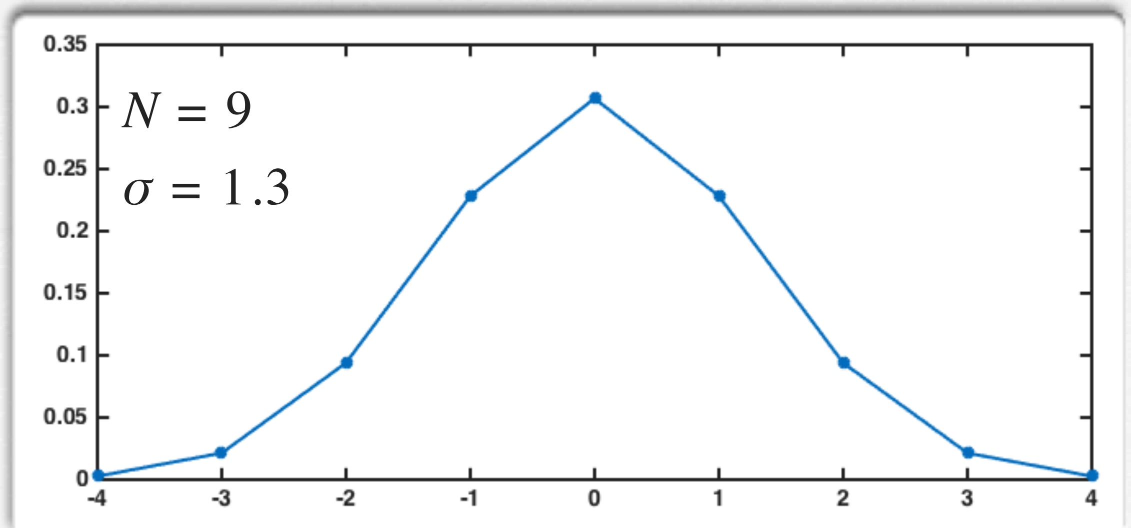
b) Pondéré

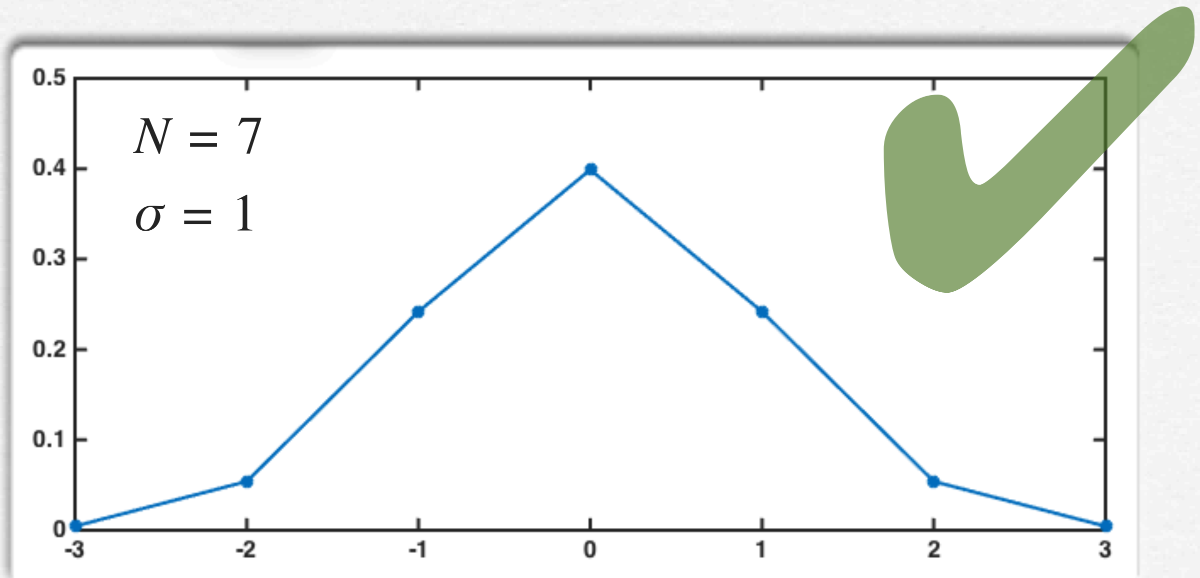
- Gaussien

Remarque :

- si on pose $\sigma < 1$, l'échantillonnage fait que les coefficients assez grands ne sont pas assez nombreux pour considérer h_b comme vraiment gaussien
  - ✓ revient à utiliser un $N < 7$







Source :

2. Réduction du bruit

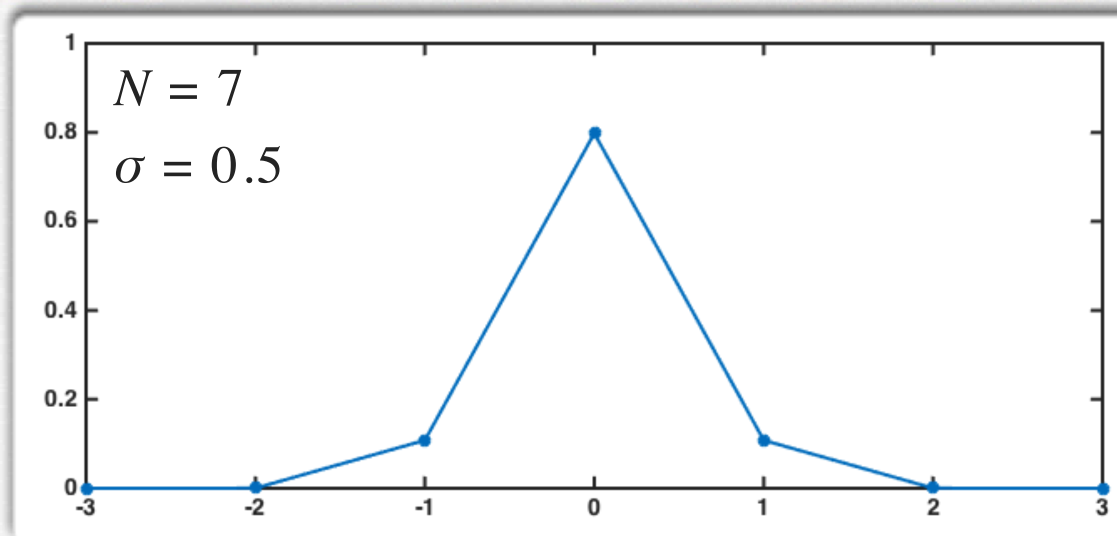
2. Filtre moyenneur

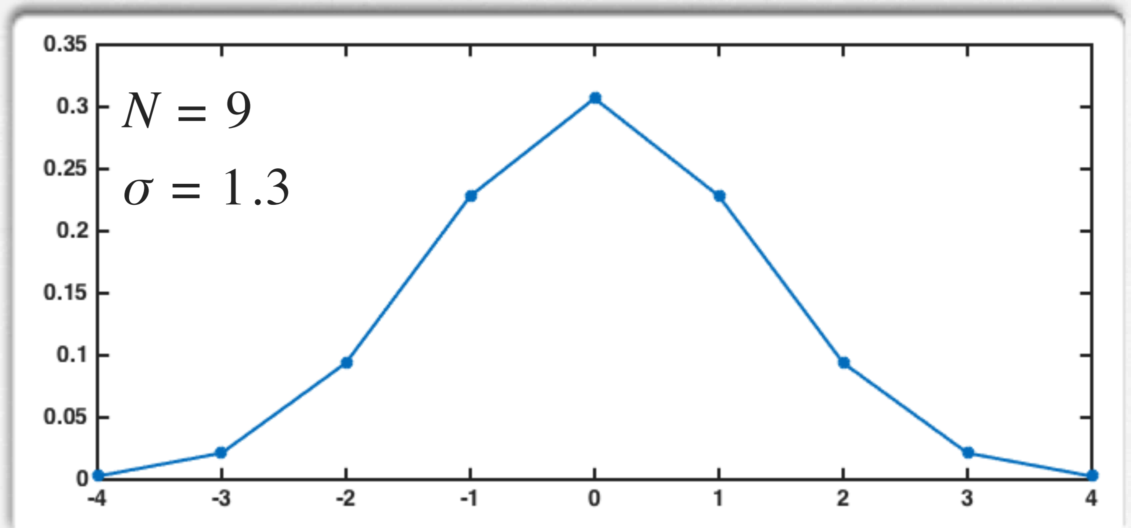
b) Pondéré

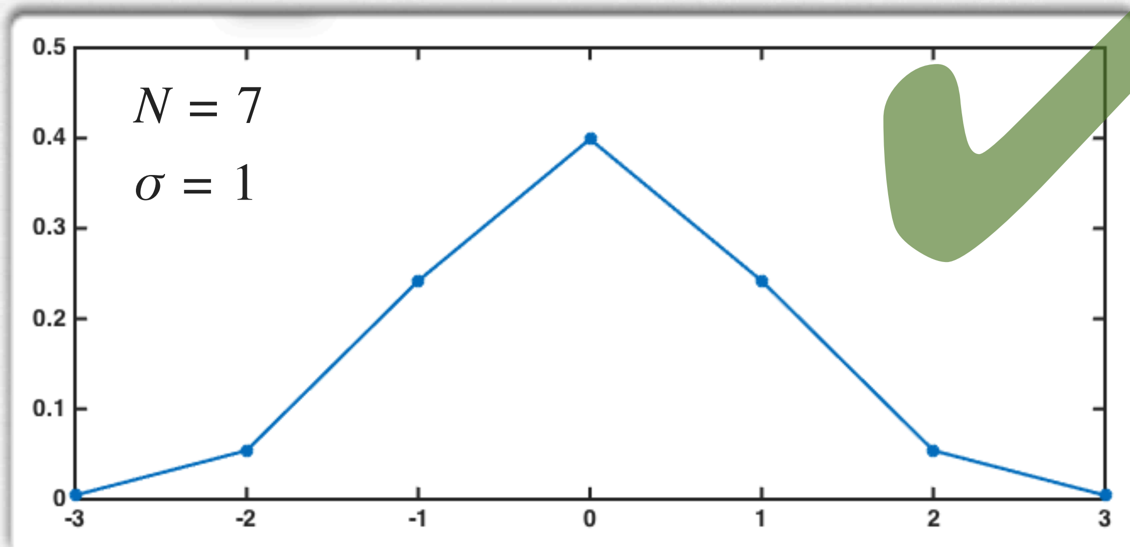
- Gaussien

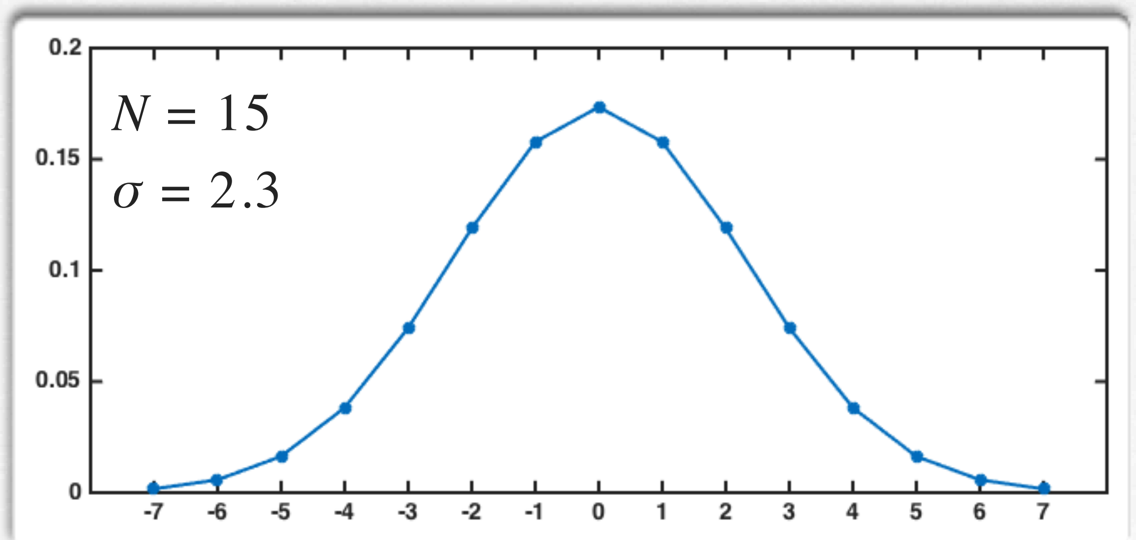
Remarque :

- si on pose $\sigma < 1$, l'échantillonnage fait que les coefficients assez grands ne sont pas assez nombreux pour considérer h_b comme vraiment gaussien
  - ✓ revient à utiliser un $N < 7$









Source :

2. Réduction du bruit

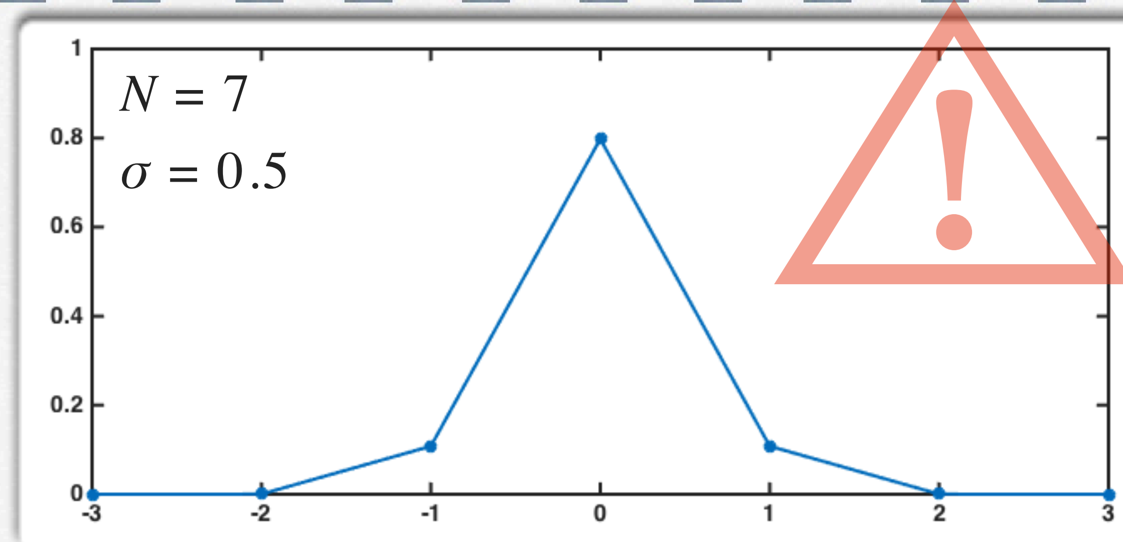
2. Filtre moyenneur

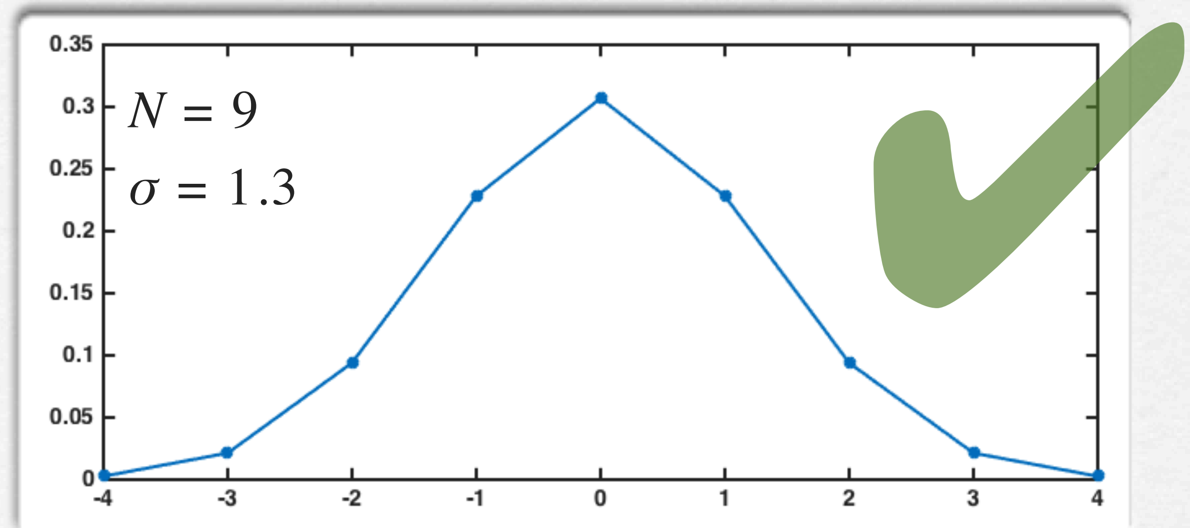
b) Pondéré

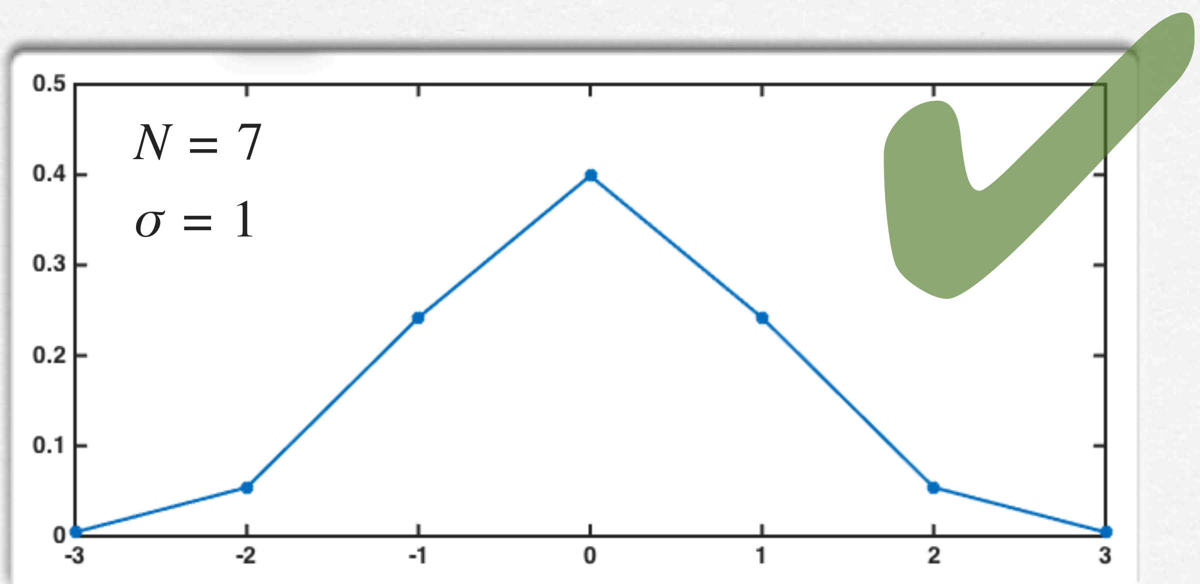
- Gaussien

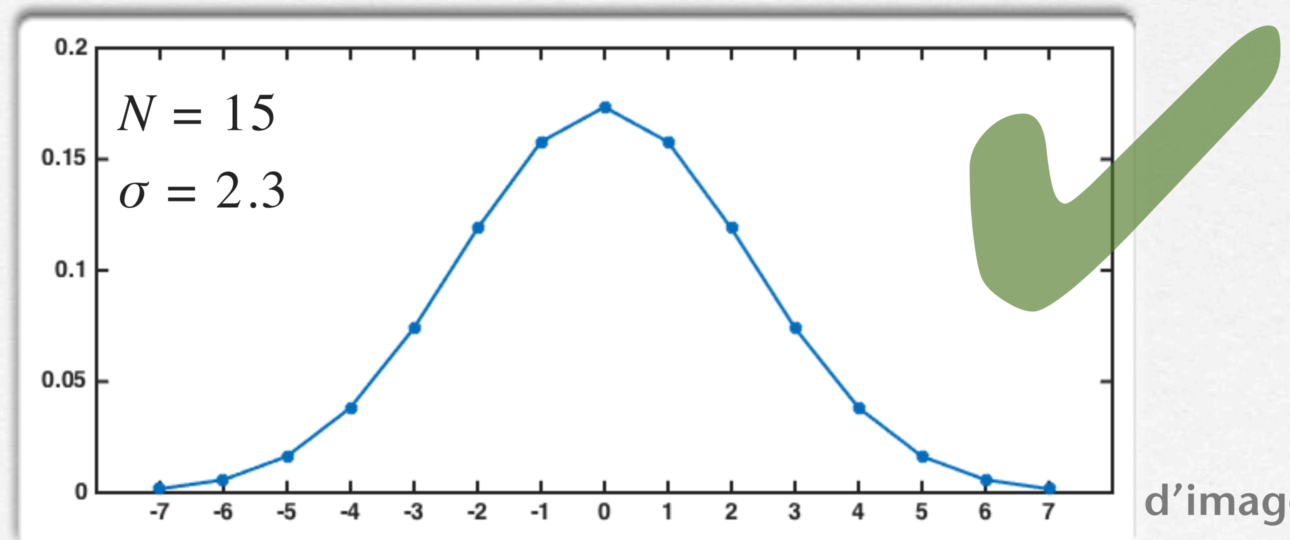
Remarque :

- si on pose $\sigma < 1$, l'échantillonnage fait que les coefficients assez grands ne sont pas assez nombreux pour considérer h_b comme vraiment gaussien
  - ✓ revient à utiliser un $N < 7$









Source :

2. Réduction du bruit

2. Filtre moyenneur

b) Pondéré

- Gaussien - *avantages*

2. RÉDUCTION DU BRUIT

2. Filtre moyenneur

b) Pondéré

- Gaussien - *avantages*
  - ✓ Isotrope (plus vrai pour *N* grand)

Source :

2. Réduction du bruit

2. Filtre moyenneur

b) Pondéré

- Gaussien - *avantages*
  - ✓ Isotrope (plus vrai pour *N* grand)
  - ✓ L'importance du coefficient central est plus grande ce qui lui donne plus d'importance

Source :

2. Réduction du bruit

2. Filtre moyenneur

b) Pondéré

- Gaussien - *avantages*
  - ✓ Isotrope (plus vrai pour N grand)
  - ✓ L'importance du coefficient central est plus grande ce qui lui donne plus d'importance
  - ✓ Relation simple et intuitive entre σ et l'effet du filtre

Source :

2. Réduction du bruit

2. Filtre moyenneur

b) Pondéré

- Gaussien - *avantages*
  - ✓ Isotrope (plus vrai pour N grand)
  - ✓ L'importance du coefficient central est plus grande ce qui lui donne plus d'importance
  - ✓ Relation simple et intuitive entre σ et l'effet du filtre
  - ✓ Séparabilité
    - ➡ Convoluer l'image avec un filtre 1D horizontal (ligne par ligne)
    - ➡ Convoluer le résultat avec un filtre 1D vertical (colonne par colonne)

Source :

2. Réduction du bruit

2. Filtre moyenneur

b) Pondéré

- Gaussien - *avantages*
  - ✓ Isotrope (plus vrai pour N grand)
  - ✓ L'importance du coefficient central est plus grande ce qui lui donne plus d'importance
  - ✓ Relation simple et intuitive entre σ et l'effet du filtre
  - ✓ Séparabilité
    - ➡ Convoluer l'image avec un filtre 1D horizontal (ligne par ligne)
    - ➡ Convoluer le résultat avec un filtre 1D vertical (colonne par colonne)
  - ✓ Convolution d'une gaussienne avec une gaussienne donne une autre gaussienne
    - ➡ Possible de filtrer une image d'abord avec une petite gaussienne;
    - ➡ Ensuite, filtrer le résultat avec une autre petite gaussienne; Le résultat global est l'équivalent de filtrer l'image originale avec une plus grande gaussienne;
    - ➡ *Exemple*, filtrer 2 fois avec une gaussienne de paramètre σ est l'équivalent de filtrer une fois avec un paramètre à $\sqrt{2}\sigma$

Source :

2. Réduction du bruit

2. Filtre moyenneur

b) Pondéré

- Gaussien - *avantages*
  - ✓ Isotrope (plus vrai pour N grand)
  - ✓ L'importance du coefficient central est plus grande ce qui lui donne plus d'importance
  - ✓ Relation simple et intuitive entre σ et l'effet du filtre
  - ✓ Séparabilité
    - ➡ Convoluer l'image avec un filtre 1D horizontal (ligne par ligne)
    - ➡ Convoluer le résultat avec un filtre 1D vertical (colonne par colonne)
  - ✓ Convolution d'une gaussienne avec une gaussienne donne une autre gaussienne
    - ➡ Possible de filtrer une image d'abord avec une petite gaussienne;
    - ➡ Ensuite, filtrer le résultat avec une autre petite gaussienne; Le résultat global est l'équivalent de filtrer l'image originale avec une plus grande gaussienne;
    - ➡ *Exemple*, filtrer 2 fois avec une gaussienne de paramètre σ est l'équivalent de filtrer une fois avec un paramètre à $\sqrt{2}\sigma$

- *désavantage*
  - ✓ Ne prend pas en compte le contenu de l'image (filtre linéaire)

Source :

2. RÉDUCTION DU BRUIT

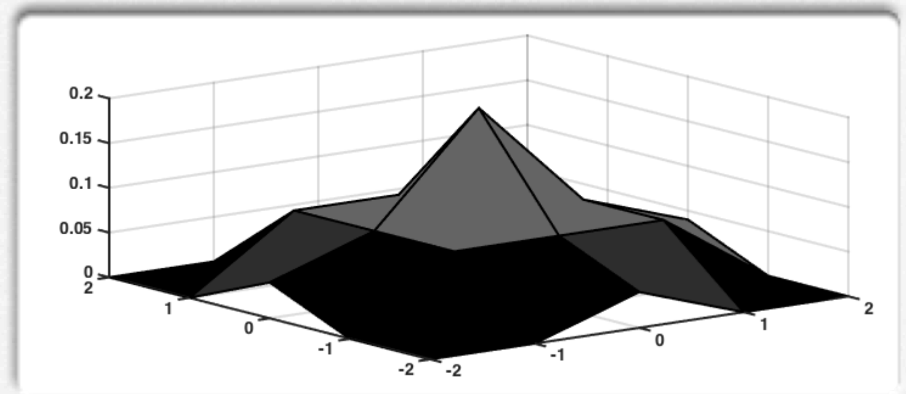
2. Filtre moyenneur

b) Pondéré

- Autres filtres moyenneur pondérés

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 2 & 2 & 0 \\ 1 & 2 & 5 & 2 & 1 \\ 0 & 2 & 2 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} / 25$$

Filtre conique



Source :

2. RÉDUCTION DU BRUIT

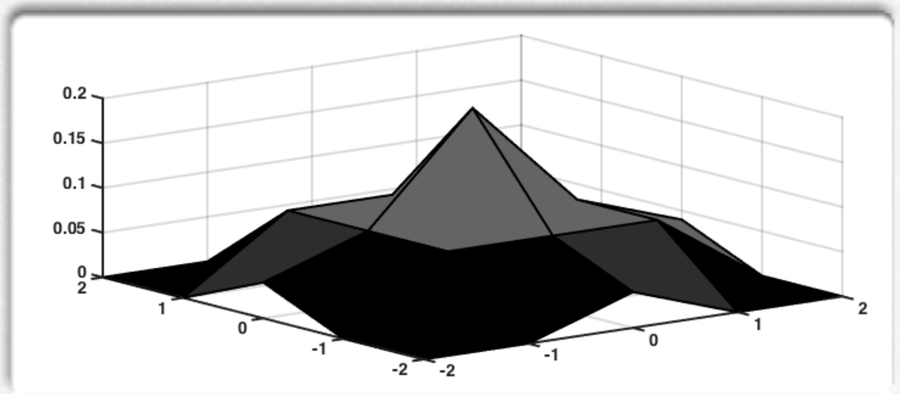
2. Filtre moyenneur

b) Pondéré

- Autres filtres moyenneur pondérés

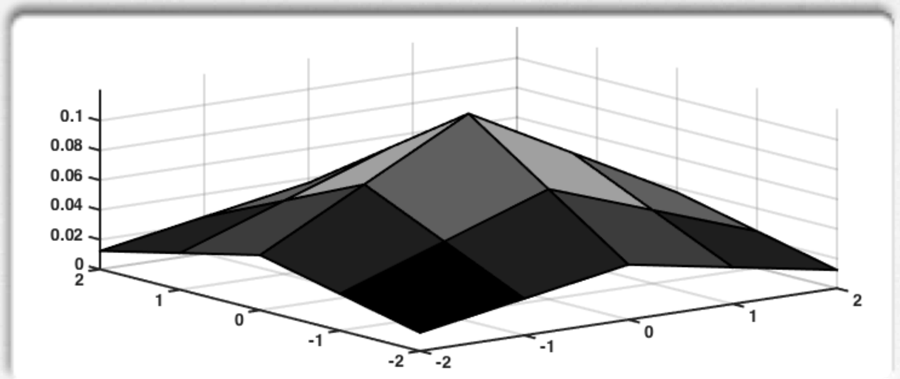
$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 2 & 2 & 0 \\ 1 & 2 & 5 & 2 & 1 \\ 0 & 2 & 2 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} / 25$$

Filtre conique



$$\begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 4 & 6 & 4 & 2 \\ 3 & 6 & 9 & 6 & 3 \\ 2 & 4 & 6 & 4 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix} / 81$$

Filtre pyramidal



Source :

2. RÉDUCTION DU BRUIT

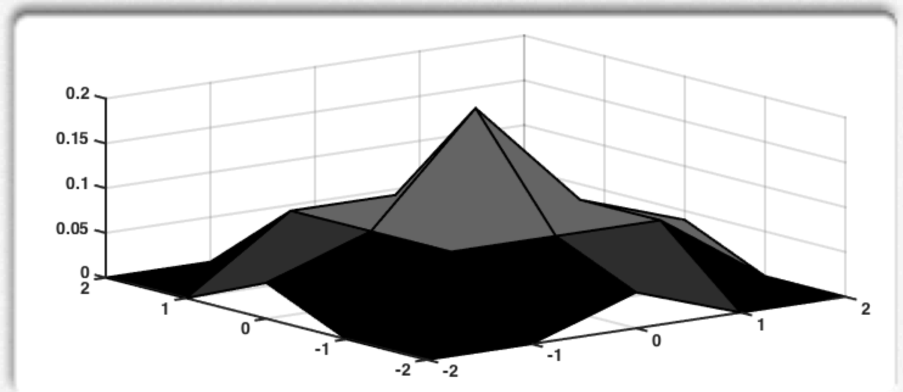
2. Filtre moyenneur

b) Pondéré

- Autres filtres moyenneur pondérés

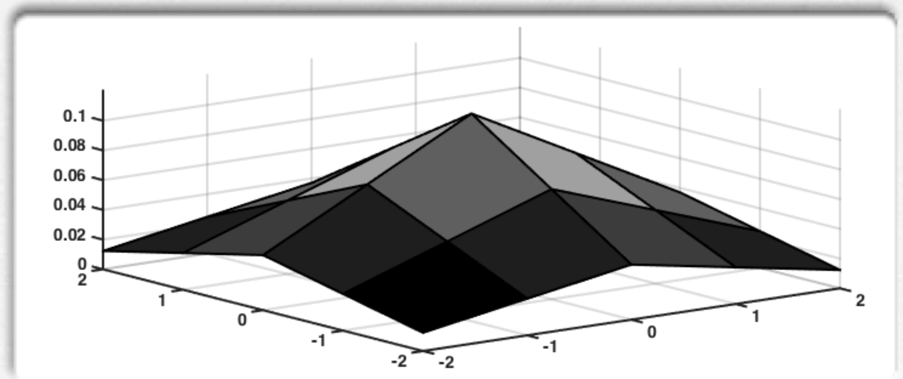
Filtre conique

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 2 & 2 & 0 \\ 1 & 2 & 5 & 2 & 1 \\ 0 & 2 & 2 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} / 25$$



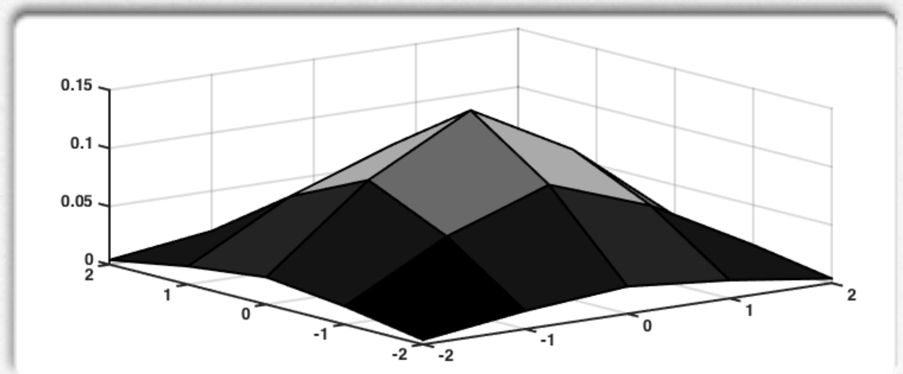
Filtre pyramidal

$$\begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 4 & 6 & 4 & 2 \\ 3 & 6 & 9 & 6 & 3 \\ 2 & 4 & 6 & 4 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix} / 81$$



Filtre binomial

$$\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} / 256$$



Source :

2. Réduction du bruit

3. Filtre médian

- *Objectif* : enlever le bruit impulsionnel (Sel et poivre) ou multiplicatif

Input Image	median	Filtered Img
7 8 4 5 5	5 5 7 8 8 9 9 9 9	8
5 9 4 3 8		
5 2 7 2 2		
6 1 9 2 4		
3 2 6 9 4		

Source : http://artemhlezin.com/2016/09/04/median.html

2. Réduction du bruit

3. Filtre médian

- *Objectif* : enlever le bruit impulsionnel (Sel et poivre) ou multiplicatif
  - ✓ *Exemples* : radar, rayon-X

Input Image					median	Filtered Img
7	8	4	5	5	5 5 7 8 8 9 9 9 9	8
5	9	4	3	8		
5	2	7	2	2		
6	1	9	2	4		
3	2	6	9	4		

Source : http://artemhlezin.com/2016/09/04/median.html

2. Réduction du bruit

3. Filtre médian

- *Objectif* : enlever le bruit impulsionnel (Sel et poivre) ou multiplicatif
  - ✓ *Exemples* : radar, rayon-X
- Le bruit impulsionnel peut être très contrasté mais très éparse
  - ✓ s'approche de valeurs aberrantes
  - ✓ si on utilisait un filtre passe-bas ou moyenneur, on ne fait qu'étaler le bruit sur ses voisins

Input Image	median	Filtered Img
7 8 4 5 5	5 5 7 8 8 9 9 9 9	8
5 9 4 3 8		
5 2 7 2 2		
6 1 9 2 4		
3 2 6 9 4		

Source : http://artemhlezin.com/2016/09/04/median.html

2. Réduction du bruit

3. Filtre médian

- *Objectif* : enlever le bruit impulsionnel (Sel et poivre) ou multiplicatif
  - ✓ *Exemples* : radar, rayon-X
- Le bruit impulsionnel peut être très contrasté mais très éparse
  - ✓ s'approche de valeurs aberrantes
  - ✓ si on utilisait un filtre passe-bas ou moyenneur, on ne fait qu'étaler le bruit sur ses voisins

Input Image	median	Filtered Img
7 8 4 5 5	5 5 7 8 8 9 9 9 9	8
5 9 4 3 8		
5 2 7 2 2		
6 1 9 2 4		
3 2 6 9 4		

Remarque :

- le filtre médian n'est pas linéaire
  - ✓ On ne peut pas utiliser la convolution, on ne peut pas passer par Fourier

Source : http://artemhlezin.com/2016/09/04/median.html

2. Réduction du bruit

3. Filtre médian

- *Objectif* : enlever le bruit impulsionnel (Sel et poivre) ou multiplicatif
  - ✓ *Exemples* : radar, rayon-X
- Le bruit impulsionnel peut être très contrasté mais très éparse
  - ✓ s'approche de valeurs aberrantes
  - ✓ si on utilisait un filtre passe-bas ou moyenneur, on ne fait qu'étaler le bruit sur ses voisins

Input Image	median	Filtered Img
7 8 4 5 5	5 5 7 8 8 9 9 9 9	8
5 9 4 3 8		
5 2 7 2 2		
6 1 9 2 4		
3 2 6 9 4		

Remarque :

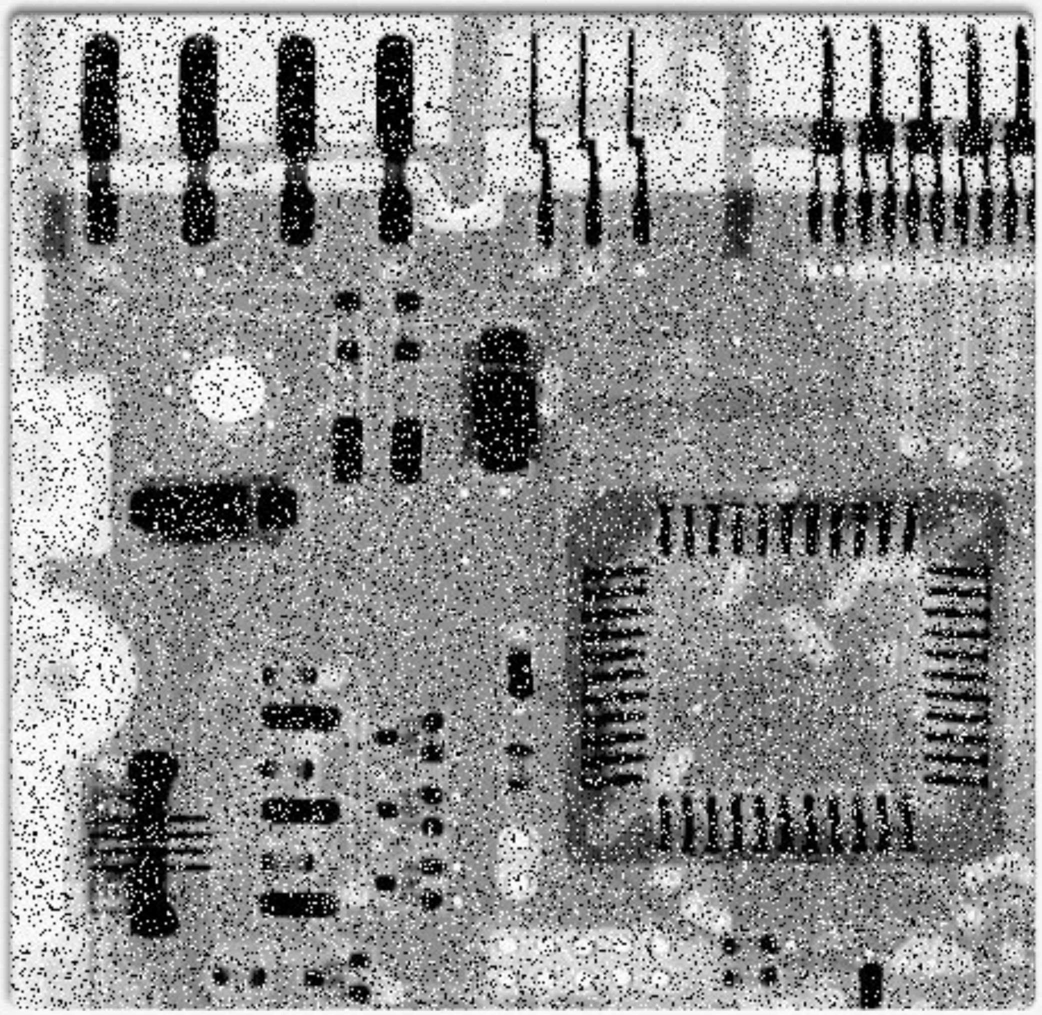
- le filtre médian n'est pas linéaire
  - ✓ On ne peut pas utiliser la convolution, on ne peut pas passer par Fourier

Source : http://artemhlezin.com/2016/09/04/median.html

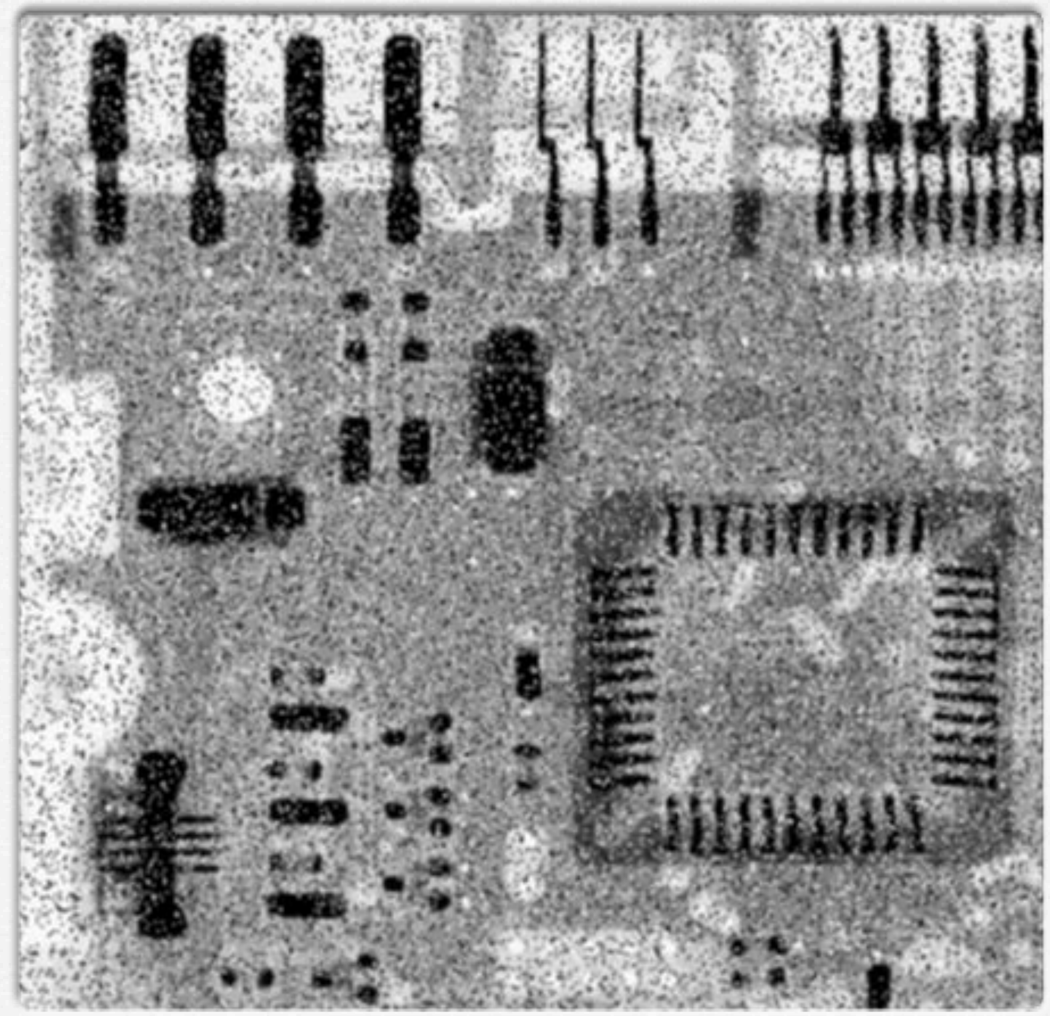
2. Réduction du bruit

3. Filtre médian

- Image rayon-X d'un circuit intégré dégradée par un bruit impulsionnel



Lissage Gaussien 3×3

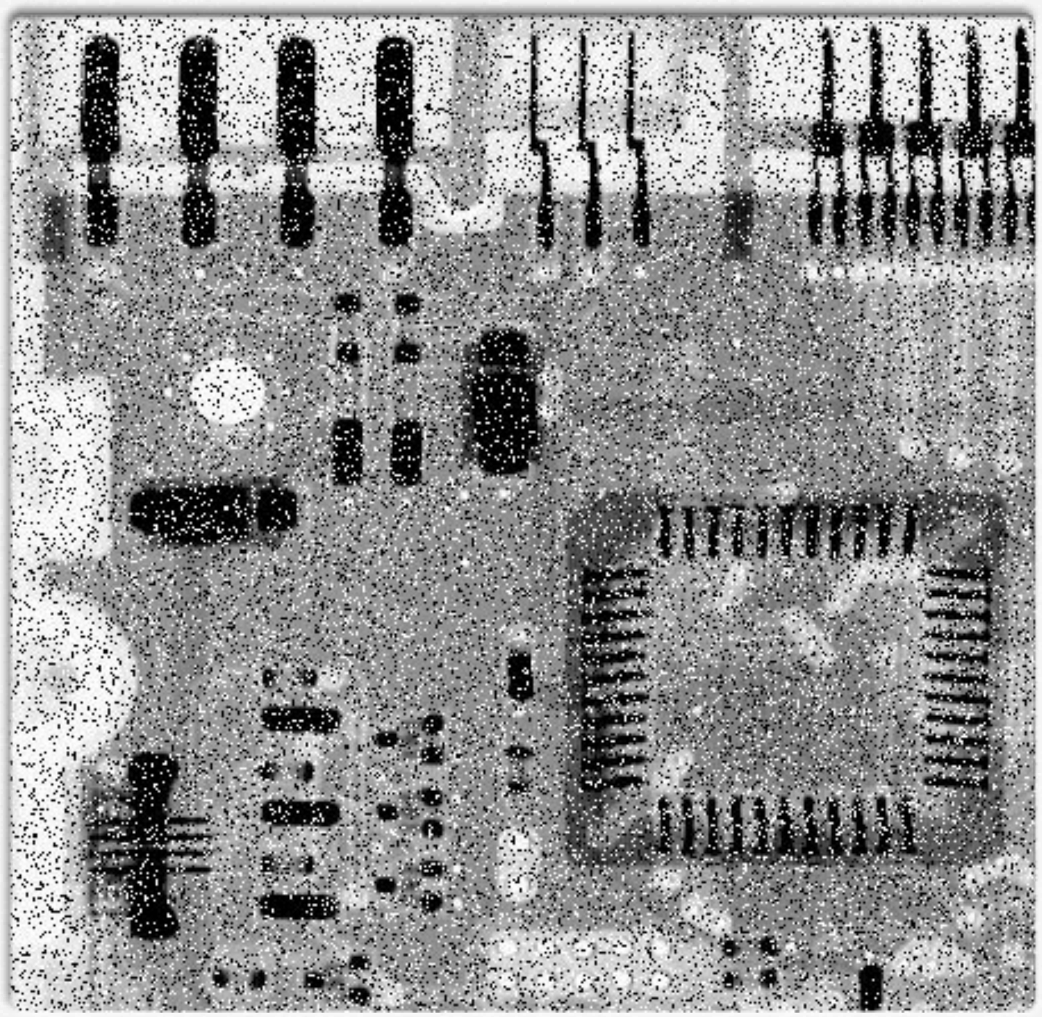


Source : DIP 2

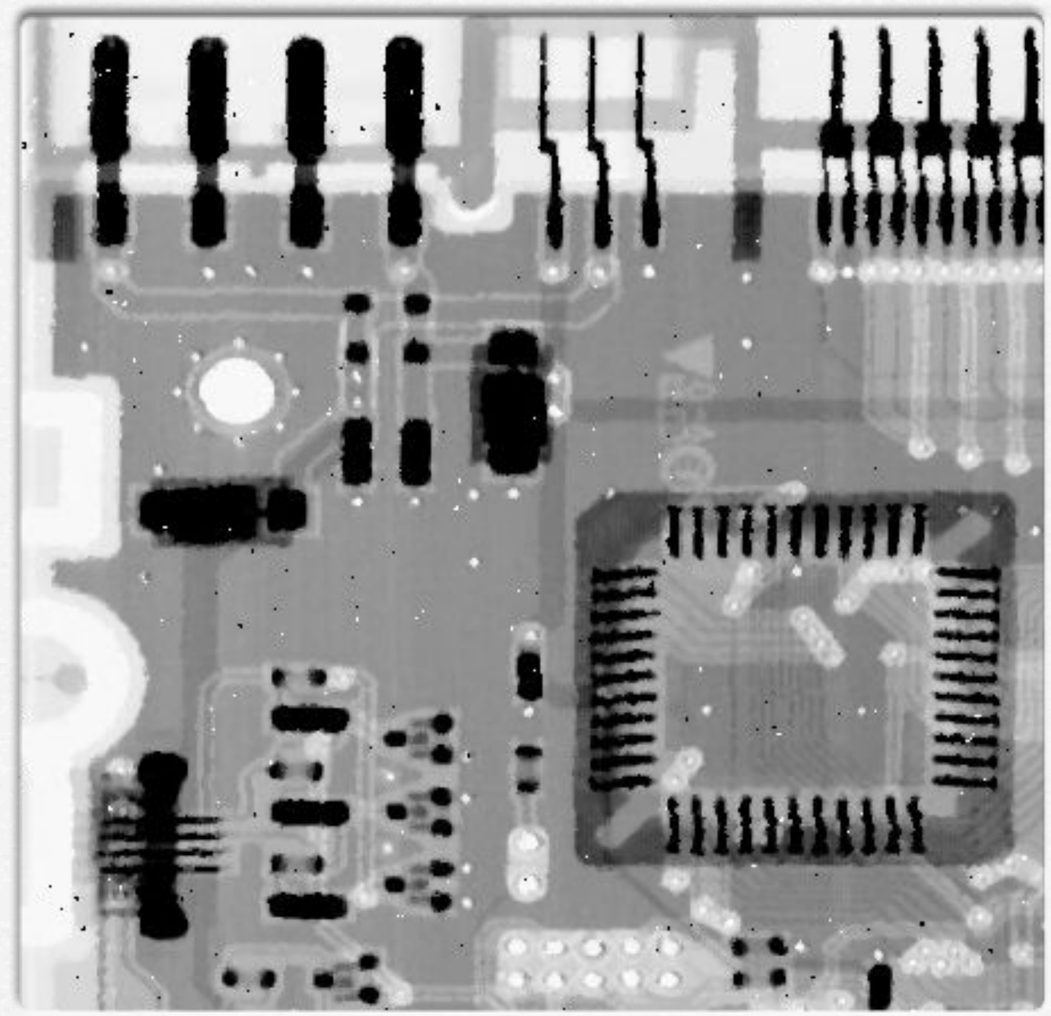
2. Réduction du bruit

3. Filtre médian

- Image rayon-X d'un circuit intégré dégradée par un bruit impulsionnel



Filtrage médian 3 × 3



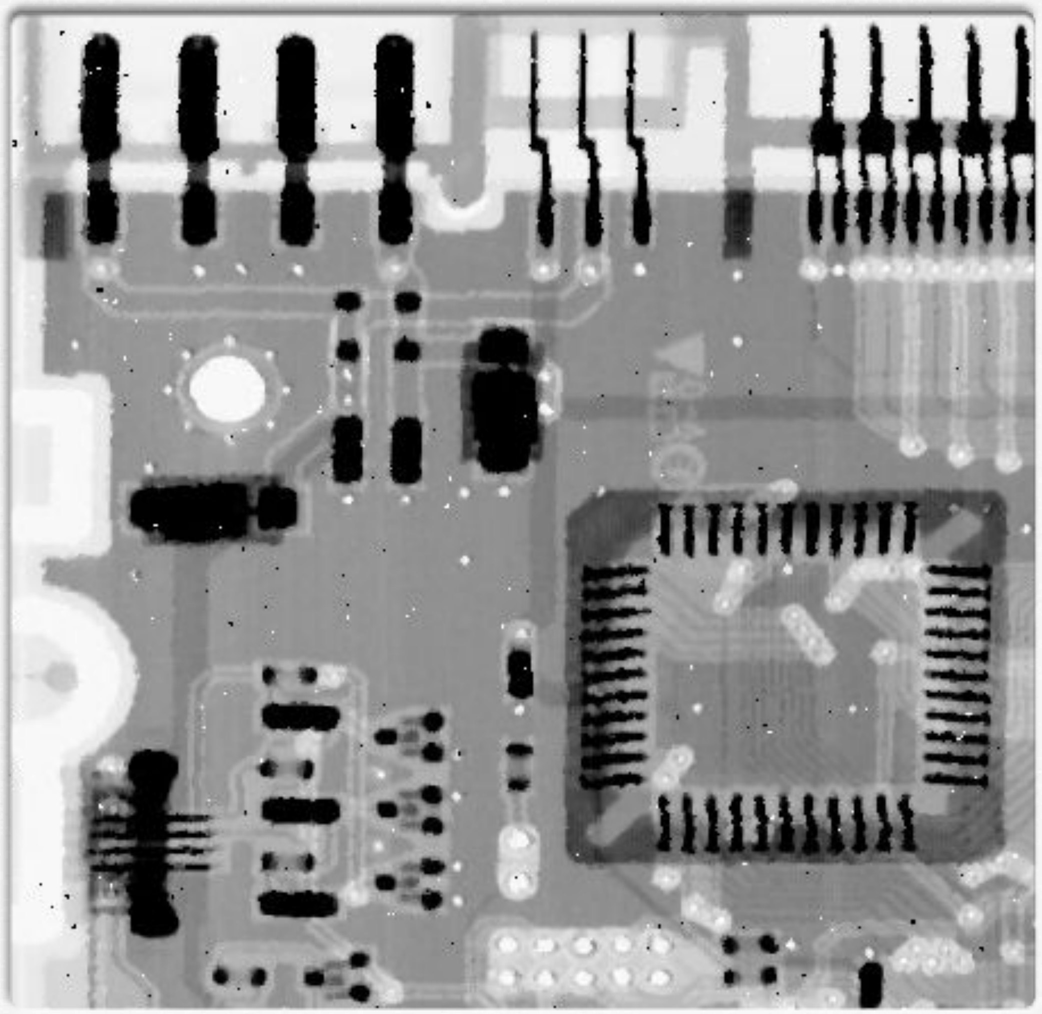
Source : DIP 2

2. RÉDUCTION DU BRUIT

3. Filtre médian

- Image rayon-X d'un circuit intégré dégradée par un bruit impulsionnel

Image filtrée une première fois



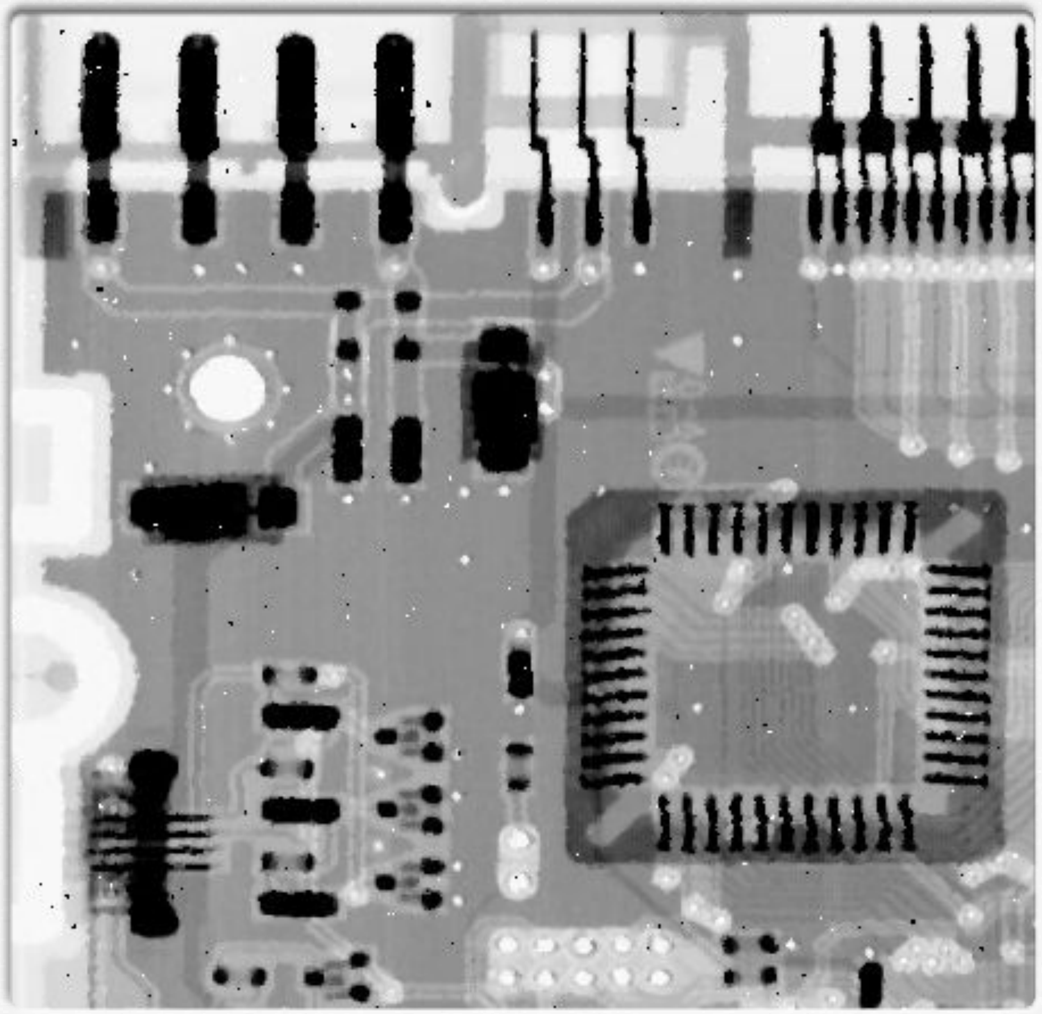
Source : DIP 2

2. Réduction du bruit

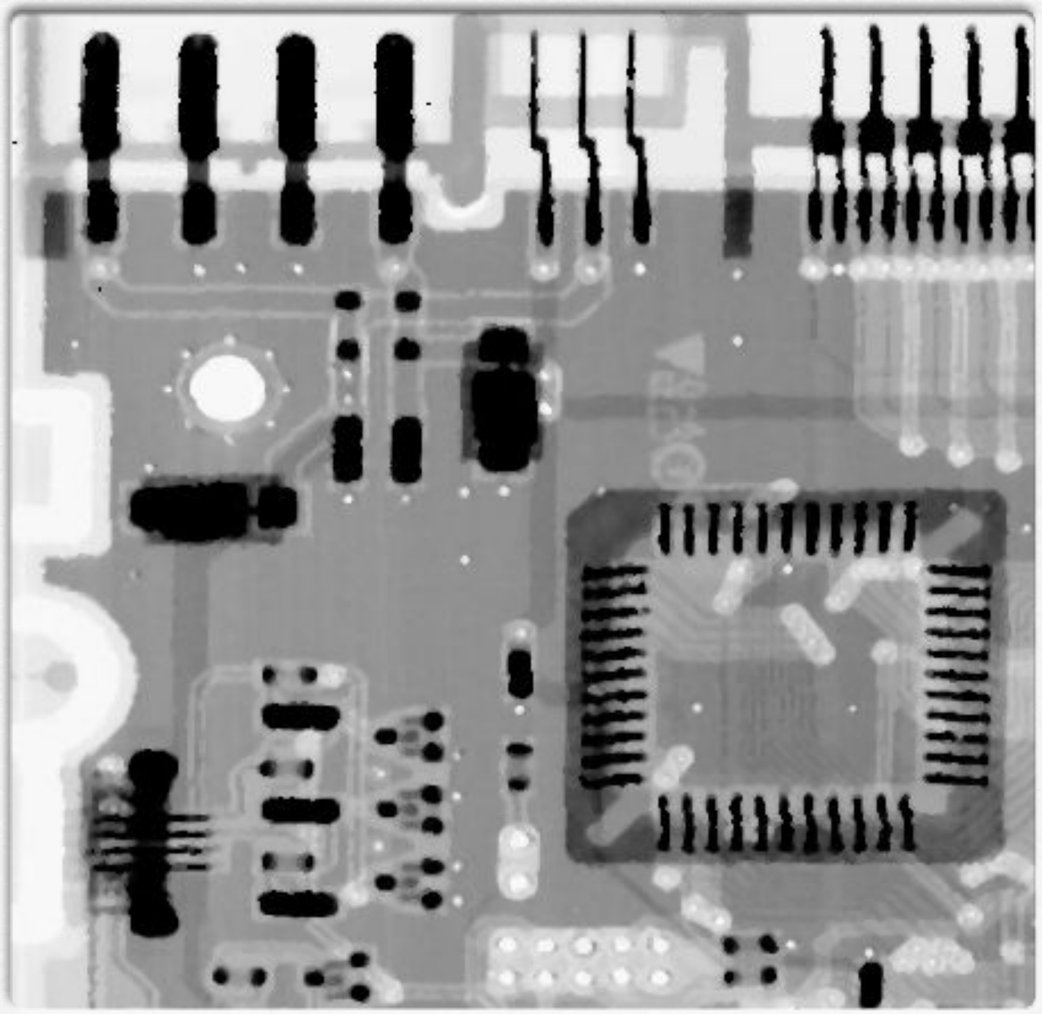
3. Filtre médian

- Image rayon-X d'un circuit intégré dégradée par un bruit impulsionnel

Image filtrée une première fois



2e Filtrage médian 3 × 3



Source : DIP 2

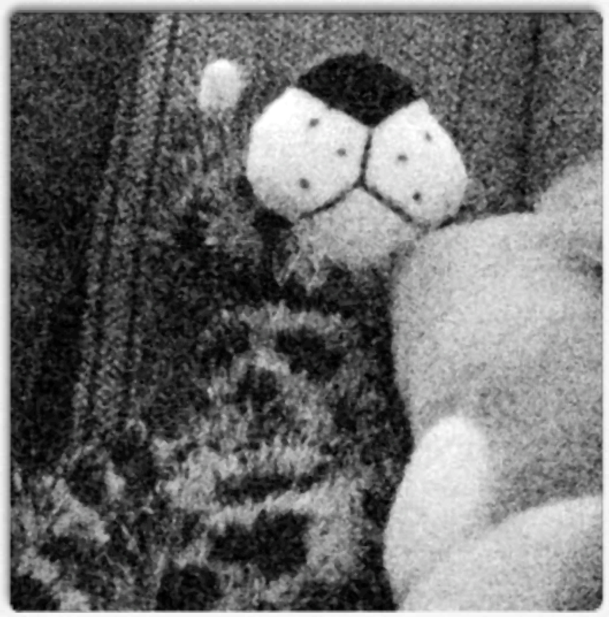
2. RÉDUCTION DU BRUIT

3. Filtre médian

- Le filtre médian est plus efficace plusieurs fois de suite que une seule fois très large

Originale

3 × 3



7 × 7

21 × 21

Source de l'image originale : Wikimedia commons

2. Réduction du bruit

3. Filtre médian

- Le filtrage médian est très efficace même avec un signal très dégradé

Source : Wikimedia commons

2. RÉDUCTION DU BRUIT

3. Filtre médian

- Le filtrage médian est très efficace même avec un signal très dégradé

Image bruitée

Filtre Gaussien $\sigma = 1$

Filtre Gaussien $\sigma = 2$

Filtre Médian 3×3

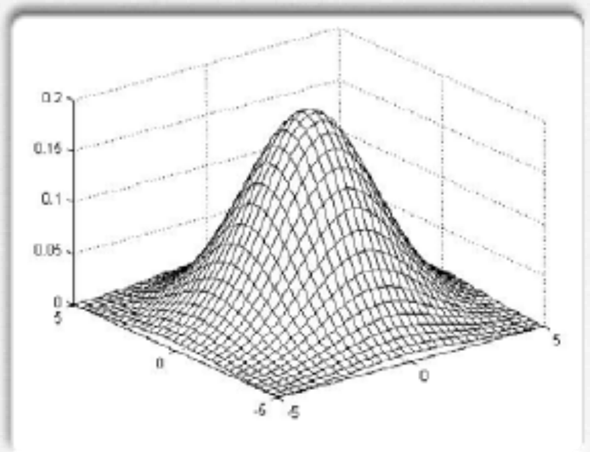
Source : PMJ

2. RÉDUCTION DU BRUIT

4. Filtre bilatéral

- Problème du filtre gaussien : Dans le cas du bruit additif, les filtres basés sur la moyenne d'un voisinage sont efficaces pour réduire le bruit mais ont pour effet de lisser aussi les contours
  - ➡ Le filtrage ne tient pas compte du contenu de l'image

$h_b[m,n]$



$\sigma = 5$

$*$

$=$

Source : PMJ

2. Réduction du bruit

4. Filtre bilatéral

- Filtre gaussien

$\mathbf{p} = [m, n]$:	coordonnée du pixel central
$\mathbf{q} = [m+k, n+l]$:	coordonnée d'un des pixels dans le voisinage de **p**

$$h_b[\mathbf{p}] = \frac{1}{W}\left(\sum_{\forall \mathbf{q} \text{ voisin de } \mathbf{p}} \left(g_{\sigma_s}[\mathbf{p}-\mathbf{q}]\, f[\mathbf{q}]\right)\right)$$

Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

2. Réduction du bruit

4. Filtre bilatéral

- Filtre gaussien

$\mathbf{p} = [m, n]$:	coordonnée du pixel central
$\mathbf{q} = [m+k, n+l]$:	coordonnée d'un des pixels dans le voisinage de **p**

$$h_b[\mathbf{p}] = \frac{1}{W}\left(\sum_{\forall \mathbf{q} \text{ voisin de } \mathbf{p}} \left(g_{\sigma_s}[\mathbf{p}-\mathbf{q}]\, f[\mathbf{q}]\right)\right)$$

- *Intensité* au point **q** du voisinage

Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

2. Réduction du bruit

4. Filtre bilatéral

- Filtre gaussien

$\mathbf{p} = [m, n]$:	coordonnée du pixel central
$\mathbf{q} = [m+k, n+l]$:	coordonnée d'un des pixels dans le voisinage de **p**

- *Distance* entre les pixels **p** et **q**

$$h_b[\mathbf{p}] = \frac{1}{W}\left(\sum_{\forall \mathbf{q} \text{ voisin de } \mathbf{p}} \left(g_{\sigma_s}[\mathbf{p}-\mathbf{q}]\, f[\mathbf{q}]\right)\right)$$

- *Intensité* au point **q** du voisinage

Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

2. RÉDUCTION DU BRUIT

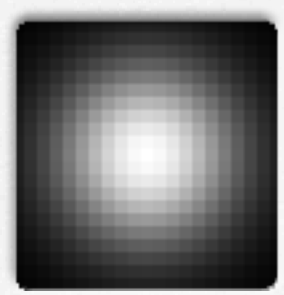
4. Filtre bilatéral

- Filtre gaussien

$\mathbf{p} = [m, n]$:	coordonnée du pixel central
$\mathbf{q} = [m+k, n+l]$:	coordonnée d'un des pixels dans le voisinage de **p**

$$h_b[\mathbf{p}] = \frac{1}{W}\left(\sum_{\forall \mathbf{q} \text{ voisin de } \mathbf{p}} \left(g_{\sigma_s}[\mathbf{p}-\mathbf{q}]\, f[\mathbf{q}]\right)\right)$$

- *Distance* entre les pixels **p** et **q**
- *Poids* spatial
  - ✓ gaussienne de paramètre σ_s
- *Intensité* au point **q** du voisinage



Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

2. Réduction du bruit

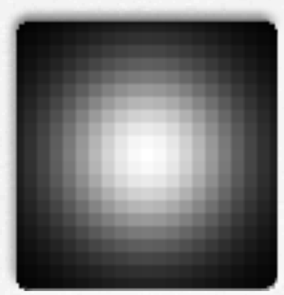
4. Filtre bilatéral

- Filtre gaussien

$\mathbf{p} = [m, n]$:	coordonnée du pixel central
$\mathbf{q} = [m+k, n+l]$:	coordonnée d'un des pixels dans le voisinage de **p**

$$h_b[\mathbf{p}] = \frac{1}{W}\left(\sum_{\forall \mathbf{q} \text{ voisin de } \mathbf{p}} \left(g_{\sigma_s}[\mathbf{p}-\mathbf{q}]\, f[\mathbf{q}]\right)\right)$$

- *Distance* entre les pixels **p** et **q**
- *Facteur de normalisation* constant peu importe **p** : peut être inclus dans le filtre
  - ✓ Somme de tous les poids pour s'assurer que la moyenne n'est pas modifiée
- *Poids* spatial
  - ✓ gaussienne de paramètre σ_s



- *Intensité* au point **q** du voisinage

Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

2. Réduction du bruit

4. Filtre bilatéral

- Filtre gaussien : pareil partout parce que linéaire

original

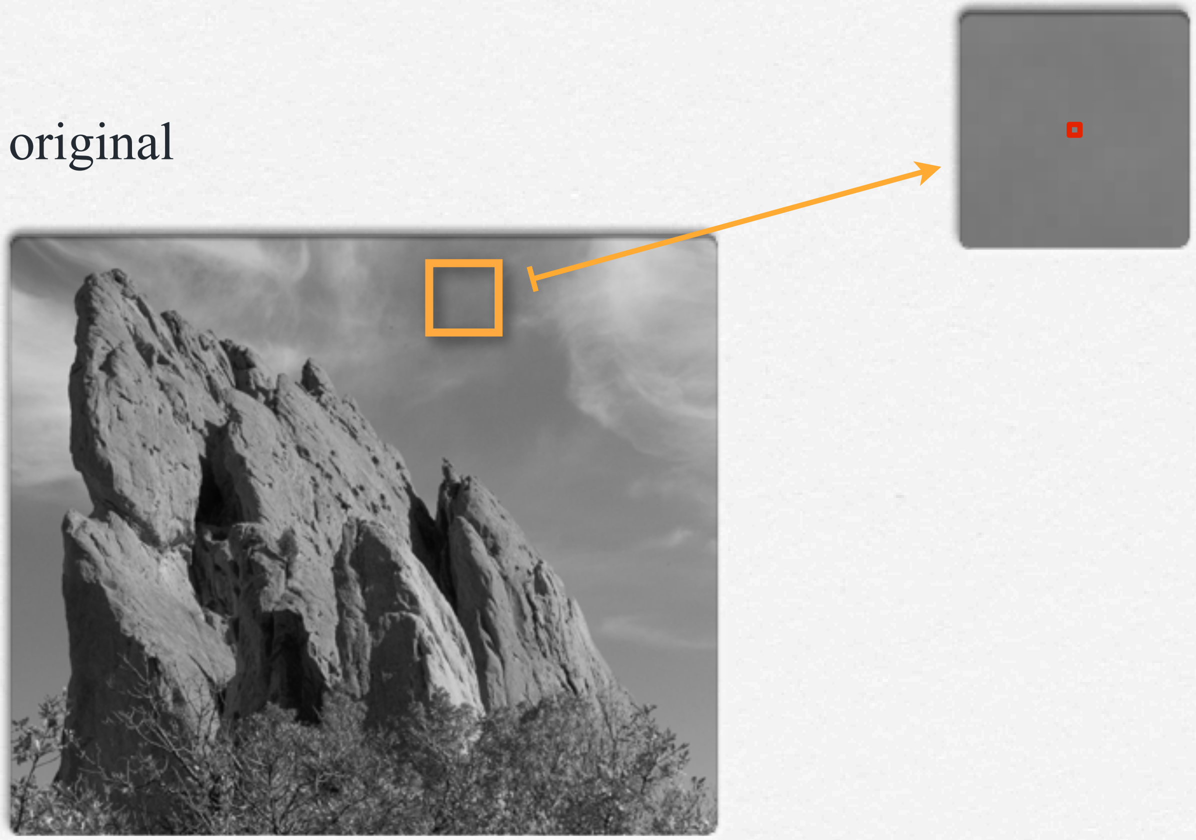
Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

2. Réduction du bruit

4. Filtre bilatéral

- Filtre gaussien : pareil partout parce que linéaire

original



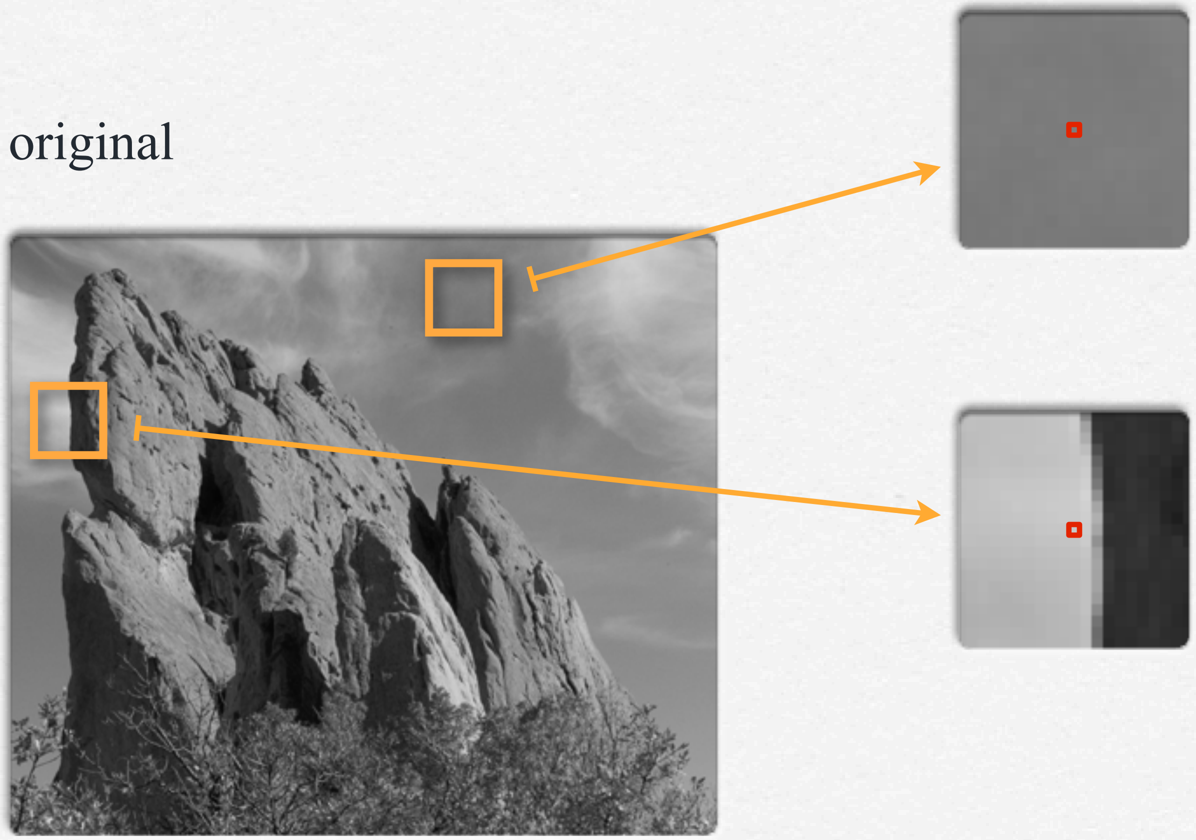
Source : S. Paris, A Gentle Introduction
to Bilateral Filtering and its Applications, SIGGRAPH'07

2. Réduction du bruit

4. Filtre bilatéral

- Filtre gaussien : pareil partout parce que linéaire

original

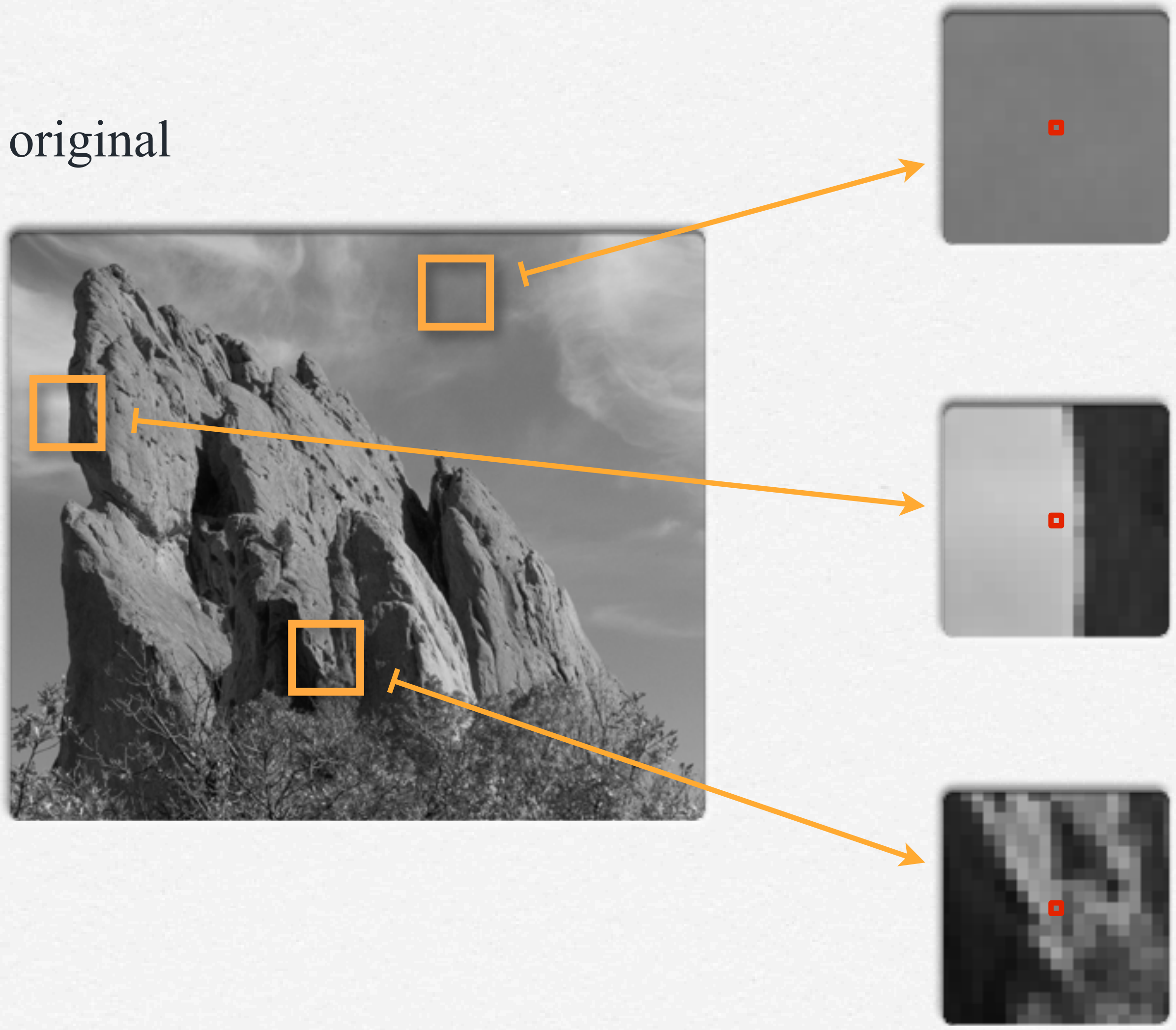


2. RÉDUCTION DU BRUIT

4. Filtre bilatéral

- Filtre gaussien : pareil partout parce que linéaire

original

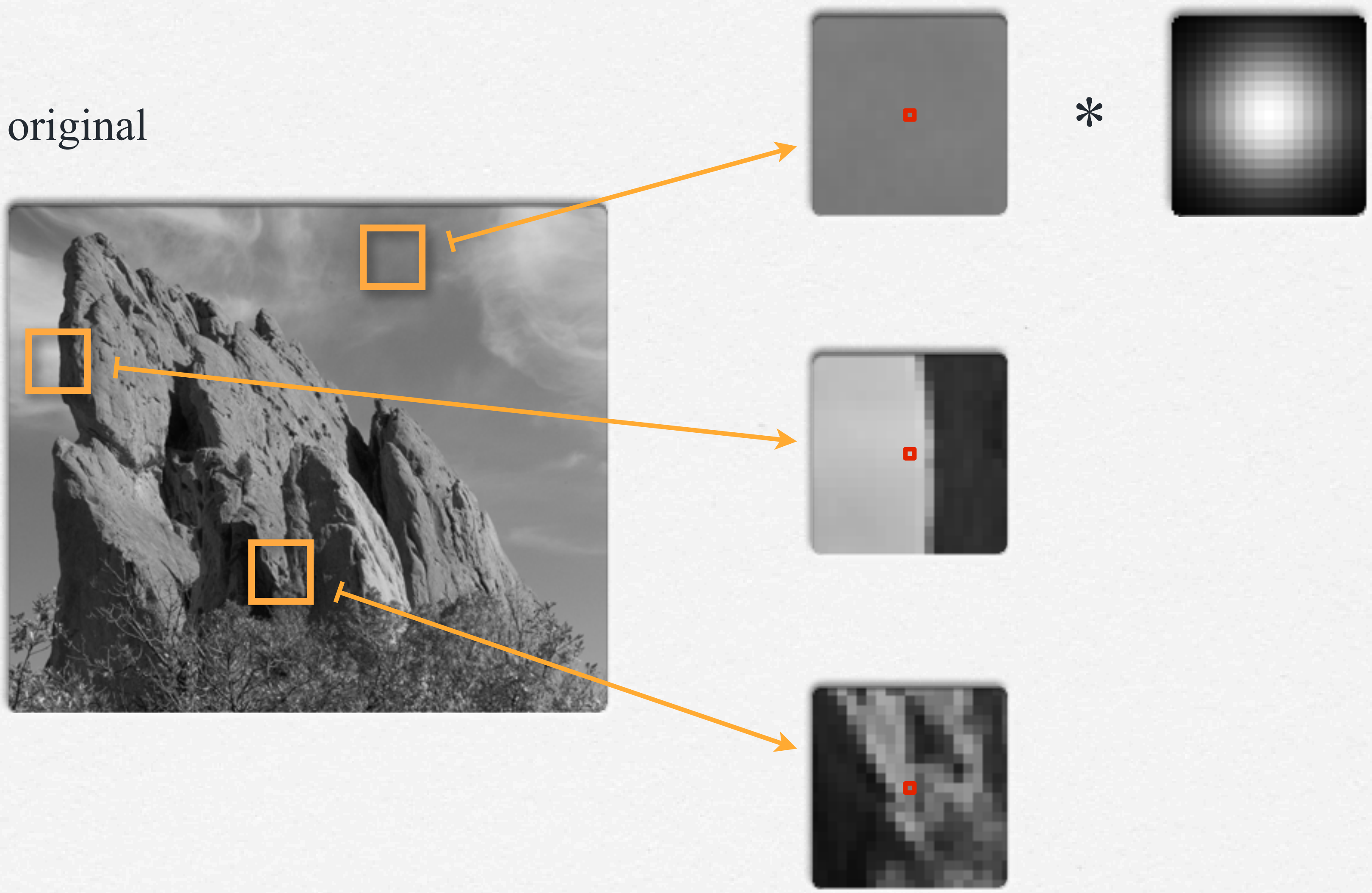


Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

2. Réduction du bruit

4. Filtre bilatéral

- Filtre gaussien : pareil partout parce que linéaire

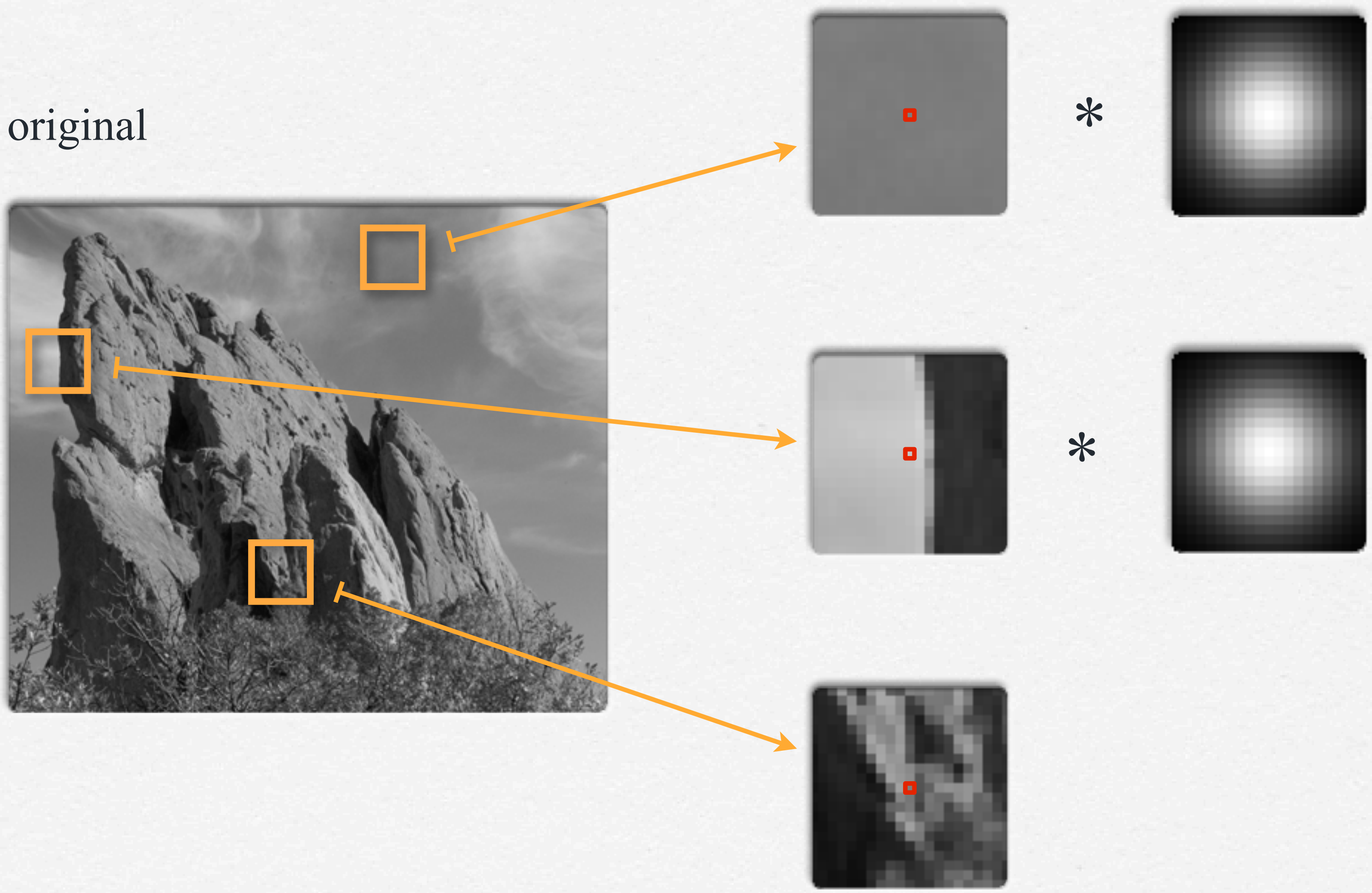


Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

2. RÉDUCTION DU BRUIT

4. Filtre bilatéral

- Filtre gaussien : pareil partout parce que linéaire



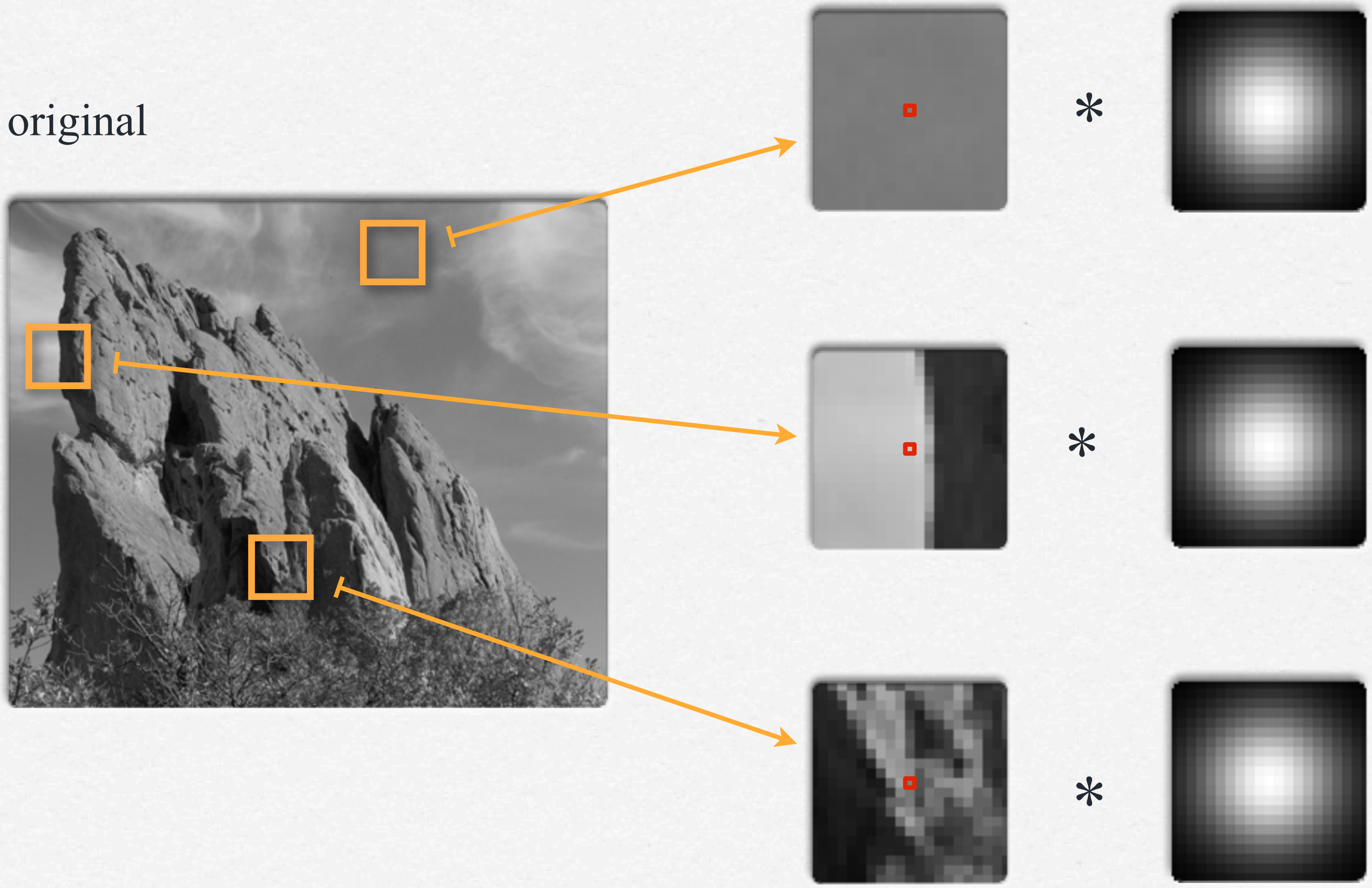
Source : S. Paris, A Gentle Introduction
to Bilateral Filtering and its Applications, SIGGRAPH'07

2. Réduction du bruit

4. Filtre bilatéral

- Filtre gaussien : pareil partout parce que linéaire

original

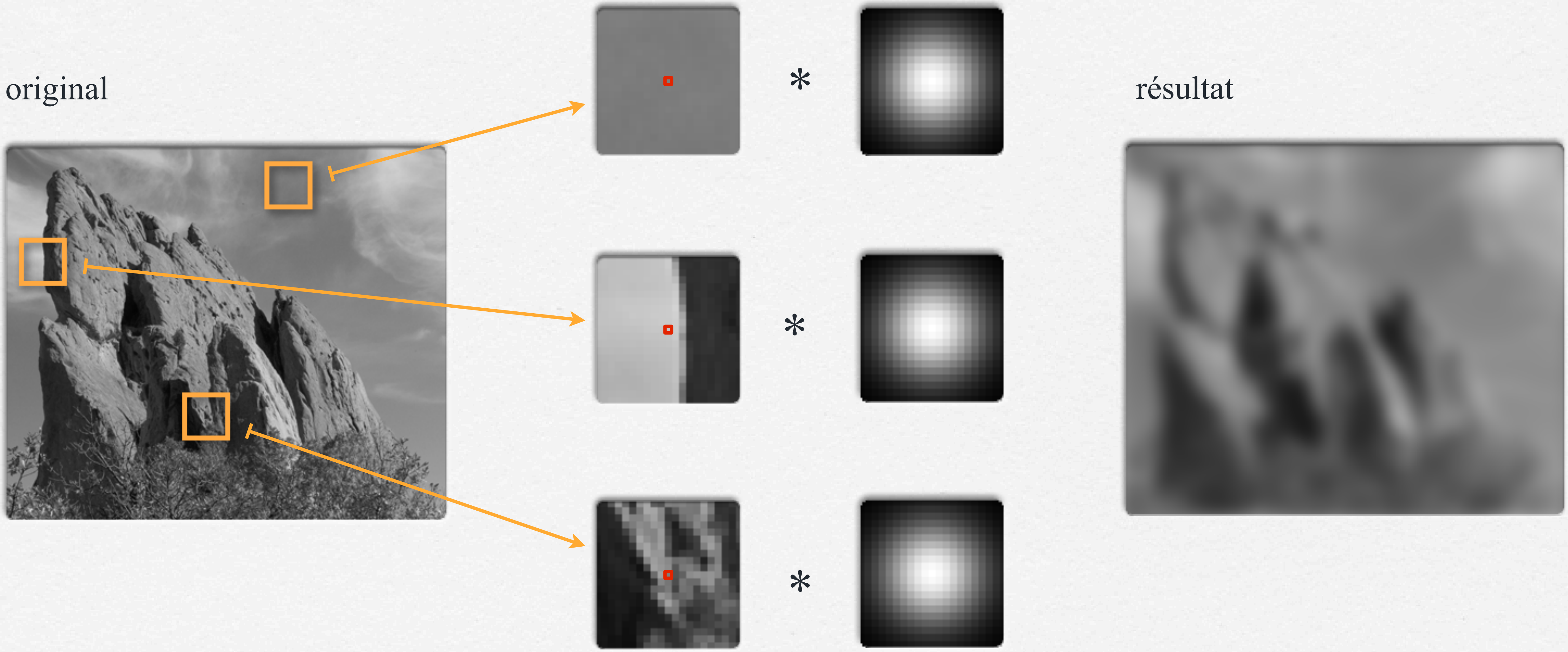


Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH’07

2. Réduction du bruit

4. Filtre bilatéral

- Filtre gaussien : pareil partout parce que linéaire

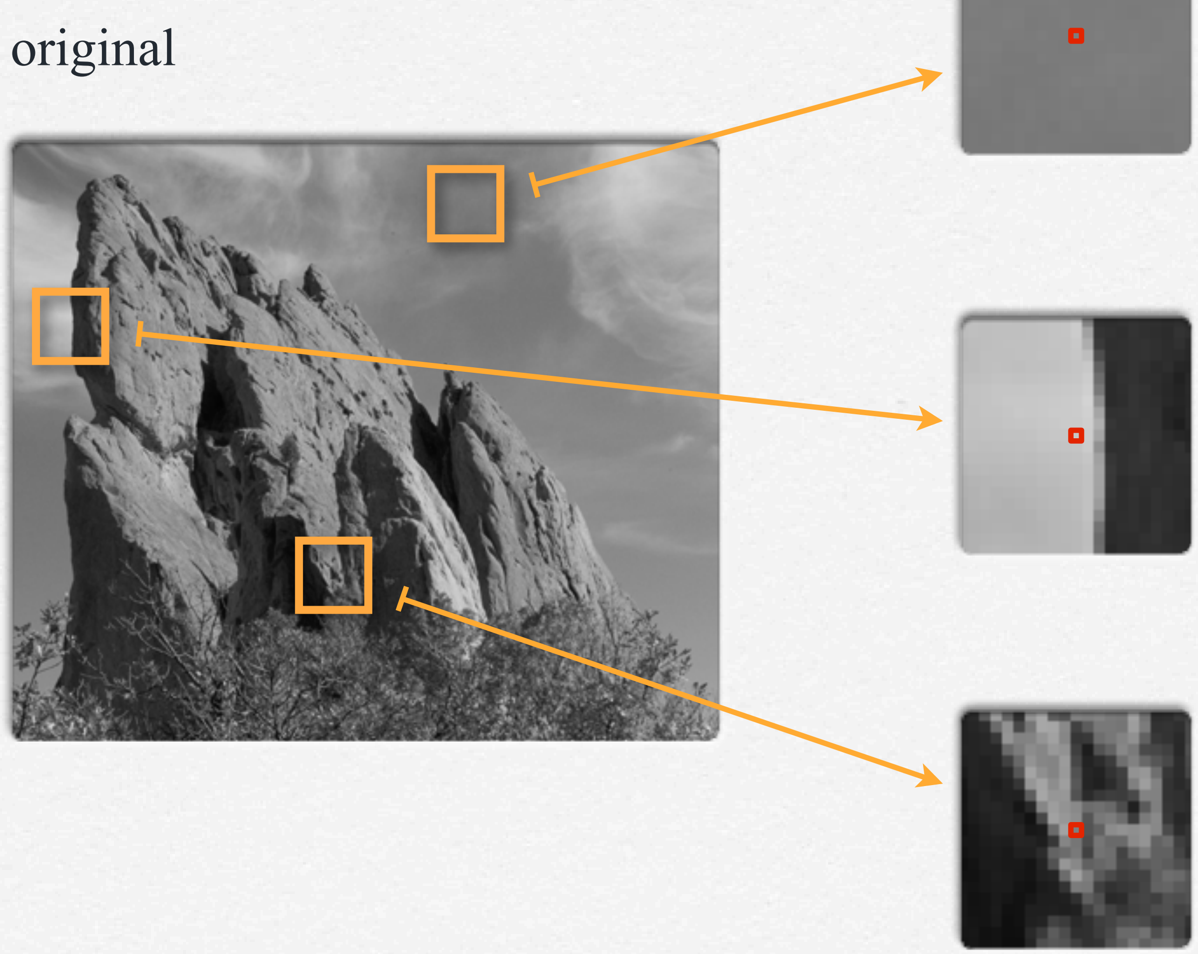


Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

2. RÉDUCTION DU BRUIT

4. Filtre bilatéral

- Filtre bilatéral : la forme du filtre s'adapte au contenu

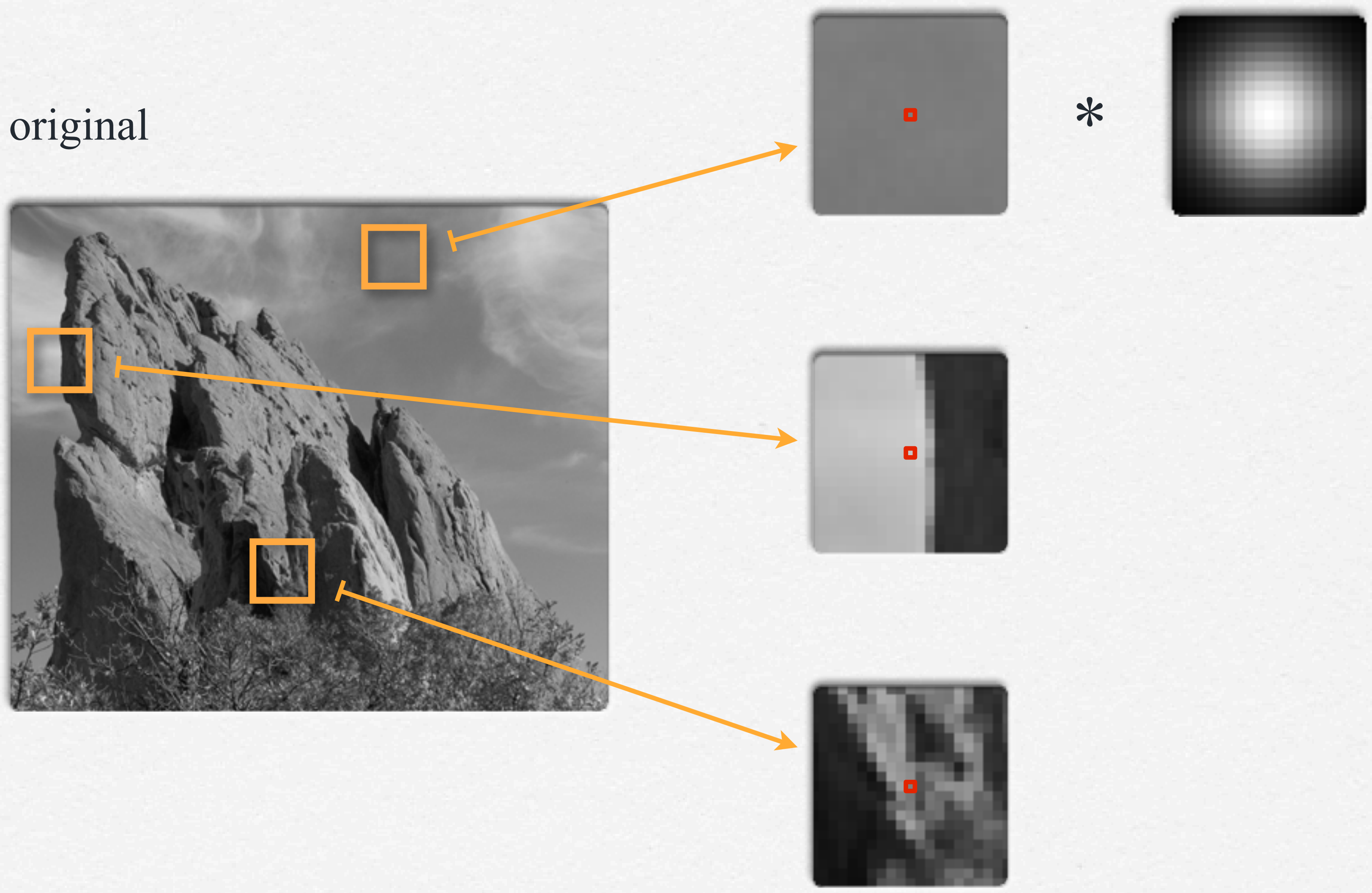


Source : S. Paris, A Gentle Introduction
to Bilateral Filtering and its Applications, SIGGRAPH'07

2. Réduction du bruit

4. Filtre bilatéral

- Filtre bilatéral : la forme du filtre s'adapte au contenu

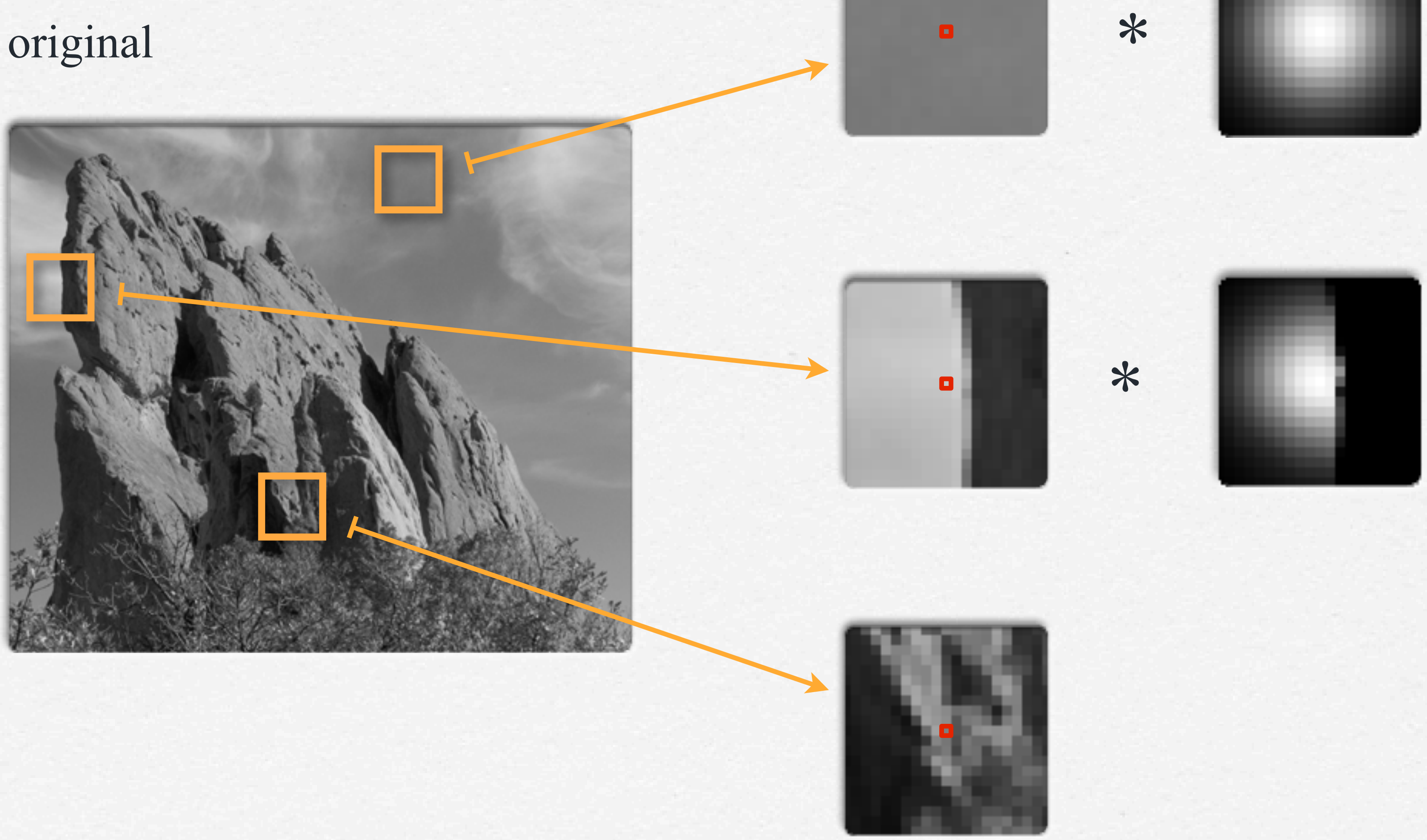


Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

2. RÉDUCTION DU BRUIT

4. Filtre bilatéral

- Filtre bilatéral : la forme du filtre s'adapte au contenu

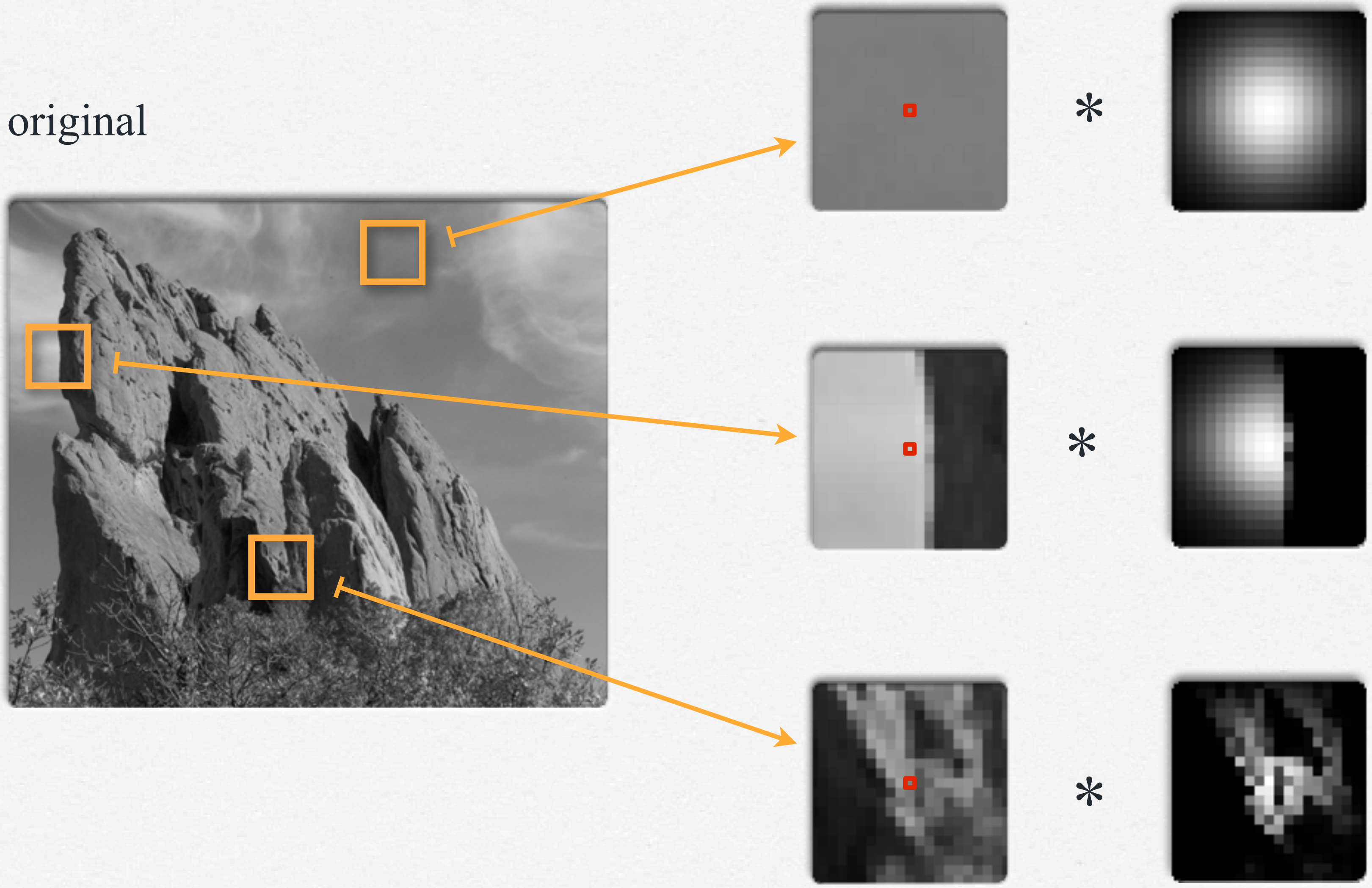


Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

2. Réduction du bruit

4. Filtre bilatéral

- Filtre bilatéral : la forme du filtre s'adapte au contenu

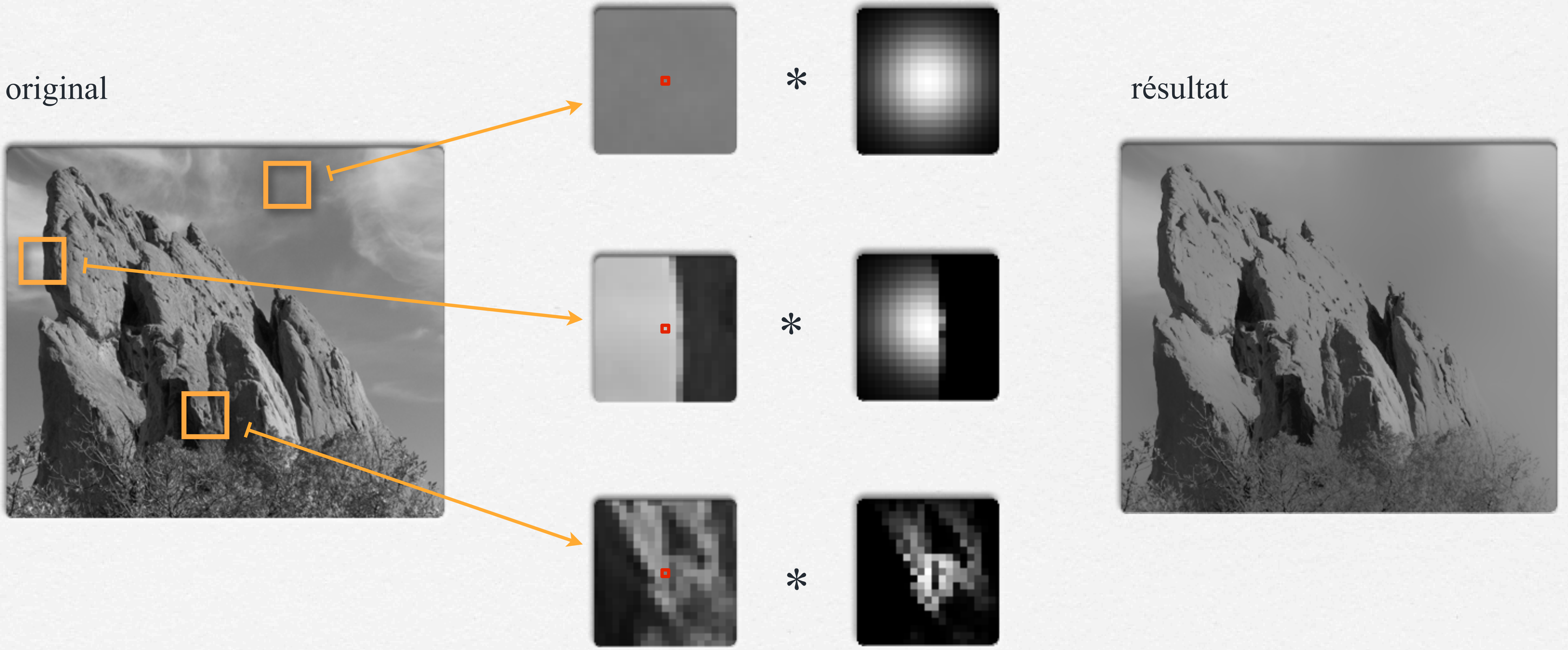


Source : S. Paris, A Gentle Introduction
to Bilateral Filtering and its Applications, SIGGRAPH'07

2. RÉDUCTION DU BRUIT

4. Filtre bilatéral

- Filtre bilatéral : la forme du filtre s'adapte au contenu



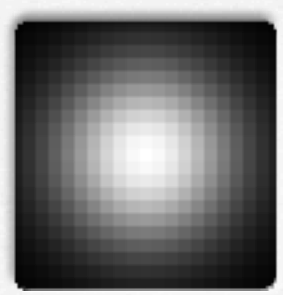
Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

$\mathbf{p} = [m, n]$:	coordonnée du pixel central
$\mathbf{q} = [m+k, n+l]$:	coordonnée d'un des pixels dans le voisinage de **p**

- Distance entre les *positions* des pixels **p** et **q**

$$h_b[\mathbf{p}] = \frac{1}{W_{\mathbf{p}}}\left(\sum_{\forall \mathbf{q}\ \text{voisin de } \mathbf{p}} \left(g_{\sigma_s}[\mathbf{p}-\mathbf{q}]\, g_{\sigma_I}\big[f[\mathbf{p}] - f[\mathbf{q}]\big]\, f[\mathbf{q}]\right)\right)$$

- *Poids spatial*
  - ✓ gaussienne de paramètre σ_s



- *Intensité* au point **q** du voisinage

Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

2. RÉDUCTION DU BRUIT

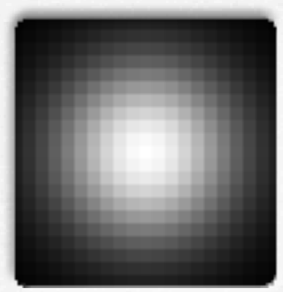
4. Filtre bilatéral

$\mathbf{p} = [m, n]$:	coordonnée du pixel central
$\mathbf{q} = [m+k, n+l]$:	coordonnée d'un des pixels dans le voisinage de **p**

- Distance entre les *positions* des pixels **p** et **q**
- Distance entre les *intensités* des pixels **p** et **q**

$$h_b[\mathbf{p}] = \frac{1}{W_{\mathbf{p}}} \left(\sum_{\forall \mathbf{q} \text{ voisin de } \mathbf{p}} \left(g_{\sigma_s}[\mathbf{p}-\mathbf{q}]\, g_{\sigma_I}\big[f[\mathbf{p}] - f[\mathbf{q}]\big]\, f[\mathbf{q}] \right) \right)$$

- *Poids spatial*
  - ✓ gaussienne de paramètre σ_s



- *Intensité* au point **q** du voisinage

Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

2. Réduction du bruit

4. Filtre bilatéral

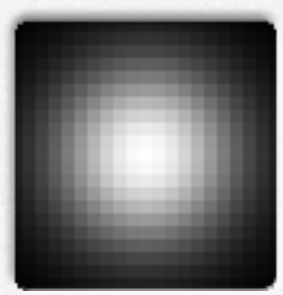
$\mathbf{p} = [m, n]$:	coordonnée du pixel central
$\mathbf{q} = [m+k, n+l]$:	coordonnée d'un des pixels dans le voisinage de **p**

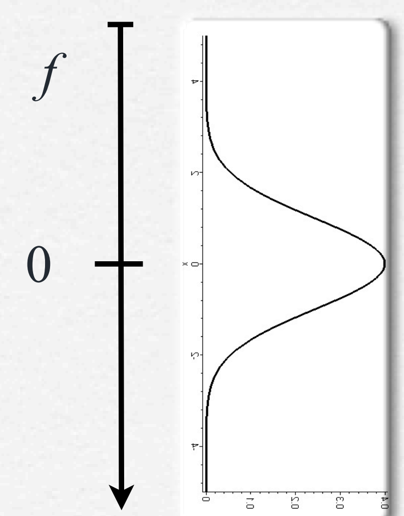
- Distance entre les *positions* des pixels **p** et **q**
- Distance entre les *intensités* des pixels **p** et **q**

$$h_b[\mathbf{p}] = \frac{1}{W_{\mathbf{p}}}\left(\sum_{\forall \mathbf{q} \text{ voisin de } \mathbf{p}} \left(g_{\sigma_s}[\mathbf{p}-\mathbf{q}]\; g_{\sigma_I}\big[f[\mathbf{p}]-f[\mathbf{q}]\big]\; f[\mathbf{q}]\right)\right)$$

- *Poids d'intensité*
  - ✓ dépend de la différence d'intensité entre les 2 pixels **p** et **q**

- *Poids spatial*
  - ✓ gaussienne de paramètre σ_s





- *Intensité* au point **q** du voisinage

Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

2. RÉDUCTION DU BRUIT

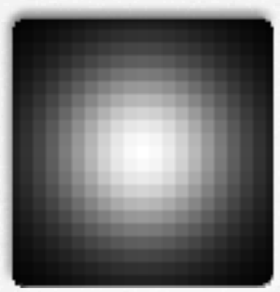
4. Filtre bilatéral

$\mathbf{p} = [m, n]$:	coordonnée du pixel central
$\mathbf{q} = [m+k, n+l]$:	coordonnée d'un des pixels dans le voisinage de **p**

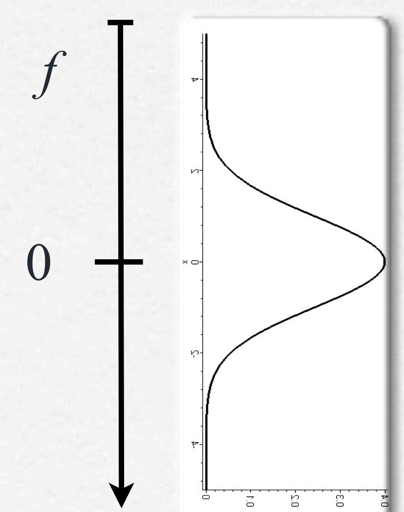
- Distance entre les *positions* des pixels **p** et **q**
- Distance entre les *intensités* des pixels **p** et **q**

$$h_b[\mathbf{p}] = \frac{1}{W_{\mathbf{p}}} \left(\sum_{\forall \mathbf{q} \text{ voisin de } \mathbf{p}} \left(g_{\sigma_s}[\mathbf{p}-\mathbf{q}]\, g_{\sigma_I}\left[f[\mathbf{p}] - f[\mathbf{q}] \right] f[\mathbf{q}] \right) \right)$$

- *Facteur de normalisation* qui dépend de **p**
  - ✓ **ne peut pas être inclus dans les poids**
- *Poids spatial*
  - ✓ gaussienne de paramètre σ_s



- *Poids d'intensité*
  - ✓ dépend de la différence d'intensité entre les 2 pixels **p** et **q**



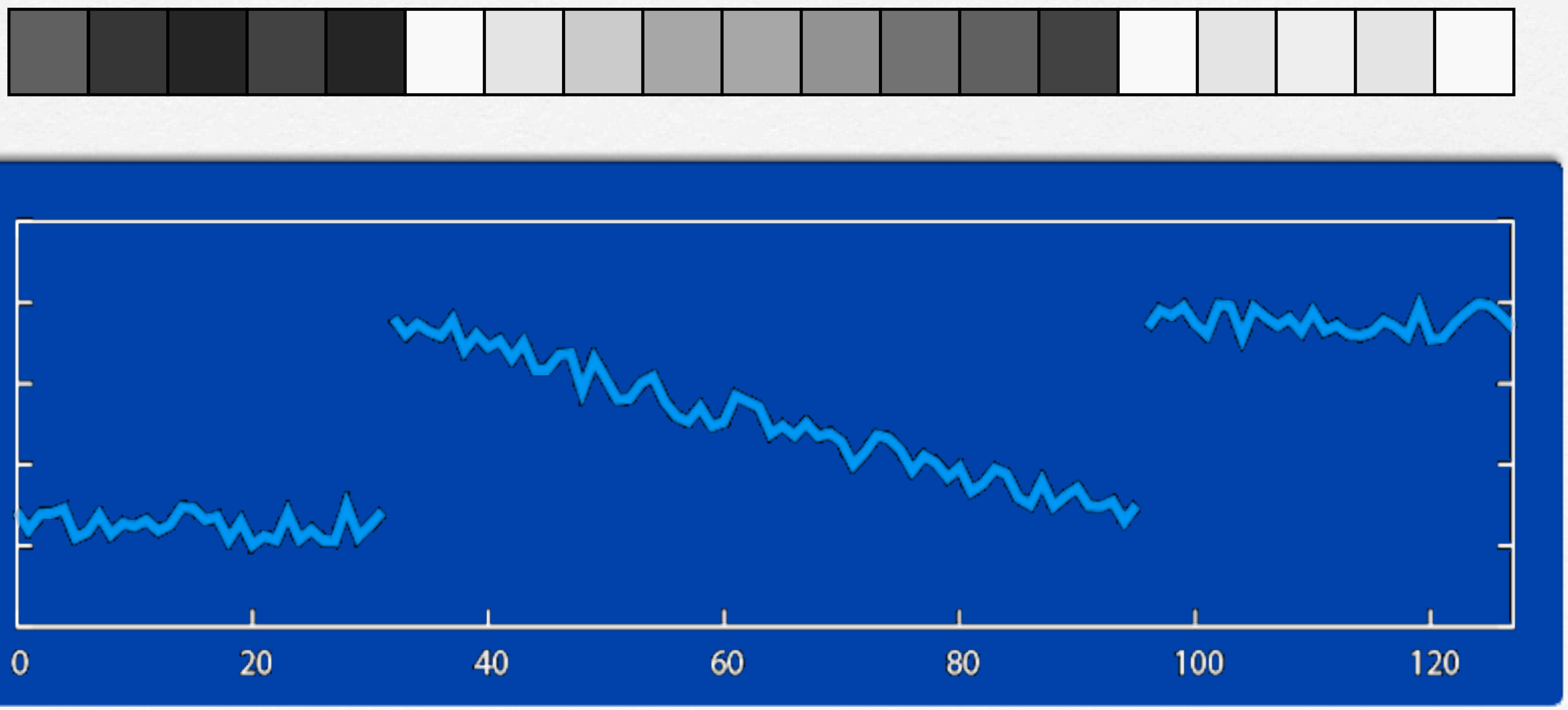
- *Intensité* au point **q** du voisinage

Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

2. Réduction du bruit

4. Filtre bilatéral

- 1D

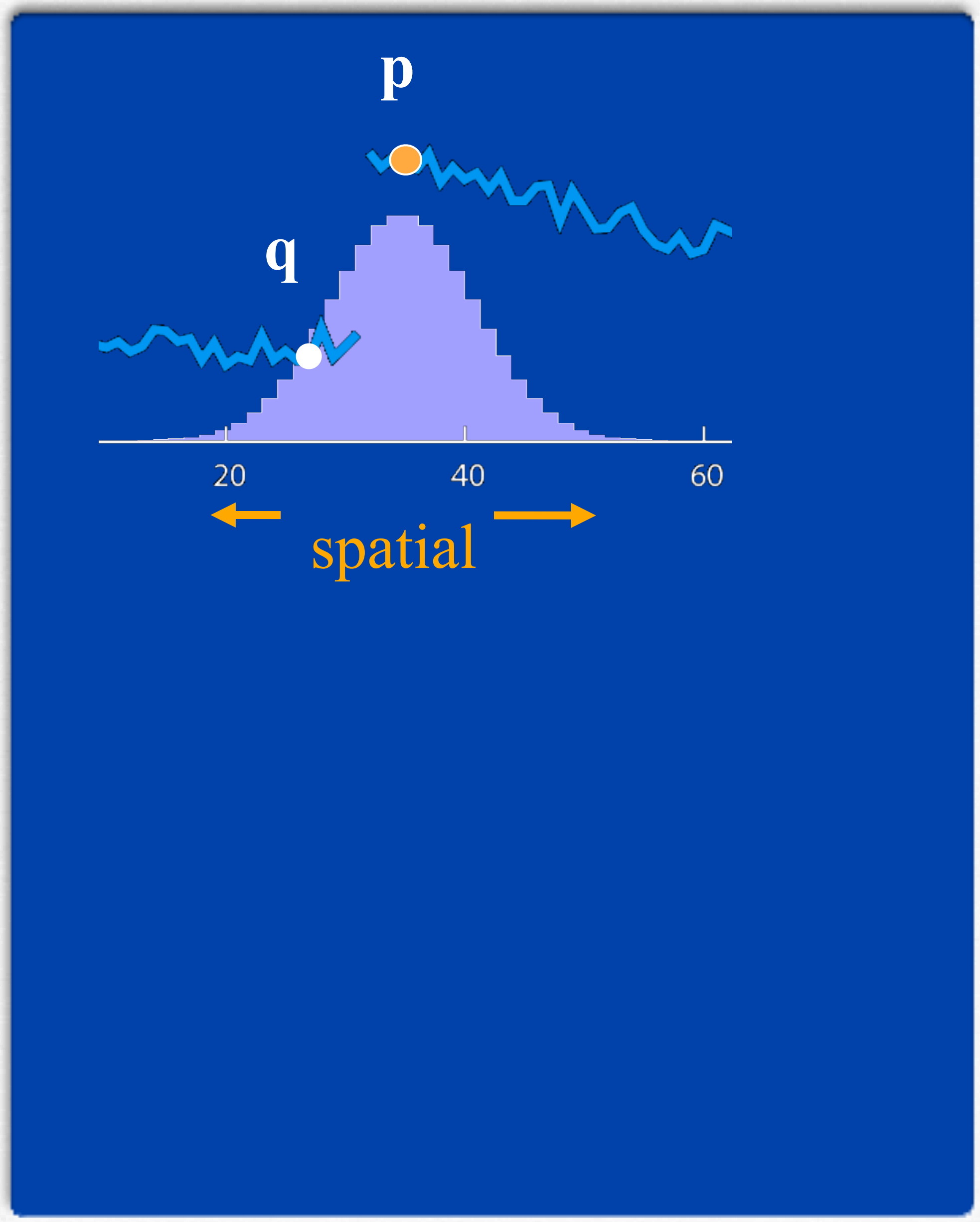


Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

2. Réduction du bruit

4. Filtre bilatéral

- 1D



Gaussien

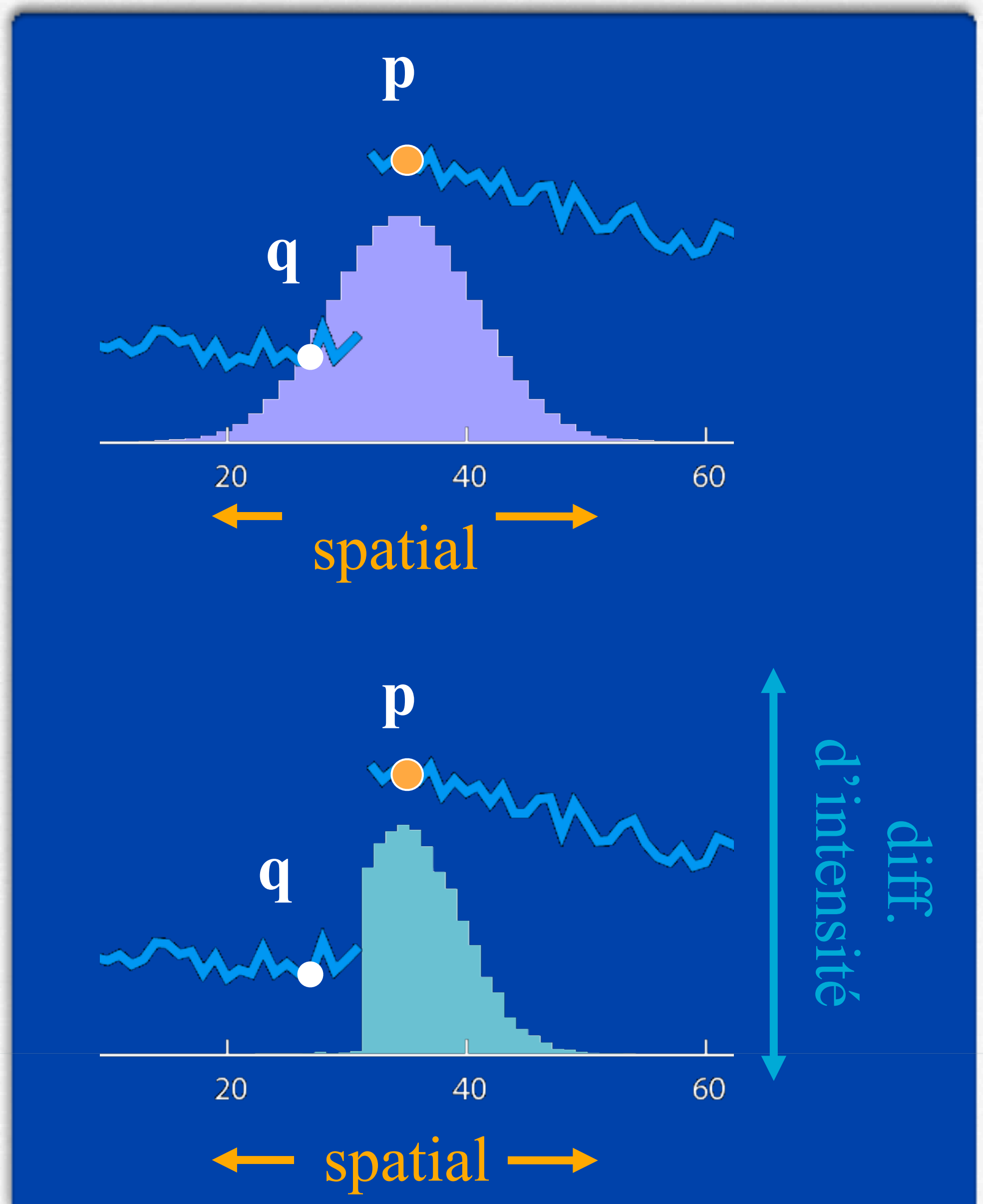
$$h_b[\mathbf{p}] = \frac{1}{W}\left(\sum_{\mathbf{q} \text{ voisin de } \mathbf{p}} \left(\underbrace{g_{\sigma_s}[\mathbf{p}-\mathbf{q}]}_{\text{poids spatial}} \right)\right)$$

Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

2. Réduction du bruit

4. Filtre bilatéral

- 1D



Gaussien

$$h_b[\mathbf{p}] = \frac{1}{W}\left(\sum_{\mathbf{q}\text{ voisin de }\mathbf{p}} \left(\underbrace{g_{\sigma_s}[\mathbf{p}-\mathbf{q}]}_{\text{poids spatial}} \right)\right)$$

Bilatéral

$$h_b[\mathbf{p}] = \frac{1}{W_{\mathbf{p}}}\left(\sum_{\mathbf{q}\text{ voisin de }\mathbf{p}} \left(\underbrace{g_{\sigma_s}[\mathbf{p}-\mathbf{q}]}_{\text{poids spatial}} \underbrace{g_{\sigma_I}\left[f[\mathbf{p}] - f[\mathbf{q}]\right]}_{\text{poids d'intensité}} \right)\right)$$

Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

2. Réduction du bruit

4. Filtre bilatéral

- 2D

$$h_b[\mathbf{p}] = \frac{1}{W_{\mathbf{p}}}\left(\sum_{\mathbf{q}\text{ voisin de }\mathbf{p}} \left(g_{\sigma_s}[\mathbf{p}-\mathbf{q}]\, g_{\sigma_I}\left[f[\mathbf{p}]-f[\mathbf{q}]\right] f[\mathbf{q}]\right)\right)$$

Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

2. RÉDUCTION DU BRUIT

4. Filtre bilatéral

- 2D

$$h_b[\mathbf{p}] = \frac{1}{W_{\mathbf{p}}}\left(\sum_{\mathbf{q}\text{ voisin de }\mathbf{p}}\left(g_{\sigma_s}[\mathbf{p}-\mathbf{q}]\, g_{\sigma_I}\left[f[\mathbf{p}]-f[\mathbf{q}]\right] f[\mathbf{q}]\right)\right)$$

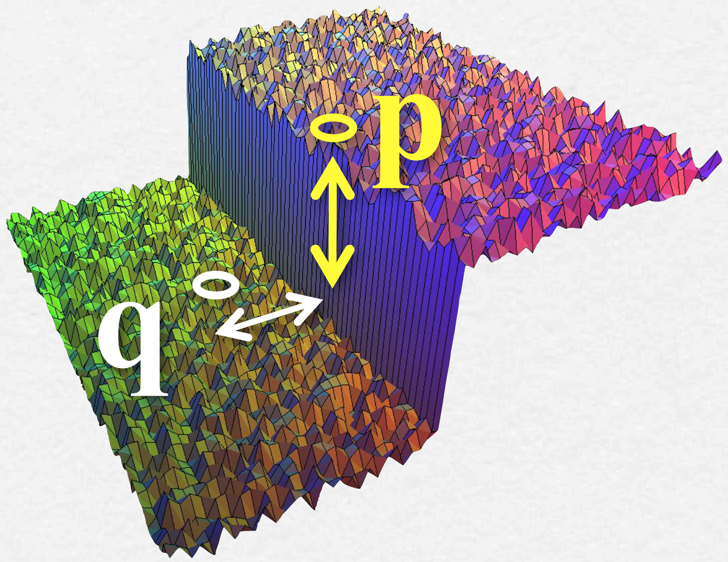


Image d'origine

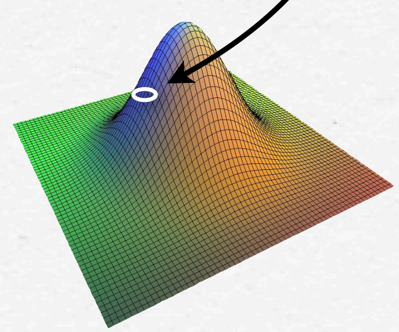
Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

2. Réduction du bruit

4. Filtre bilatéral

- 2D

$$h_b[\mathbf{p}] = \frac{1}{W_{\mathbf{p}}}\left(\sum_{\mathbf{q}\text{ voisin de }\mathbf{p}} \left(\underbrace{g_{\sigma_s}[\mathbf{p}-\mathbf{q}]}\, g_{\sigma_I}\left[f[\mathbf{p}]-f[\mathbf{q}]\right] f[\mathbf{q}]\right)\right)$$



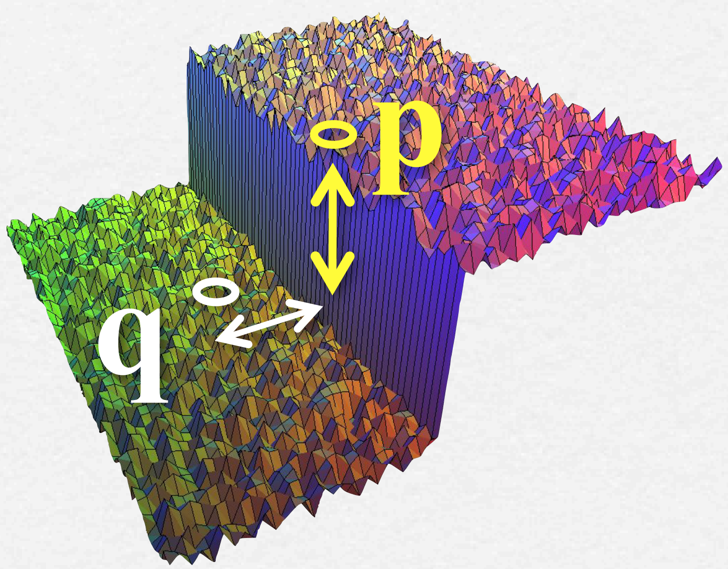


Image d'origine

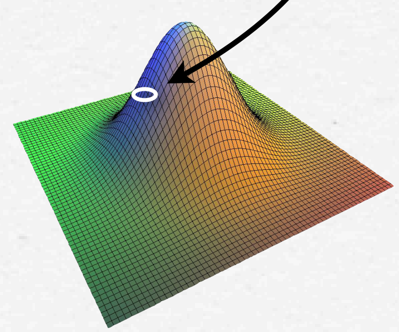
Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

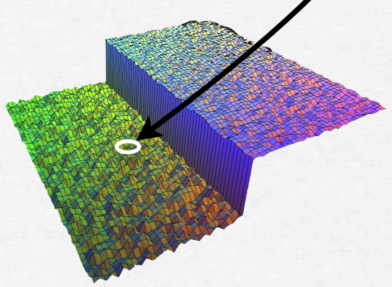
2. Réduction du bruit

4. Filtre bilatéral

- 2D

$$h_b[\mathbf{p}] = \frac{1}{W_{\mathbf{p}}}\left(\sum_{\mathbf{q}\text{ voisin de }\mathbf{p}} \left(\underbrace{g_{\sigma_s}[\mathbf{p}-\mathbf{q}]}\,\underbrace{g_{\sigma_I}\left[f[\mathbf{p}]-f[\mathbf{q}]\right]}\,f[\mathbf{q}]\right)\right)$$





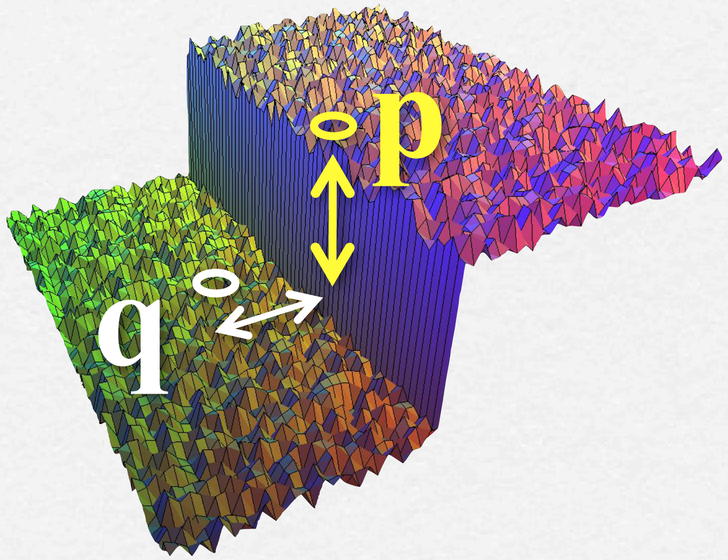


Image d'origine

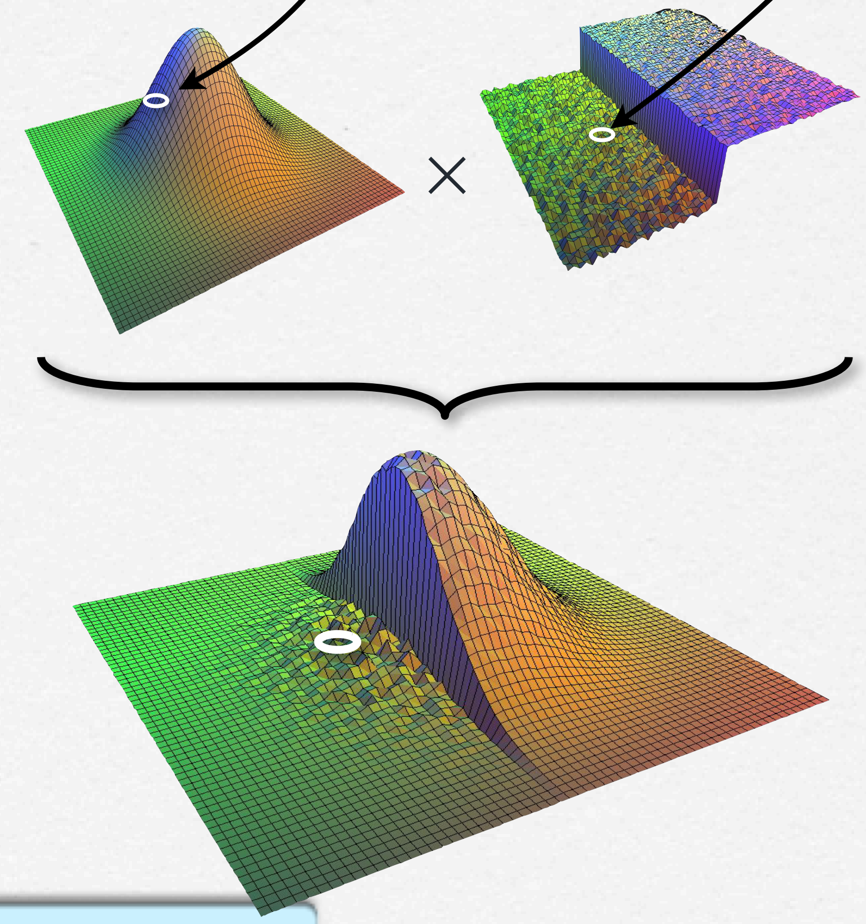
Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

2. Réduction du bruit

4. Filtre bilatéral

- 2D

$$h_b[\mathbf{p}] = \frac{1}{W_{\mathbf{p}}}\left(\sum_{\mathbf{q}\ \text{voisin de}\ \mathbf{p}} \left(g_{\sigma_s}[\mathbf{p}-\mathbf{q}]\, g_{\sigma_I}\left[f[\mathbf{p}] - f[\mathbf{q}]\right] f[\mathbf{q}]\right)\right)$$



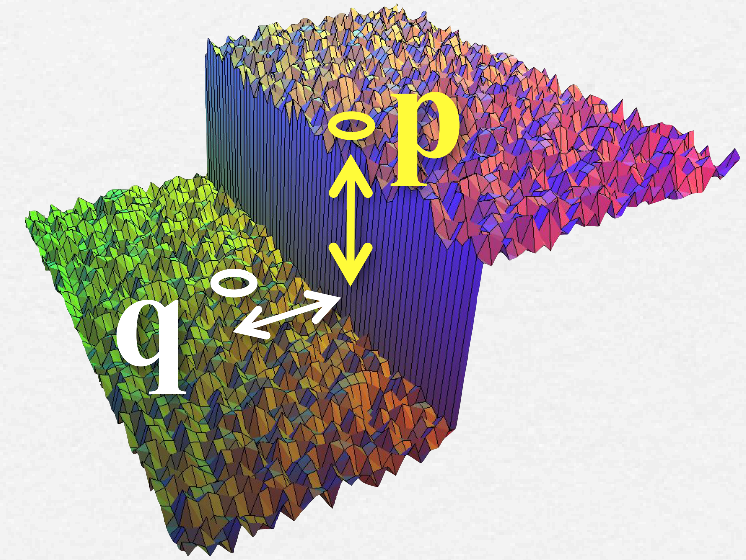


Image d'origine

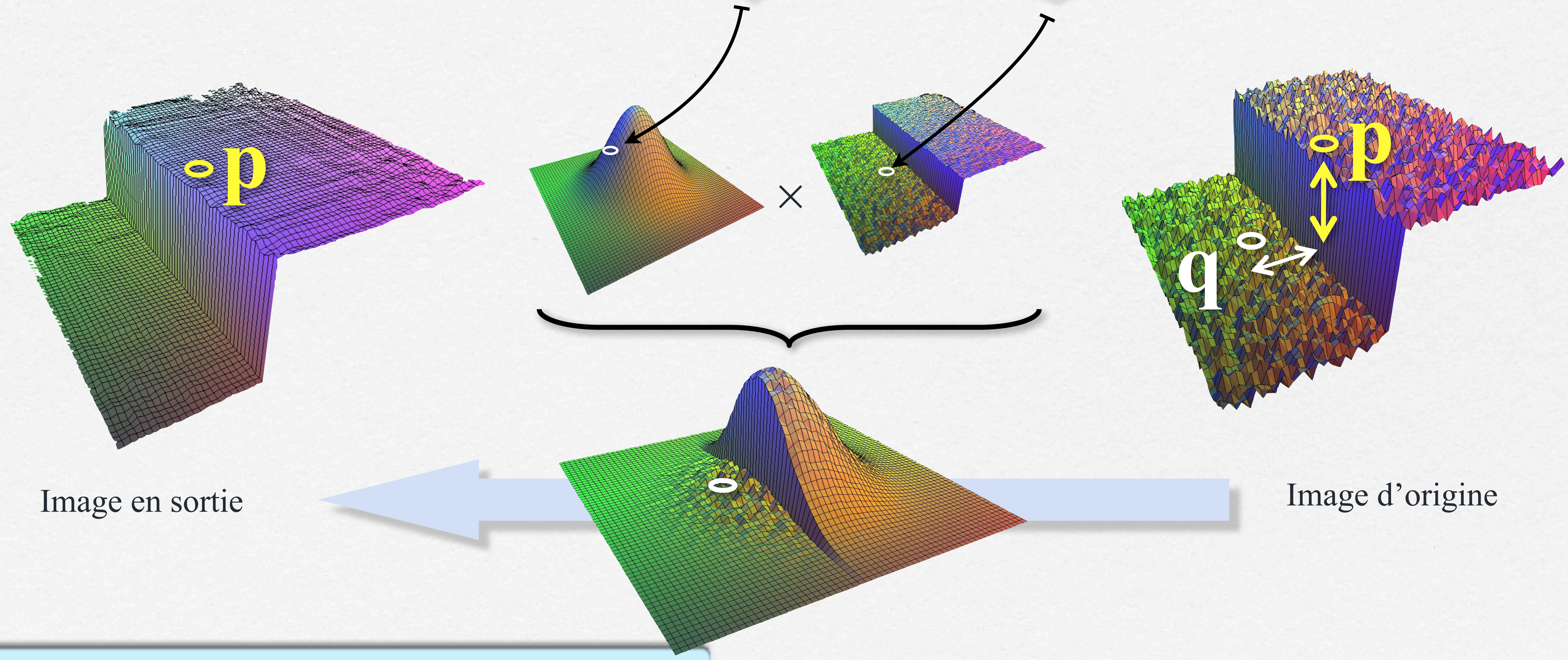
Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

2. Réduction du bruit

4. Filtre bilatéral

- 2D

$$h_b[\mathbf{p}] = \frac{1}{W_\mathbf{p}}\left(\sum_{\mathbf{q}\text{ voisin de }\mathbf{p}}\left(g_{\sigma_s}[\mathbf{p}-\mathbf{q}]\, g_{\sigma_I}\left[f[\mathbf{p}]-f[\mathbf{q}]\right] f[\mathbf{q}]\right)\right)$$



Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH’07

2. RÉDUCTION DU BRUIT

4. Filtre bilatéral

- 2D

$\sigma_i = 0.1$

$\sigma_s = 2$

$\sigma_s = 6$

$\sigma_s = 18$

Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

2. RÉDUCTION DU BRUIT

4. Filtre bilatéral

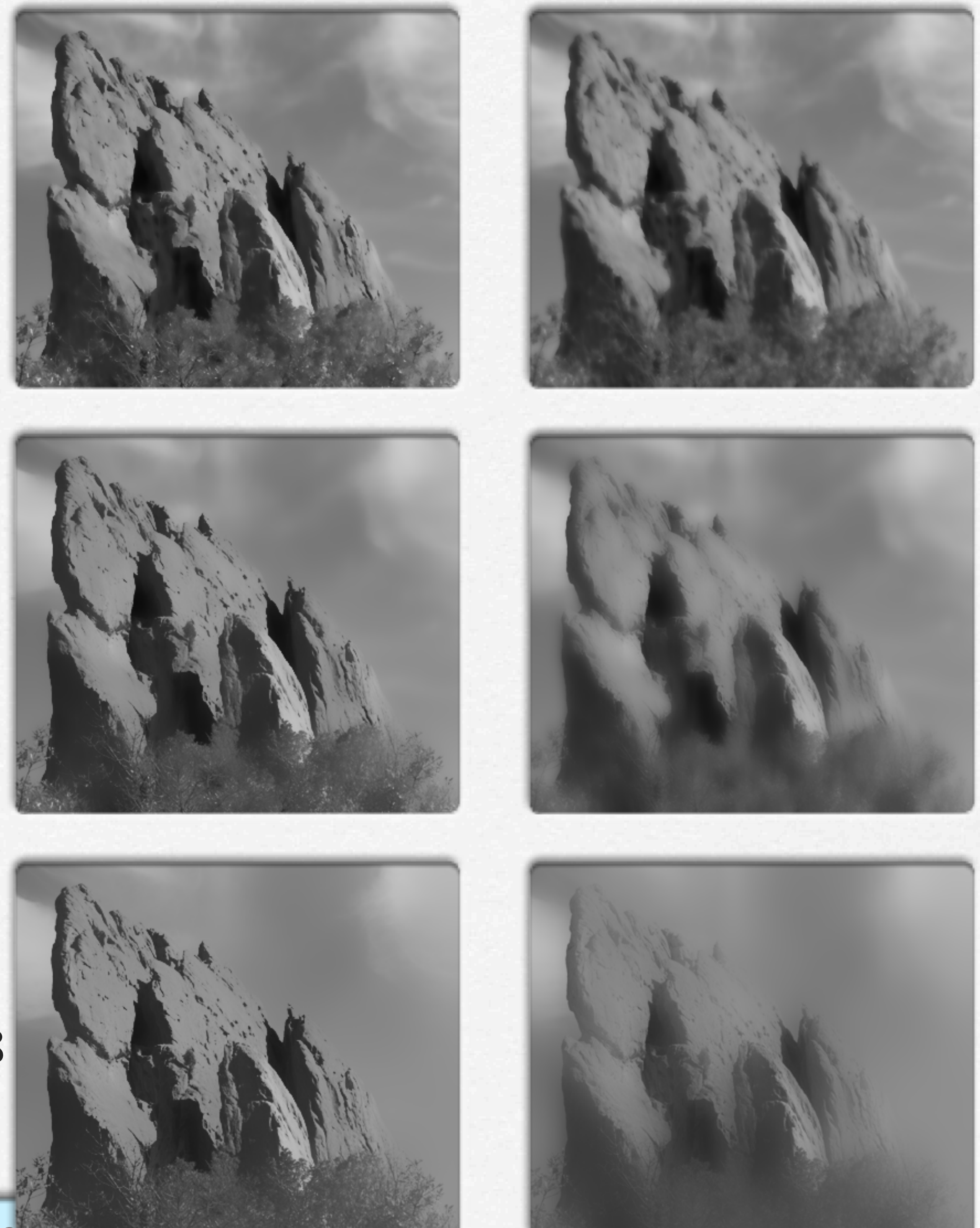
- 2D

$\sigma_i = 0.1$ $\sigma_i = 0.25$

$\sigma_s = 2$

$\sigma_s = 6$

$\sigma_s = 18$



Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

2. RÉDUCTION DU BRUIT

4. Filtre bilatéral

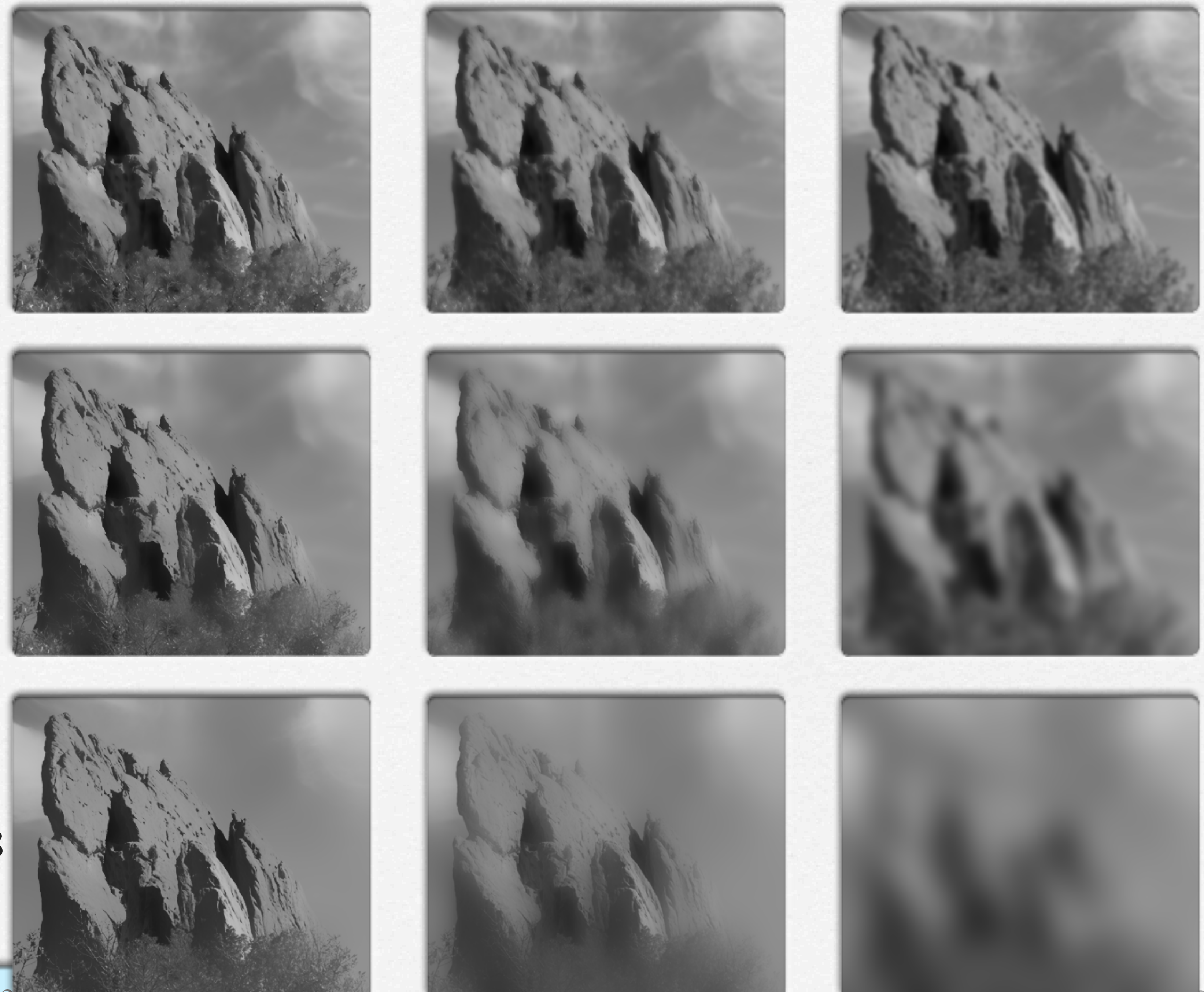
- 2D

$\sigma_i = 0.1$ | $\sigma_i = 0.25$ | $\sigma_i = \infty$ ➡ lissage gaussien

$\sigma_s = 2$

$\sigma_s = 6$

$\sigma_s = 18$



Source : S. Paris, A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH'07

2. Réduction du bruit

4. Filtre bilatéral

- *Inconvénients*
  - ✓ Impossible à effectuer dans Fourier car non linéaire

Source : PM Jodoin

2. Réduction du bruit

4. Filtre bilatéral

- *Inconvénients*
  - ✓ Impossible à effectuer dans Fourier car non linéaire
  - ✓ Lent : il faut recalculer les poids et la normalisation à chaque pixel

Source : PM Jodoin

2. RÉDUCTION DU BRUIT

4. Filtre bilatéral

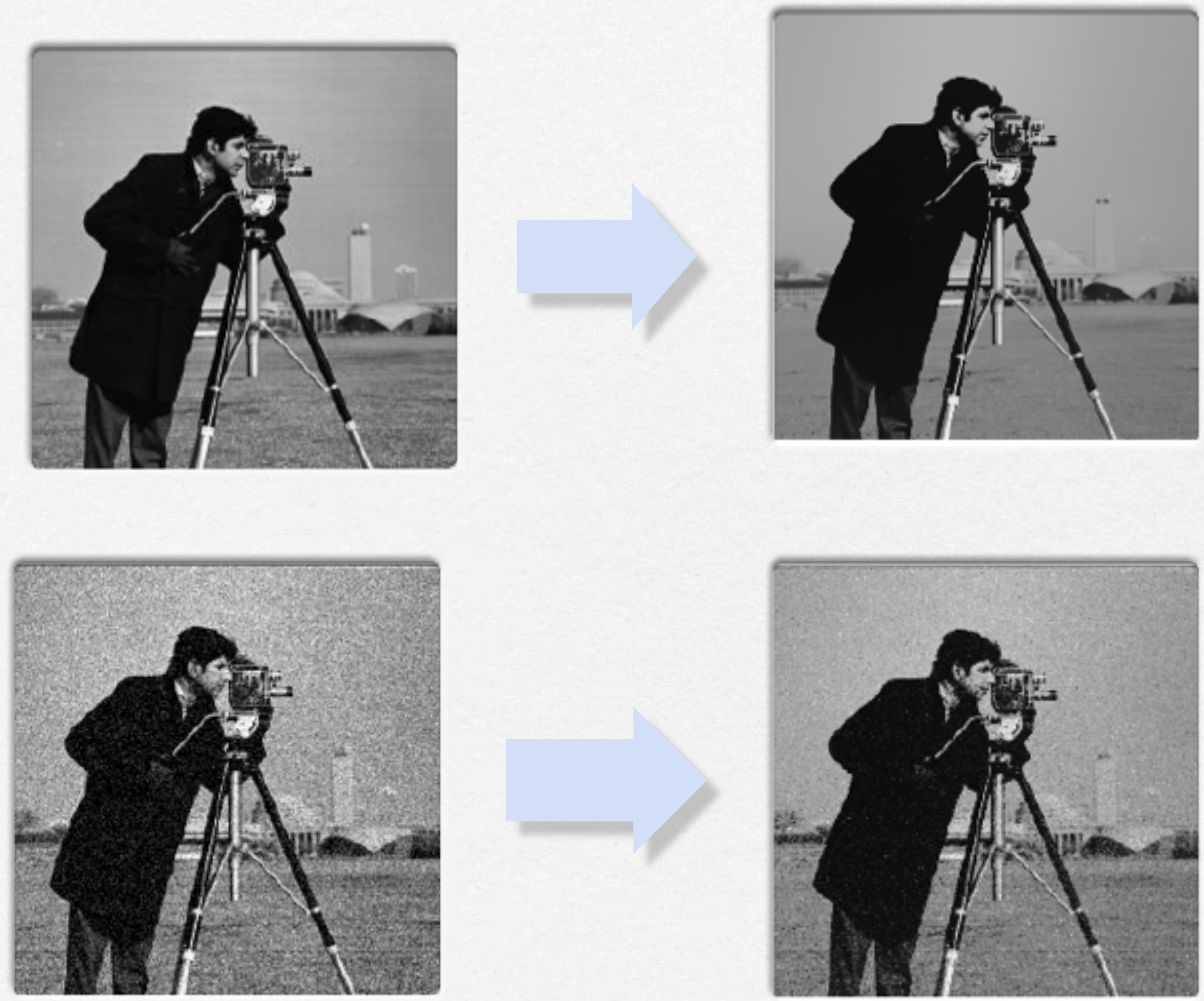
- *Inconvénients*
  - ✓ Impossible à effectuer dans Fourier car non linéaire
  - ✓ Lent : il faut recalculer les poids et la normalisation à chaque pixel
  - ✓ Fonctionne mal en présence de fort bruit

Source : PM Jodoin

2. Réduction du bruit

4. Filtre bilatéral

- *Inconvénients*
  - ✓ Impossible à effectuer dans Fourier car non linéaire
  - ✓ Lent : il faut recalculer les poids et la normalisation à chaque pixel
  - ✓ Fonctionne mal en présence de fort bruit



Source : PM Jodoin

2. RÉDUCTION DU BRUIT

4. Filtre bilatéral

- Peut être itératif

Source : PM Jodoin

2. Réduction du bruit

4. Filtre bilatéral

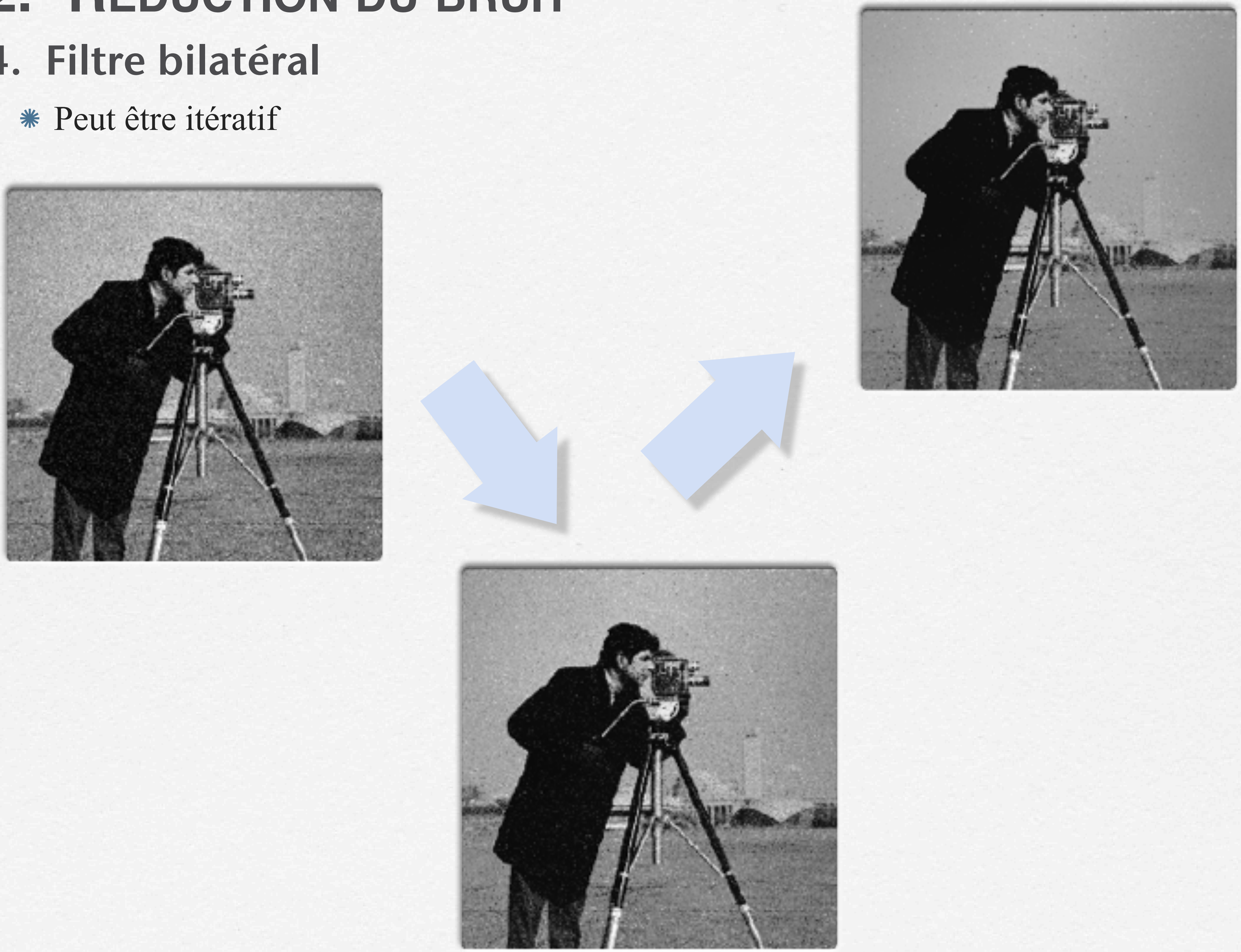
- Peut être itératif

Source : PM Jodoin

2. Réduction du bruit

4. Filtre bilatéral

- Peut être itératif



Source : PM Jodoin

2. Réduction du bruit

4. Filtre bilatéral

- Peut être itératif

Source : PM Jodoin

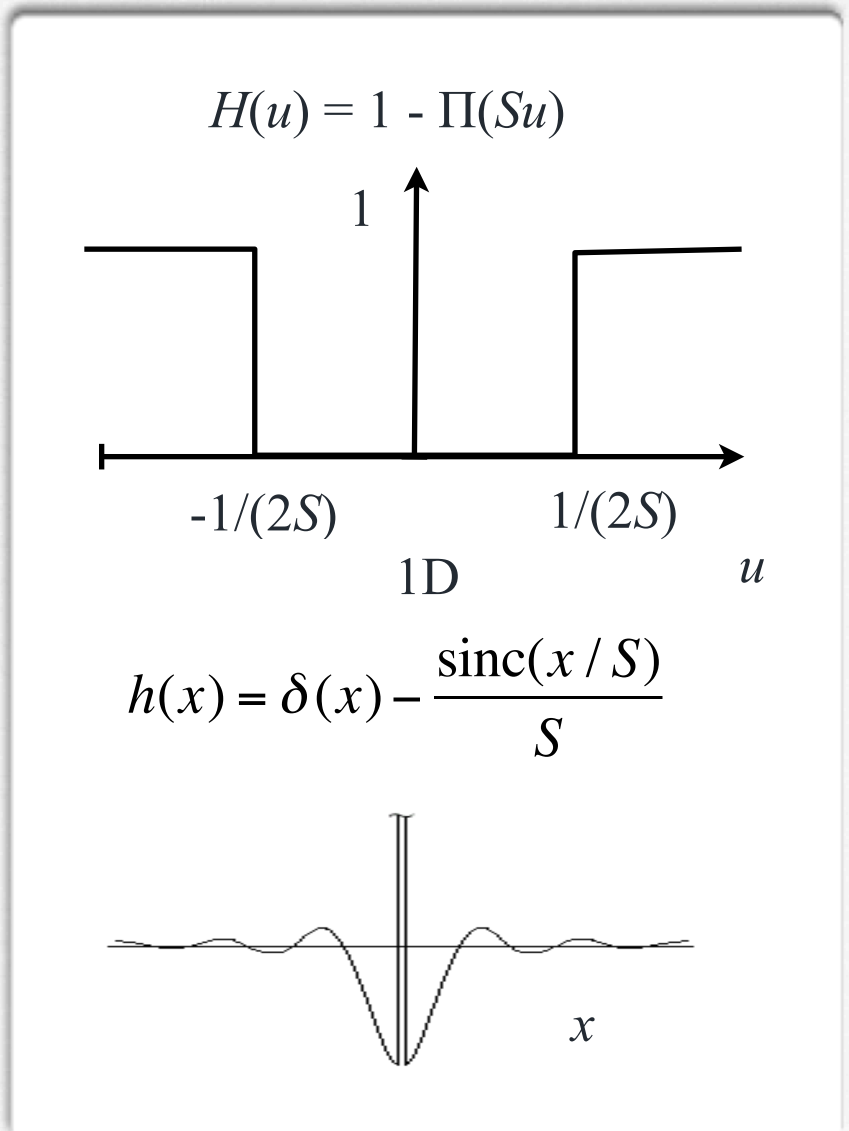
PLAN

1. Rehaussement du contraste
2. Réduction du bruit
3. Amélioration de la netteté
   3.1. Filtrage passe-haut
   3.2. Rehaussement des contours
   3.3. 2e dérivée de l'image
4. Corrélation

3. AMÉLIORATION DE LA NETTETÉ

1. Filtrage passe-haut

- Rappel du filtre idéal
  - ✓ Élimine toutes les fréquences situées en deçà d'un certain seuil



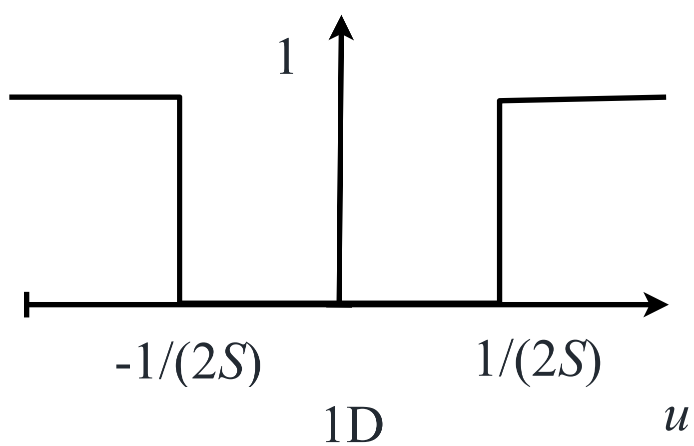
Source : DIP 2

3. AMÉLIORATION DE LA NETTETÉ

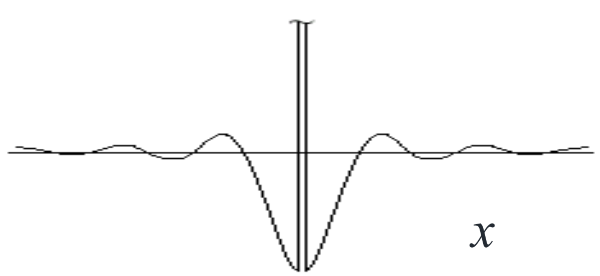
1. Filtrage passe-haut

- Rappel du filtre idéal
  - ✓ Élimine toutes les fréquences situées en deçà d'un certain seuil

$H(u) = 1 - \Pi(Su)$

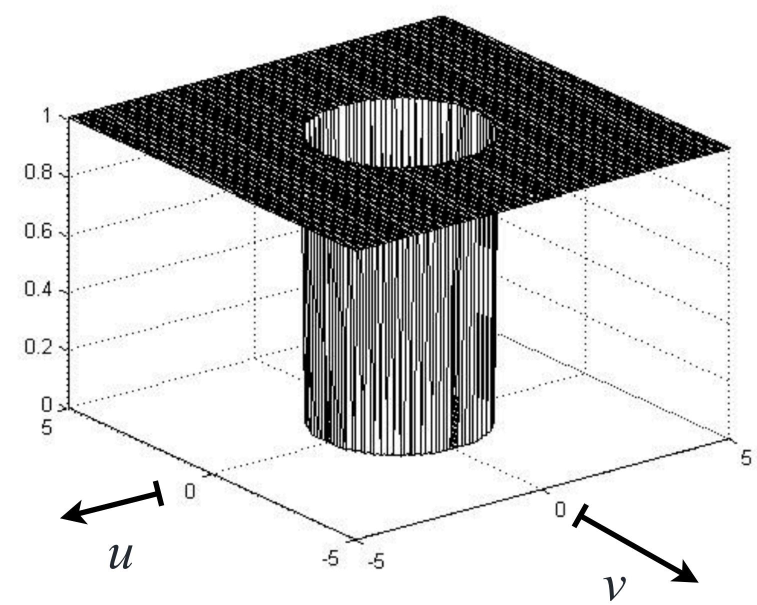


$$h(x) = \delta(x) - \frac{\operatorname{sinc}(x/S)}{S}$$

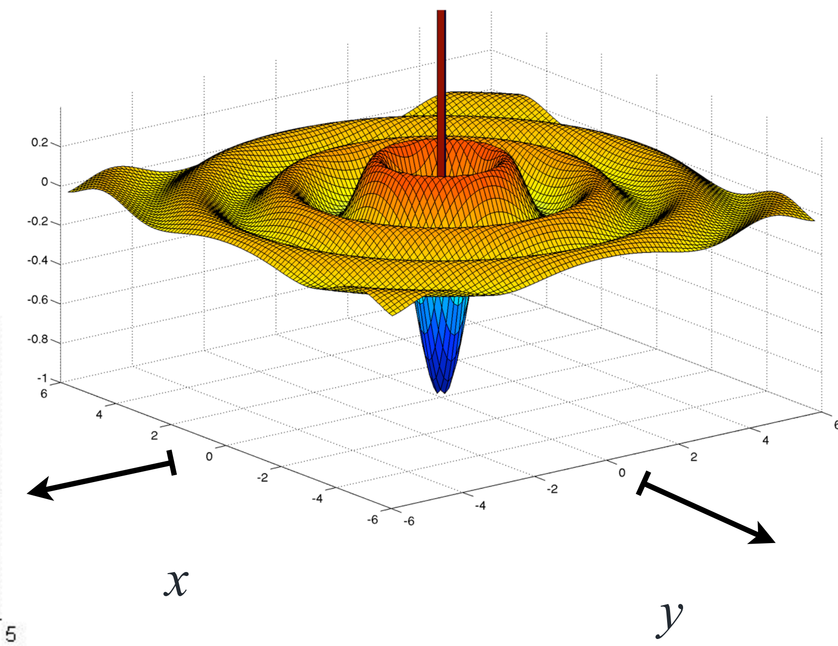


fonction porte ronde

$$H(u,v) = 1 - \Pi\left(S\sqrt{u^2+v^2}\right)$$



$$h(x,y) = \delta(x,y) - \frac{\operatorname{sinc}\left(\sqrt{(x/S)^2 + (y/S)^2}\right)}{S^2}$$



Source : DIP 2

3. Amélioration de la netteté

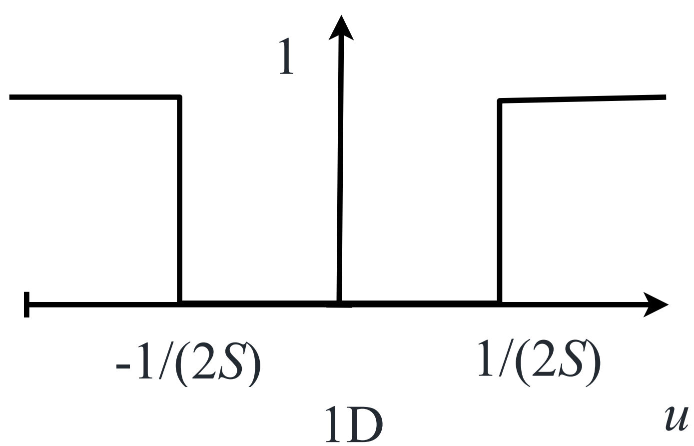
1. Filtrage passe-haut

- Rappel du filtre idéal
  - ✓ Élimine toutes les fréquences situées en deçà d'un certain seuil

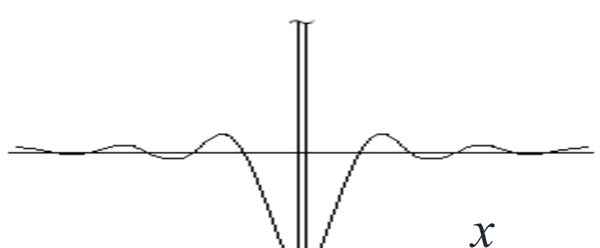
Objectif :

- Utilisation d'un filtre passe-haut pour rehausser les contours

$$H(u) = 1 - \Pi(Su)$$

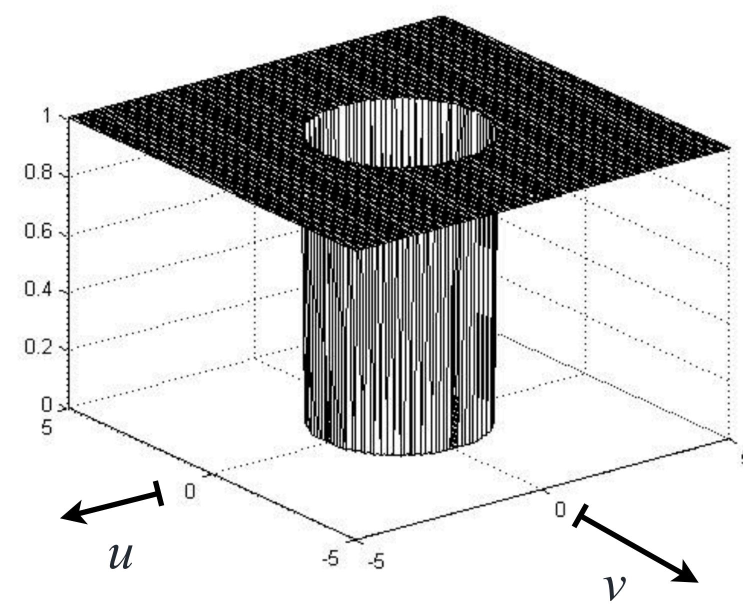


$$h(x) = \delta(x) - \frac{\mathrm{sinc}(x/S)}{S}$$

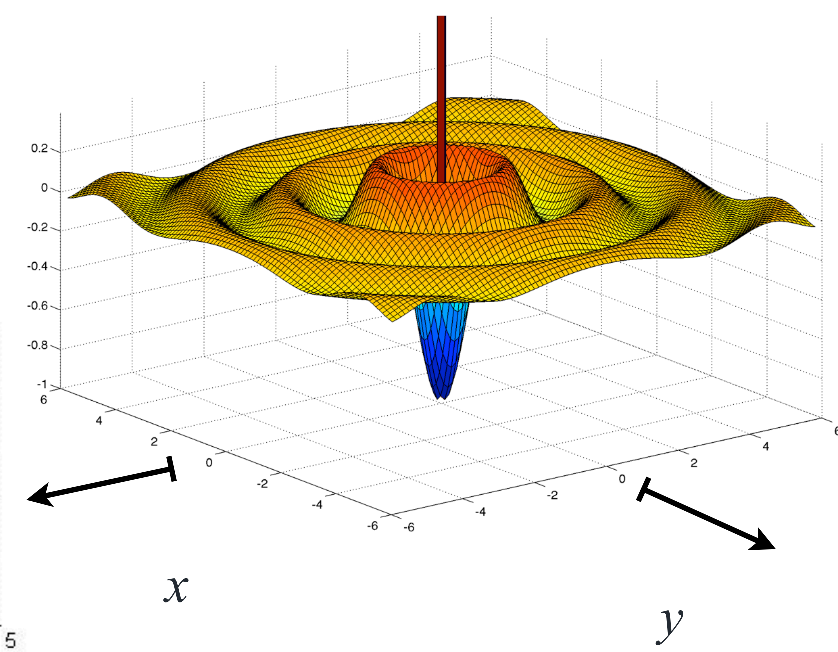


fonction porte ronde

$$H(u,v) = 1 - \Pi\left(S\sqrt{u^2+v^2}\right)$$



$$h(x,y) = \delta(x,y) - \frac{\mathrm{sinc}\left(\sqrt{(x/S)^2+(y/S)^2}\right)}{S^2}$$

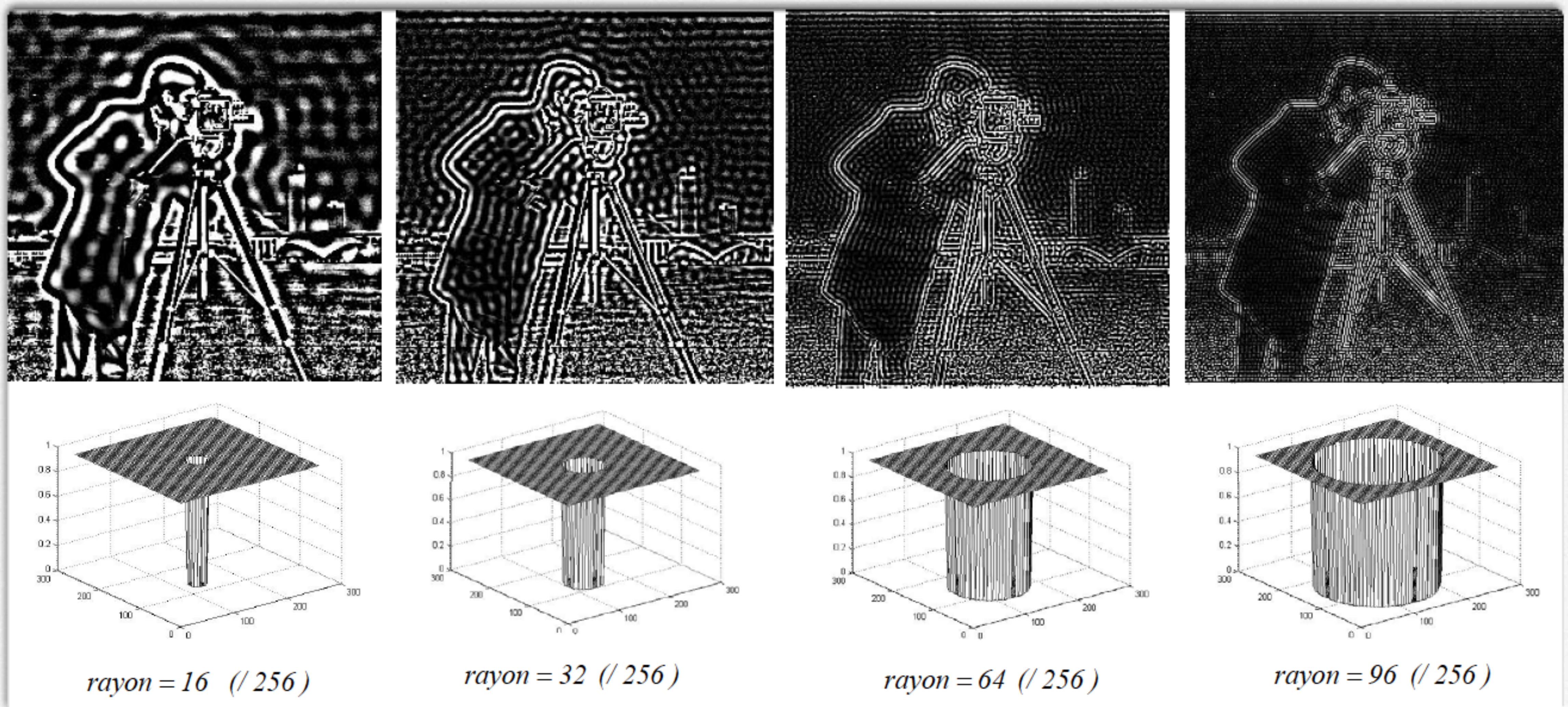


Source : DIP 2

3. AMÉLIORATION DE LA NETTETÉ

1. Filtrage passe-haut

- Problème du filtre idéal
  - ✓ La forme spatiale nous donne la puce à l'oeil



Filtres passe-haut

Filtres utilisés lors de filtrages spatiaux et fréquentiels

« cameraman »

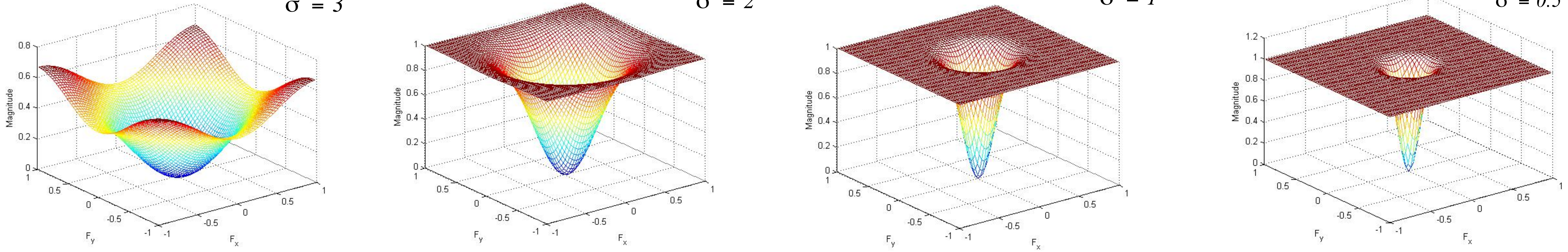
Filtre passe-haut gaussien:

$$h(x,y) = \begin{pmatrix} -0.0297 & -0.1331 & -0.2194 & -0.1331 & -0.0297 \\ -0.1331 & -0.5963 & -0.9832 & -0.5963 & -0.1331 \\ -0.2194 & -0.9832 & 8.3790 & -0.9832 & -0.2194 \\ -0.1331 & -0.5963 & -0.9832 & -0.5963 & -0.1331 \\ -0.0297 & -0.1331 & -0.2194 & -0.1331 & -0.0297 \end{pmatrix} \approx \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

pour $\sigma = 1$

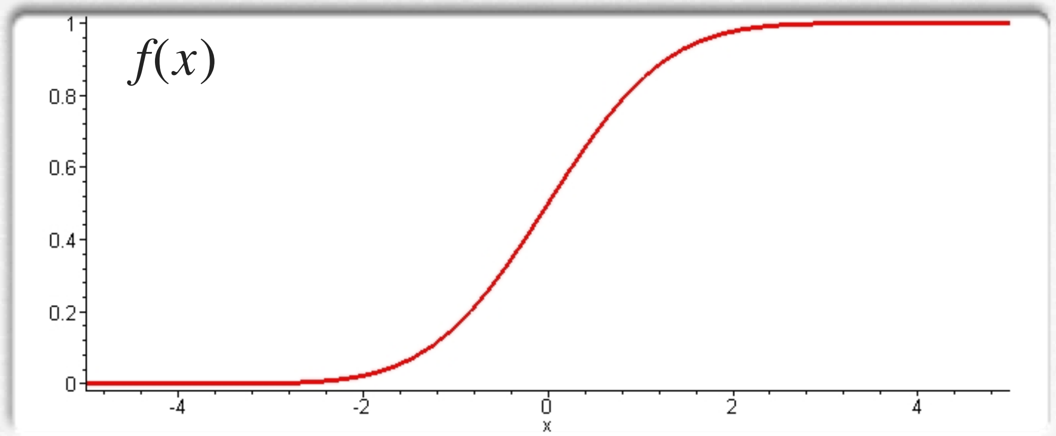
Exemple :

$\sigma = 3$ $\sigma = 2$ $\sigma = 1$ $\sigma = 0.5$



3. Amélioration de la netteté

- Image originale $f(x)$: contour de type marche
  - ✓ non net mais non bruité

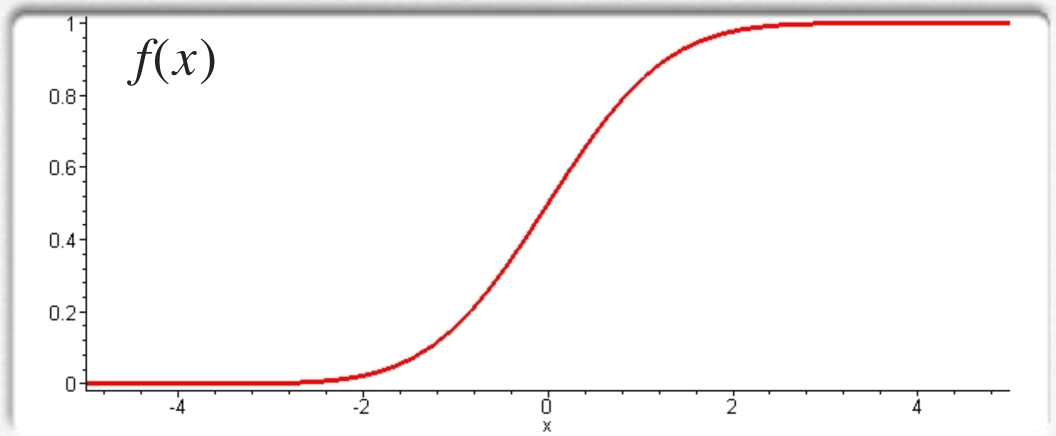


Source :

3. Amélioration de la netteté

1. Filtrage passe-haut

- Image originale $f(x)$: contour de type marche
  - ✓ non net mais non bruité

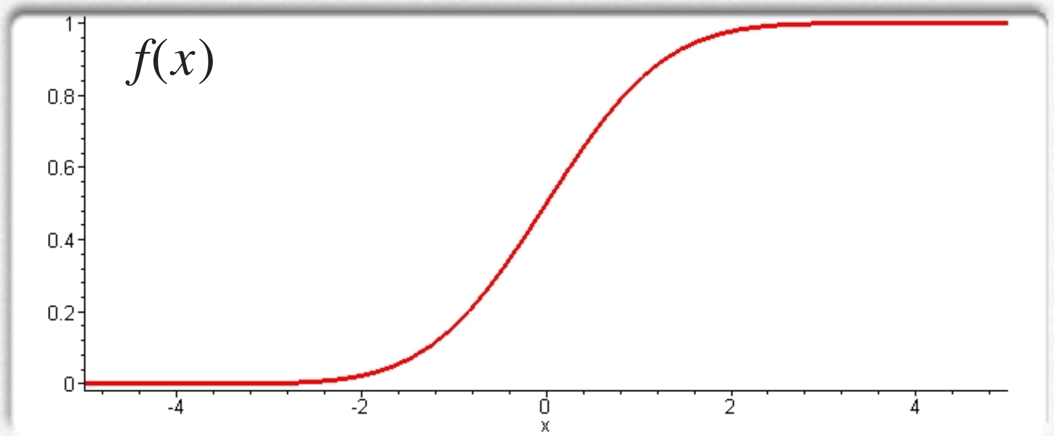


Source :

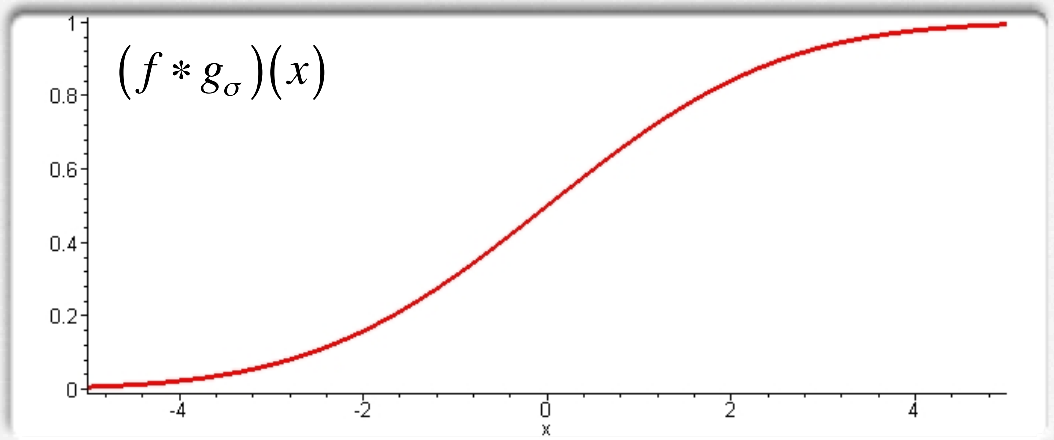
3. Amélioration de la netteté

1. Filtrage passe-haut

- Image originale $f(x)$: contour de type marche
  - ✓ non net mais non bruité



- Image filtrée par un filtre passe-bas

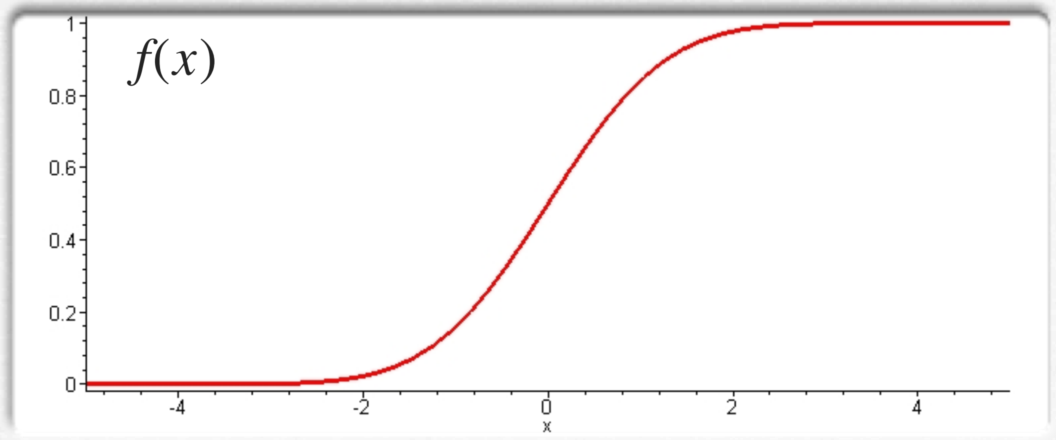


Source :

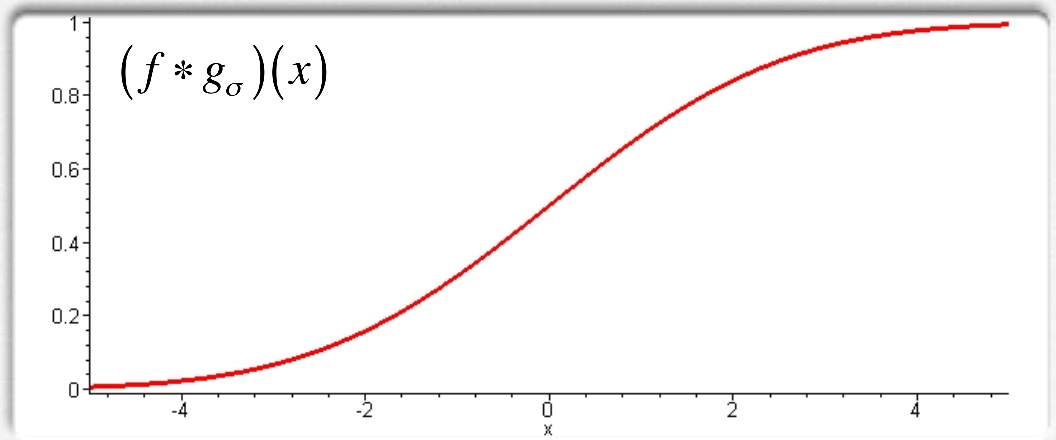
3. Amélioration de la netteté

1. Filtrage passe-haut

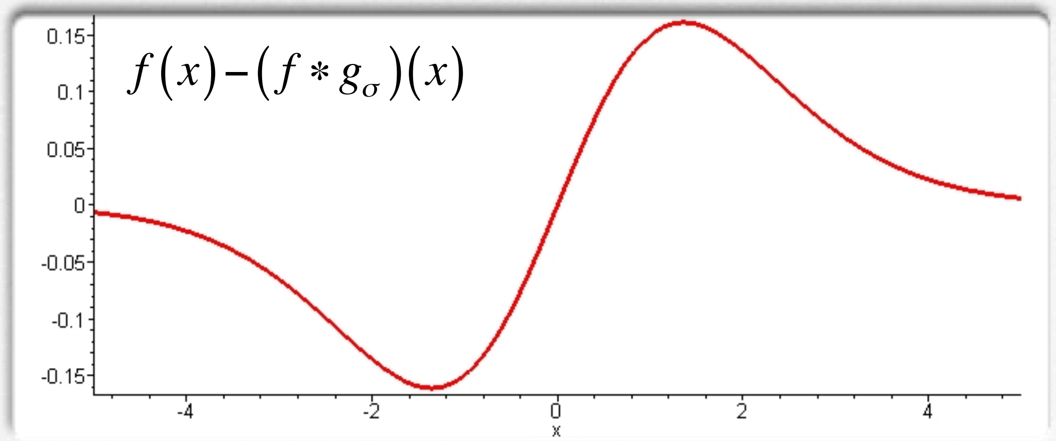
- Image originale $f(x)$: contour de type marche
  - ✓ non net mais non bruité



- Image filtrée par un filtre passe-bas



- Image filtrée par un filtre passe-haut



Source :

3. Amélioration de la netteté

1. Filtrage passe-haut

Source :

3. AMÉLIORATION DE LA NETTETÉ

1. Filtrage passe-haut

Source :

3. Amélioration de la netteté

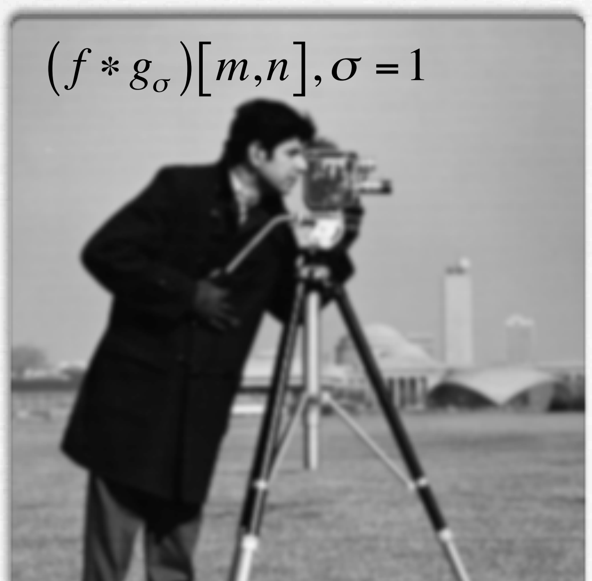
1. Filtrage passe-haut

$f[m,n]-(f*g_\sigma)[m,n]$

Source :

3. Amélioration de la netteté

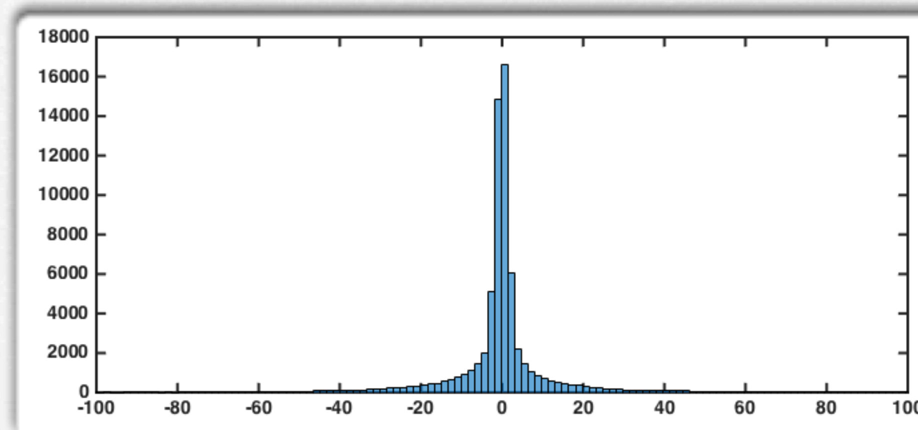
1. Filtrage passe-haut



$f[m,n]-(f * g_\sigma)[m,n]$

Soustraction de 2 images

- il y a des négatifs



Source :

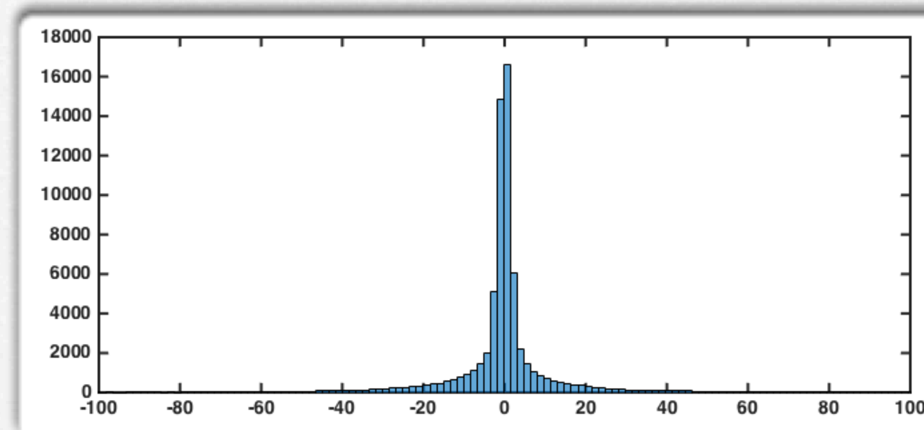
3. Amélioration de la netteté

1. Filtrage passe-haut

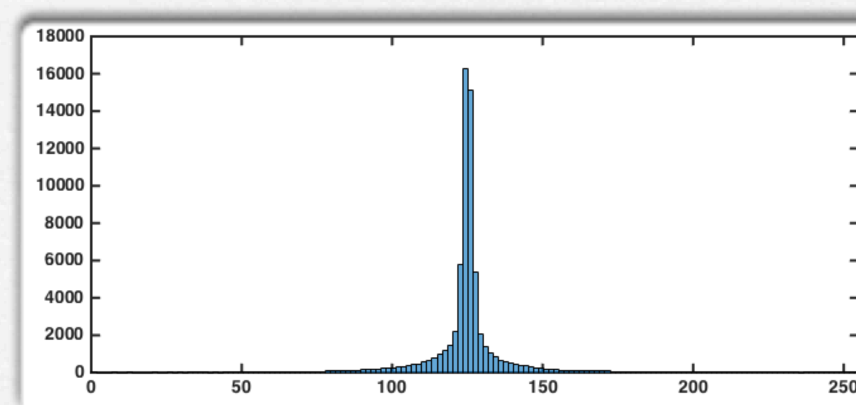
$f[m,n]-(f * g_\sigma)[m,n]$

Soustraction de 2 images

- il y a des négatifs



après changement de dynamique linéaire



Source :

3. AMÉLIORATION DE LA NETTETÉ

2. Rehaussement des contours

Image originale

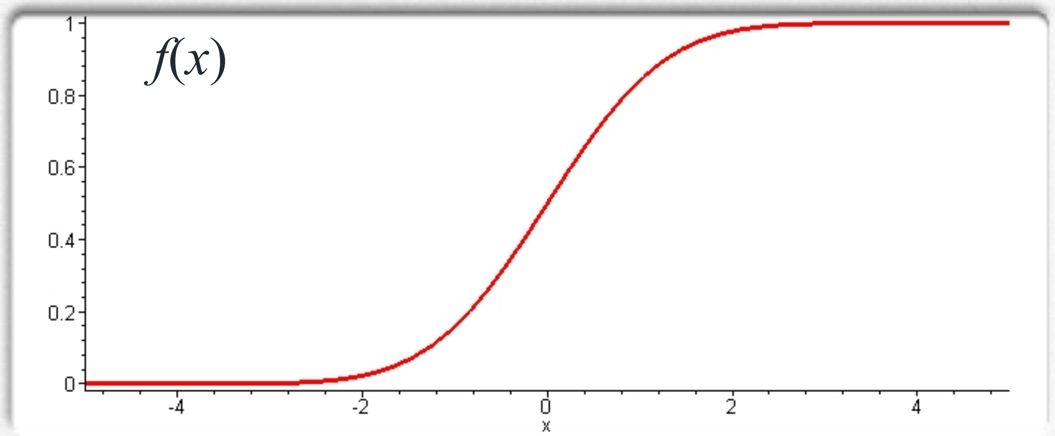
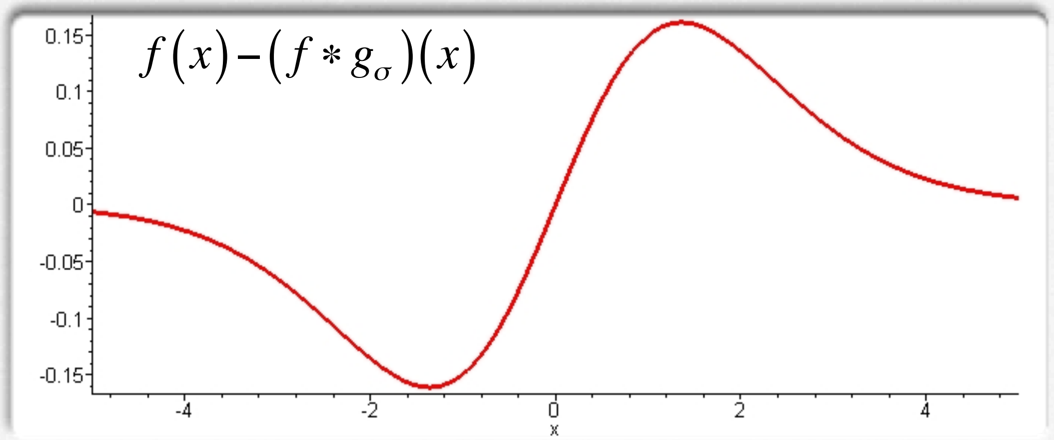


Image filtrée par un filtre passe-haut



Source :

3. Amélioration de la netteté

2. Rehaussement des contours

Image originale

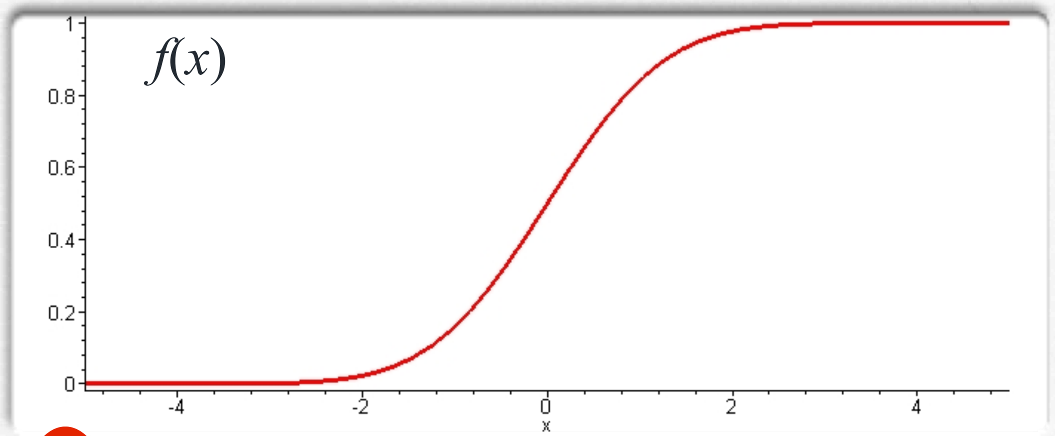
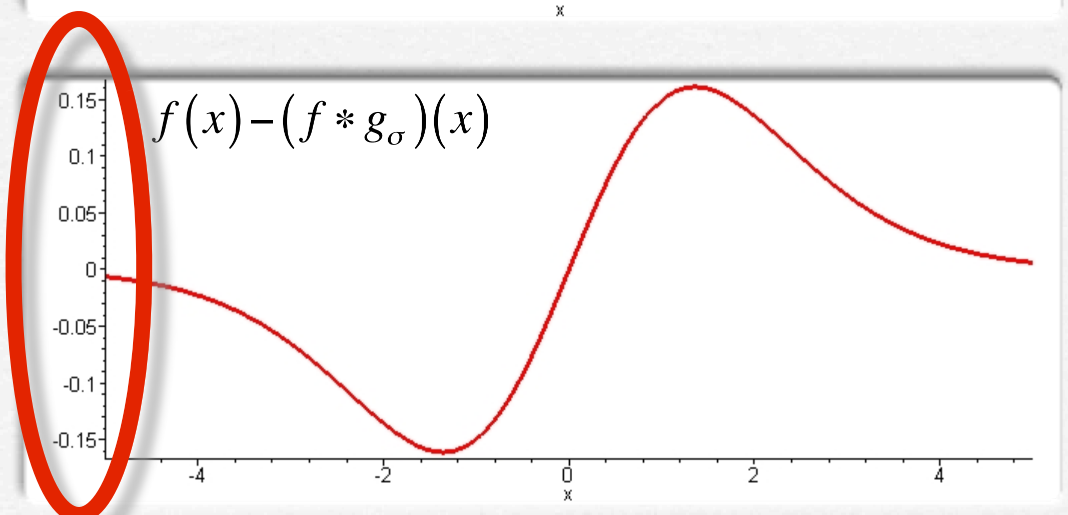


Image filtrée par un filtre passe-haut



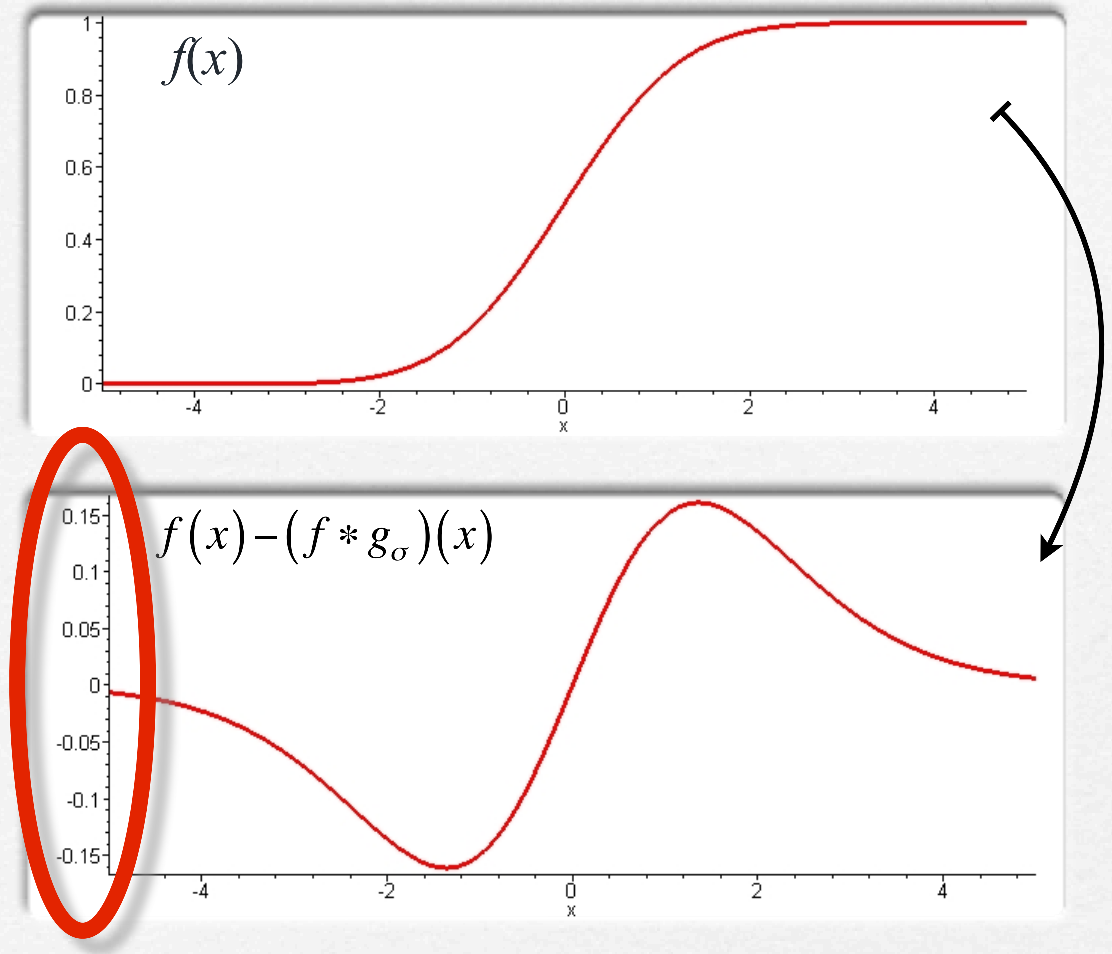
Source :

3. Amélioration de la netteté

2. Rehaussement des contours

Image originale

Image filtrée par un filtre passe-haut



Source :

3. Amélioration de la netteté

2. Rehaussement des contours

Image originale

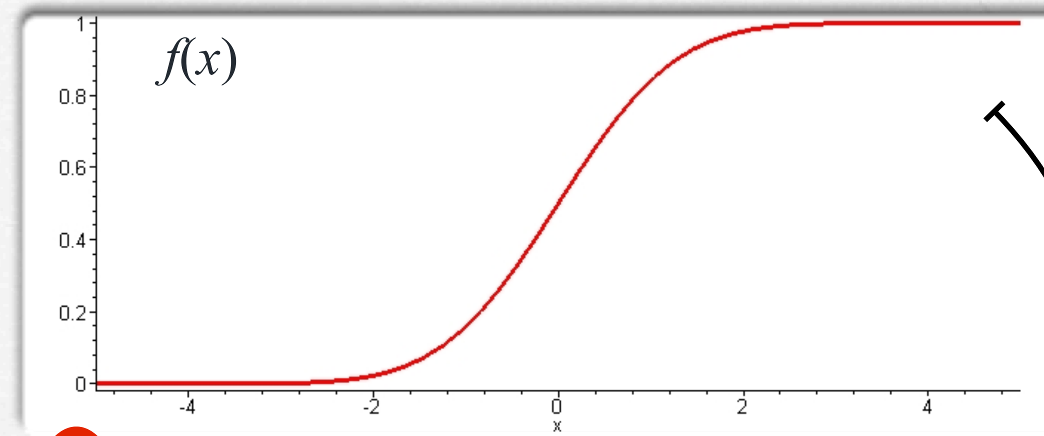
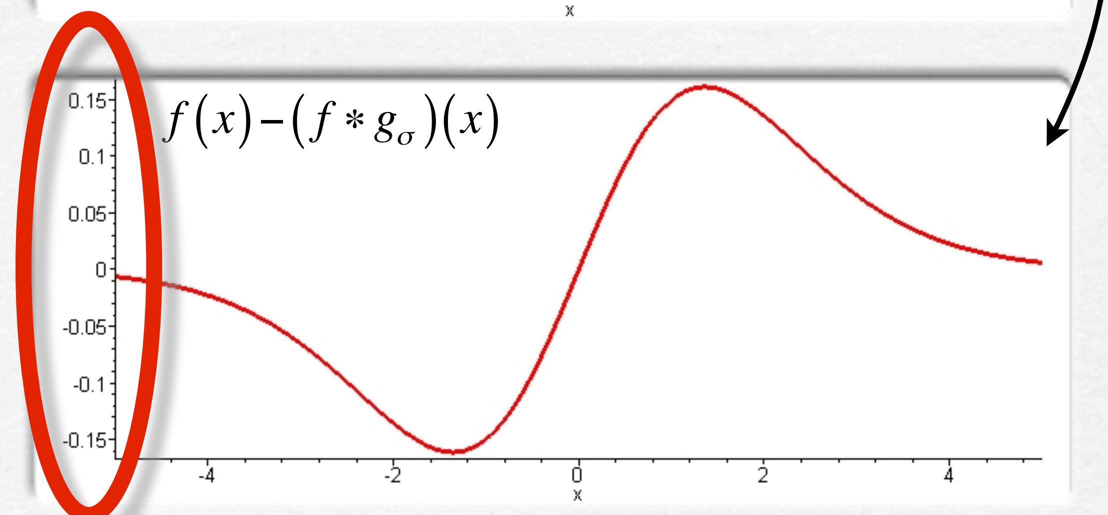
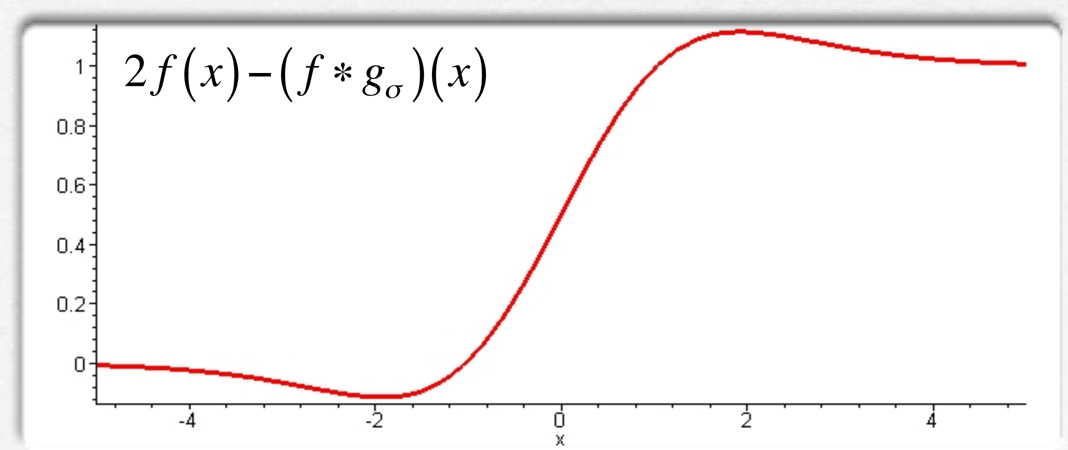


Image filtrée par un filtre passe-haut



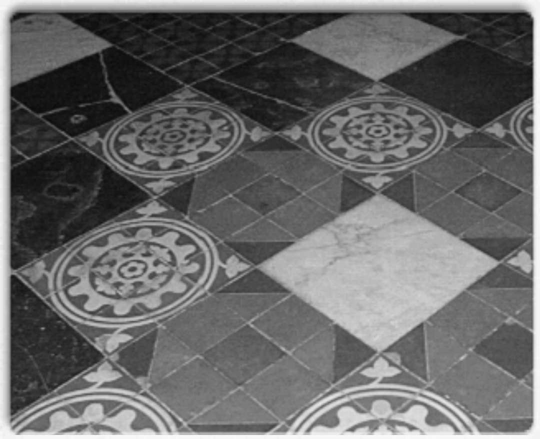
Correction du niveau de gris moyen



Source :

3. Amélioration de la netteté

2. Rehaussement des contours

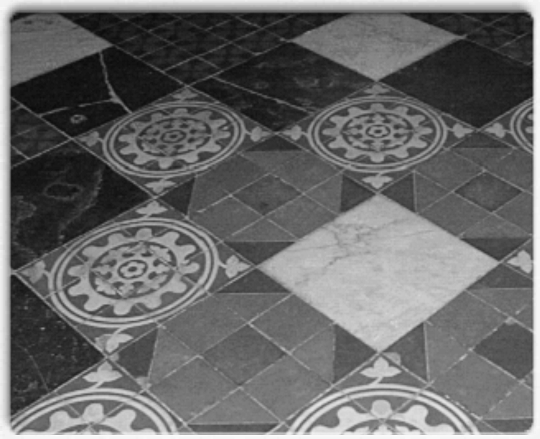


$f[m, n]$

Source :

3. Amélioration de la netteté

2. Rehaussement des contours



$-$

$f[m, n]$

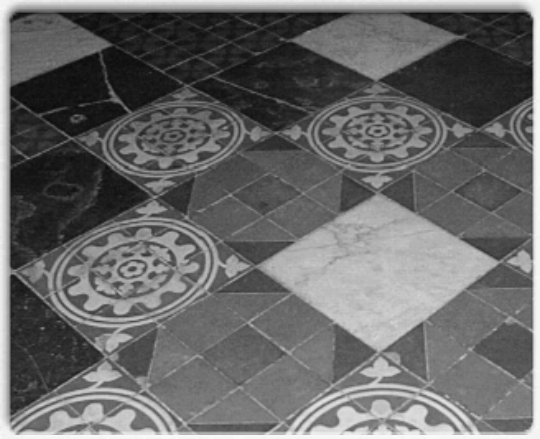
$f_b[m,n] = (f * g_\sigma)[m,n]$

avec $\sigma = 1.3$ $(N = 9)$

Source :

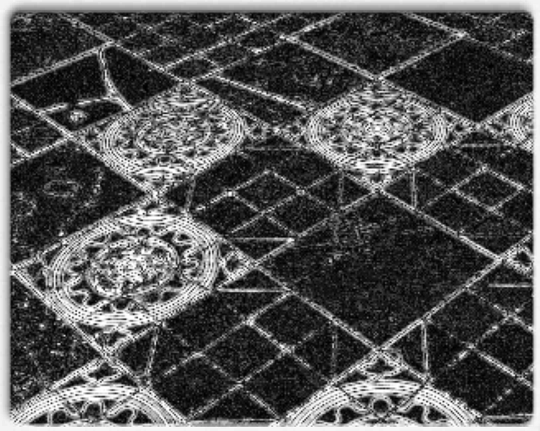
3. Amélioration de la netteté

2. Rehaussement des contours



$-$

$=$



$f[m, n]$

$f_b[m,n] = (f * g_\sigma)[m,n]$

avec $\sigma = 1.3$ $(N = 9)$

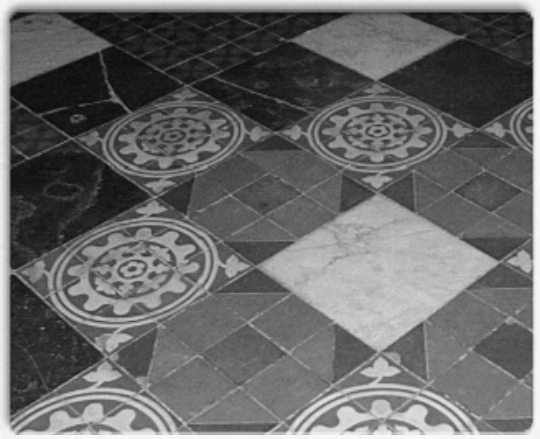
$f[m,n] - f_b[m,n]$

Source :

3. AMÉLIORATION DE LA NETTETÉ

2. Rehaussement des contours

Valeur absolue avec changement de la dynamique



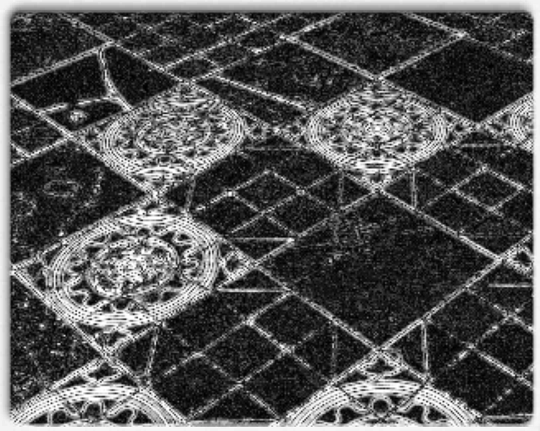
$f[m, n]$

$-$

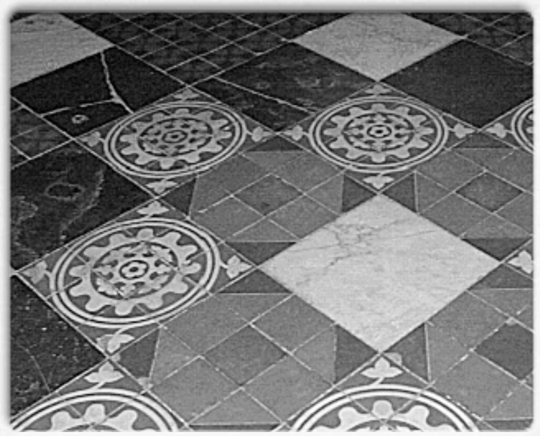
$f_b[m,n] = (f * g_\sigma)[m,n]$

avec $\sigma = 1.3$ $(N = 9)$

$=$



$f[m,n] - f_b[m,n]$



$f[m, n]$ rehaussée

$2f[m,n] - f_b[m,n]$

Source :

3. Amélioration de la netteté

2. Rehaussement des contours

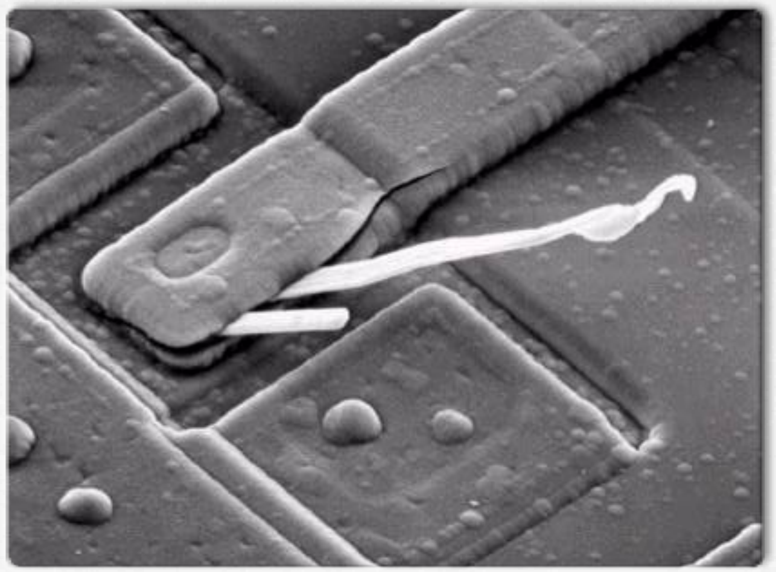


Image originale : $f[m, n]$

Source de l'image originale : DIP2

3. Amélioration de la netteté

2. Rehaussement des contours

Correction du niveau de gris moyen

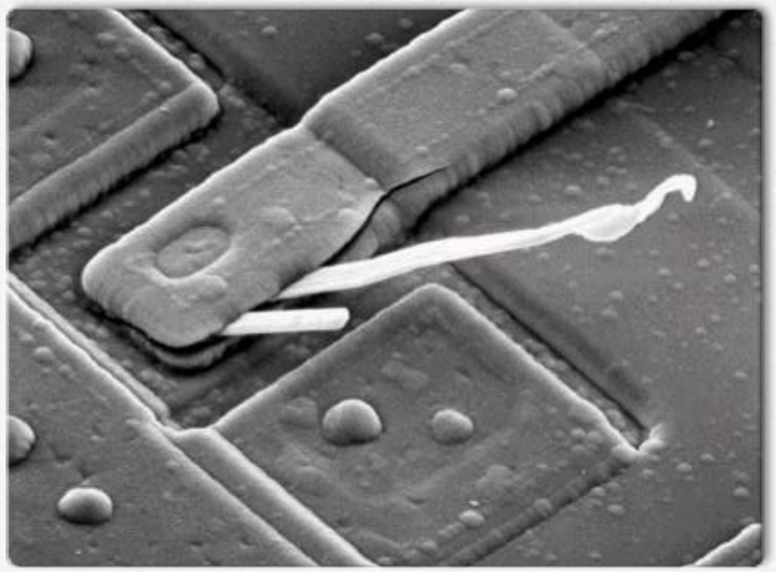


Image originale : $f[m, n]$

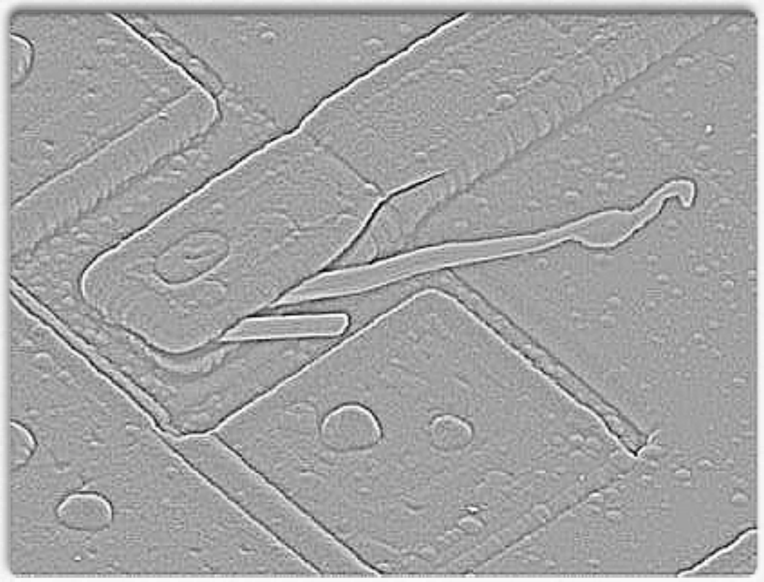


Image filtrée par un filtre passe-haut avec changement de dynamique pour la visualisation

$$f[m, n] - (f * g_\sigma)[m, n], \quad \sigma = 1$$

Source de l'image originale : DIP2

3. AMÉLIORATION DE LA NETTETÉ

2. Rehaussement des contours

Correction du niveau de gris moyen

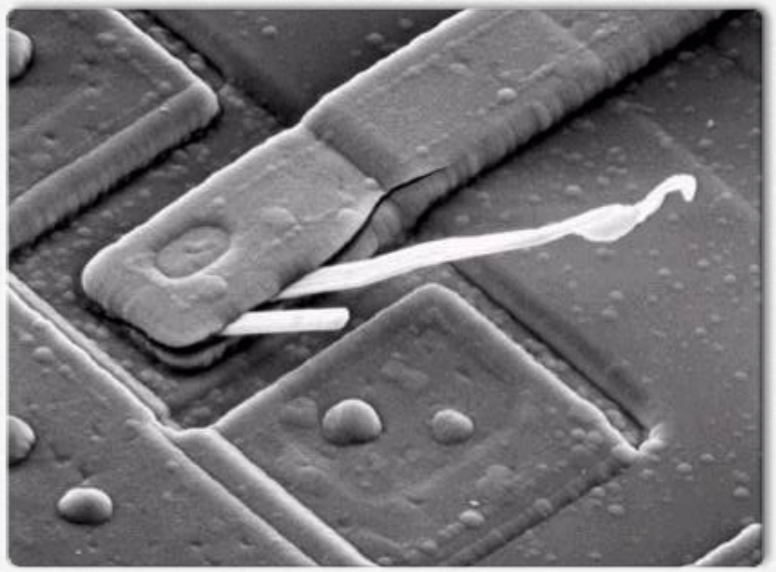


Image originale : $f[m, n]$

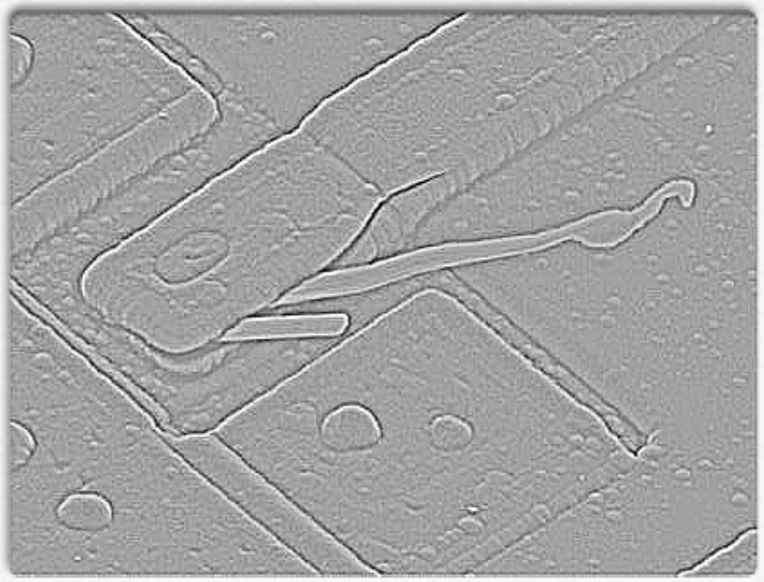
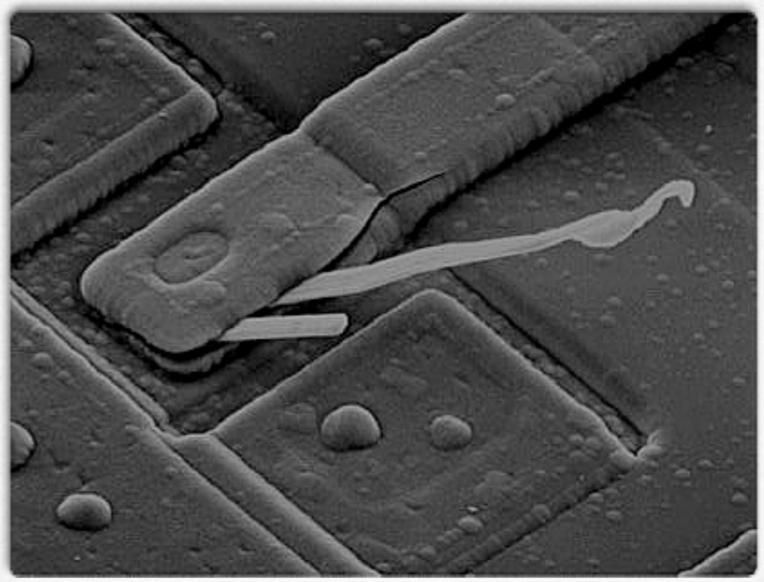


Image filtrée par un filtre passe-haut avec changement de dynamique pour la visualisation

$$f[m, n] - (f * g_\sigma)[m, n], \quad \sigma = 1$$



$$1.5 f[m, n] - (f * g_\sigma)[m, n]$$

Source de l'image originale : DIP2

3. AMÉLIORATION DE LA NETTETÉ

2. Rehaussement des contours

Correction du niveau de gris moyen

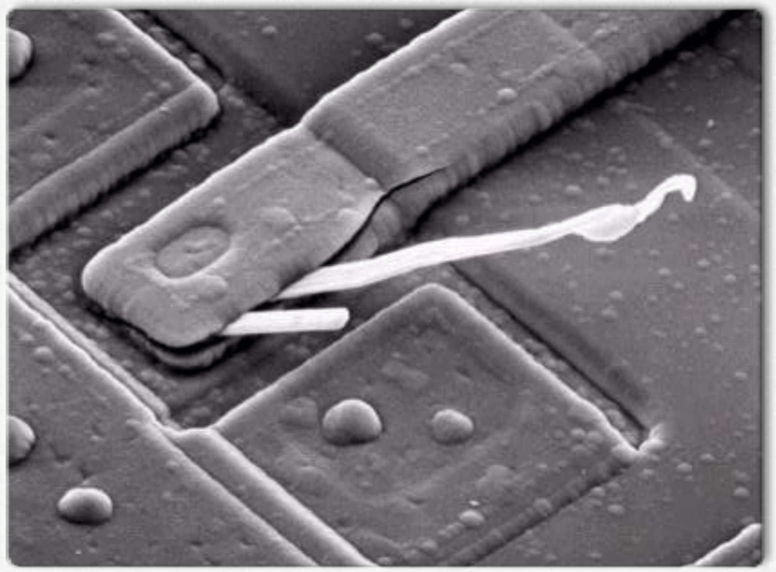


Image originale : $f[m, n]$

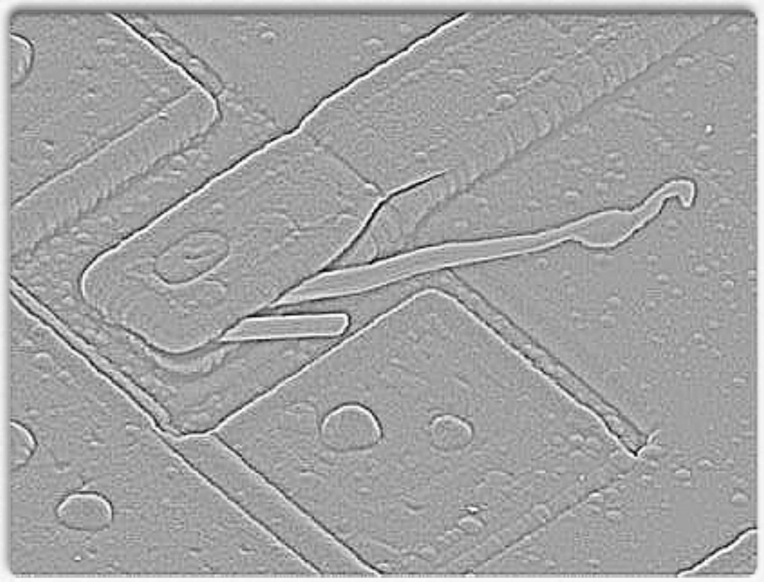
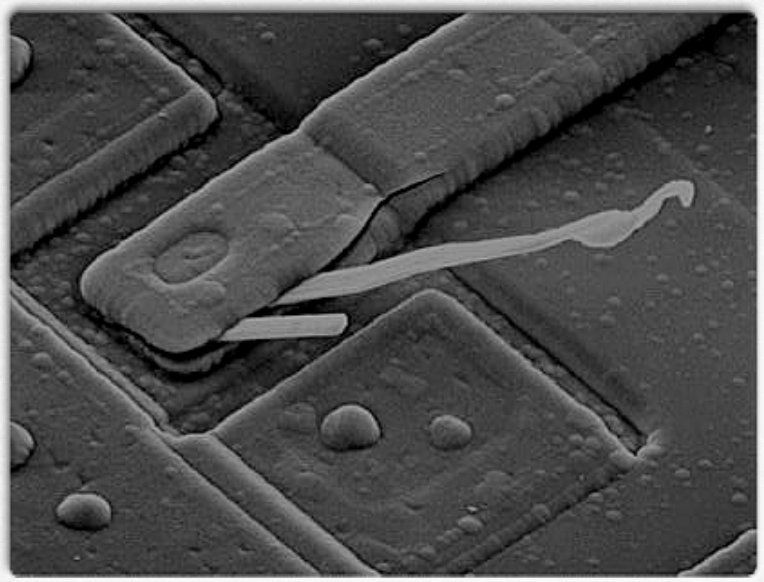
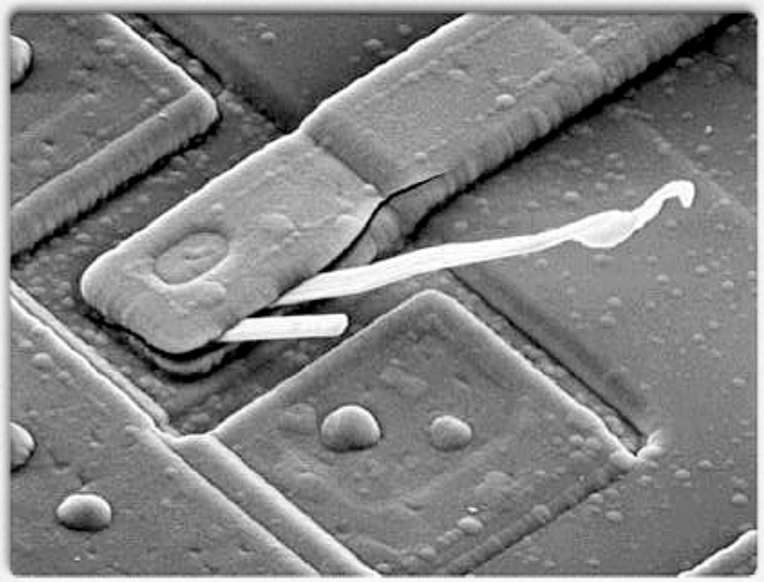


Image filtrée par un filtre passe-haut avec changement de dynamique pour la visualisation

$$f[m, n] - (f * g_\sigma)[m, n], \quad \sigma = 1$$



$1.5 f[m, n] - (f * g_\sigma)[m, n]$



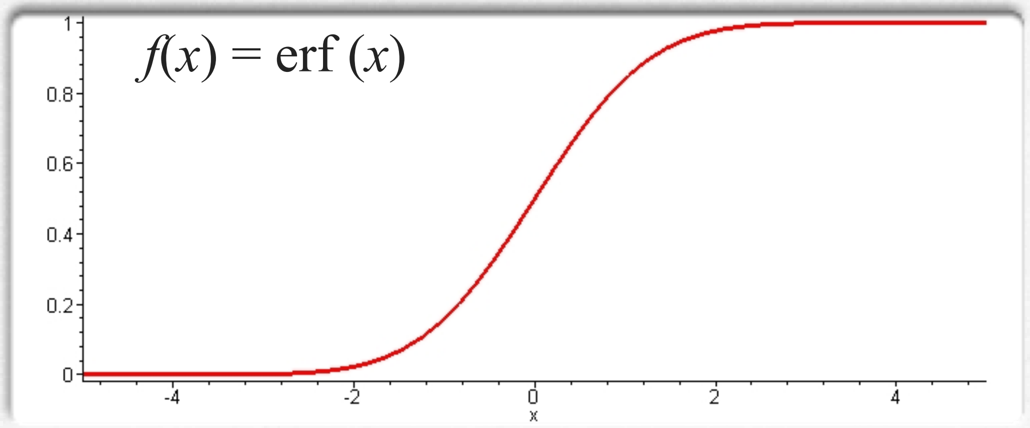
$2 f[m, n] - (f * g_\sigma)[m, n]$

Source de l'image originale : DIP2

3. AMÉLIORATION DE LA NETTETÉ

3. 2e dérivée de l'image

Image originale
Contour modélisé par erf(x)



Source :

3. AMÉLIORATION DE LA NETTETÉ

3. 2e dérivée de l'image

Image originale
Contour modélisé par erf(x)

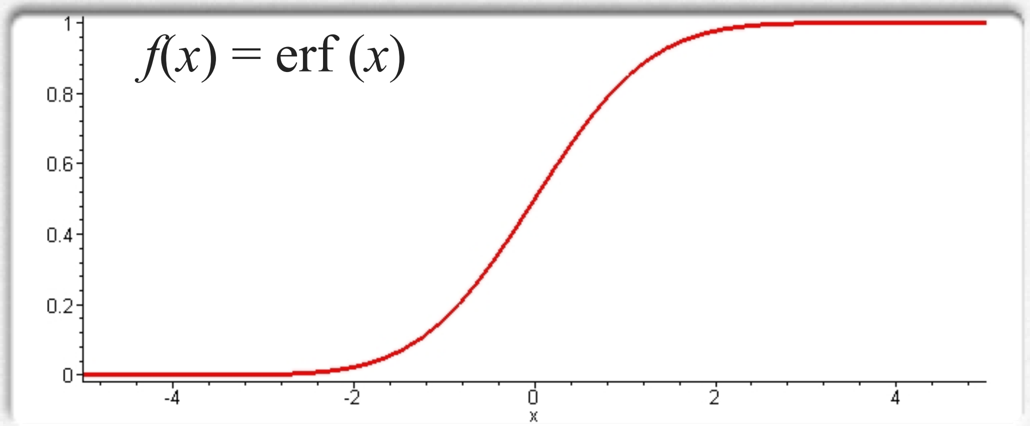
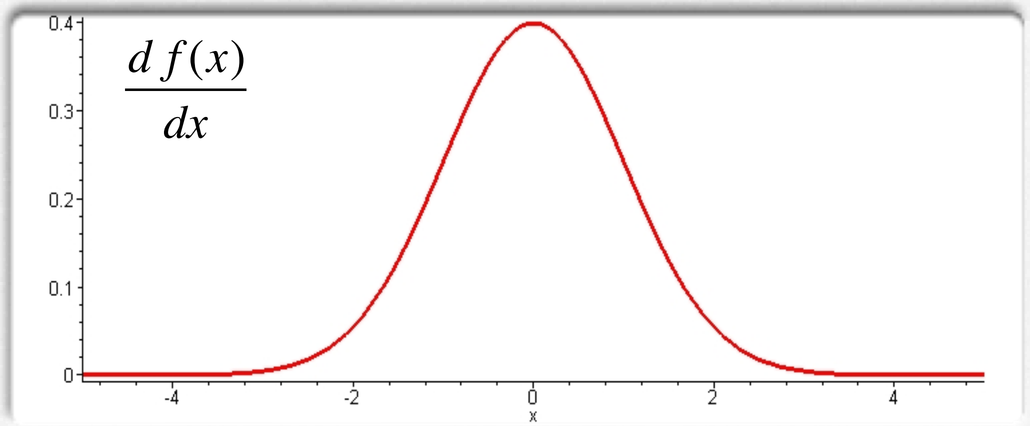


Image dérivée selon x



Source :

3. AMÉLIORATION DE LA NETTETÉ

3. 2e dérivée de l'image

Image originale
Contour modélisé par erf(x)

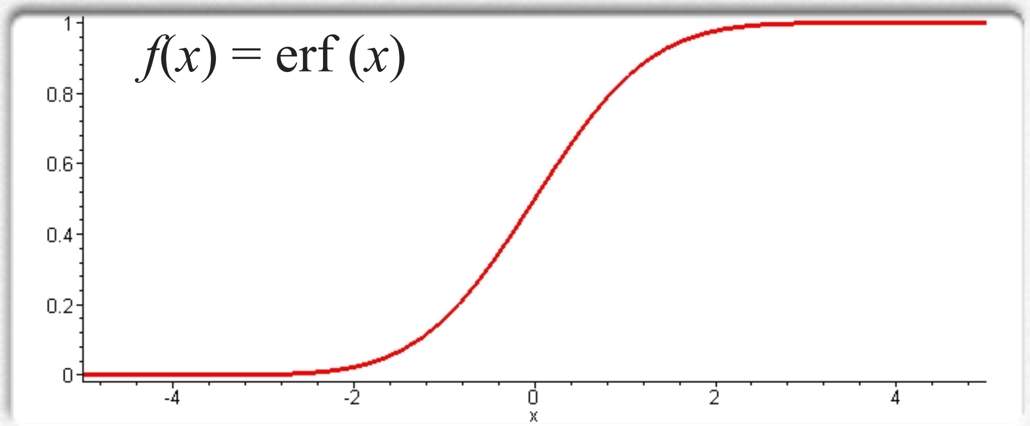
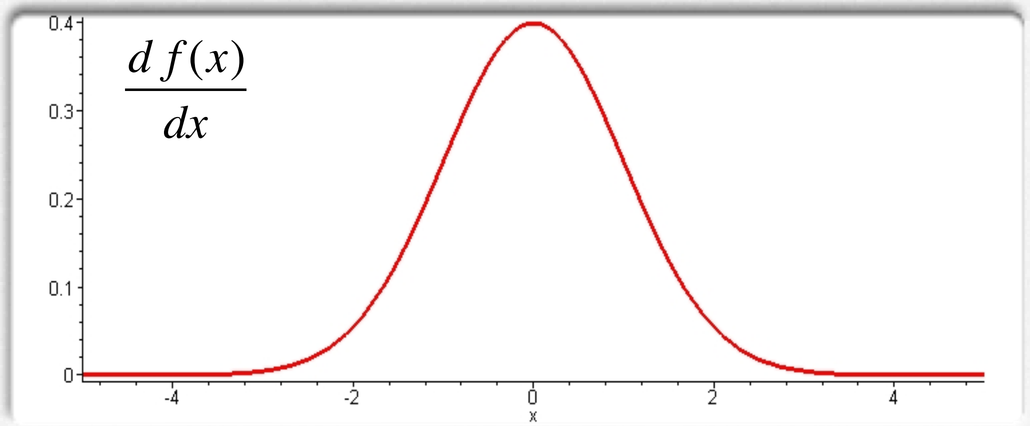
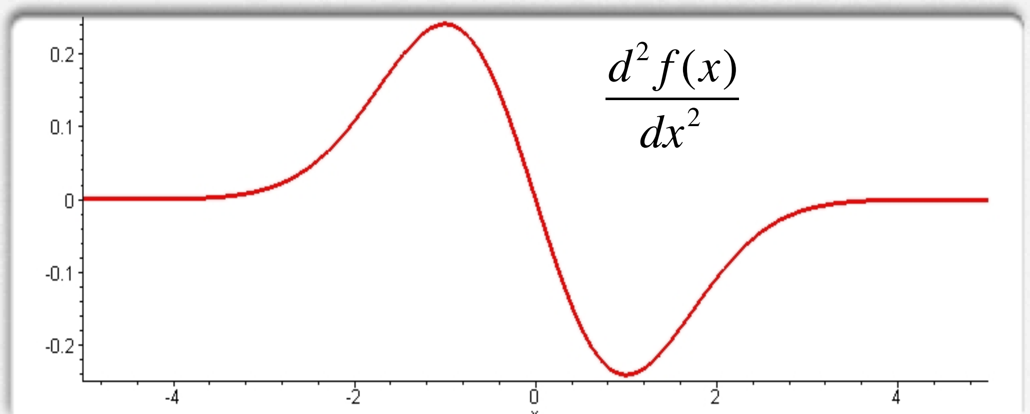


Image dérivée selon x





Deuxième dérivée selon x

Source :

3. AMÉLIORATION DE LA NETTETÉ

3. 2e dérivée de l'image

Image originale
Contour modélisé par erf(x)

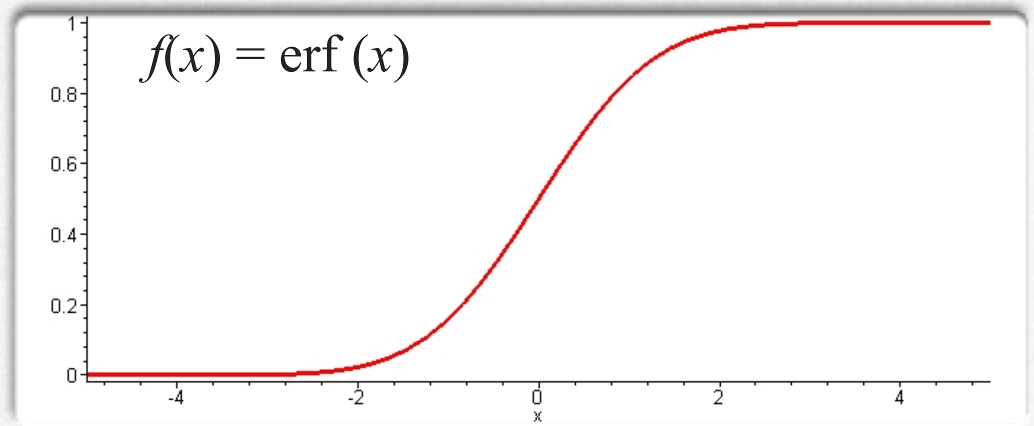
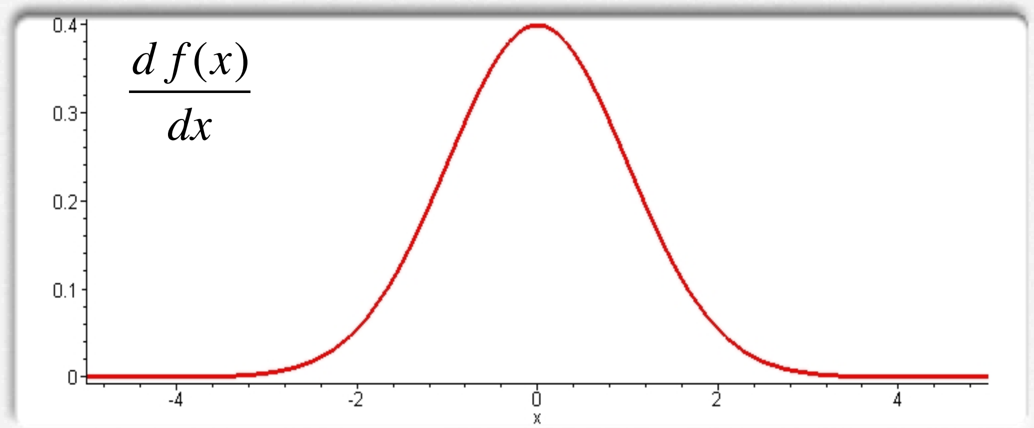
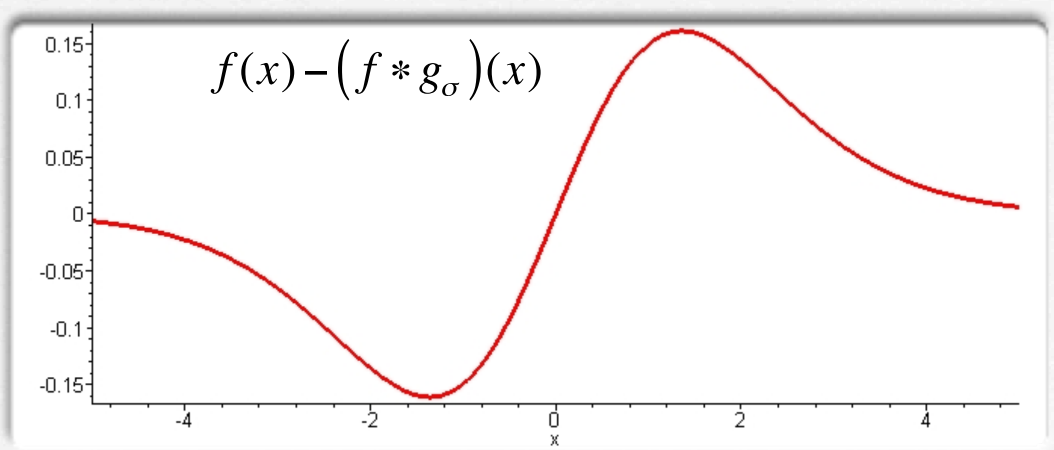
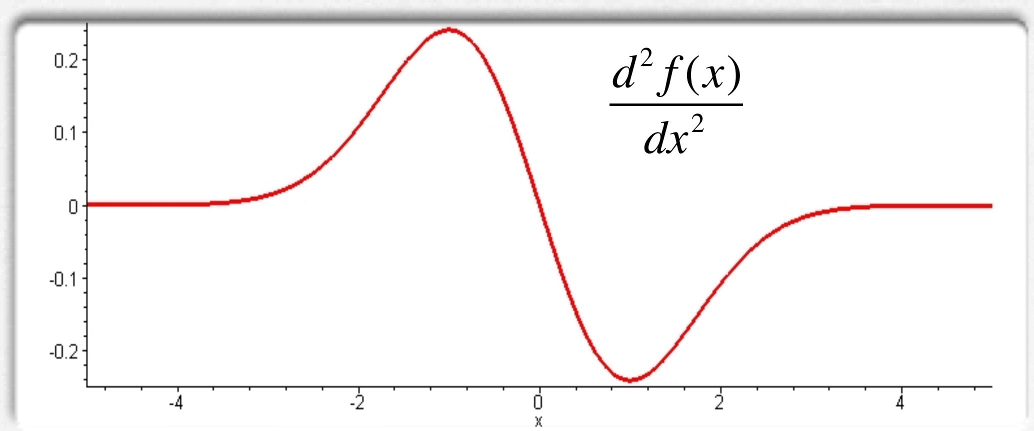


Image dérivée selon x







Deuxième dérivée selon x

Source :

3. AMÉLIORATION DE LA NETTETÉ

3. 2e dérivée de l'image

Image originale
Contour modélisé par erf(x)

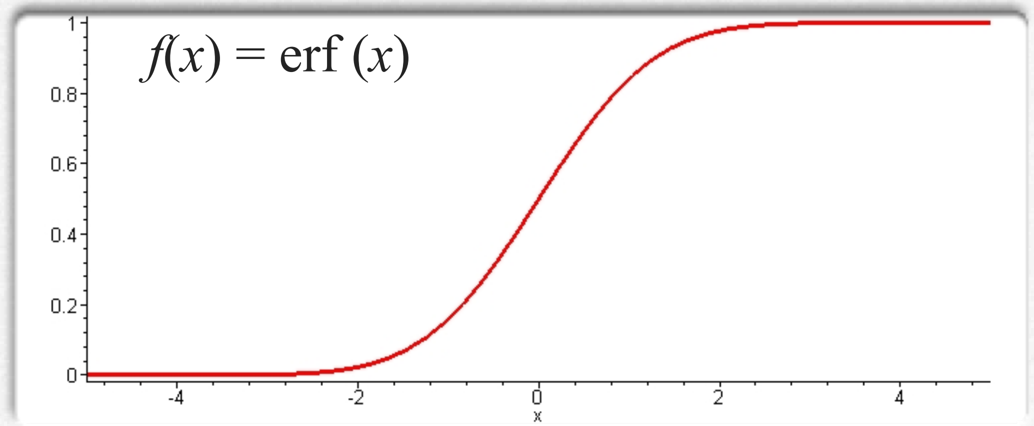
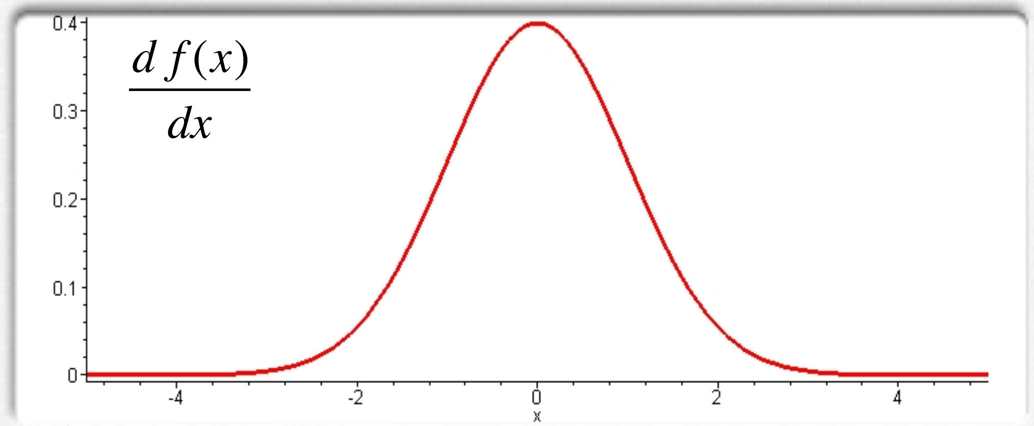
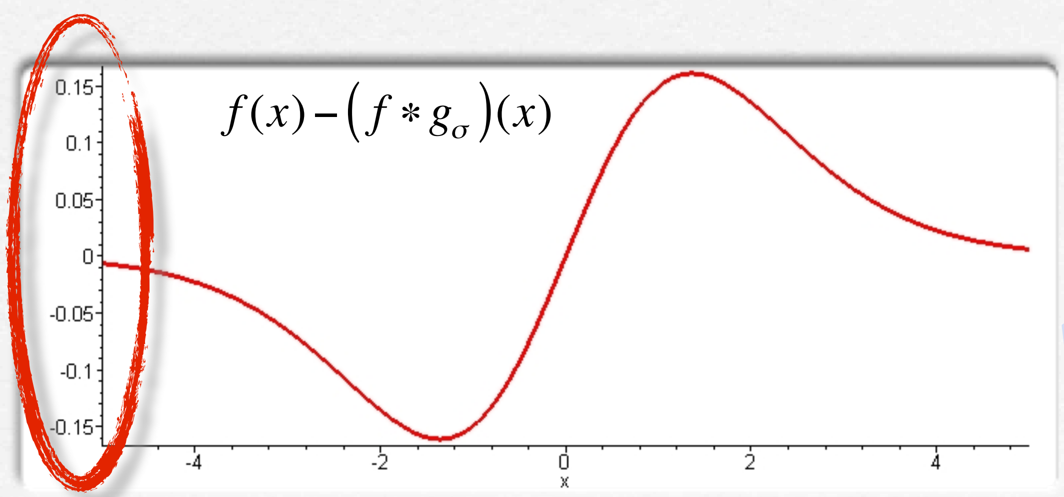
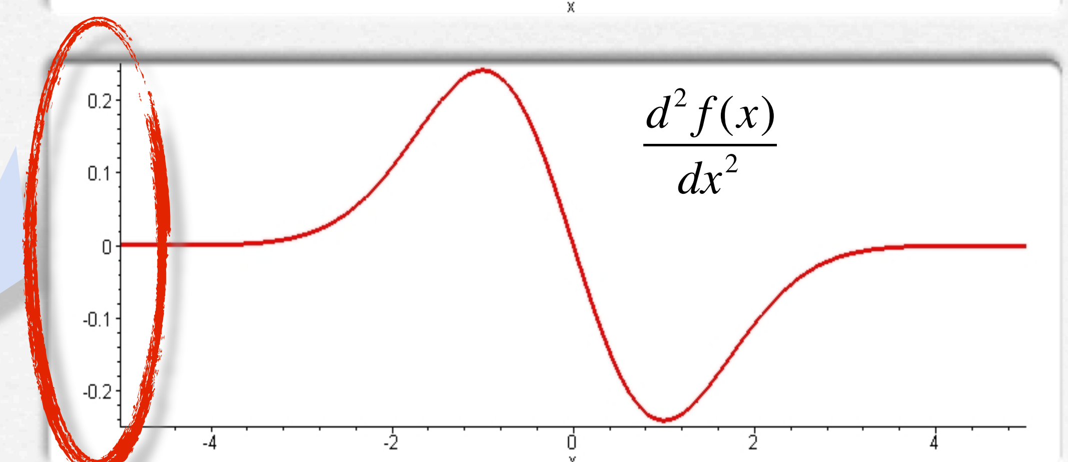


Image dérivée selon x







Deuxième dérivée selon x

3. AMÉLIORATION DE LA NETTETÉ

3. 2e dérivée de l'image

Image originale
Contour modélisé par erf(x)

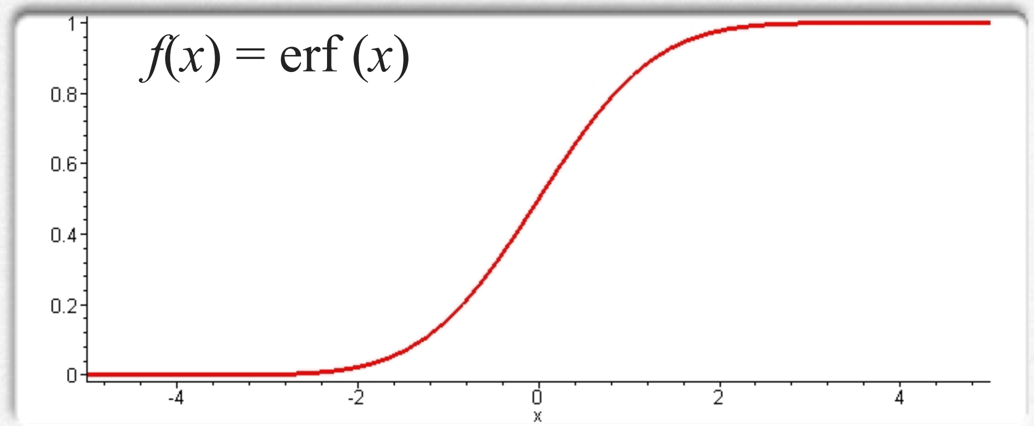
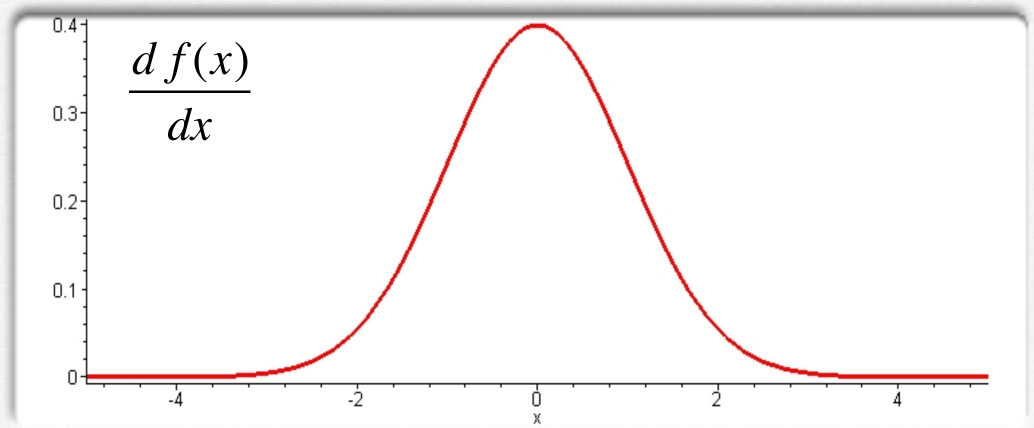
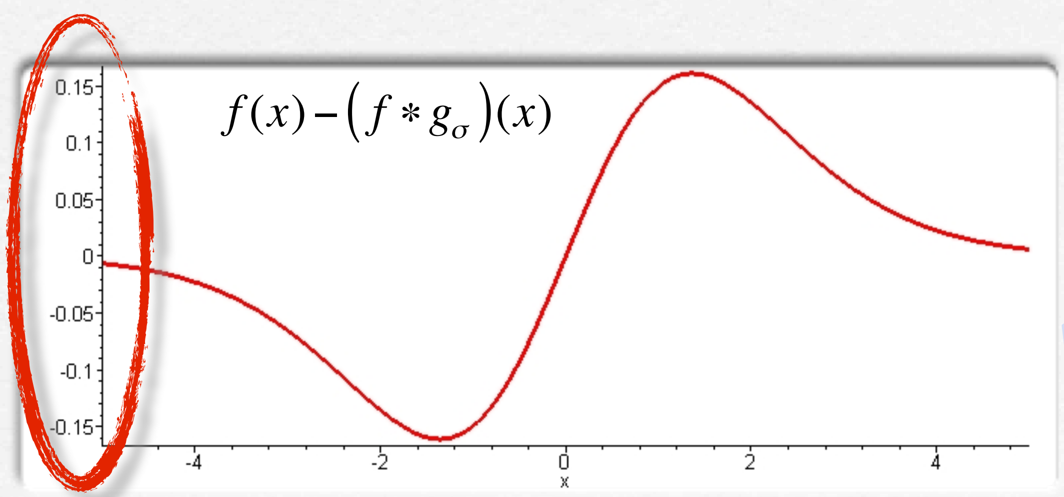
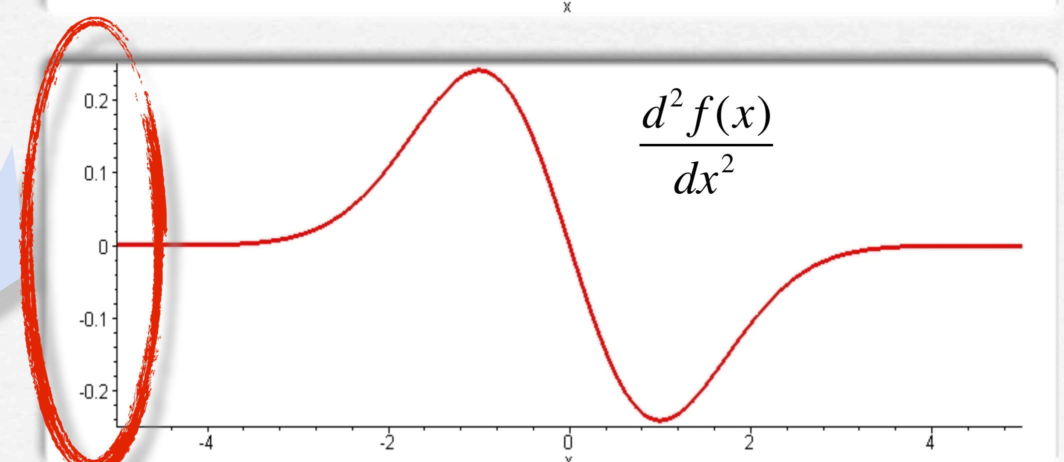


Image dérivée selon x





$$\Rightarrow \frac{d^2 f(x)}{dx^2} = A\left(f(x)-\left(f*g_\sigma\right)(x)\right)$$



Deuxième dérivée selon x

Source :

3. Amélioration de la netteté

3. 2e dérivée de l'image

- Utilisation du Laplacien pour rehausser la netteté

Image originale : $f[m, n]$

Source : DIP 2

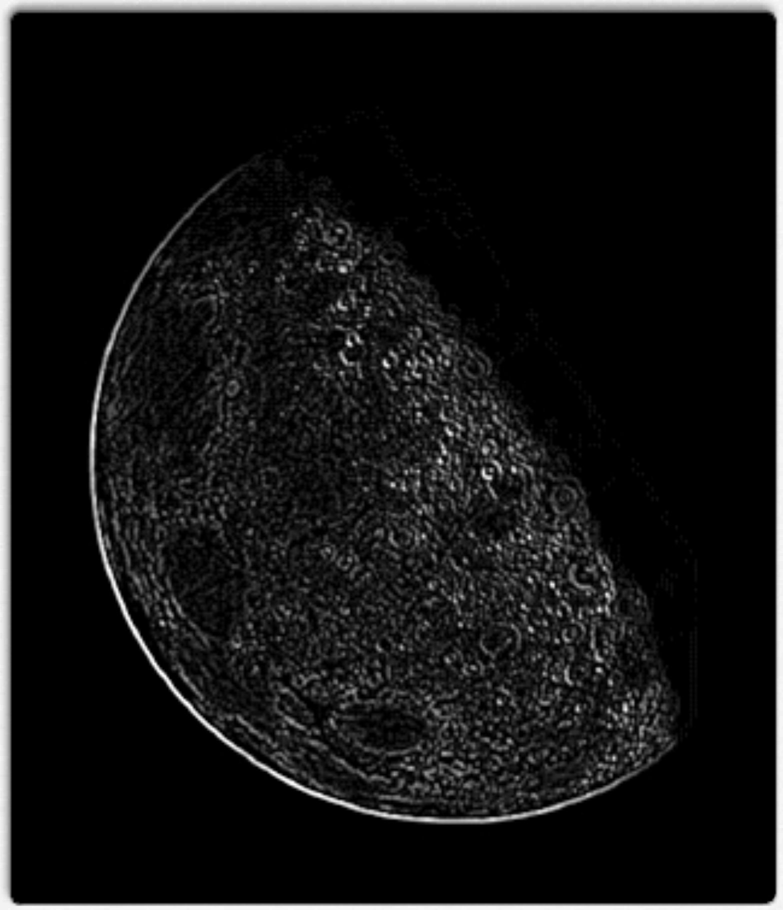
3. Amélioration de la netteté

3. 2e dérivée de l'image

- Utilisation du Laplacien pour rehausser la netteté

Laplacien : $\nabla^2 f[m,n] = \dfrac{\partial^2 f[m,n]}{\partial m^2} + \dfrac{\partial^2 f[m,n]}{\partial n^2}$

Image originale : $f[m, n]$



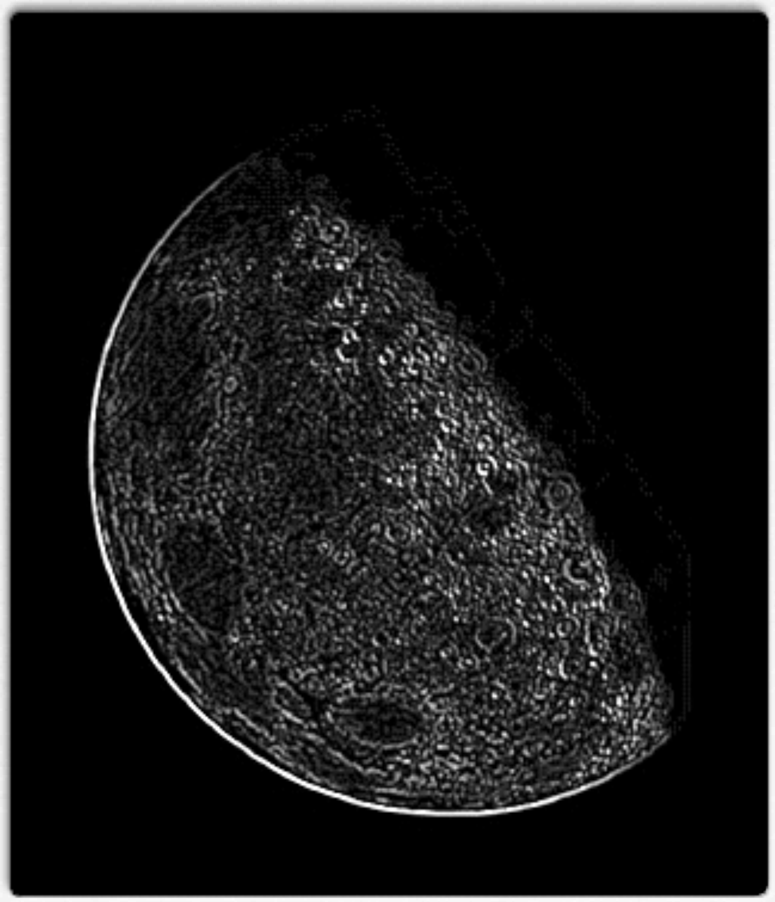
(valeur ***absolue***, dynamique rehaussée)

Source : DIP 2

3. Amélioration de la netteté

3. 2e dérivée de l'image

- Utilisation du Laplacien pour rehausser la netteté



$$\text{Laplacien} : \nabla^2 f[m,n] = \frac{\partial^2 f[m,n]}{\partial m^2} + \frac{\partial^2 f[m,n]}{\partial n^2}$$

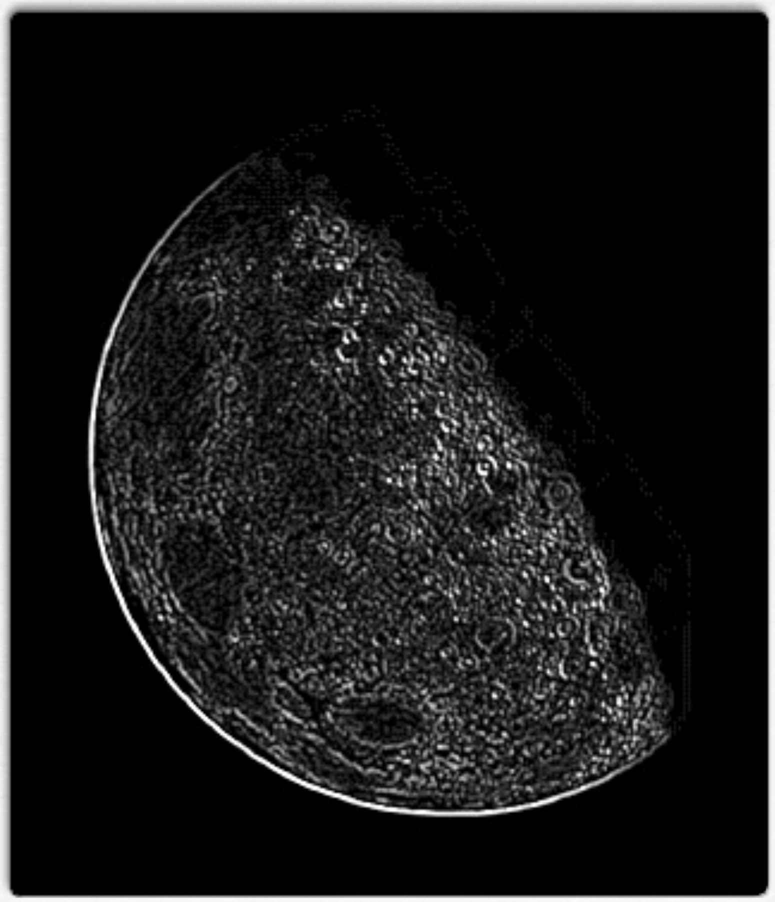
(valeur ***absolue***, dynamique rehaussée)

Source : DIP 2

3. AMÉLIORATION DE LA NETTETÉ

3. 2e dérivée de l'image

- Utilisation du Laplacien pour rehausser la netteté



Laplacien avec changement de dynamique pour fins de visualisation. Le 0 est au niveau de gris 128

$$\text{Laplacien} : \nabla^2 f[m,n] = \frac{\partial^2 f[m,n]}{\partial m^2} + \frac{\partial^2 f[m,n]}{\partial n^2}$$

(valeur ***absolue***, dynamique rehaussée)

Source : DIP 2

3. AMÉLIORATION DE LA NETTETÉ

3. 2e dérivée de l'image

- Utilisation du Laplacien pour rehausser la netteté

Image originale : $f[m, n]$

Source : DIP 2

3. Amélioration de la netteté

3. 2e dérivée de l'image

- Utilisation du Laplacien pour rehausser la netteté

Image originale : $f[m, n]$

Image rehaussée : $f[m,n]-\nabla^2 f[m,n]$

Source : DIP 2

3. AMÉLIORATION DE LA NETTETÉ

3. 2e dérivée de l'image

- Utilisation du Laplacien pour rehausser la netteté

Image originale : $f[m, n]$

Source :

3. AMÉLIORATION DE LA NETTETÉ

3. 2e dérivée de l'image

- Utilisation du Laplacien pour rehausser la netteté

Image originale : $f[m, n]$

Image rehaussée $f * \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$

Source :

3. AMÉLIORATION DE LA NETTETÉ

3. 2e dérivée de l'image

- Utilisation du Laplacien pour rehausser la netteté

Image originale : $f[m, n]$

Image rehaussée $f * \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4.5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$

Source :

3. AMÉLIORATION DE LA NETTETÉ

3. 2e dérivée de l'image

- Utilisation du Laplacien pour rehausser la netteté

Image originale : $f[m, n]$

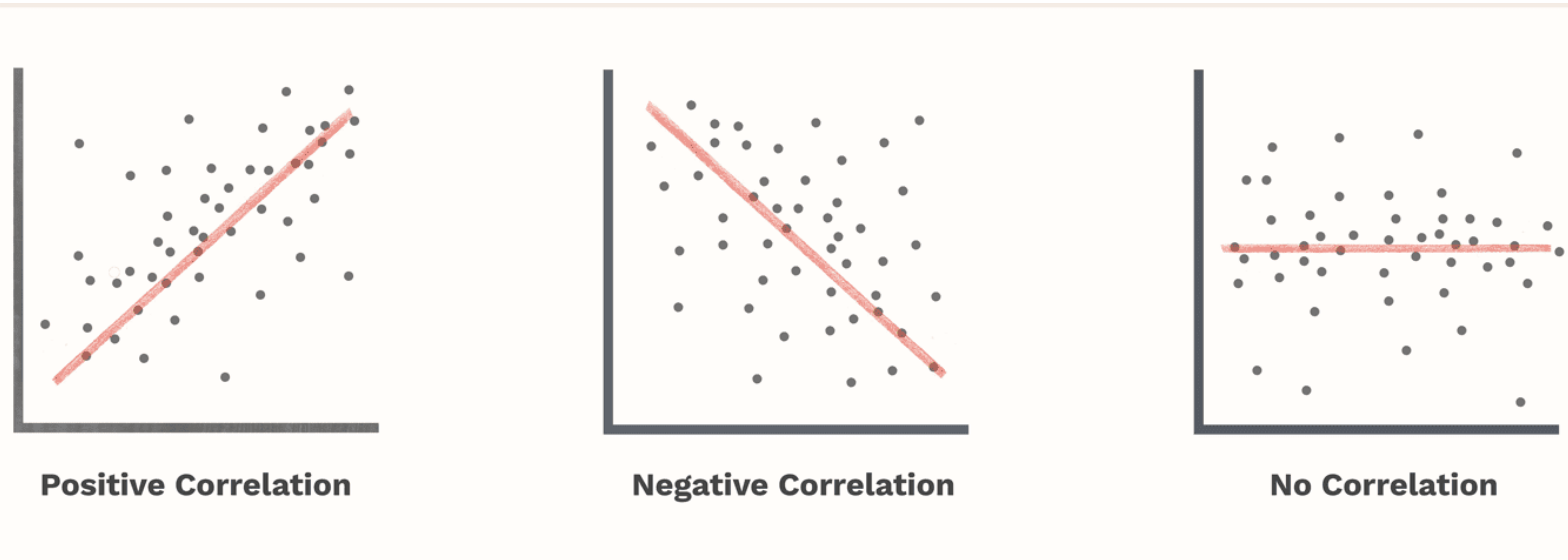
Image rehaussée $f * \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5.5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$

Source :

PLAN

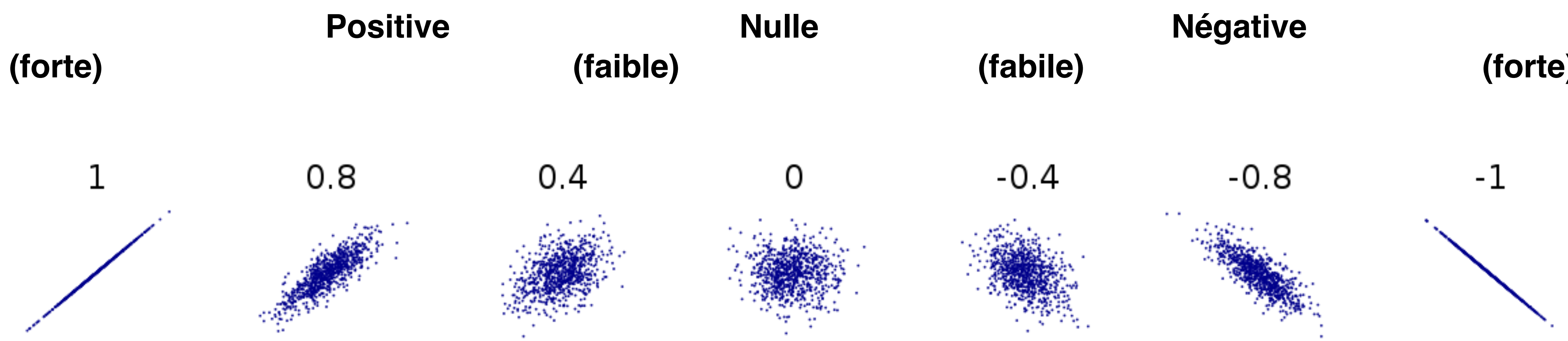
1. Rehaussement du contraste
2. Réduction du bruit
3. Amélioration de la qualité
4. Corrélation

Corrélation



Corrélation

- Indique si le nuage de point est bien aligné



- Rappel: n'indique PAS la pente de la droite!
(sauf le signe)

4. Corrélation

- *Exemple* 1:

Question :

- Est-ce que cette image contient  ?

Source : DIP2

4. Corrélation

- *Exemple* 1:

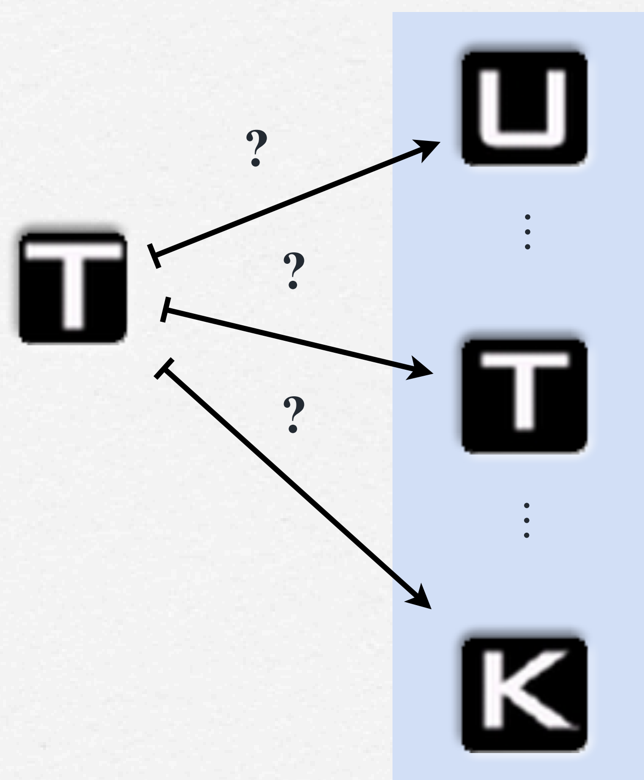
Question :

- Est-ce que cette image contient  ?

- *Exemple* 2 :

Question :

- Est-ce que cette image est dans la banque d'images?



Source : DIP2

4. CORRÉLATION

2. 2D

- Implantation directe de la corrélation

Patron à rechercher : *h*

Image : *f*

Source : DIP2

4. Corrélation

2. 2D

- Implantation directe de la corrélation

Patron à rechercher : h

Image : f

f remplie avec des 0

Source : DIP2

4. CORRÉLATION

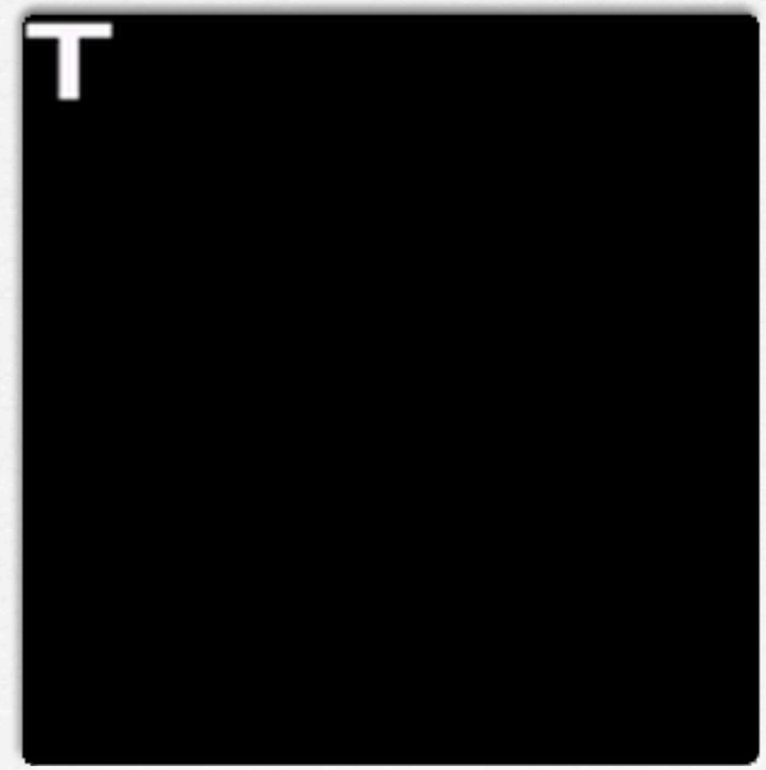
2. 2D

- Implantation directe de la corrélation

Patron à rechercher : *h*

Image : *f*

f remplie avec des 0



h remplie avec des 0

Source : DIP2

4. Corrélation

2. 2D

Image de corrélation : C

Source : DIP2

4. Corrélation

2. 2D

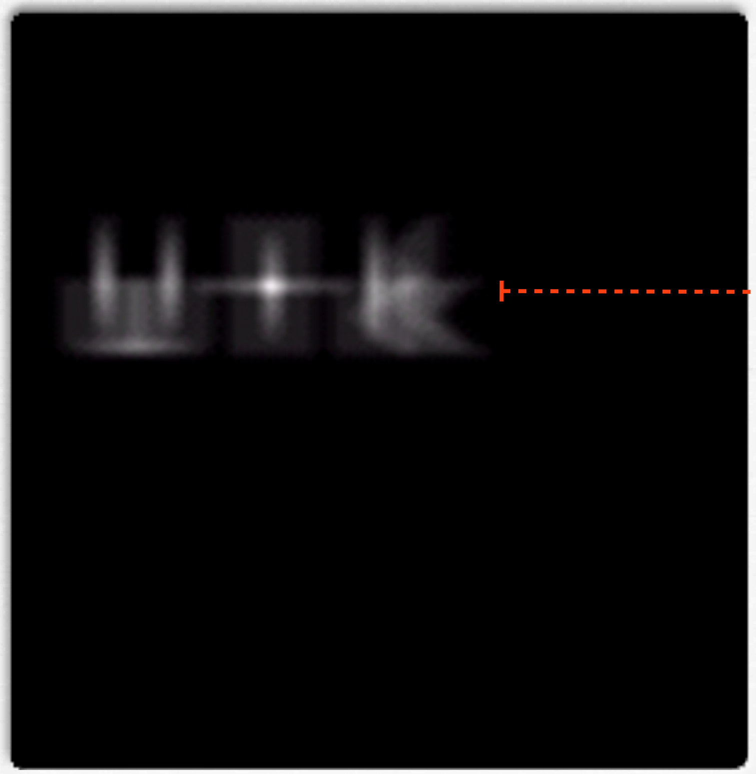
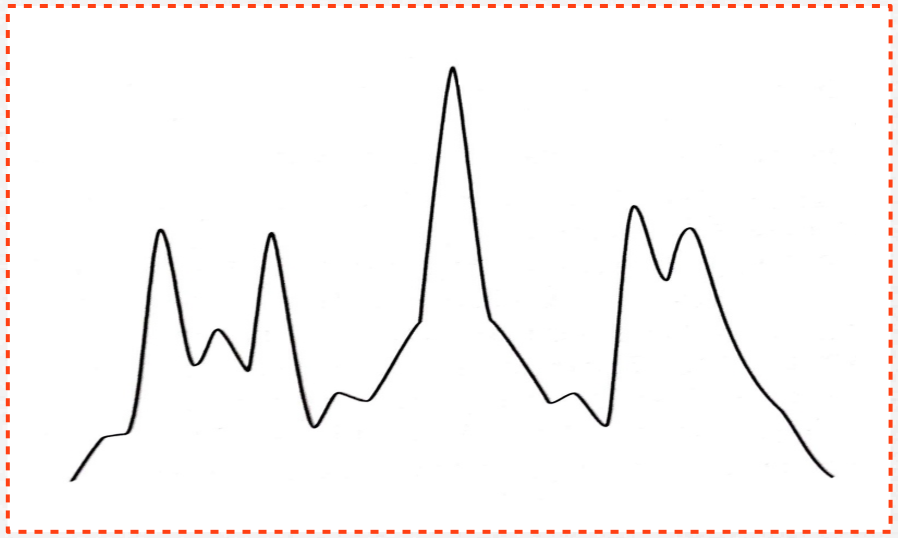


Image de corrélation : C



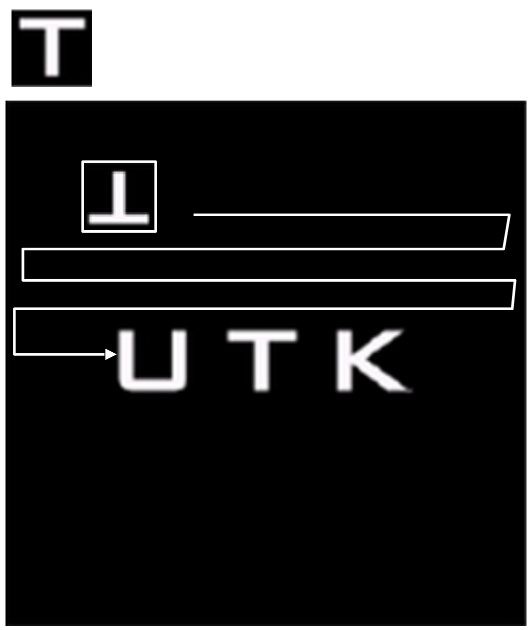
Une ligne de C
où la fonction est
maximale

Source : DIP2

Corrélation

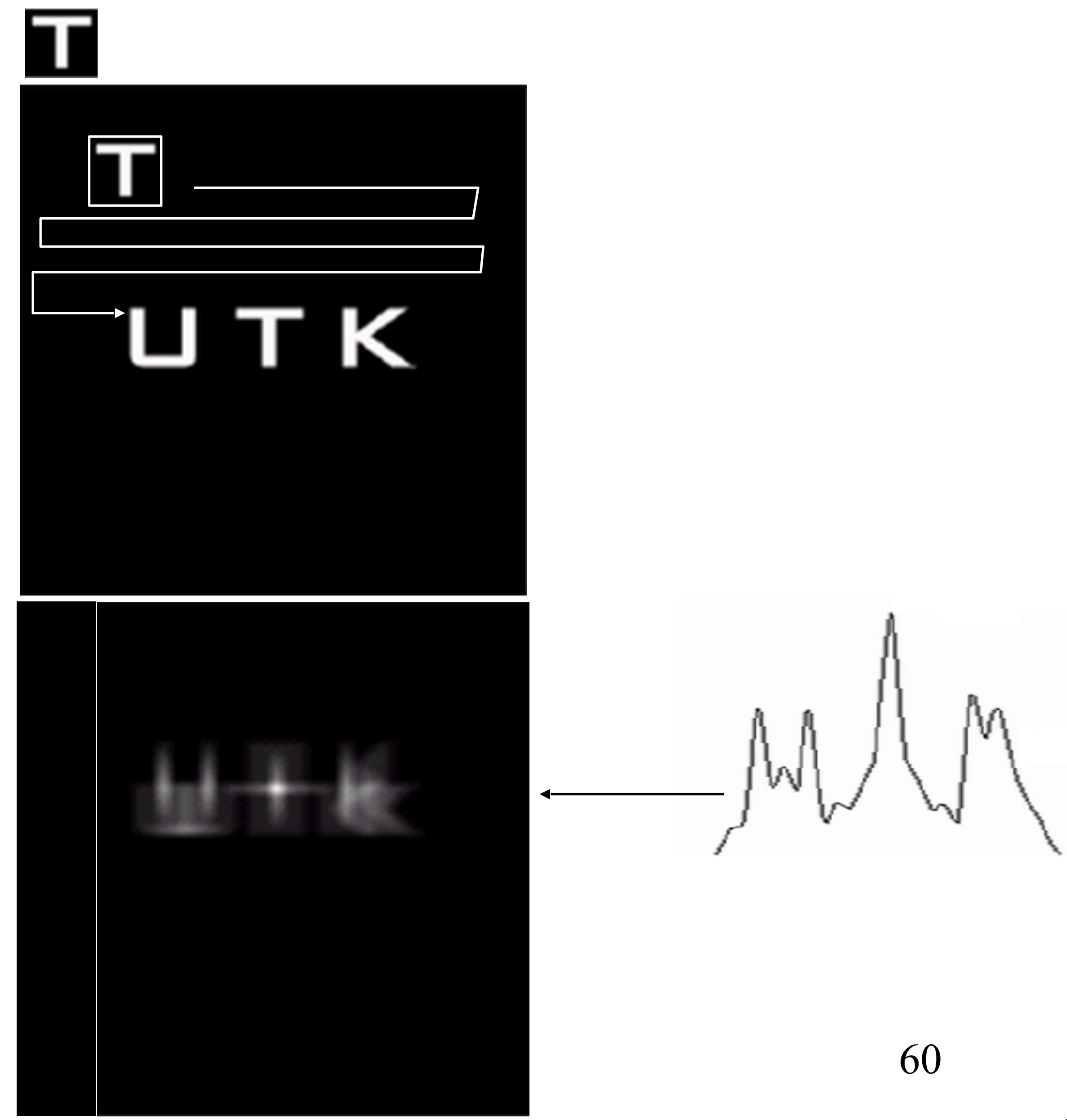
Convolution

$$(f * h)(x,y) = \sum_r \sum_t f(t,r)h(x-t, y-r)$$

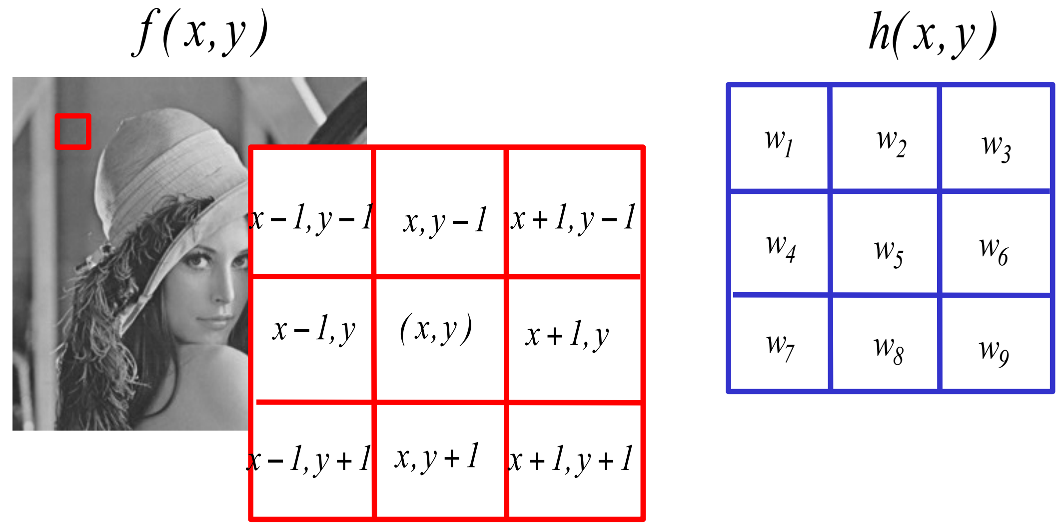


Corrélation

$$(f \circ h)(x,y) = \sum_r \sum_t f(t,r)h(x+t, y+r)$$



Corrélation



Convolution	Corrélation
$(f*h)(x,y)=\sum_r\sum_t f(t,r)h(x-t,y-r)$	$(f\circ h)(x,y)=f(x,y)*h(-x,-y)=\sum_r\sum_t f(t,r)h(x+t,y+r)$
$(f*h)(x,y)=w_9f(x-1,y-1)+w_8f(x,y-1)+w_7f(x+1,y-1)$ $+w_6f(x-1,y)\quad+w_5f(x,y)\quad+w_4f(x+1,y)$ $+w_3f(x-1,y+1)+w_2f(x,y+1)+w_1f(x+1,y+1)$	$(f*h)(x,y)=w_1f(x-1,y-1)+w_2f(x,y-1)+w_3f(x+1,y-1)$ $+w_4f(x-1,y)\quad+w_5f(x,y)\quad+w_6f(x+1,y)$ $+w_7f(x-1,y+1)+w_8f(x,y+1)+w_9f(x+1,y+1)$

Propriétés de la corrélation:

$$(f\circ h)(x,y)\ \overset{\mathfrak{I}}{\longrightarrow}\ (\overline{F}H)(u,v)$$

4. CORRÉLATION

2. 2D

Algorithme 2.2 : Corrélation directe

Entrées : f $(M \times N)$ **et** h $(M_h \times N_h)$ **2 images** // **cas** $M_f \gg M_h$ **et** $N_f \gg N_h$

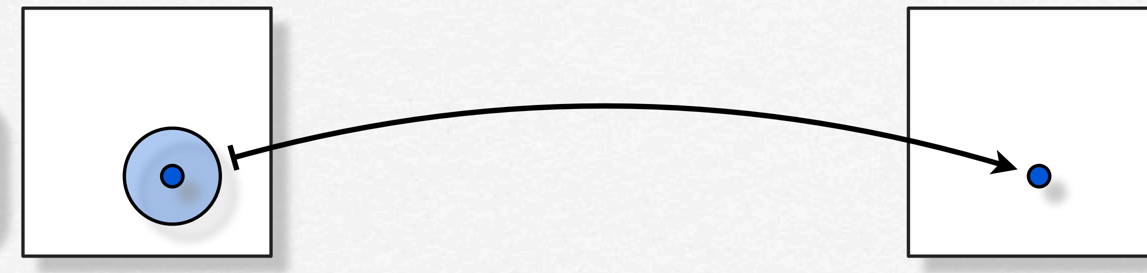
Sortie : $R(M \times N)$ → **Image du résultat de la corrélation**

```
Pour tous [m, n]                                      // tous les pixels
  R[m, n] = 0                                         // Initialiser le résultat à 0
Pour tous m de Mh/2 à Mf–Mh/2                         // tous les pixels de f sauf les bords
  Pour tous n de Nh/2 à Nf–Nh/2
    Pour tous k de –Mh/2 à Mh/2                       // tous les pixels de h
      Pour tous l de –Nh/2 à Nh/2
        R[m, n] += f[m + k, n + l] h[Mh/2 + k, Nh/2 + l]
    R[m, n] = R[m, n]/(Mh × Nh)                       // division par le nombre de pixels dans le
                                                         masque h
```

4. Corrélation

2. 2D

La corrélation est une opération locale



Algorithme 2.2 : Corrélation directe

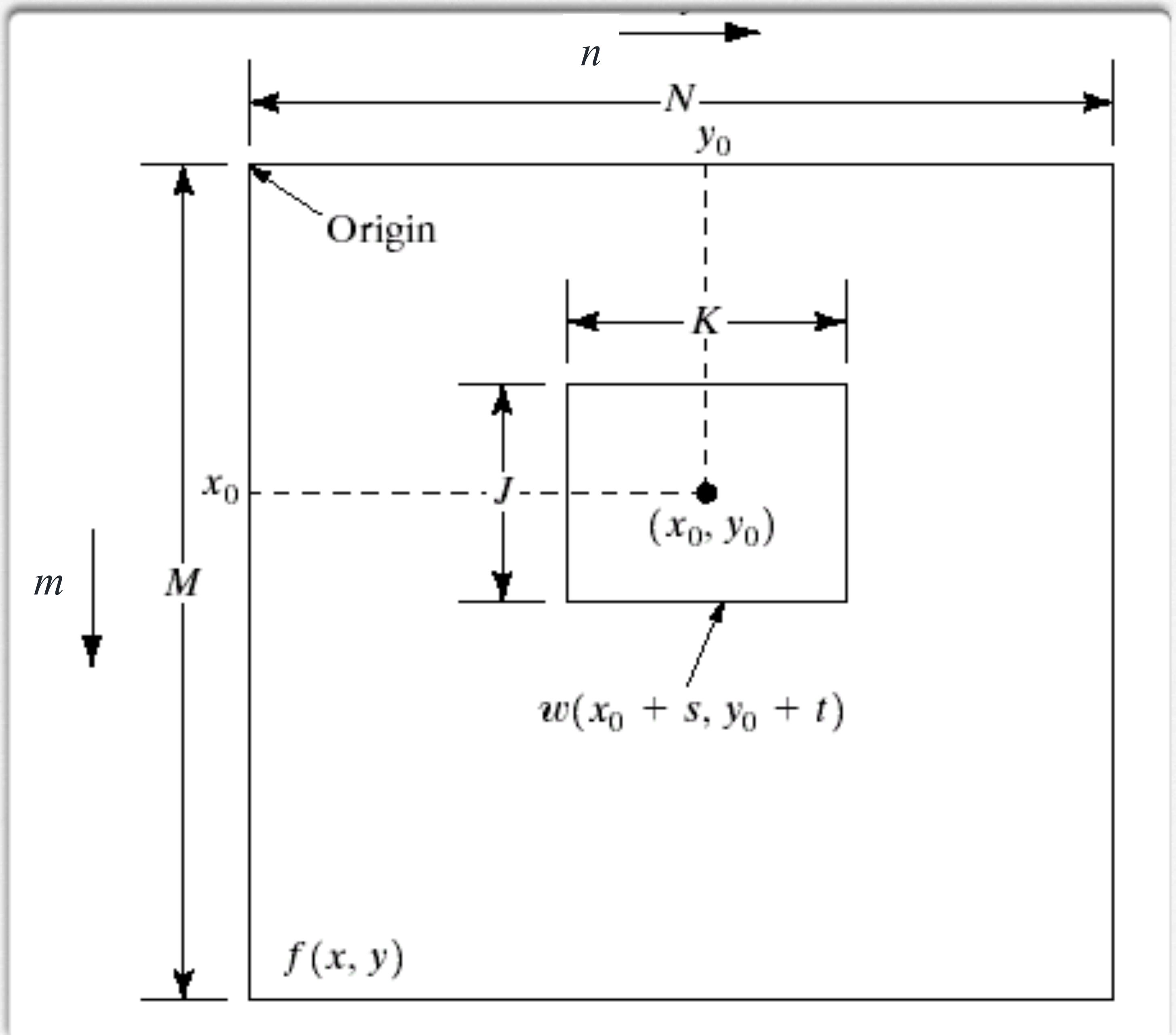
Entrées : f $(M \times N)$ **et** h $(M_h \times N_h)$ **2 images** // **cas** $M_f \gg M_h$ **et** $N_f \gg N_h$

Sortie : $R(M \times N) \rightarrow$ **Image du résultat de la corrélation**

```
Pour tous [m, n]                                        // tous les pixels
  R[m, n] = 0                                           // Initialiser le résultat à 0
Pour tous m de M_h/2 à M_f–M_h/2                        // tous les pixels de f sauf les bords
  Pour tous n de N_h/2 à N_f–N_h/2
    Pour tous k de –M_h/2 à M_h/2                       // tous les pixels de h
      Pour tous l de –N_h/2 à N_h/2
        R[m, n] += f[m + k, n + l] h[M_h/2 + k, N_h/2 + l]
    R[m, n] = R[m, n]/(M_h × N_h)                       // division par le nombre de pixels dans le
                                                           masque h
```

4. CORRÉLATION

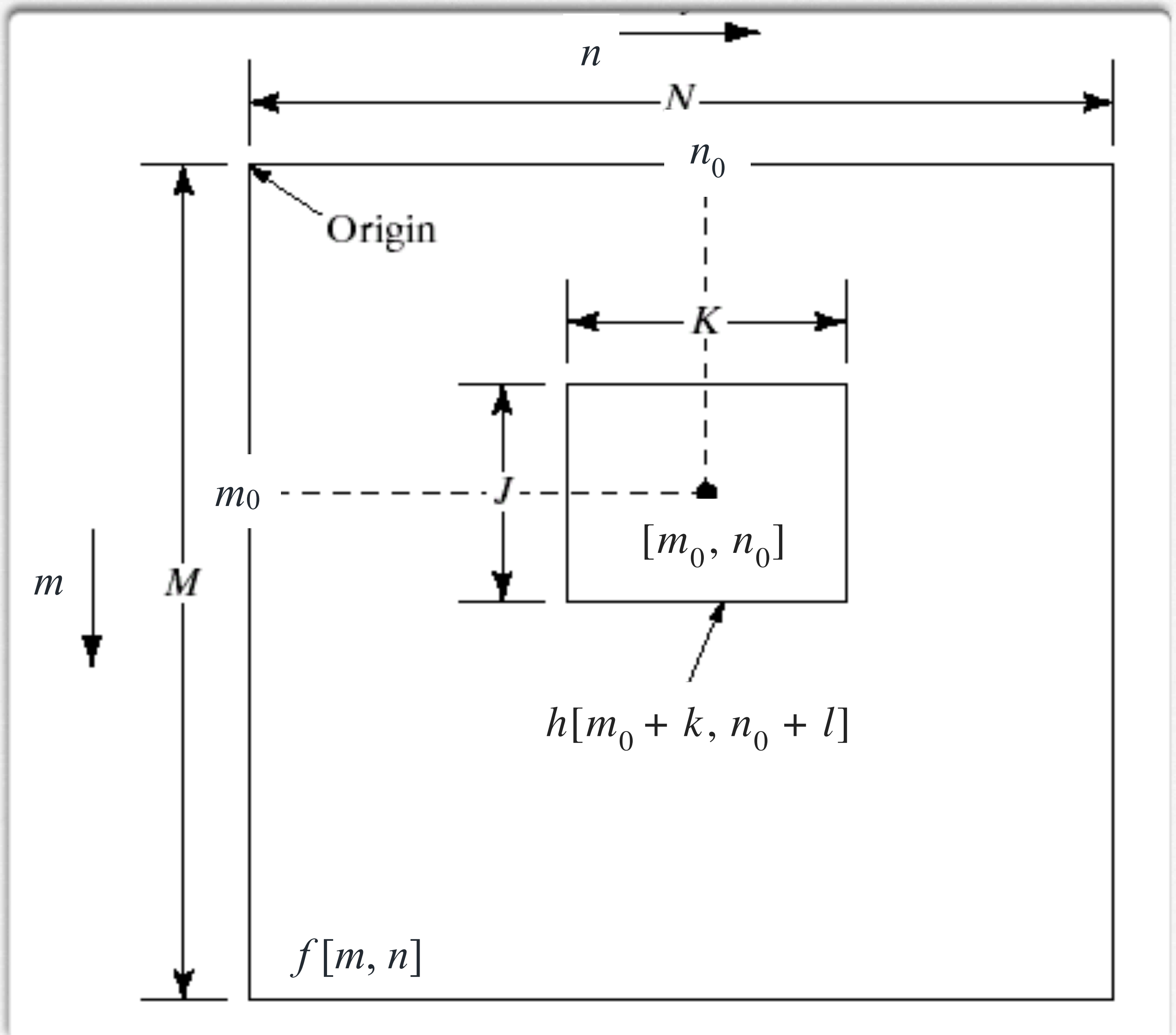
2. 2D



Source : DIP2

4. CORRÉLATION

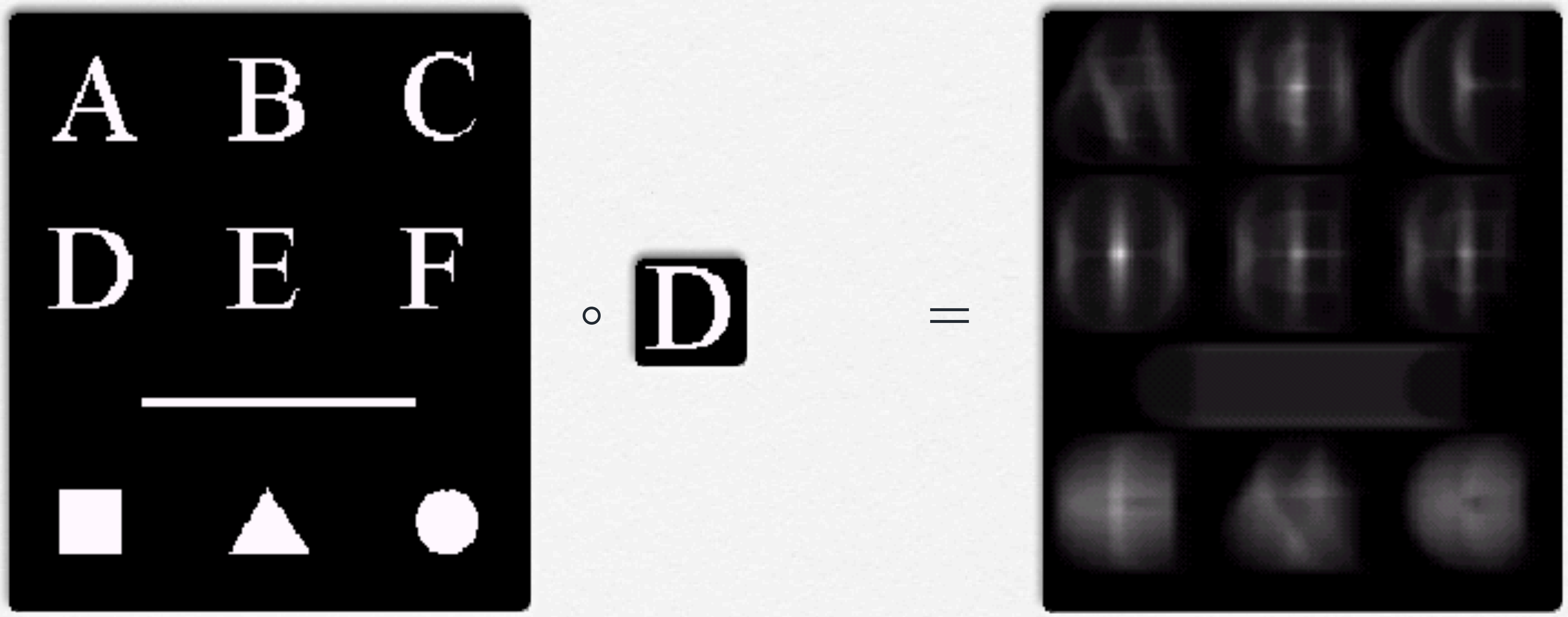
2. 2D



Source : DIP2

4. Corrélation

2. 2D



Source : DIP2

4. Corrélation

1. 1D

a) continue

- Posons 2 signaux 1D $f(t)$ et $h(t)$

$$\left(f \circ h\right)(t) = \int_{-\infty}^{\infty} f(x)h(x+t)\,dx$$

est la corrélation

Source :

4. Corrélation

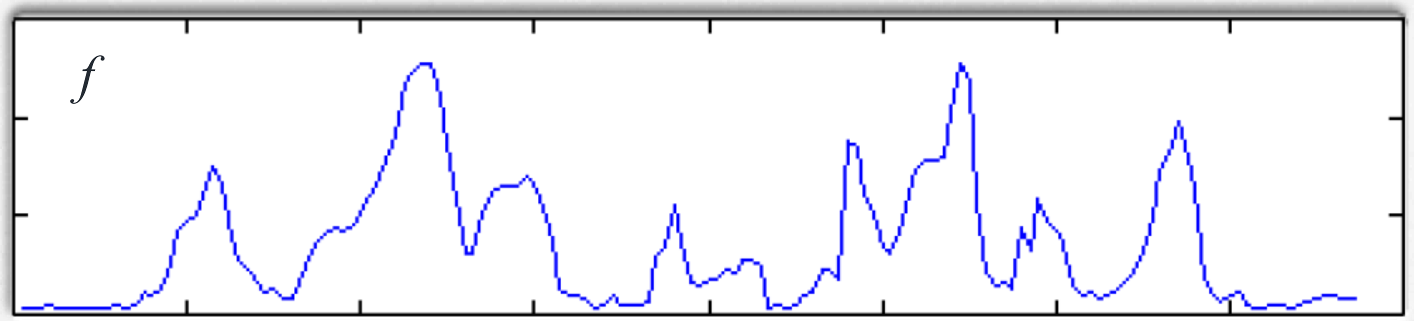
1. 1D

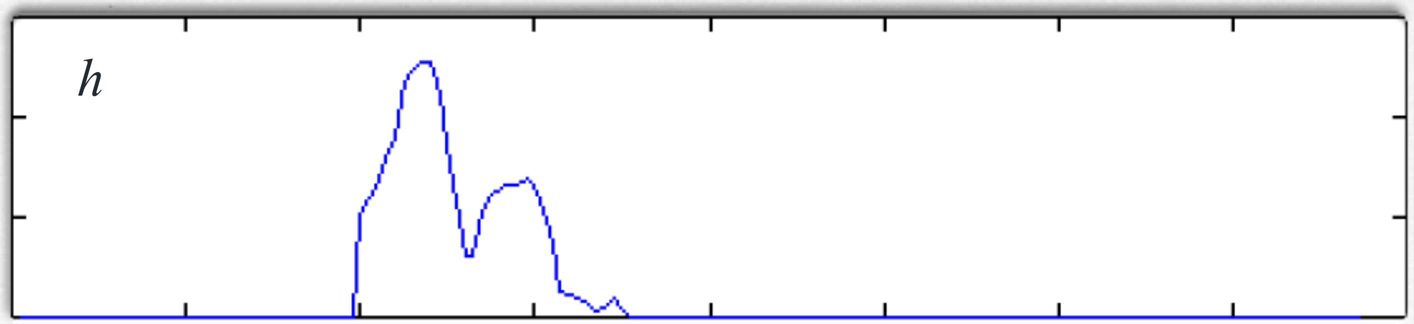
a) continue

- Posons 2 signaux 1D $f(t)$ et $h(t)$

$$(f \circ h)(t) = \int_{-\infty}^{\infty} f(x)h(x+t)\,dx$$

est la corrélation





Source :

4. Corrélation

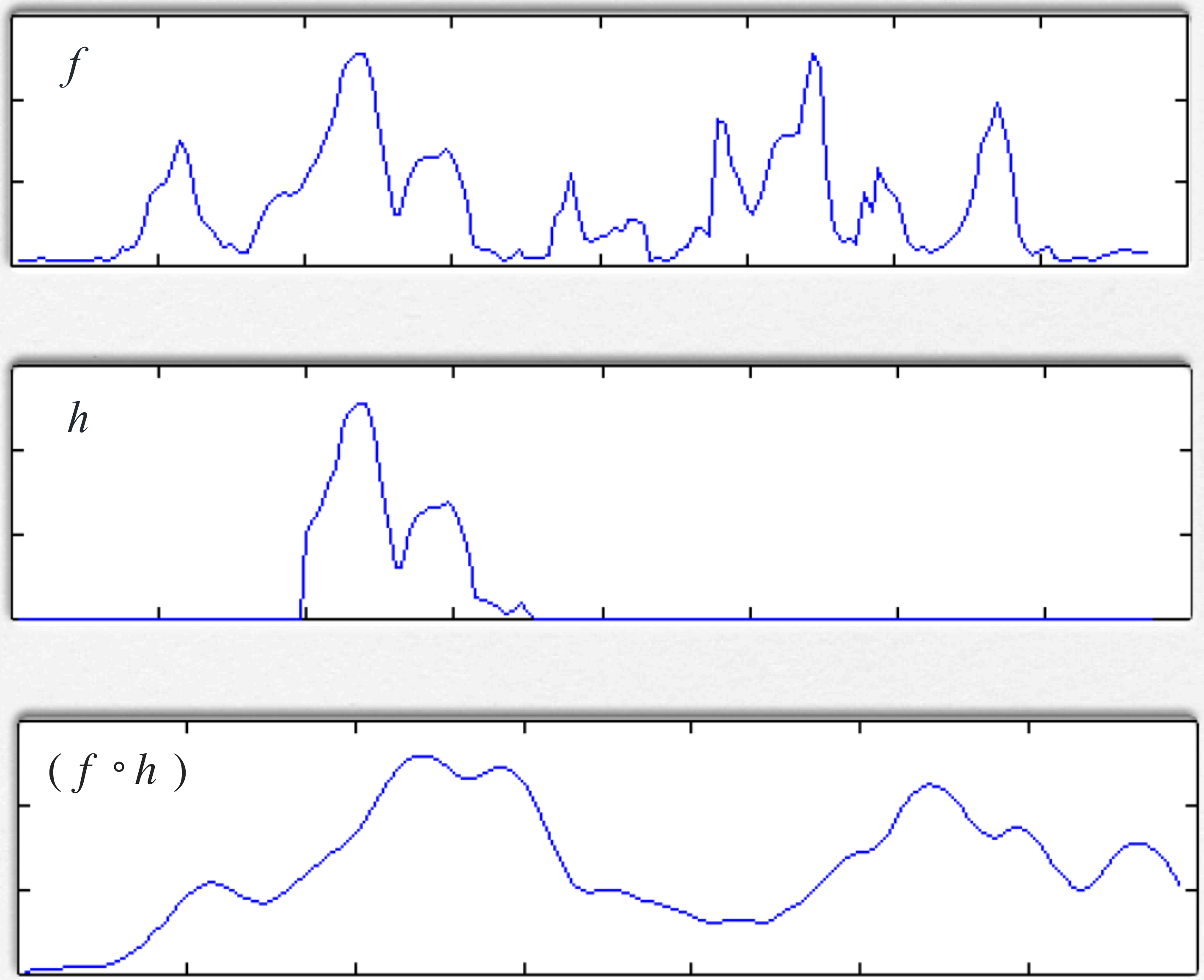
1. 1D

a) continue

- Posons 2 signaux 1D $f(t)$ et $h(t)$

$$\left(f \circ h\right)(t) = \int_{-\infty}^{\infty} f(x)h(x+t)\,dx$$

est la corrélation



4. CORRÉLATION

2. 2D

- *Exemple* d'application : repérer les cimes

Image originale
h est entouré de pointillés

Source : DIP2

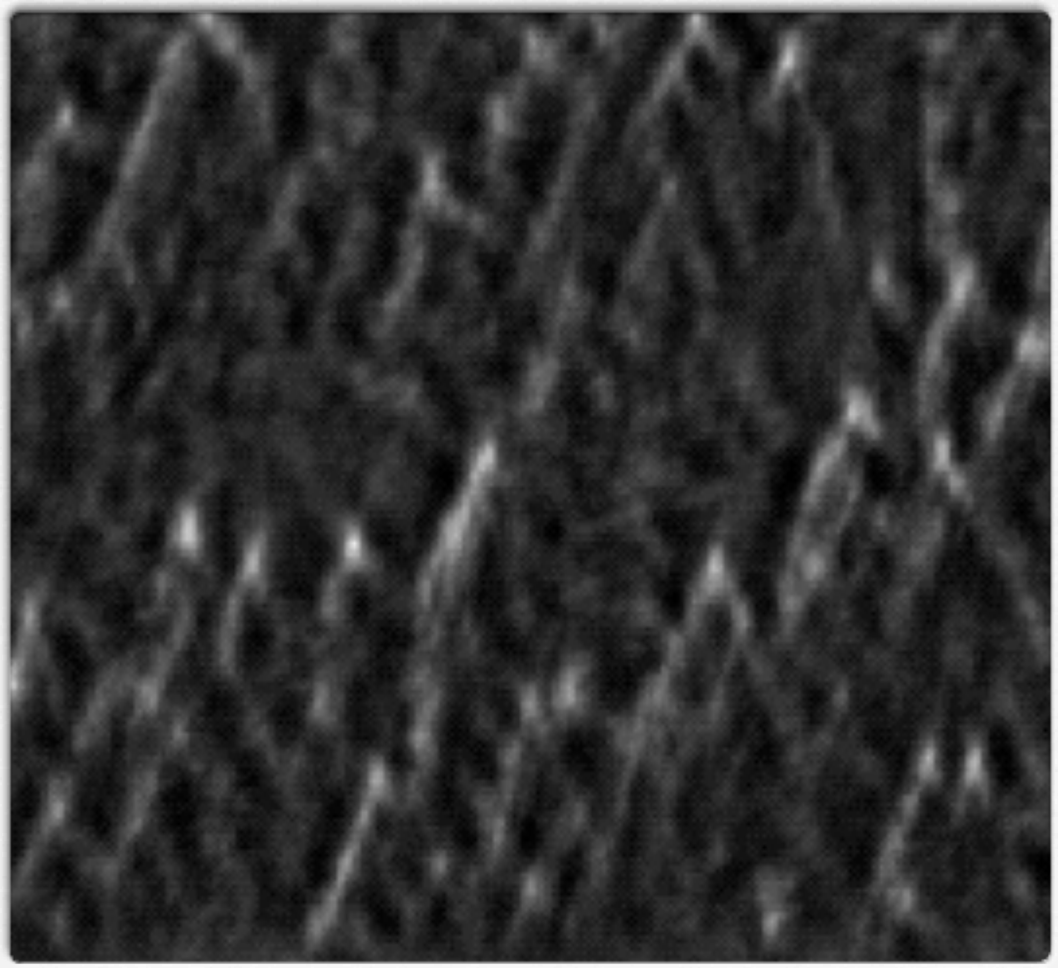
4. CORRÉLATION

2. 2D

- *Exemple* d'application : repérer les cimes

Image originale
h est entouré de pointillés

Image de corrélation



Source : DIP2

4. CORRÉLATION

2. 2D

a) Corrélation normalisée

4. Corrélation

2. 2D

a) Corrélation normalisée

- *Problème*: la corrélation est très sensible aux changement d’intensités

4. CORRÉLATION

2. 2D

a) Corrélation normalisée

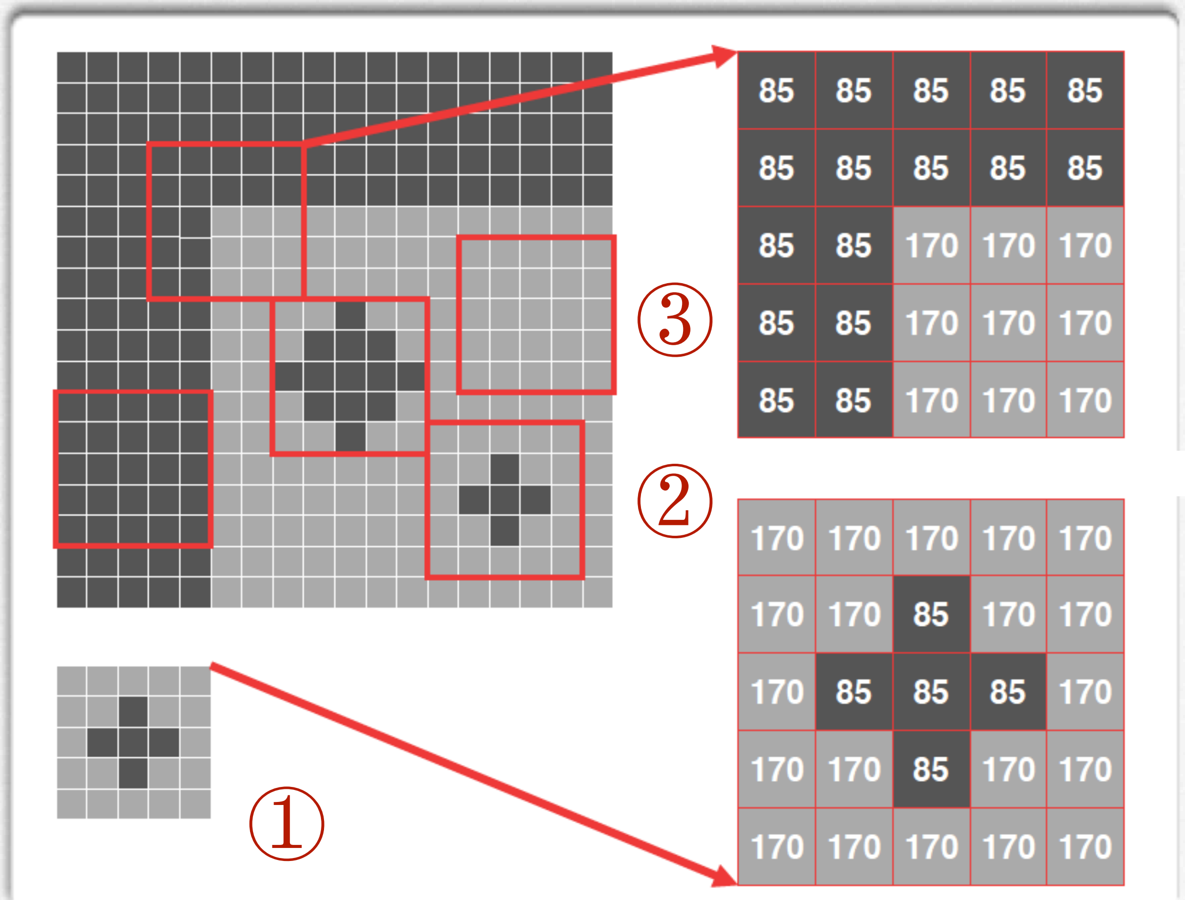
- *Problème*: la corrélation est très sensible aux changement d'intensités
  - ✓ Le résultat risque d'être maximum si l'intensité de *f* est grand.

4. Corrélation

2. 2D

a) Corrélation normalisée

- *Problème*: la corrélation est très sensible aux changement d'intensités
  - ✓ Le résultat risque d'être maximum si l'intensité de f est grand.
  - ✓ *Exemple* :



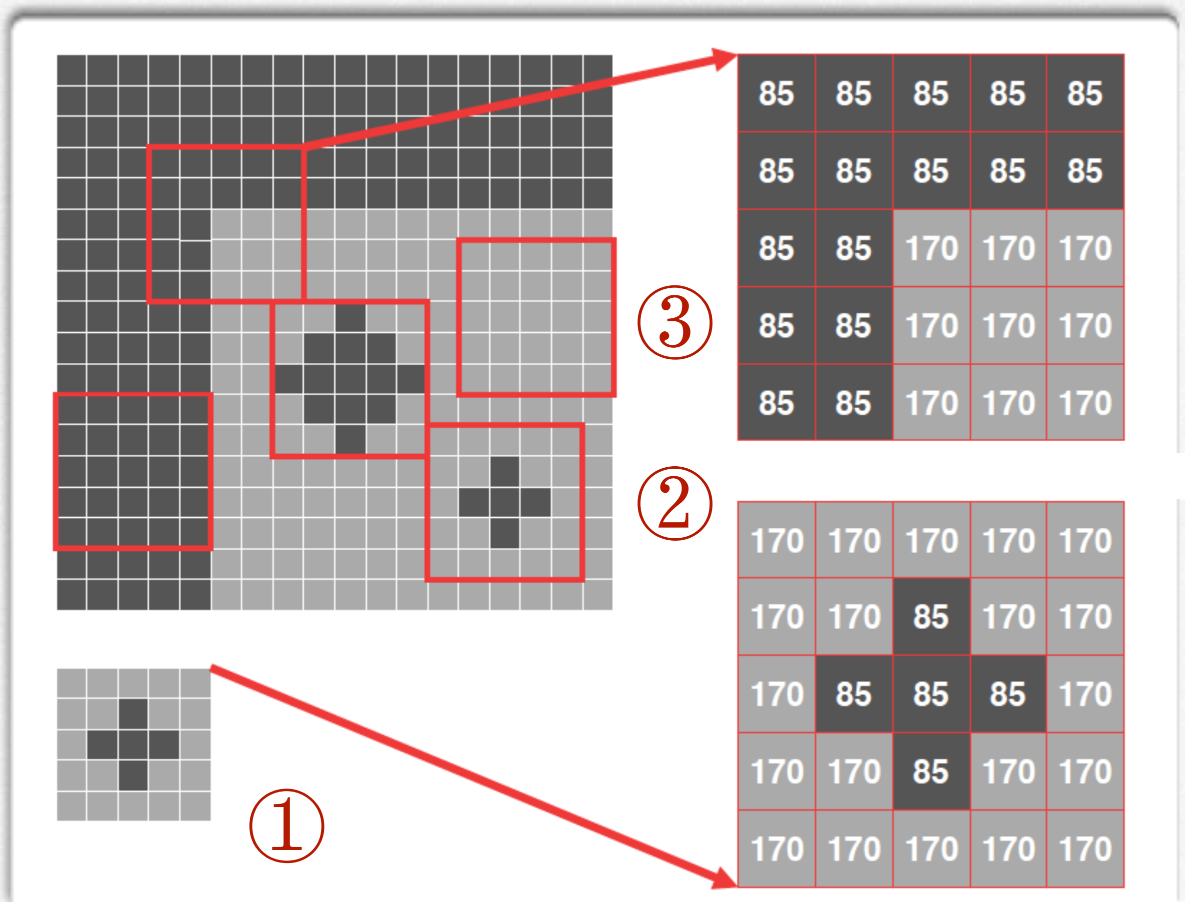
Source : N. Vandenbroucke, LISIC, france

4. Corrélation

2. 2D

a) Corrélation normalisée

- *Problème*: la corrélation est très sensible aux changement d'intensités
  - ✓ Le résultat risque d'être maximum si l'intensité de *f* est grand.
  - ✓ *Exemple* :



$$② \circ ① = 20\times170\times170 + 5\times85\times85$$

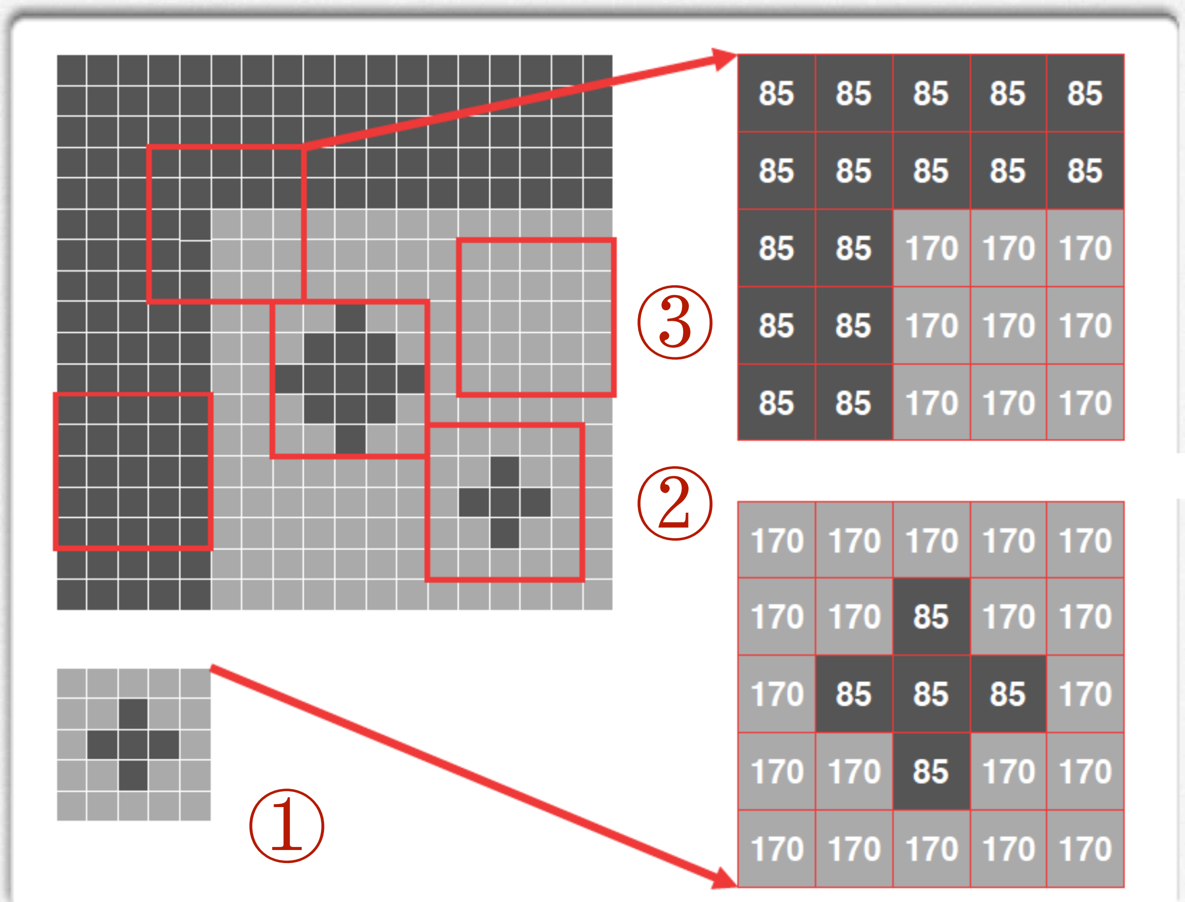
Source : N. Vandenbroucke, LISIC, france

4. Corrélation

2. 2D

a) Corrélation normalisée

- *Problème*: la corrélation est très sensible aux changement d'intensités
  - ✓ Le résultat risque d'être maximum si l'intensité de *f* est grand.
  - ✓ *Exemple* :



$$② \circ ① = 20\times170\times170 + 5\times85\times85$$

$$<$$

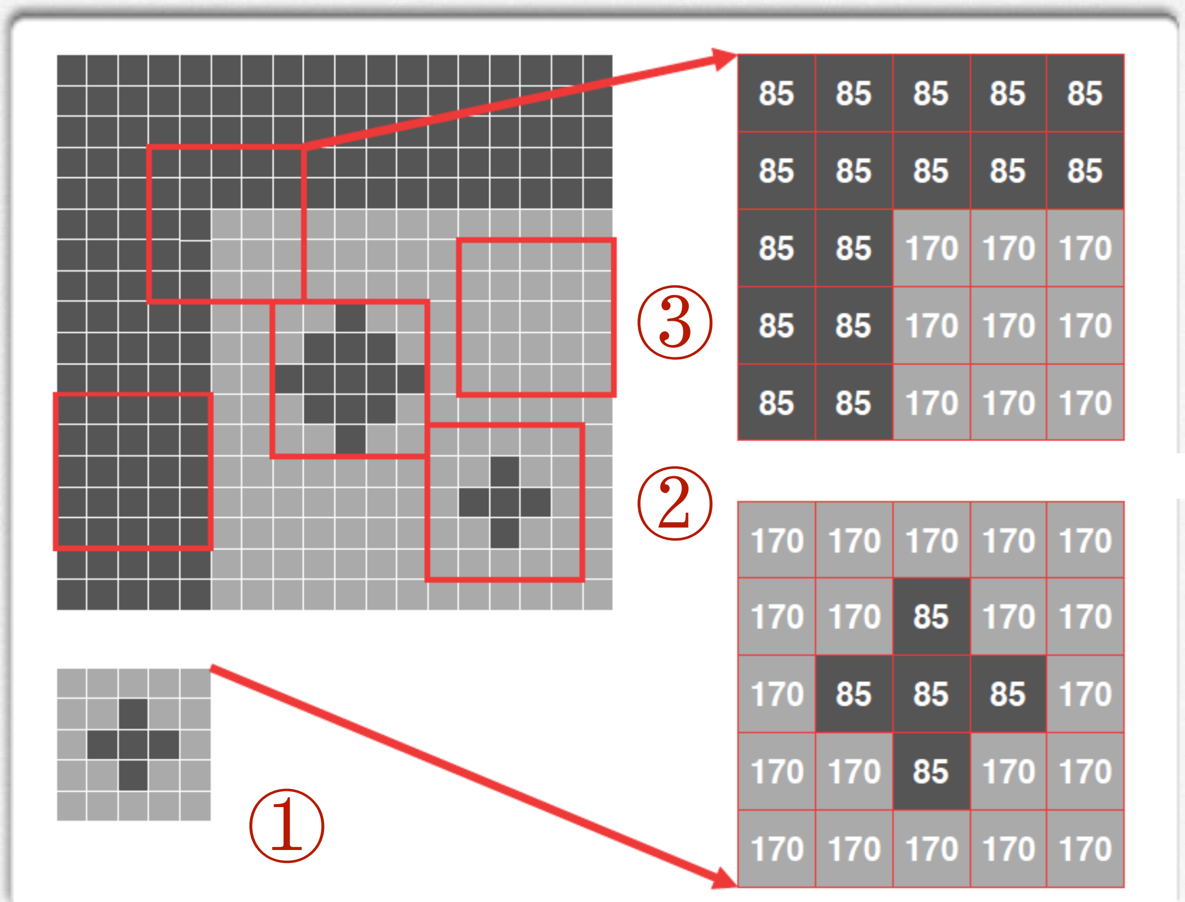
Source : N. Vandenbroucke, LISIC, france

4. Corrélation

2. 2D

a) Corrélation normalisée

- *Problème*: la corrélation est très sensible aux changement d'intensités
  - ✓ Le résultat risque d'être maximum si l'intensité de *f* est grand.
  - ✓ *Exemple* :



$$② \circ ① = 20 \times 170 \times 170 + 5 \times 85 \times 85$$

$$<$$

$$③ \circ ① = 20 \times 170 \times 170 + 5 \times 85 \times 170$$

Source : N. Vandenbroucke, LISIC, france

4. Corrélation

2. 2D

a) Corrélation normalisée

- *Solution* : on normalise la corrélation par rapport à la moyenne et la variance

$$\mu_f[m,n] = \frac{1}{M_h \cdot N_h} \sum_{k=-M_h/2}^{M_h/2} \sum_{l=-N_h/2}^{N_h/2} f[m+k,n+l]$$

$$\mu_h = \frac{1}{M_h \cdot N_h} \sum_{k=-M_h/2}^{M_h/2} \sum_{l=-N_h/2}^{N_h/2} h[k,l]$$

$$\gamma[m,n] = \left(\frac{1}{\sigma_f[m,n] \cdot \sigma_h}\right) \sum_{k=-M_h/2}^{M_h/2} \sum_{l=-N_h/2}^{N_h/2} \left(f[m+k,n+l] - \mu_f[m,n]\right) \cdot \left(h[k,l] - \mu_h\right)$$

$$\sigma_h = \left(\frac{1}{M_h \cdot n_h} \sum_k \sum_l \left(h[k,l] - \mu_h\right)^2\right)^{1/2}$$

$$\sigma_f[m,n] = \left(\frac{1}{M_h \cdot N_h} \sum_k \sum_l \left(f[m+,n+] - \mu_f[m,n]\right)^2\right)^{1/2}$$

Source :

4. Corrélation

2. 2D

a) Corrélation normalisée

- *Solution* : on normalise la corrélation par rapport à la moyenne et la variance

moyenne de h

$$\mu_f[m,n] = \frac{1}{M_h \cdot N_h} \sum_{k=-M_h/2}^{M_h/2} \sum_{l=-N_h/2}^{N_h/2} f[m+k,n+l]$$

$$\mu_h = \frac{1}{M_h \cdot N_h} \sum_{k=-M_h/2}^{M_h/2} \sum_{l=-N_h/2}^{N_h/2} h[k,l]$$

$$\gamma[m,n] = \left(\frac{1}{\sigma_f[m,n] \cdot \sigma_h}\right) \sum_{k=-M_h/2}^{M_h/2} \sum_{l=-N_h/2}^{N_h/2} \left(f[m+k,n+l] - \mu_f[m,n]\right) \cdot \left(h[k,l] - \mu_h\right)$$

$$\sigma_h = \left(\frac{1}{M_h \cdot n_h} \sum_k \sum_l \left(h[k,l] - \mu_h\right)^2\right)^{1/2}$$

$$\sigma_f[m,n] = \left(\frac{1}{M_h \cdot N_h} \sum_k \sum_l \left(f[m+,n+] - \mu_f[m,n]\right)^2\right)^{1/2}$$

Source :

4. CORRÉLATION

2. 2D

a) Corrélation normalisée

- *Solution* : on normalise la corrélation par rapport à la moyenne et la variance

moyenne de h

moyenne de f vis-à-vis h

$$\mu_f[m,n] = \frac{1}{M_h \cdot N_h} \sum_{k=-M_h/2}^{M_h/2} \sum_{l=-N_h/2}^{N_h/2} f[m+k,n+l]$$

$$\mu_h = \frac{1}{M_h \cdot N_h} \sum_{k=-M_h/2}^{M_h/2} \sum_{l=-N_h/2}^{N_h/2} h[k,l]$$

$$\gamma[m,n] = \left(\frac{1}{\sigma_f[m,n] \cdot \sigma_h}\right) \sum_{k=-M_h/2}^{M_h/2} \sum_{l=-N_h/2}^{N_h/2} \left(f[m+k,n+l] - \mu_f[m,n]\right) \cdot \left(h[k,l] - \mu_h\right)$$

$$\sigma_h = \left(\frac{1}{M_h \cdot n_h} \sum_k \sum_l \left(h[k,l] - \mu_h\right)^2\right)^{1/2}$$

$$\sigma_f[m,n] = \left(\frac{1}{M_h \cdot N_h} \sum_k \sum_l \left(f[m+,n+] - \mu_f[m,n]\right)^2\right)^{1/2}$$

Source :

4. Corrélation

2. 2D

a) Corrélation normalisée

- *Solution* : on normalise la corrélation par rapport à la moyenne et la variance

moyenne de h

moyenne de f vis-à-vis h

$$\mu_f[m,n] = \frac{1}{M_h \cdot N_h} \sum_{k=-M_h/2}^{M_h/2} \sum_{l=-N_h/2}^{N_h/2} f[m+k,n+l]$$

$$\mu_h = \frac{1}{M_h \cdot N_h} \sum_{k=-M_h/2}^{M_h/2} \sum_{l=-N_h/2}^{N_h/2} h[k,l]$$

$$\gamma[m,n] = \left(\frac{1}{\sigma_f[m,n] \cdot \sigma_h} \right) \sum_{k=-M_h/2}^{M_h/2} \sum_{l=-N_h/2}^{N_h/2} \left(f[m+k,n+l] - \mu_f[m,n] \right) \cdot \left(h[k,l] - \mu_h \right)$$

$$\sigma_h = \left(\frac{1}{M_h \cdot n_h} \sum_k \sum_l \left(h[k,l] - \mu_h \right)^2 \right)^{1/2}$$

$$\sigma_f[m,n] = \left(\frac{1}{M_h \cdot N_h} \sum_k \sum_l \left(f[m+,n+] - \mu_f[m,n] \right)^2 \right)^{1/2}$$

variance de h

Source :

4. Corrélation

2. 2D

a) Corrélation normalisée

- *Solution* : on normalise la corrélation par rapport à la moyenne et la variance

moyenne de h

moyenne de f vis-à-vis h

$$\mu_f[m,n] = \frac{1}{M_h \cdot N_h} \sum_{k=-M_h/2}^{M_h/2} \sum_{l=-N_h/2}^{N_h/2} f[m+k,n+l]$$

$$\mu_h = \frac{1}{M_h \cdot N_h} \sum_{k=-M_h/2}^{M_h/2} \sum_{l=-N_h/2}^{N_h/2} h[k,l]$$

$$\gamma[m,n] = \left(\frac{1}{\sigma_f[m,n] \cdot \sigma_h}\right) \sum_{k=-M_h/2}^{M_h/2} \sum_{l=-N_h/2}^{N_h/2} \left(f[m+k,n+l] - \mu_f[m,n]\right) \cdot \left(h[k,l] - \mu_h\right)$$

$$\sigma_h = \left(\frac{1}{M_h \cdot n_h} \sum_k \sum_l \left(h[k,l] - \mu_h\right)^2\right)^{1/2}$$

$$\sigma_f[m,n] = \left(\frac{1}{M_h \cdot N_h} \sum_k \sum_l \left(f[m+,n+] - \mu_f[m,n]\right)^2\right)^{1/2}$$

variance de h

variance de f vis-à-vis h

Source :

Coefficients de la corrélation

$$\gamma(x,y) = \frac{\sum_r \sum_t (f(t,r) - f_M(t,r))(h(x+t,y+r) - h_M)}{\sqrt{A}}$$

$f_M(t,r)$ = moyenne de l'image f sur la région couverte par h

h_M = moyenne du filtre h

$\sqrt{A}$ est un terme de normalisation pour ramener $\gamma(x,y)$ dans l'interval [-1,1]

$$A = \sum_r \sum_t \left(f(t,r) - f_M(t,r)\right)^2 \times \sum_r \sum_t \left(h(x+t,y+r) - h_M\right)^2$$

Image originale : h est entouré de pointillés

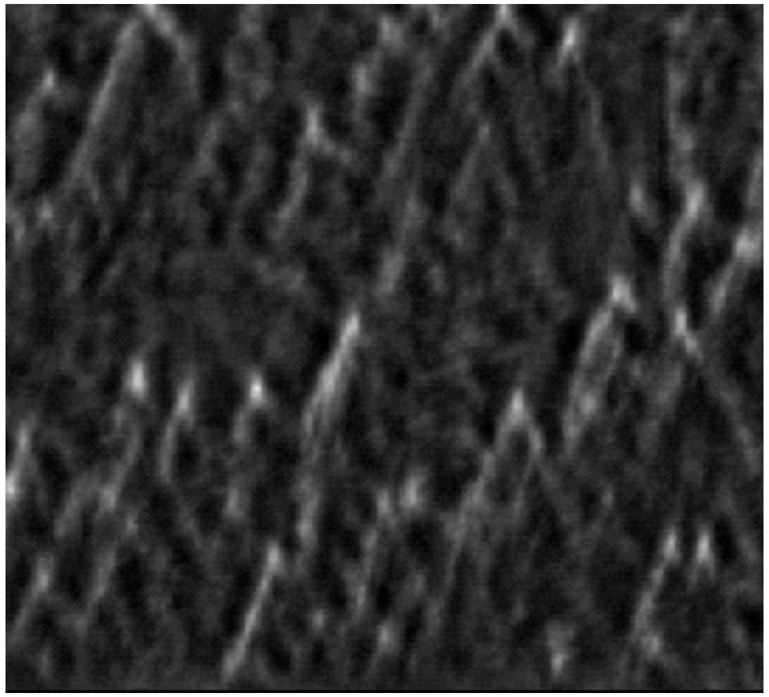


Image de corrélation

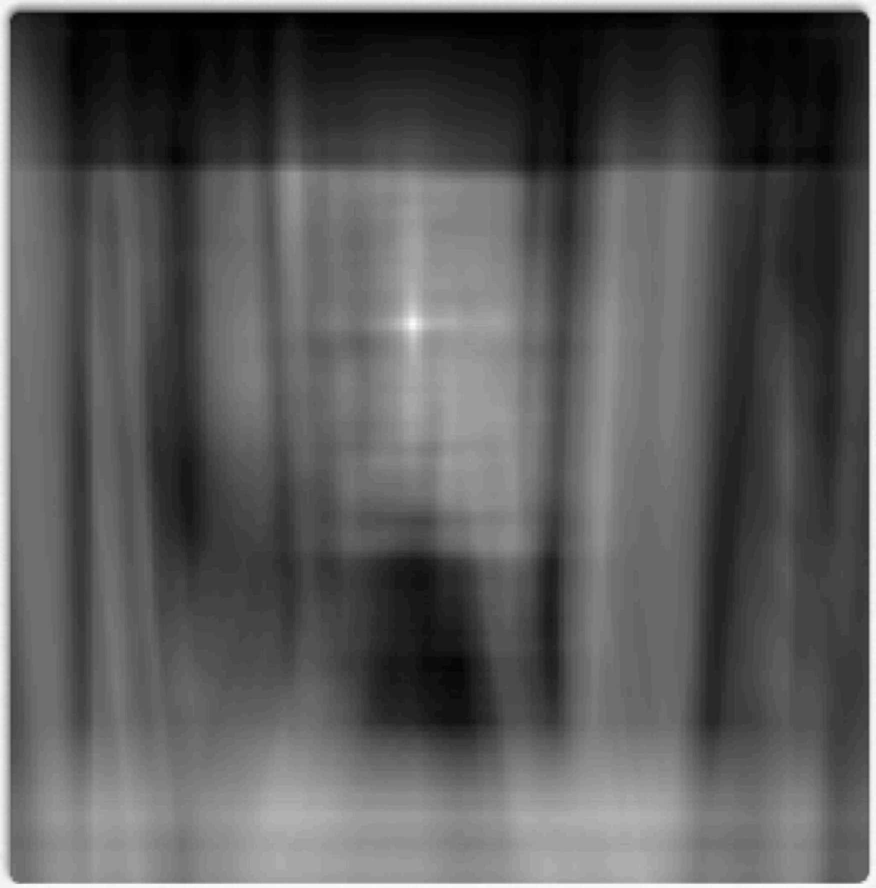
Application : repérer les cimes

4. CORRÉLATION

Ceci est une chaise

Trouver la chaise

Résultat d'une corrélation normalisée



Source : PM Jodoin

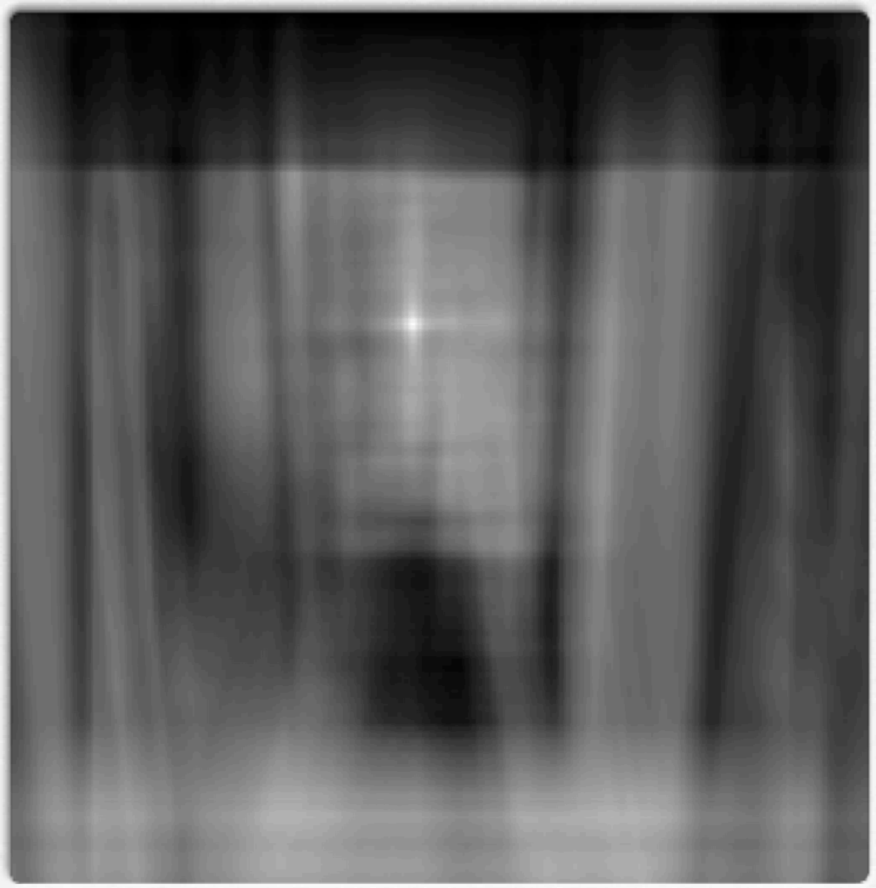
4. CORRÉLATION

1. 2D

Ceci est une chaise

Trouver la chaise

Résultat d'une corrélation normalisée



Source : PM Jodoin

4. CORRÉLATION

2. 2D

Ceci est une bouteille

Source :

4. Corrélation

2. 2D

Trouver la bouteille

Ceci est une bouteille

Source :

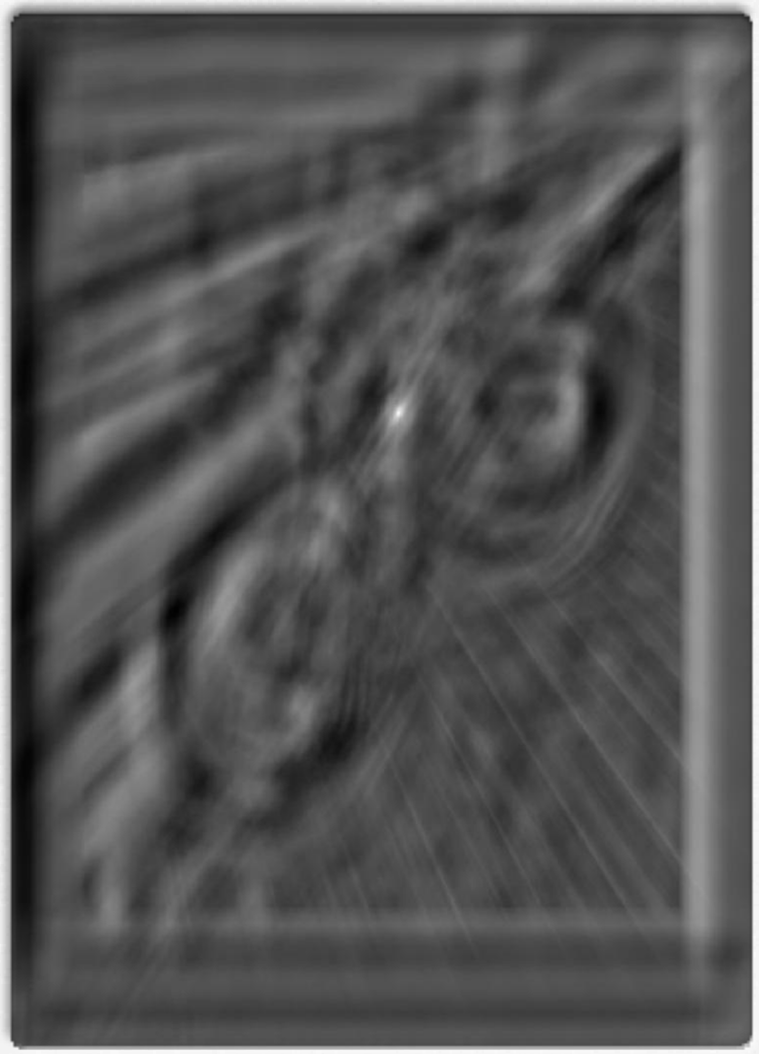
4. Corrélation

2. 2D

Ceci est une bouteille

Trouver la bouteille

Résultat d’une corrélation normalisée



Source :

4. Corrélation

2. 2D

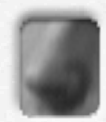
 $\circ$  $=$

Source :

4. CORRÉLATION

2. 2D

∘



=

Image de corrélation

Source :

4. CORRÉLATION

2. 2D

Réponse maximale

Image de corrélation

$\circ$  $=$ 

Source :

4. Corrélation

2. 2D

Réponse maximale

Image de corrélation

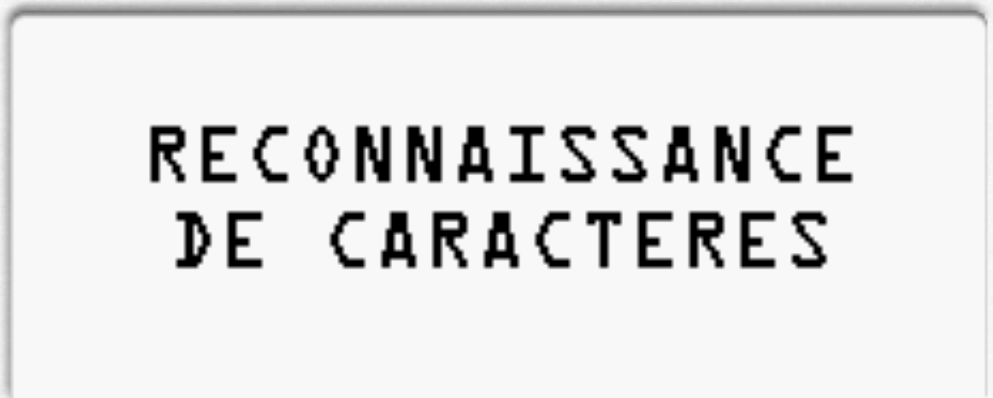
$\circ$

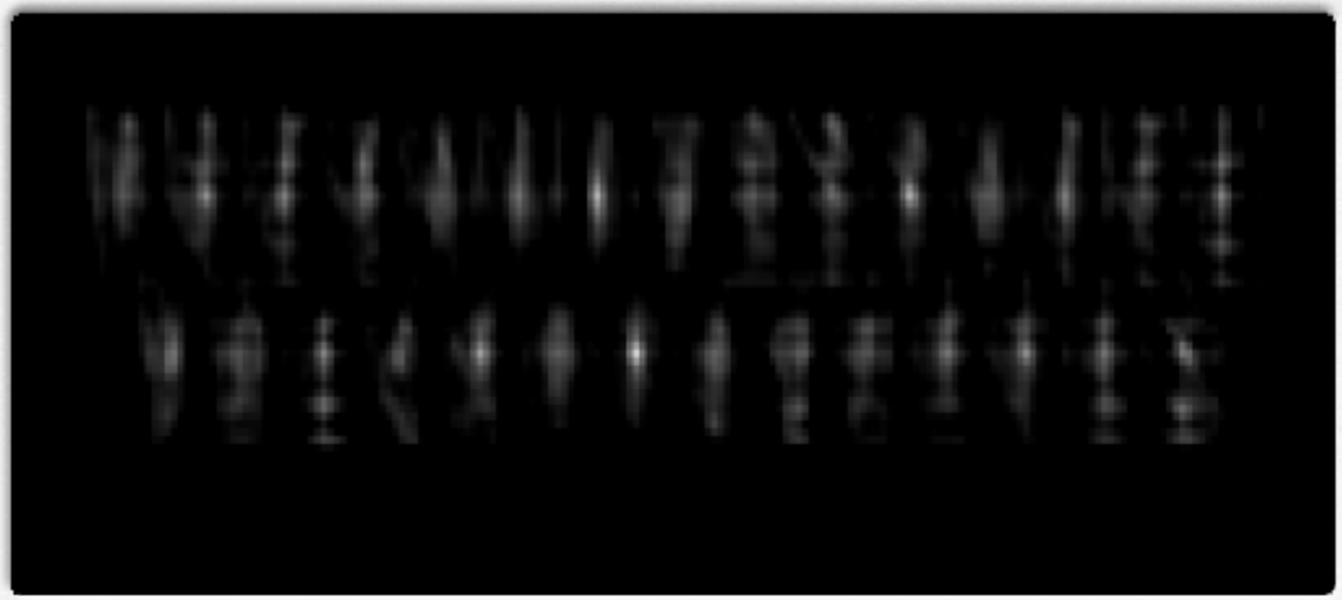
$=$

Source :

4. Corrélation

2. 2D



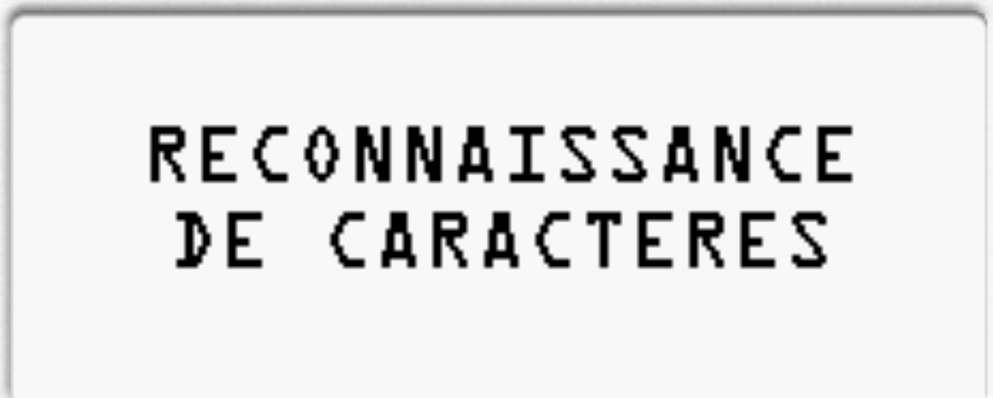


Reconnaissance de caractère

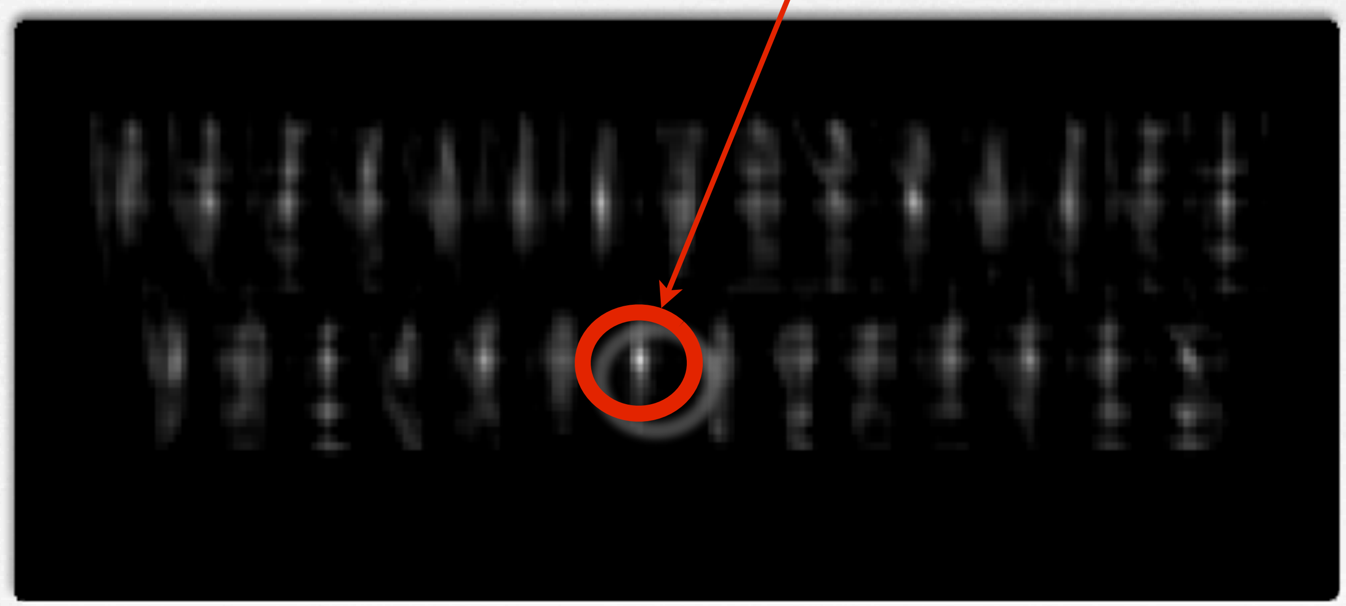
Source : N. Vandenbroucke, LISIC, france

4. Corrélation

2. 2D



Réponse maximale



Reconnaissance de caractère

Source : N. Vandenbroucke, LISIC, france

Corrélation & Convolution

Il y a certains filtres pour lesquels la **convolution = corrélation**. C'est entre autre le cas de :

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} / 9 \qquad \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} / 25 \qquad \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \qquad \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 2 & 2 & 0 \\ 1 & 2 & 5 & 2 & 1 \\ 0 & 2 & 2 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} / 25 \qquad \begin{pmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 4 & 6 & 4 & 1 \\ 3 & 6 & 9 & 6 & 3 \\ 2 & 4 & 6 & 4 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{pmatrix} / 81 \qquad \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix} / 256 \qquad (\ldots)$$