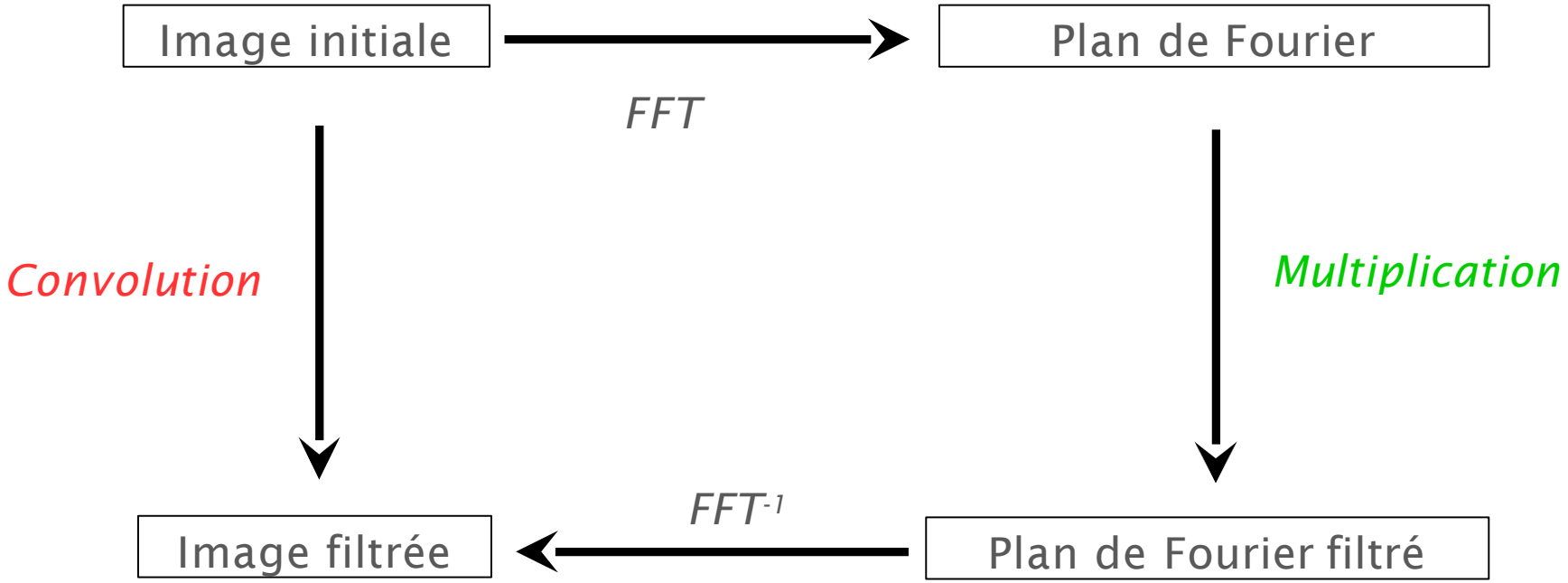
# Filtrage spatial

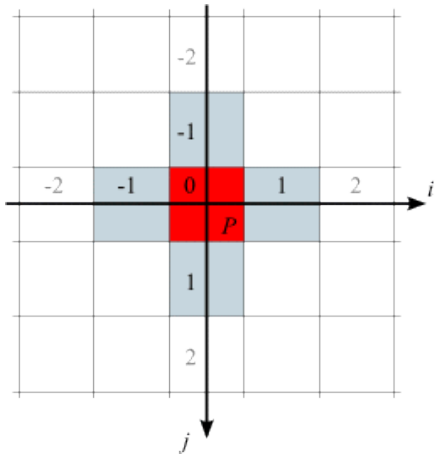
# Filtrage



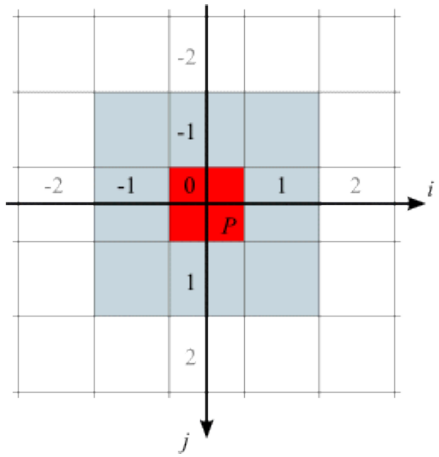

Dans le **domaine spatial**, le filtrage se fait par **convolution**.
Dans le **domaine spectral**, il se fait par **multiplication** (ou **masquage** de l'image).

# Voisinage V d'un pixel P

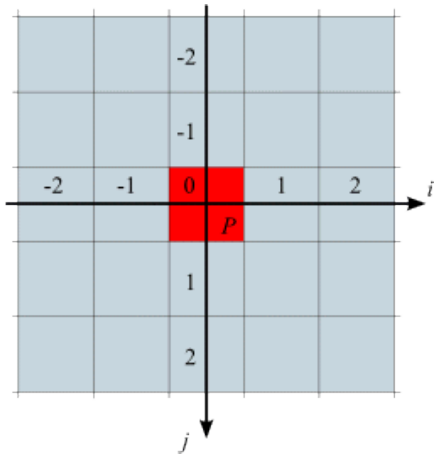
- Hypothèses :
  - V  est centré en *P*
  - les pixels sont disposés selon une maille carrée
- Définition : V (*P*) est l'ensemble des pixels *Q* situés à moins d'une certaine distance de *P*
- La forme du voisinage (et le nombre de voisins) de *P* dépendent de la distance considérée.
- Voisinages les plus usités en traitement d'images :




4-voisinage




8-voisinage (voisinage 3x3)




voisinage 5x5

# Convolution discrète 2D

## Principe :

- L'image $I$ est une fonction de 2 variables discrètes ($x$ et $y$) à support fini.
- Le filtre de convolution $H$ appliqué sur $I$ est lui aussi à 2D. Il est donné par une matrice de poids (ou coefficients) de taille $(2w+1)x(2w+1)$.
- $H$ est appelé filtre, masque, noyau ou fenêtre de convolution.

## Formule et mise en œuvre :

$$I'(x,y) = (I * H)(x,y) = \sum_{i=-w}^{w} \sum_{j=-w}^{w} I(x-i, y-j).H(i,j)$$

Le filtrage de $I$ consiste à déplacer $H$ sur chaque pixel et à remplacer la valeur de ce pixel d'analyse par la combinaison des valeurs de ses voisins donnée par la formule de convolution

# Convolution discrète 2D : calcul

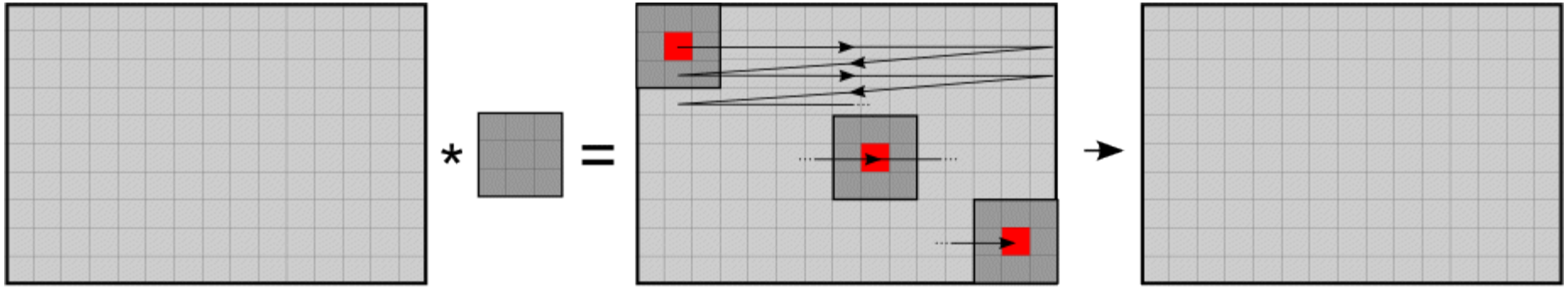


Image source $I$ Filtre $H$ Filtrage de $I$ par $H$ Image résultat $I'$

En chaque pixel $(x,y)$ :

- Tourner la matrice de poids [H($i,j$)] de 180° autour de son centre (0,0)
- Superposer le masque obtenu à l'image $I$ de sorte qu'il soit centré en $(x,y)$
- Multiplier chaque coefficient du masque par le niveau du pixel sous-jacent
- Additionner chacun de ces produits pour obtenir $I'(x,y)$.

$H$

| | | |
|---|---|---|
| $h_{-1,-1}$ | $h_{0,-1}$ | $h_{+1,-1}$ |
| $h_{-1,0}$ | $h_{0,0}$ | $h_{+1,0}$ |
| $h_{-1,+1}$ | $h_{0,+1}$ | $h_{+1,+1}$ |

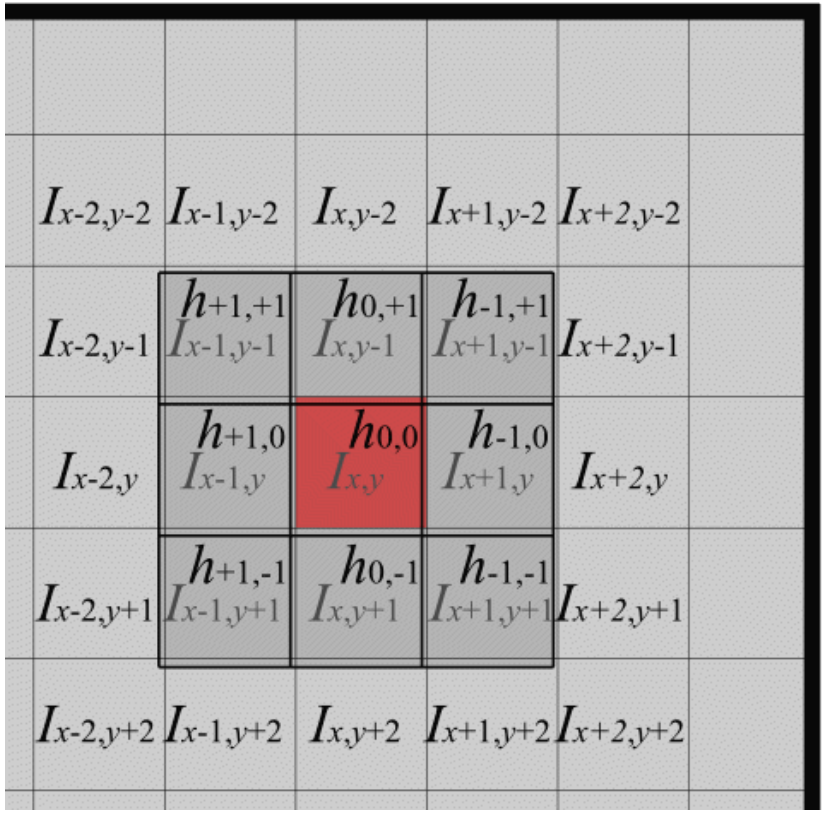
$H_{\theta=180°}$

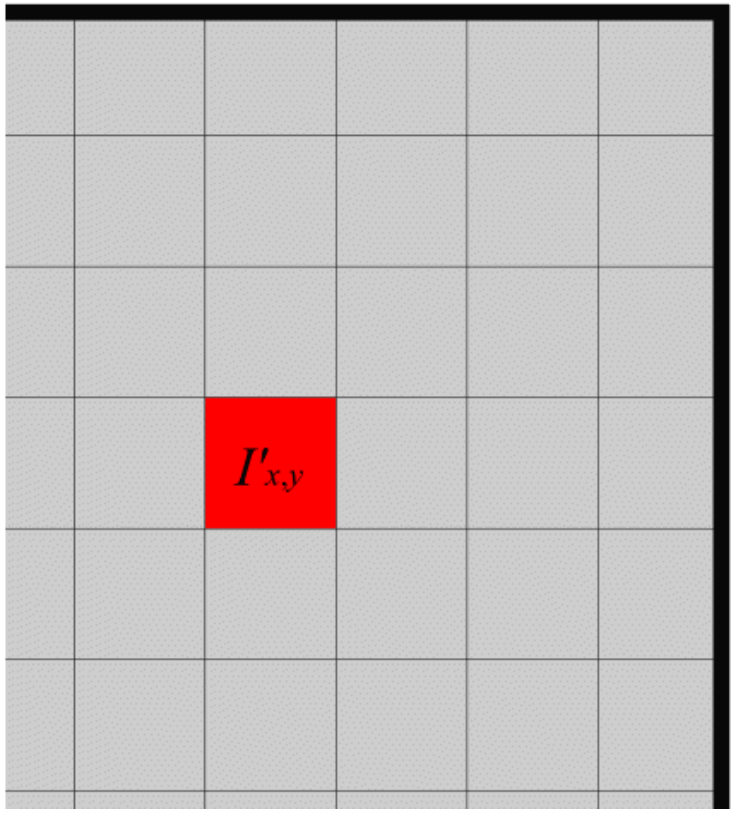
| | | |
|---|---|---|
| $h_{+1,+1}$ | $h_{0,+1}$ | $h_{-1,+1}$ |
| $h_{+1,0}$ | $h_{0,0}$ | $h_{-1,0}$ |
| $h_{+1,-1}$ | $h_{0,-1}$ | $h_{-1,-1}$ |

$H_{\theta=180°}$ et $I$

| | | |
|---|---|---|
| $h_{+1,+1}$ $I_{x-1,y-1}$ | $h_{0,+1}$ $I_{x,y-1}$ | $h_{-1,+1}$ $I_{x+1,y-1}$ |
| $h_{+1,0}$ $I_{x-1,y}$ | $h_{0,0}$ $I_{x,y}$ | $h_{-1,0}$ $I_{x+1,y}$ |
| $h_{+1,-1}$ $I_{x-1,y+1}$ | $h_{0,-1}$ $I_{x,y+1}$ | $h_{-1,-1}$ $I_{x+1,y+1}$ |

# Convolution discrète 2D : calcul




$$I'_{x,y} = I_{x-1,y-1} \cdot h_{+1,+1} + I_{x-1,y} \cdot h_{+1,0} + I_{x-1,y+1} \cdot h_{+1,-1} + I_{x,y-1} \cdot h_{0,+1} + I_{x,y} \cdot h_{0,0} + I_{x,y+1} \cdot h_{0,-1} + I_{x+1,y-1} \cdot h_{-1,+1} + I_{x+1,y} \cdot h_{-1,0} + I_{x+1,y+1} \cdot h_{-1,-1}$$

# Propriétés de la convolution

- Additivité / distributivité : $(I * h_1) + (I * h_2) = I * (h_1 + h_2)$
- Commutativité : $I * h = h * I$
- Associativité : $(I * h_1) * h_2 = I * (h_1 * h_2)$
- Un filtre de convolution est dit séparable si : $h(x, y) = h_x(x) * h_y(y)$

$$[a \quad b \quad c] * \begin{bmatrix} a' \\ b' \\ c' \end{bmatrix} = \begin{bmatrix} aa' & ba' & ca' \\ ab' & bb' & cb' \\ ac' & bc' & cc' \end{bmatrix}$$

- Élément neutre : $\delta = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \rightarrow (\delta * I)(x, y) = I(x, y)$

# Effets de bords

- Problème : comment traiter les pixels situés aux bords de l'image ?

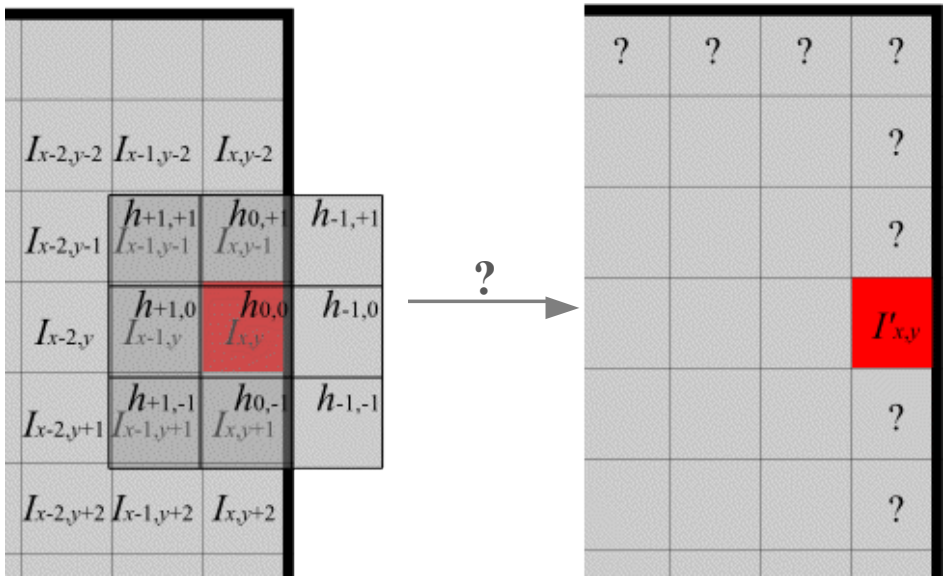
- Stratégies possibles :

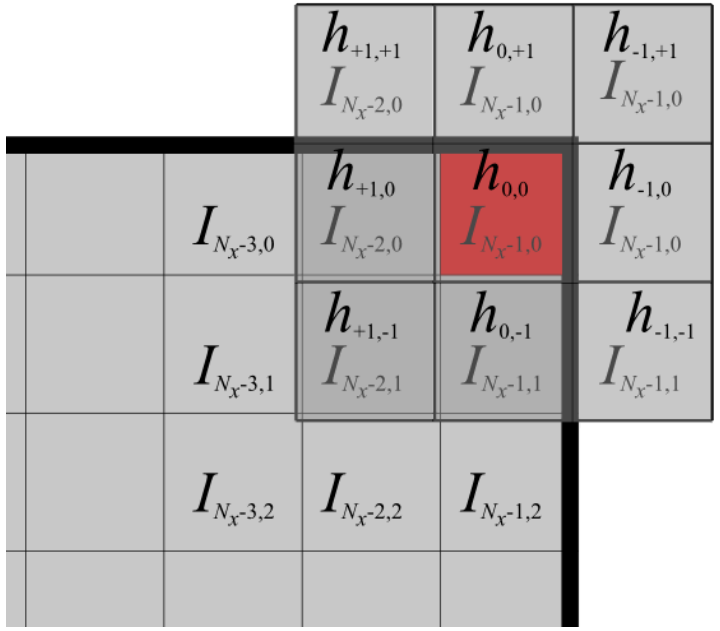
  - laissés inchangés
  - pixels extérieurs = 0
  - miroir (virtuel) de l'image :

    $I(-i-1,y)=I(i,y)$ sur le bord gauche
    $I(N_x+i,y)=I(N_x-i-1,y)$ sur le bord droit
    $I(x,-j-1)=I(x,j)$ sur le bord haut
    $I(x,N_y+j)=I(x,N_y-j-1)$ sur le bord bas

  - image enroulée

  - etc



Complétion avec des zéros

Périodisation

Reproduire le bord

Miroir

# Masque de convolution

- Caractéristiques
  - Ses caractéristiques (coefficients, taille) déterminent l'effet du filtre
  - Souvent carré et de taille impaire (3x3, 5x5, etc) pour être centré sans ambiguïté sur le pixel d'analyse
  - Souvent à valeurs symétriques centrées : h(-1,-1)=h(+1,+1), h(0,-1)=h(0,+1), ...

- Normalisation

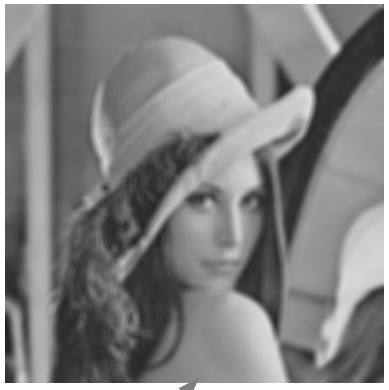
  Soit S la somme des coefficients du masque.

  Diviseur : normalisation à 1.

  - Si l'on veut conserver la luminance de l'image (moyenne des niveaux de gris), on doit avoir S=1.
  - Si les coefficients sont tous positifs, on doit donc les diviser par S

# Types de filtres

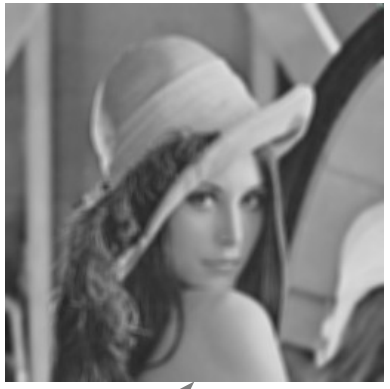
- Filtres passe-bas, ou de lissage
  - Principe : moyenne pondérée des valeurs du voisinage
  - Effet : lissage de l'image (variations atténuées)
  - Avantage : atténuation du bruit
  - Inconvénient : atténuation des  détails, flou
  - Caractérisation : coefficients tous positifs
- Filtres passe-haut, ou de contours
  - Principe : dérivation de la fonction image
  - Effet : accentuation des détails de l'image
  - Avantage : mise en évidence des contours/détails
  - Inconvénient : accentuation du bruit
  - Caractérisation : coefficients de somme nulle

# Types de filtres

- Filtres passe-bas, ou de lissage
  - Principe : moyenne pondérée des valeurs du voisinage
  - Effet : lissage de l'image (variations atténuées)
  - Avantage : atténuation du bruit
  - Inconvénient : atténuation des détails, flou
  - Caractérisation : coefficients tous positifs

lissage



- Filtres passe-haut, ou de contours
  - Principe : dérivation de la fonction image
  - Effet : accentuation des détails de l'image
  - Avantage : mise en évidence des contours/détails
  - Inconvénient : accentuation du bruit
  - Caractérisation : coefficients de somme nulle

## À propos du bruit

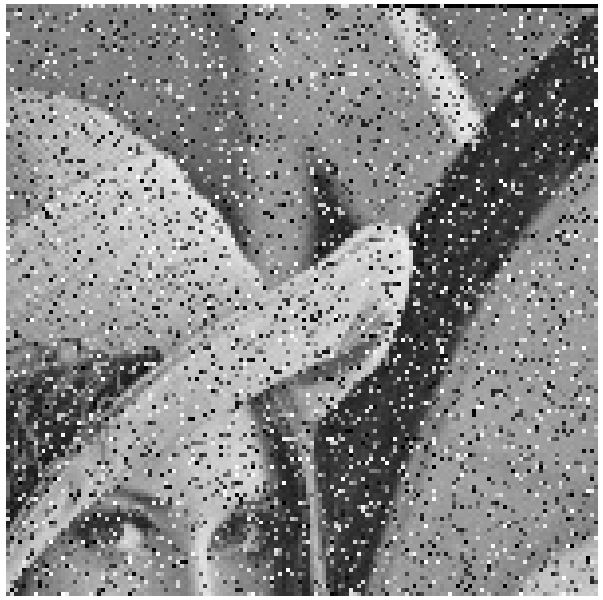
- La restauration de l'image I vise à en réduire le bruit et à trouver l'image idéale I’ qui aurait été obtenue avec un système d'acquisition parfait.
- Le bruit $b(x,y)$ est souvent considéré comme aléatoire.

  - Bruit additif : $I(x,y) = I'(x,y) + b(x,y)$
  - Bruit multiplicatif : $I(x,y) = I'(x,y) \cdot b(x,y)$

# Bruits dans une image : exemples

Bruit additif
Gaussien ($\sigma = 25$)

Bruit multiplicatif
Gaussien ($\sigma = 25$)



Bruit poivre et sel
(20 % de pixels
affectés)

# Filtres de lissage

- Principe
  - Utilité : restauration de l'image (élimination du bruit) par lissage.
  - Inconvénient : suppression des hautes fréquences (filtres passe-bas), d'où dégradation des contours et effet de flou.
- Variétés
  - Plusieurs types de filtres possédant chacun des avantages propres.
  - Plusieurs tailles possibles, selon l'étendue du voisinage à considérer : 3x3, 5x5, ... l'effet de flou est d'autant plus marqué que la taille est grande.
- Principaux filtres de lissage
  - Linéaires
    - Caractérisés par un masque (réalisables par convolution).
    - Exemples : filtres moyenneurs, gaussiens
  - Non-linéaires
    - Caractérisés par un opérateur non-linéaire (non réalisables par convolution).
    - Exemple : filtre médian

# Filtres de lissage linéaires : filtres moyenneurs

- Configuration : dépend de l'importance à donner au pixel d'analyse et à ses voisins :

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} \quad \text{ou} \quad \frac{1}{10}\begin{bmatrix}1 & 1 & 1\\ 1 & 2 & 1\\ 1 & 1 & 1\end{bmatrix} \quad \text{ou} \quad \frac{1}{5}\begin{bmatrix}0 & 1 & 0\\ 1 & 1 & 1\\ 0 & 1 & 0\end{bmatrix}$$

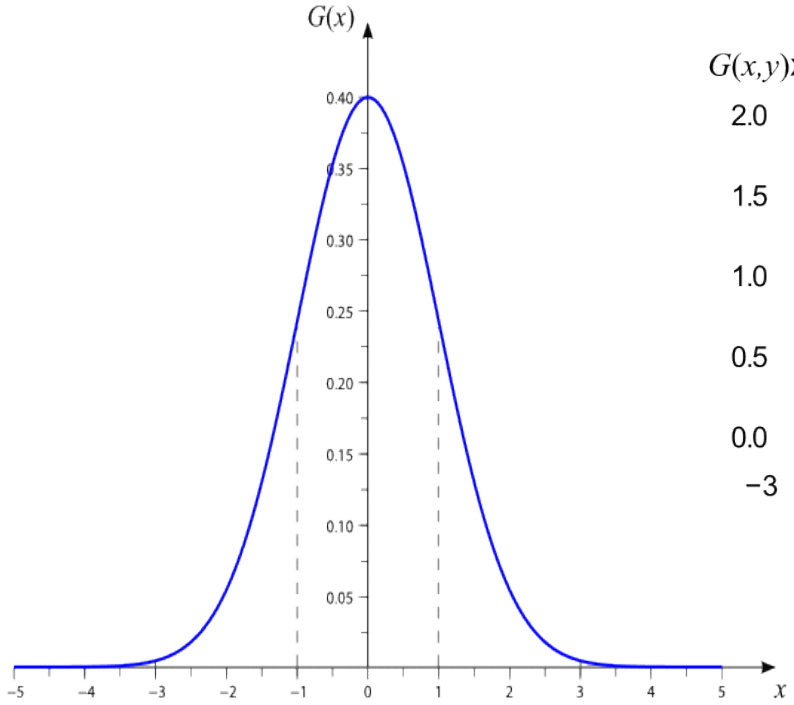
- Remarque : $|S|=1$, donc préservation de la luminance

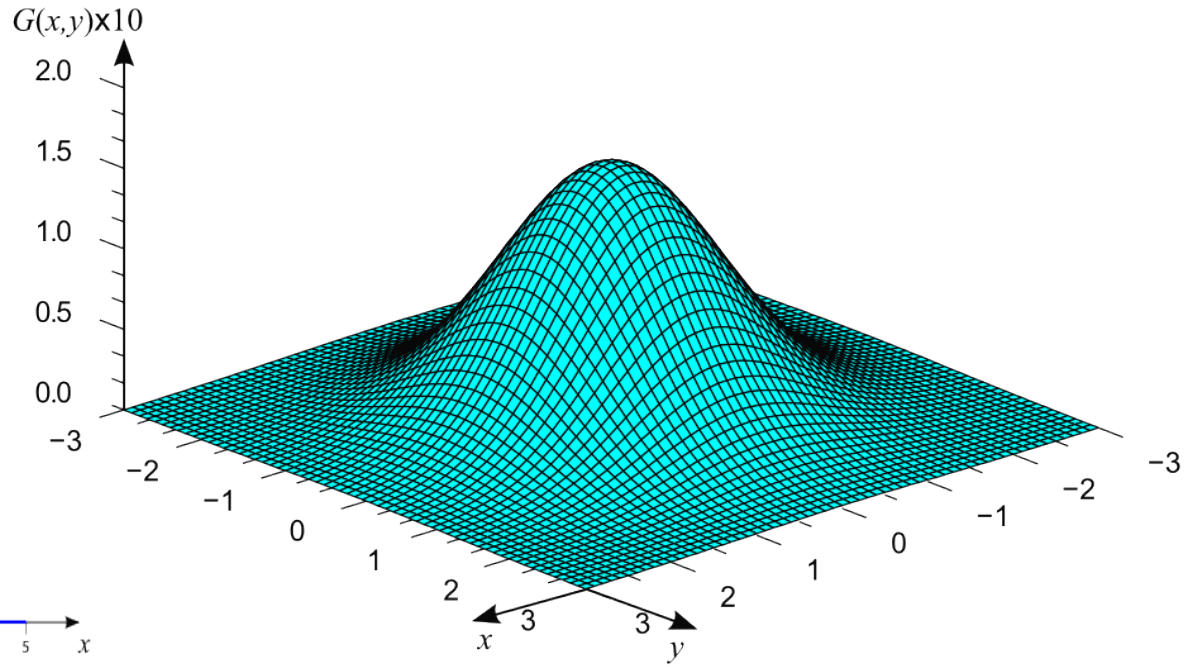
- Inconvénients :
  - forte atténuation des contours (limite la performance des traitements ultérieurs)

# Filtres de lissage linéaires : filtres gaussiens

- Paramètres : moyenne $\mu$, écart-type $\sigma$.

*En 1 D :* $G(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ *En 2 D :* $G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{((x-\mu)^2+(y-\mu)^2)}{2\sigma^2}\right)$



# Filtres de lissage linéaires : filtres gaussiens

- Avantage : limite l'effet de flou (contours mieux conservés)

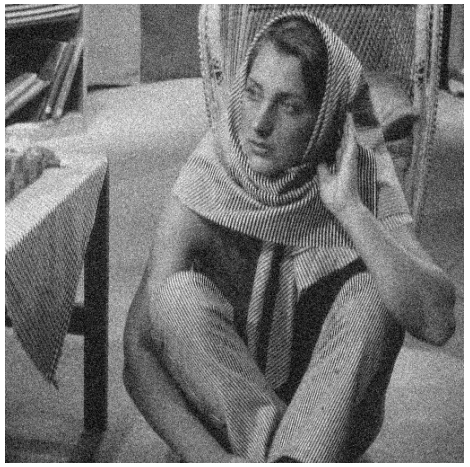
- Configurations : approximations discrètes de la distribution gaussienne de moyenne μ=0 et d'écart-type σ dans un filtre fini. Exemple pour σ=0.8 :

$$H_G = \frac{1}{2025}\begin{bmatrix} 1 & 10 & 22 & 10 & 1 \\ 10 & 106 & 231 & 106 & 10 \\ 22 & 231 & 504 & 231 & 22 \\ 10 & 106 & 231 & 106 & 10 \\ 1 & 10 & 22 & 10 & 1 \end{bmatrix}$$

- Écart-type :
  - Détermine le degré de lissage
  - Impose la taille du masque (idéalement $\lceil 6\sigma \rceil \times \lceil 6\sigma \rceil$)

# Filtres de lissage : exemples

Image originale



Bruit gaussien σ=20

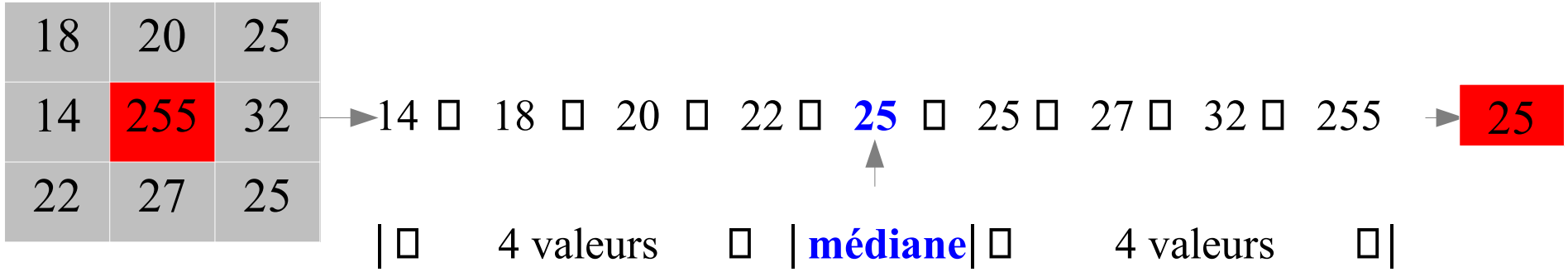
Moy3x3

Moy5x5

Filtre gaussien σ=2

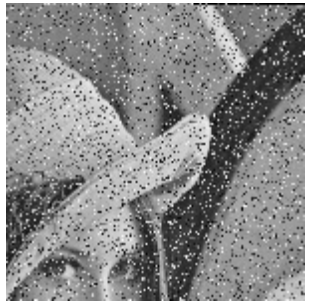
# Filtres de lissage non-linéaires : filtre médian

- Principe
  - Le niveau de gris résultat est le niveau de gris médian des pixels voisins.
  - Filtre non-linéaire, donc non réalisable par masque de convolution.
- Calcul
  - Trier les niveaux par ordre croissant.
  - Donner au pixel d'analyse le niveau situé au milieu des niveaux triés.
- Exemple

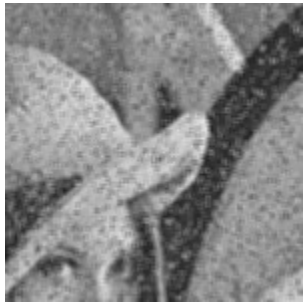


# Filtres de lissage non-linéaires : filtre médian

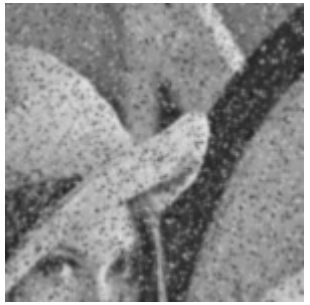
- Avantages par rapport aux filtres moyenneur et Gaussien
  - Filtre mieux le bruit impulsionnel (type « poivre et sel »).
  - Ne crée pas de nouveau niveau ; préserve mieux les contours sans altérer le fond.



*I*



moy 3x3



gauss 5x5

méd 3x3

- Limites et inconvénients
  - Supprime les détails fins qui ne sont pas du bruit.
  - Détruit les coins.
  - Coûteux en temps de calcul (tri).

# Filtres de lissage non-linéaires : filtre min-max

- Principe
  - Garantit que la valeur de tout pixel appartient à l'intervalle des valeurs de ses voisins  débruitage efficace.
  - Préserve encore mieux les contours que le filtre médian.
- Calcul

$$I'(x,y)=\begin{cases} I(x,y) & \text{si } i_{min} \leqslant I(x,y) \leqslant i_{max} \\ i_{min} & \text{si } I(x,y) < i_{min} \\ i_{max} & \text{si } I(x,y) > i_{max} \end{cases}$$

- Exemple

| | | |
|---|---|---|
| 124 | 126 | 127 |
| 120 | 255 | 125 |
| 115 | 119 | 123 |

$i_{min}$=115, $i_{max}$= 127
moyenne = **137**
médiane = **124**
(min-)max = **127**

# Mesure de distorsion

- PSNR (Peak Signal to Noise Ratio)
  - Le PSNR (unité décibel dB) permet de mesurer la similarité entre deux images $I_1$ et $I_2$ codées sur 8 bits et de définition M x N pixels.
  - Il peut être utilisé pour quantifier la puissance du bruit ajouté à l'image ou pour mesurer la qualité du débruitage.
- Calcul

$$PSNR(I_1, I_2) = 10 \log_{10} \left( \frac{255^2}{EQM(I_1, I_2)} \right)$$

où EQM désigne l'erreur quadratique moyenne entre les 2 images :

$$EQM(I_1, I_2) = \frac{1}{M \times N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (I_1(x, y) - I_2(x, y))^2$$

- NB : Si $I_1 = I_2$ alors $PSNR(I_1, I_2) = +\infty$