Chapitre 4 –
Transformations d’intensités
et filtrage d’images dans le domaine spatial

ÉCOLE
POLYTECHNIQUE
MONTRÉAL

Plan

- Du 1D vers le 2D
- Histogramme d’une image
- Transformation d’histogramme
- Filtres passe-bas
- Filtres passe-haut
- Filtrage en présence de bruit
- Filtres morphologiques

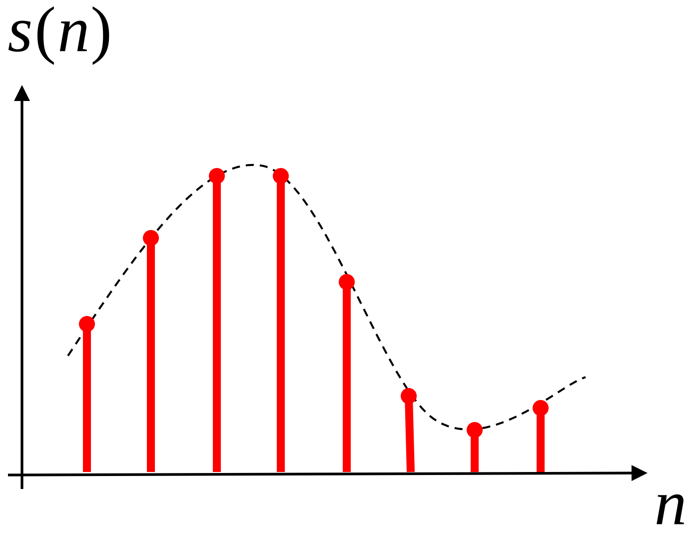
Du 1D vers le 2D

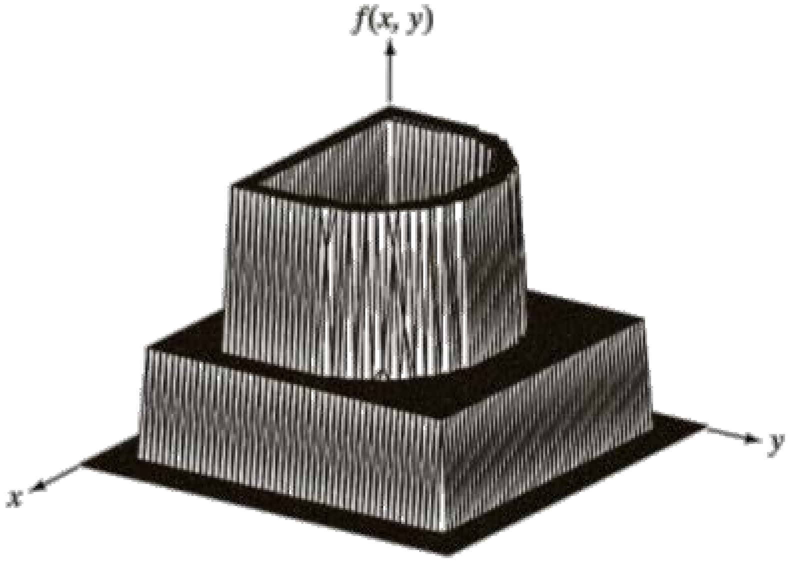
Du signal 1D à l'image numérique

Signal 2D

Une image numérique est un signal à **deux dimensions discret** et **borné.** Elle est représentée mathématiquement par :

$$f(x, y) \in N, x \in N, y \in N$$

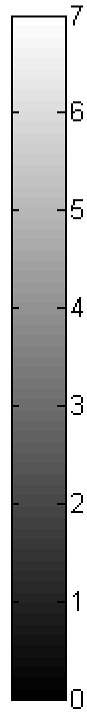




L’image numérique

1	1	1	1	1	1	1	1
1	5	5	5	5	5	5	1
1	5	3	3	3	3	5	1
1	5	3	6	6	3	5	1
1	5	3	6	6	3	5	1
1	5	3	3	3	3	5	1
1	5	5	5	5	5	5	1
1	1	1	1	1	1	1	1

Matrice, notée f



Palette de couleurs

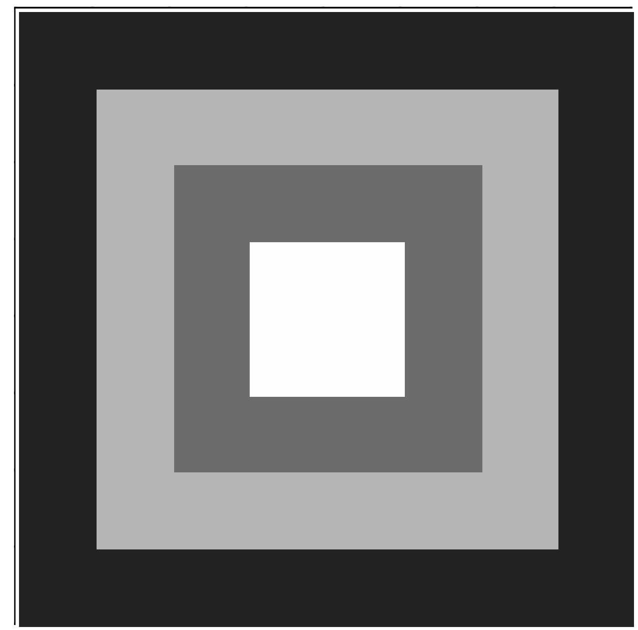


Image numérique

Généralisation

Généralisation

Une image numérique est un signal à **deux dimensions discret** et **borné**

Toutes les définitions et propriétés des Transformées de Fourier discrète, inversées, convolution, etc. peuvent être facilement généralisées. Néanmoins nous n'expliciterons pas ces formules dans le cours.

>>fft2	**TF 2D**
>>ifft2	**TF inverse 2D**
>>fftshift	**recentrage des fft pour affichage**
>>filter2 (ou conv2)	**filtrage par convolution**

Transformées de Fourier 2D

TABLE 2.1

[redacted]

Signal	Fourier Transform
1	$\delta(u, v)$
$\delta(x, y)$	1
$\delta(x - x_0, y - y_0)$	$e^{-j2\pi(ux_0 + vy_0)}$
$\delta_s(x, y; \Delta x, \Delta y)$	$\mathrm{comb}(u\Delta x, v\Delta y)$
$e^{j2\pi(u_0 x + v_0 y)}$	$\delta(u - u_0, v - v_0)$
$\sin[2\pi(u_0 x + v_0 y)]$	$\frac{1}{2j}[\delta(u - u_0, v - v_0) - \delta(u + u_0, v + v_0)]$
$\cos[2\pi(u_0 x + v_0 y)]$	$\frac{1}{2}[\delta(u - u_0, v - v_0) + \delta(u + u_0, v + v_0)]$
$\mathrm{rect}(x, y)$	$\mathrm{sinc}(u, v)$
$\mathrm{sinc}(x, y)$	$\mathrm{rect}(u, v)$
$\mathrm{comb}(x, y)$	$\mathrm{comb}(u, v)$
$e^{-\pi(x^2 + y^2)}$	$e^{-\pi(u^2 + v^2)}$

Représentation d'un système linéaire invariant

1D $\longrightarrow$ **2D**

$$s(n) \longrightarrow \boxed{h(n)} \longrightarrow u(n)$$

$$u(n) = s(n) * h(n)$$

$$f(x,y) \longrightarrow \boxed{h(x,y)} \longrightarrow g(x,y)$$

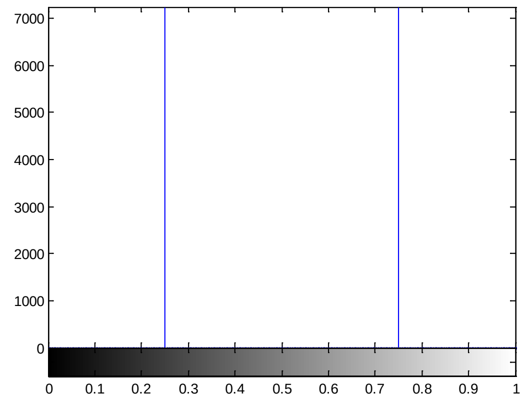
$$g(x,y) = f(x,y) * h(x,y)$$

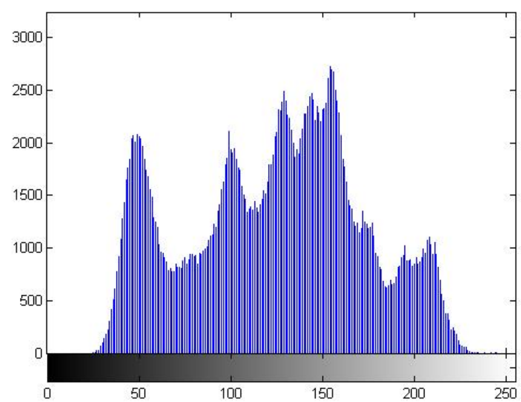
Histogramme d'une image

Histogramme des niveaux de gris d'une image

Définition

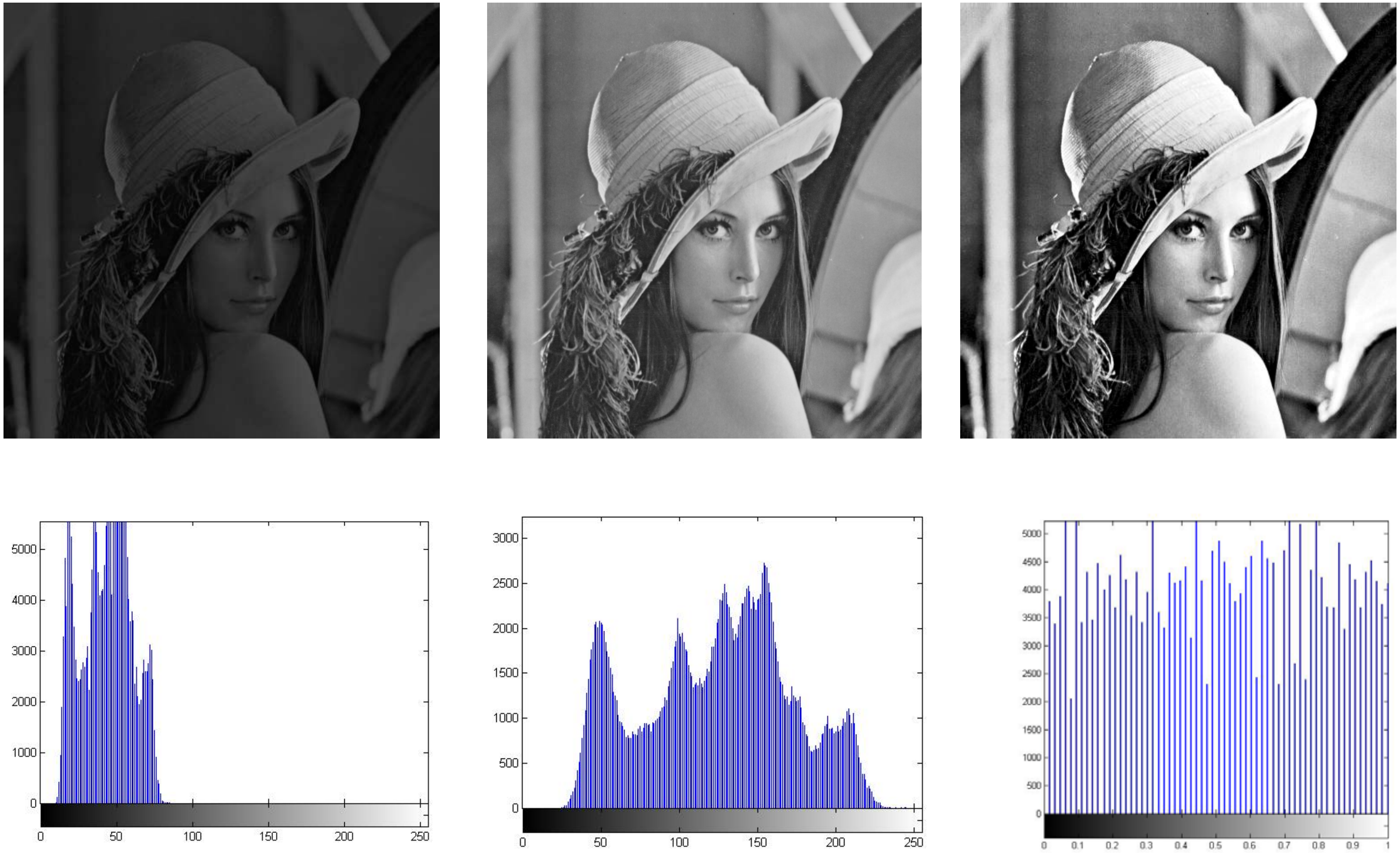
L'histogramme des niveaux de gris d'une image est la distribution des valeurs de niveaux de gris



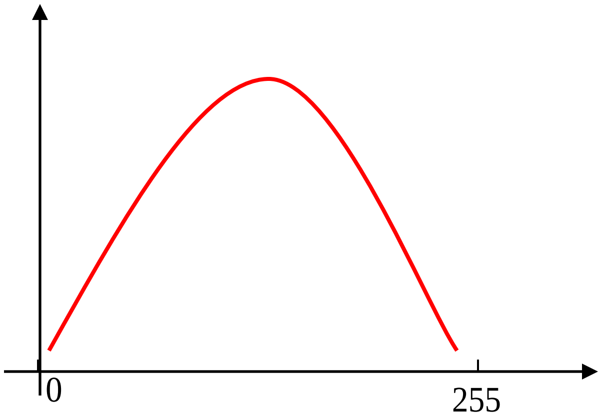


>>imhist(I)

Histogramme et contraste

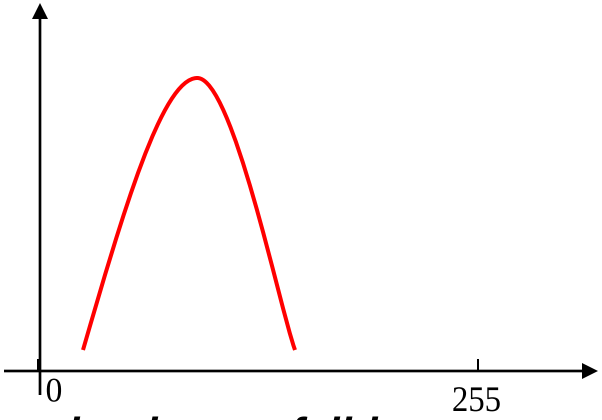


Histogramme et contraste

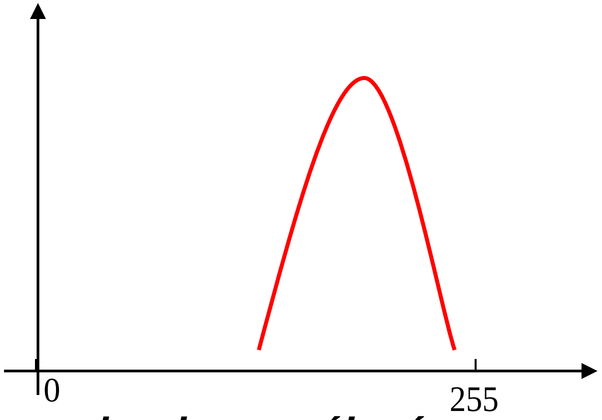


contraste élevé

contraste faible



luminance faible



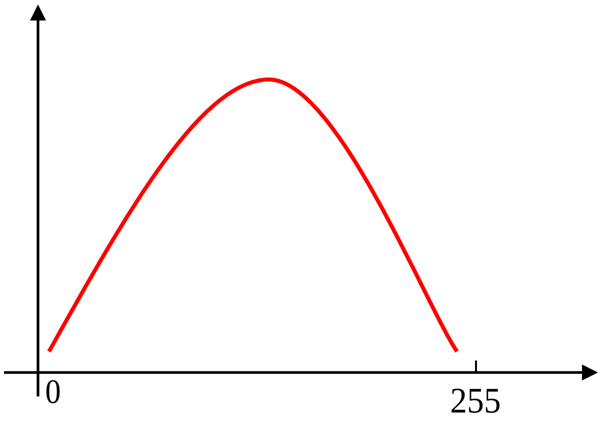
luminance élevée

Transformation d'histogramme

Comment passer de ....

transformation d'histogramme !

.... à ?



Transformation d'histogramme

Définition

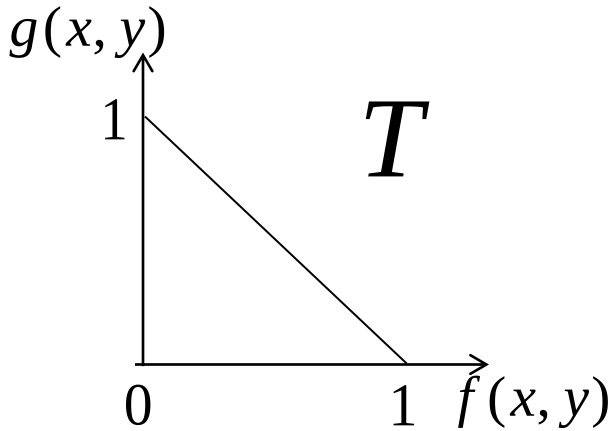
La transformation d'histogramme est une fonction pas nécessairement linéaire transformant un histogramme en un autre.

Cas général

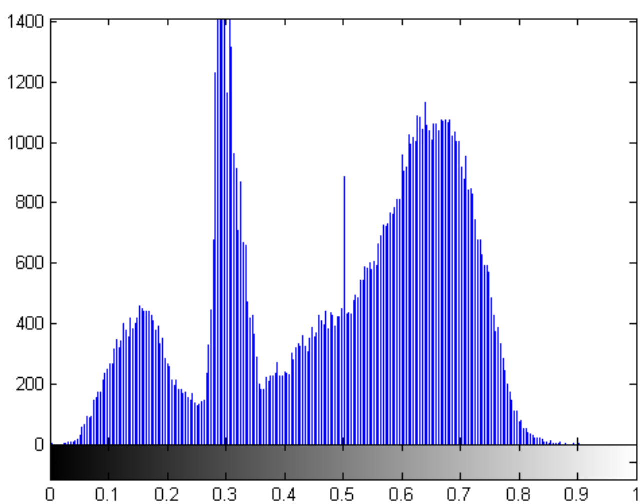
$$g(x,y) = T[f(x,y)]$$

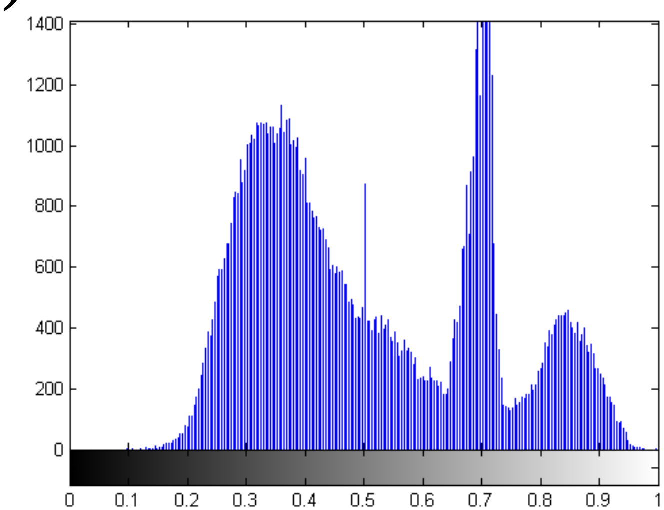
g : Image transformée f : Image originale

Exemple : Inversion des niveaux de gris



$$g(x, y) = 1 - f(x, y)$$





Égalisation d'histogramme

Définition

L'égalisation d'histogramme est une fonction non linéaire transformant un histogramme en un histogramme plus uniformes.

On peut démontrer que la fonction de cette transformation est égale à l'histogramme **normalisé cumulatif** de départ. Alors l'image d'arrivée est donnée par:

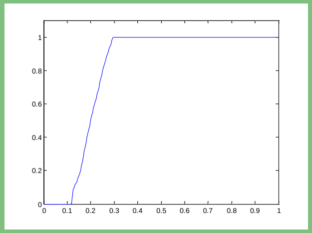
$$I_{eq}(i, j) = 255 * T[I(i, j)]$$

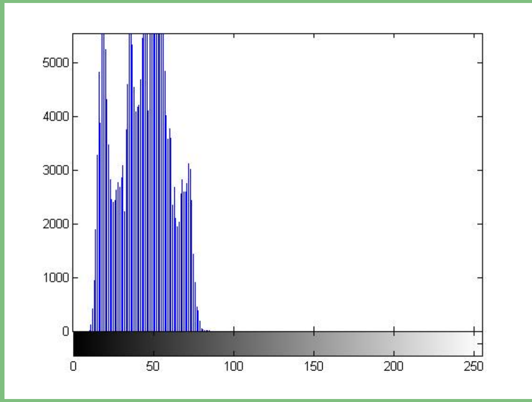
où $T(i) = \sum_{j=0}^{i} h_n(j)$ est l'histogramme cumulatif normalisé de l'image de départ

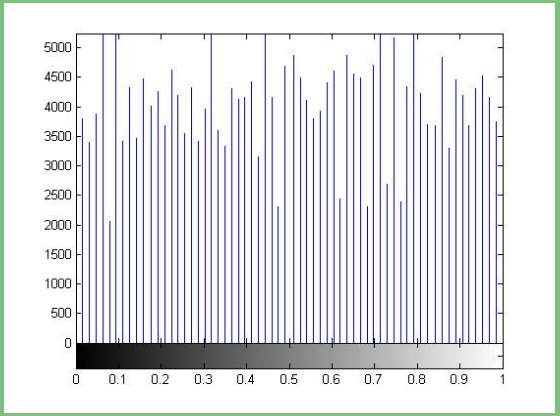
Egalisation d’histogramme

>>I_eq=histeq(I)

histogramme cumulatif normalisé de I







Plan

- Du 1D vers le 2D
- Histogramme d’une image
- Transformation d’histogramme
- Filtres passe-bas
- Filtres passe-haut
- Filtres morphologiques
- Filtrage en présence de bruit

Filtrage spatial

$$f(x,y) \longrightarrow \boxed{h(x,y)} \longrightarrow g(x,y)$$

Image originale

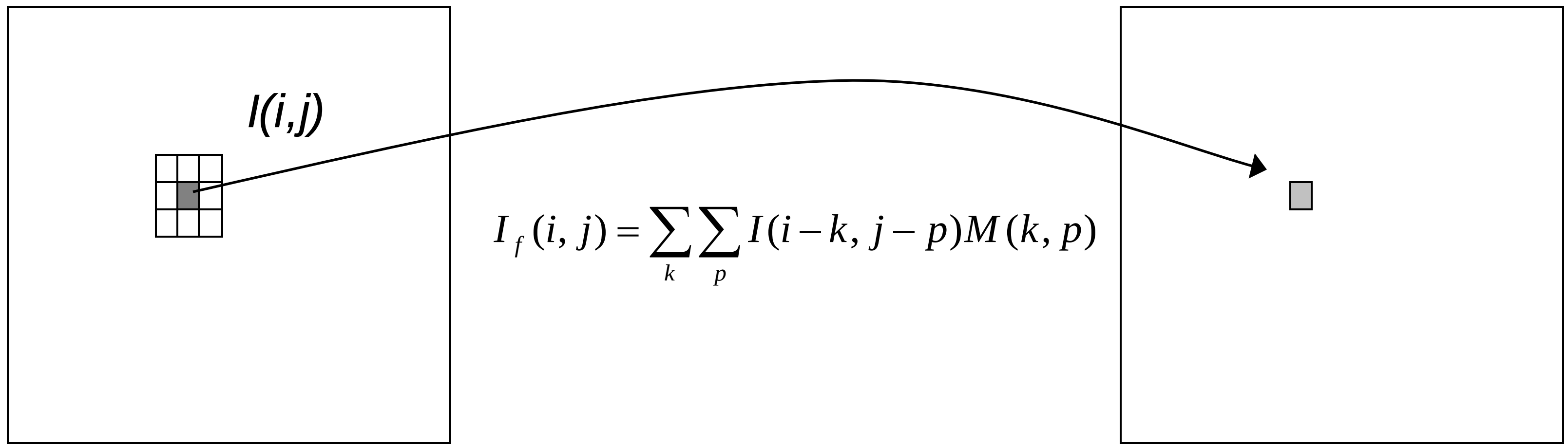
Image filtrée

$$g(x,y) = f(x,y) * h(x,y)$$

Convolution 2D

La convolution 2D d'une image par un « masque » (=filtre spatiale) correspond à une transformation glissante basée sur le voisinage des points

<u>Ex.</u> avec un masque M(k,p) de taille 3*3:



$$I_f(i,j) = \sum_k \sum_p I(i-k, j-p) M(k,p)$$

image originale *image filtrée*

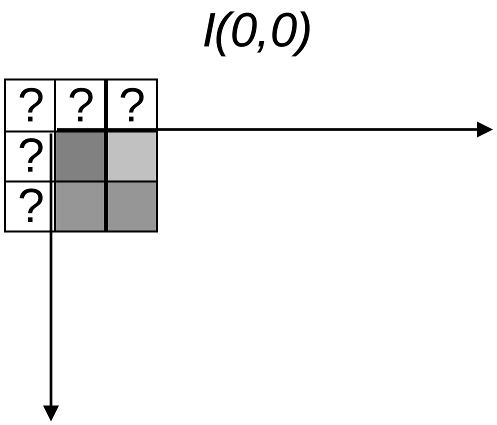
<u>Remarques</u>.

* les masques sont souvent symétriques et de dimensions **impaires**

* pour conserver la moyenne de I (et donc éviter une altération de la luminance moyenne), la **somme** des coefficients du masque est souvent fixée à **1**

Convolution 2D

<u>Effet de bords:</u>
Au bord il manque des valeurs (signal borné) exemple dans le coin:



<u>Solutions:</u>
* ignorer les bords → mais l'image devient + petite ou inchangée sur les bords
* considérer l'image périodique
* considérer l'image symétrisée (mirroir aux bords)
* ajouter des valeurs constantes aux bords

→ aucune solution parfaite

Filtrage passe-bas

Filtre moyen

Un filtre moyen est un filtre de convolution avec un masque moyenneur non pondéré. Il est exactement l'équivalent du filtre moyenneur en signal 1D...

Ex.: filtre moyen de taille 3*3

$$I * \frac{1}{9} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

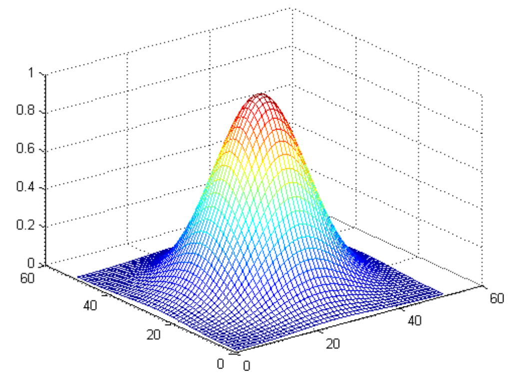
Comme en signal 1D, ce filtre est un filtre passe-bas:

→ il diminue le bruit
→ dans les images, on remarque qu'il dégrade les contours
→ il diffuse le bruit impulsionnel au lieu de l'éliminer
→ les lobes de sa TF 2D créent des fréquences parasites (aliasing)..

Filtre Gaussien

Une Gaussienne 2D continue centrée d'écart type σ est une fonction de la forme:

$$G(x,y) = e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)}$$



Pour les images numériques, on approxime une telle Gaussienne continue par un masque 2D borné idéalement de taille (6σ)* (6σ) pour couvrir la majeure partie non nulle de la Gaussienne.

ex: une Gaussienne de taille 3*3

$$I * \frac{1}{16} \cdot \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Avantage par rapport au filtre moyenneur:
- la TF d'une Gaussienne est une Gaussienne donc sans lobes (pas d'aliasing)
- le calcul peut être séparé en deux Gaussiennes 1D plus efficaces

Filtre Median

Définition

En chaque pixel, on remplace la valeur par la valeur médiane prise sur un voisinage du pixel

Ex: voisinage 3*3

10	11	9
12	200	10
11	11	11

valeur bruit

$$I_f(i,j) = med(9,10,10,11,11,11,11,12,200) = 11$$

Filtre Median

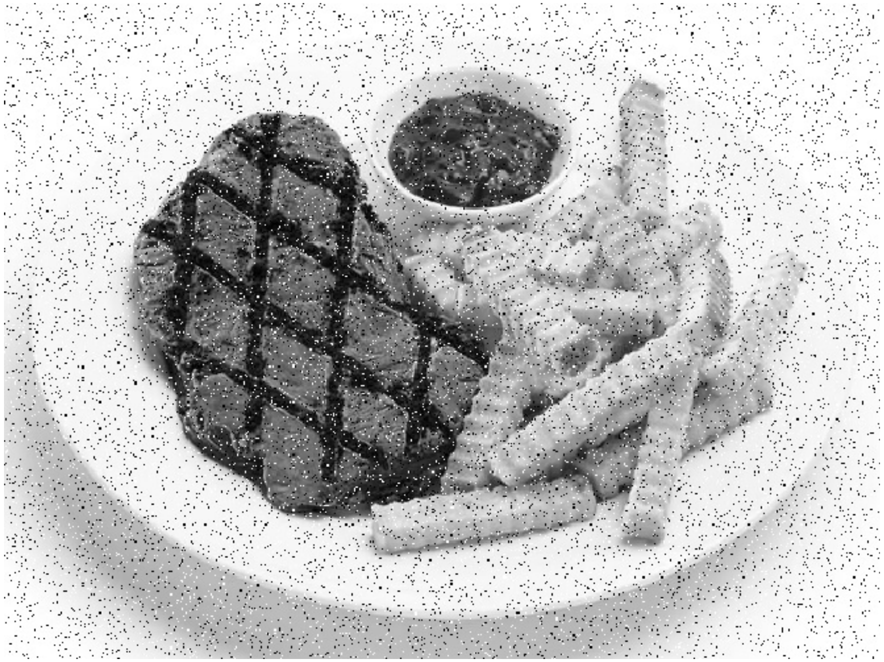


Image avec beaucoup de bruit poivre & sel

image filtrée avec un filtre médian de taille 3*3

>>medfilt2

Remarques:

- très adapté au bruit impulsionnel type « **poivre et sel** »
- préserve les contours
- les filtres Gaussien et moyenneur sont des filtres linéaires. Le filtre médian est un filtre **non linéaire**.

Exemple filtre médian

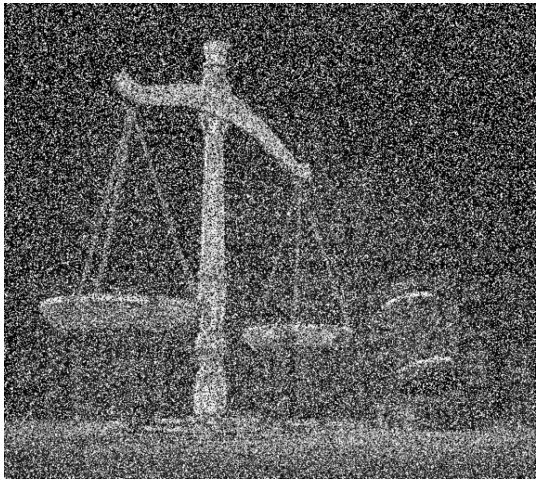
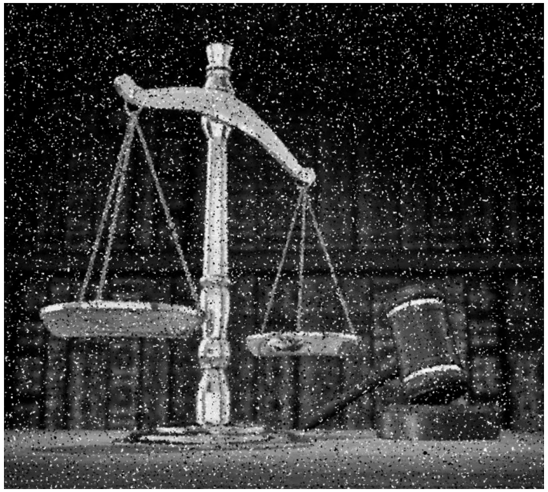
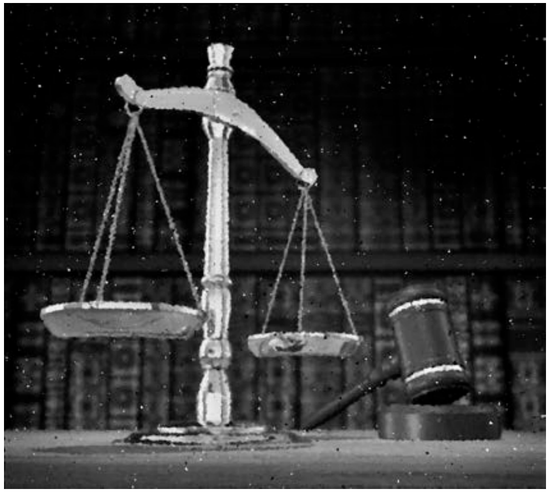


image bruitée sel&poivre



filtre médian 3*3



filtre médian 5*5

filtre médian 7*7

Filtre médian adaptatif

Idée: faire varier localement la taille [n,m] pour améliorer les résultats

L'objectif est aussi de conserver les qualités du filtre médian en améliorant son comportement au niveau des contours:
→ filtrer le bruit poivre et sel
→ réduire aussi le bruit Gaussien
→ ne pas trop dégrader les contours

Filtre médian adaptatif

Algorithme:

Pour chaque pixel (i,j) de l'image à filtrer
n=m=1
tant que n*m<seuil
    voisinage S_{nm} autour de (i,j)
    si $g_{min}<g_{med}<g_{max}$ sur S_{nm}
        si $g_{min}<g_{ij}<g_{max}$ sur S_{nm}
            on garde g_{ij}
        sinon
            on prend g_{med}
        fin
    sinon on augmente le voisinage n++, m++
fin
si la fenêtre a dépassé le seuil
    on prend g_{med}
fin

: si la valeur médiane n'est pas une impulsion

: et si la valeur en (i,j) n'est pas une impulsion
→ alors on garde cette valeur

→ sinon on prend la valeur médiane (qui n'est pas du bruit)

: si la valeur médiane est une impulsion
→ on augmente la taille de la fenêtre

Filtre médian adaptatif: exemple

image originale

image bruitée

filtre médian 9*9

filtre médian adaptatif
taille max 9*9

→ préserve mieux les contours

Filtrage passe-haut

Filtre Passe haut à partir d'un passe-bas

Principe

Comme en 1D, on peut obtenir une image filtrée passe-haut en retranchant l'image filtrée passe-bas à l'image originale.

$$f_{PH}(x,y) = f(x,y) - f_{PB}(x,y)$$

On peut également construire un filtre passe haut à partir d'un filtre passe bas :

$$h_{PH}(x,y) = \delta(x,y) - h_{PB}(x,y)$$

<u>Ex:</u> à partir du filtre moyenneur 3*3

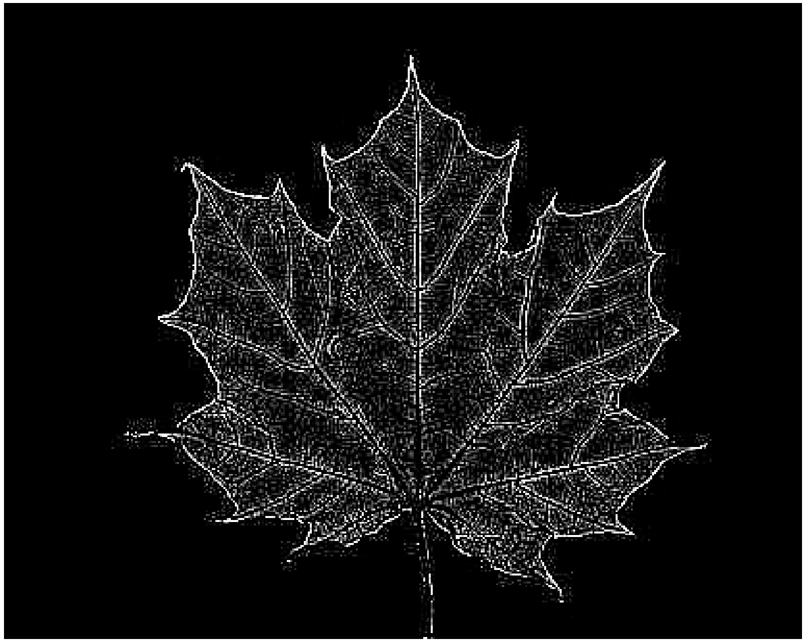
$$\frac{1}{9}\cdot\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Filtre Passe Haut

<u>Ex:</u> à partir du filtre moyenneur 3*3

$$\frac{1}{9}\cdot\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$\longrightarrow$



Filtre High-Boost

Principe

On souhaite rehausser les contours d'une image. On utilise alors le filtre High-boost

Étape 1 : $$f_{PH}(x,y) = f(x,y) - f_{PB}(x,y)$$

Étape 2 : $$f_{HB}(x,y) = f(x,y) + K.f_{PH}(x,y)$$

K est un facteur multiplicatif (high boost) pour accentuer les contours

Filtre Gradient

Définition

En continu 2D, le **vecteur** gradient est défini par:

$$\nabla I = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial I}{\partial x} \\ \dfrac{\partial I}{\partial y} \end{bmatrix}$$

<u>Remarques:</u>

→ c'est un vecteur, d'opérateurs linéaires

→ il pointe dans la direction de la plus forte variation d'intensité en (x,y)

→ le module de ce vecteur donne l'amplitude de cette variation:

$$M(x,y) = \sqrt{g_x^2 + g_y^2} \approx |g_x| + |g_y|$$

Filtre Gradient

Comment calculer le vecteur gradient en pratique (**cas discret**) ?
On approxime les dérivées:

Sur un voisinage autour de I(0,0)=I(x,x)

$$\begin{array}{cc} & \xrightarrow{\hspace{6em}} y \\ x \downarrow & \begin{bmatrix} I_{-1,-1} & I_{-1,0} & I_{-1,1} \\ I_{0,-1} & I_{0,0} & I_{0,1} \\ I_{1,-1} & I_{1,0} & I_{1,1} \end{bmatrix} \end{array}$$

<u>ex:</u>

$$\frac{\partial I}{\partial x} \approx I(x+1, y) - I(x, y)$$

$$\frac{\partial I}{\partial y} \approx I(x, y+1) - I(x, y)$$

$$g_x \approx I_{1,0} - I_{0,0} \approx I_{0,0} - I_{-1,0} \approx I_{1,0} - I_{-1,0}$$

$$g_y \approx I_{0,1} - I_{0,0} \approx I_{0,0} - I_{0,-1} \approx I_{0,1} - I_{0,-1}$$

Filtre Gradient

Masques de Gradient simples:

g_x $$\begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

g_y $$[1 \quad -1],[-1 \quad 1],[1 \quad 0 \quad -1],[-1 \quad 0 \quad 1]$$

Filtre de Roberts (diagonales): $$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

→ sur une seule colonne: assez **sensible au bruit**
→ si dimension paire = **pas symétrique**

Filtre Gradient

<u>Masques de Gradient 3*3</u>

Pour rendre moins sensible au bruit on peut introduire un lissage:

+ lissage moyenneur

+ lissage Gaussien

$$g_x \quad \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} * [1 \quad 1 \quad 1] = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

$$g_y \quad \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} * [1 \quad 0 \quad -1] = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

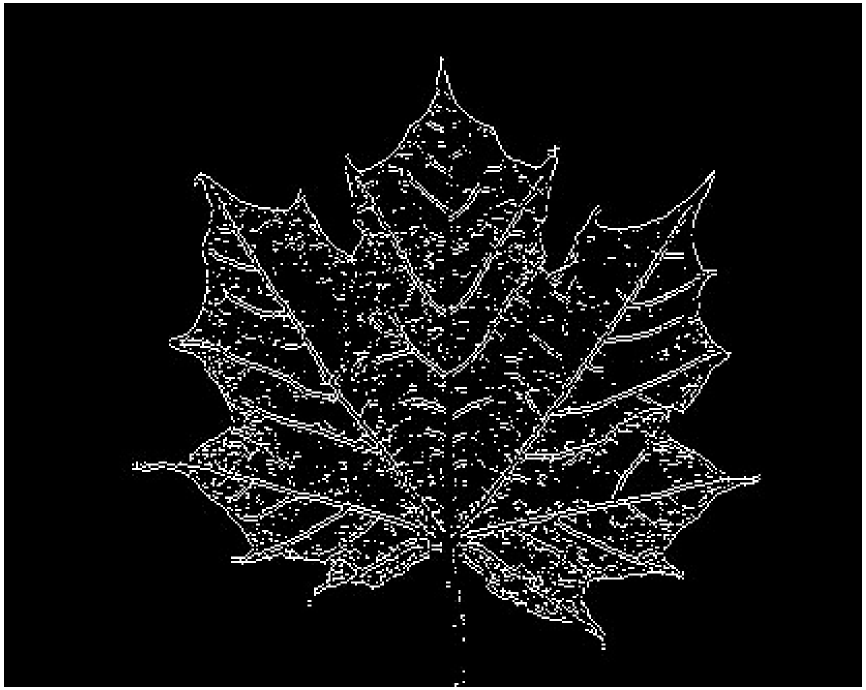
filtres de Prewitt

filtres de Sobel

Filtre Gradient

```
>>edge(I,'sobel')
>>imgradientxy
```

Exemples filtre de Sobel:



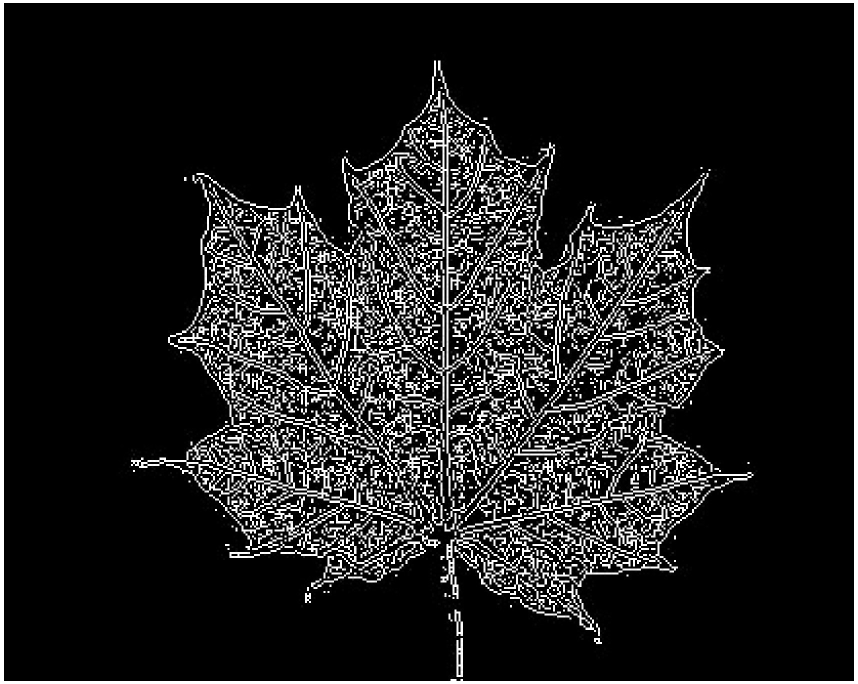
$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

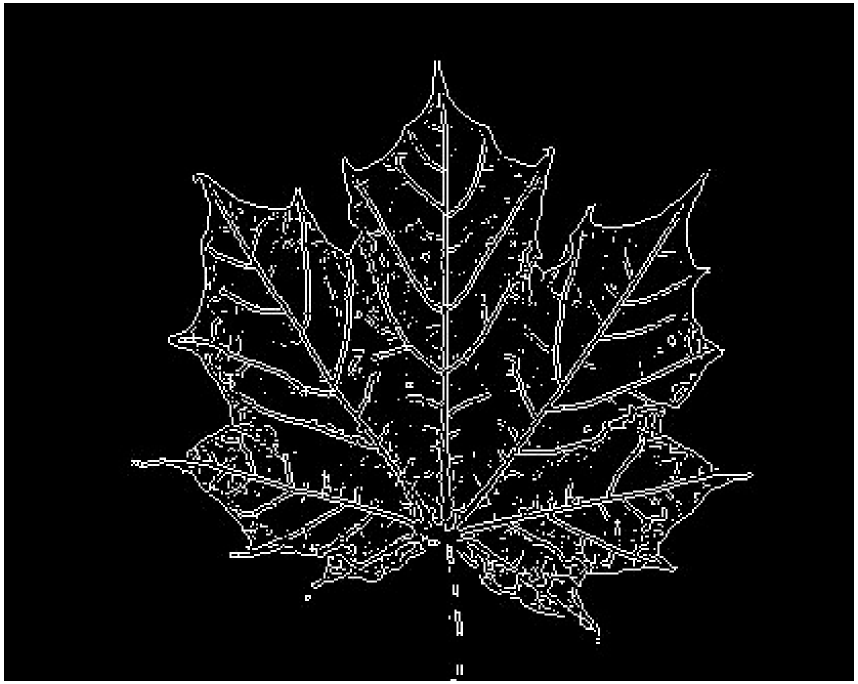
Filtre Gradient

>>edge(I,'prewitt')

Masque de Prewitt (amplitude du Gradient)



amplitude du gradient...



idem mais seuillé à 100

→ vers la **détection de contours**...

Filtre Laplacien

Définition

En continu 2D, le **laplacien** est défini par:

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

$$\frac{\partial^2 I}{\partial x^2} = I(x+1, y) + I(x-1, y) - 2I(x, y)$$

$$\frac{\partial^2 I}{\partial y^2} = I(x, y+1) + I(x, y-1) - 2I(x, y)$$

Filtre Laplacien

Masques possibles:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \text{ ou } \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

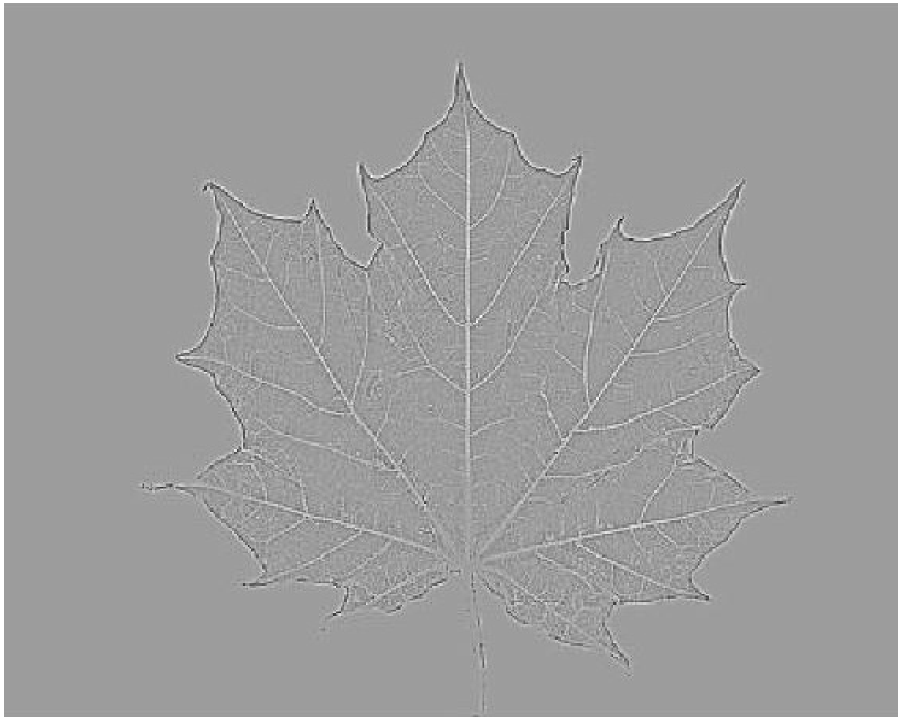
$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \text{ ou } \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

En incluant des termes sur les diagonales

→ invariant par rotation de 90°

→ invariant par rotation de 45°

Filtre Laplacien



Laplacien mis à l'échelle:

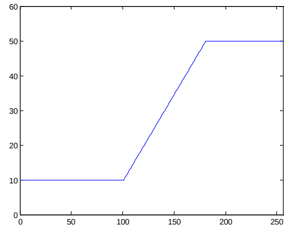
$$J = J - \min(J)$$

$$J = 255 * J / \max(J)$$

Filtre Laplacien

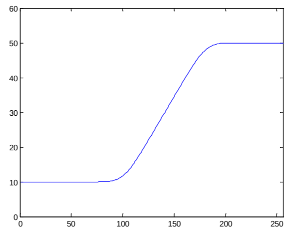
Rehaussement des contours : filtre Gaussien + Laplacien

I_1



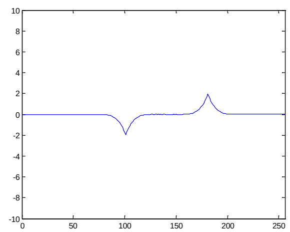
contour vu de profil (1D)

I_2



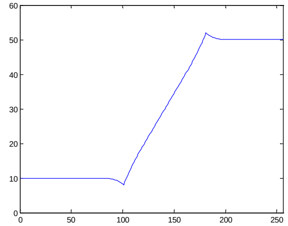
filtrage Gaussien

I_3



Laplacien (ou filtrage high-boost= différence entre I_1 et I_2)

I_4



$I_{1+}\, I_3$

Filtre Laplacien

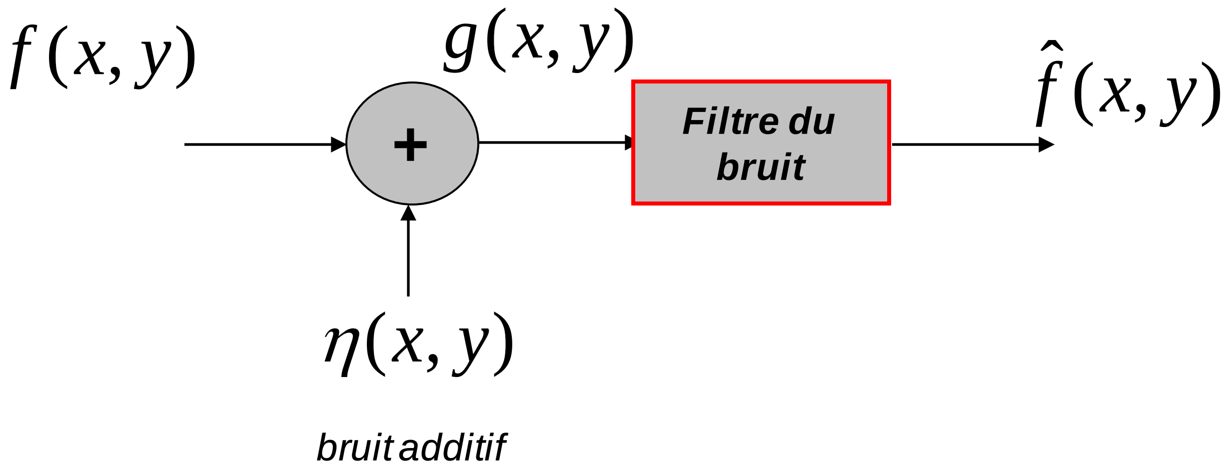
Rehaussement des contours: filtre Gaussien + Laplacien

$$I_g = G * I$$

$$I_r = I_g + \nabla^2 I_g$$

Filtrage en présence de bruit

Modèle de dégradation



$$g(x,y) = f(x,y) + \eta(x,y)$$

$$G(u,v) = F(u,v) + N(u,v)$$

Modèles de bruit

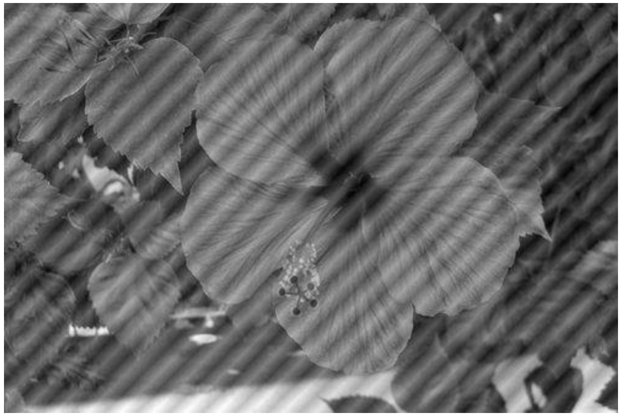
$$g(x,y) = f(x,y) + \eta(x,y)$$

Le bruit *n(x,y)* est généralement inséré dans la chaîne d'imagerie au moment de **l'acquisition** ou de la **transmission**.

Deux types de propriétés du bruit:

→ **spatiales** :le bruit est-il indépendant des coordonnées spatiales ?
On suppose que c'est le cas le plus souvent, **sauf** pour les bruits périodiques spatialement (c.f. Chapitre 5, pour éliminer ce type de bruit)

→ **fréquentielles**: de quelle forme est la TF du bruit ?
ex: si TF constante → bruit dit ***blanc***

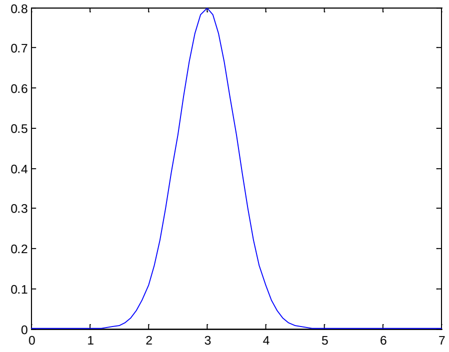


bruit spatial périodique

Densités de probabilités de bruit typique

Il existe plusieurs modèles de bruit classiques qui permettent ensuite de reconnaître ou d'estimer le bruit dans une image:

$$p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(z-\bar{z})^2/2\sigma^2}$$



Densité de probabilité
bruit Gaussien

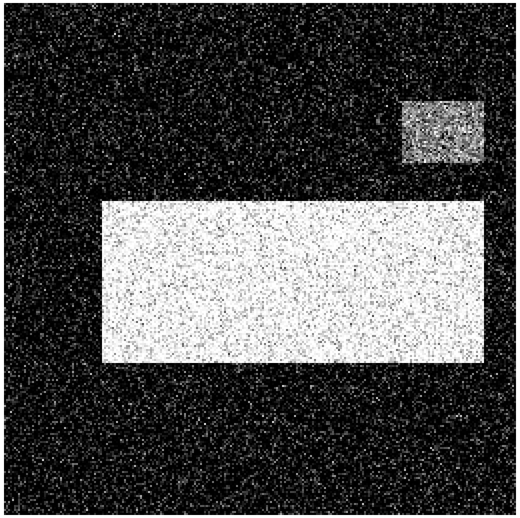
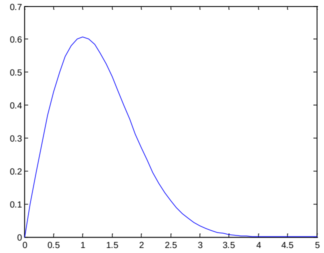


Image synthétique avec bruit Gaussien
σ=0.05

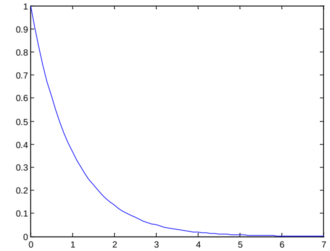
Densités de probabilités de bruit typique

$$p(z)=\begin{cases}\dfrac{2}{b}(z-a)e^{-(z-a)^2/b} & z\geq a\\ 0, & z<a\end{cases}$$



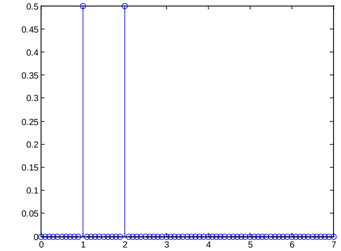
Densité de probabilité
bruit Rayleigh

$$p(z)=\begin{cases}ae^{-az}, & z\geq 0\\ 0, & z<0\end{cases}$$



Densité de probabilité
bruit exponentiel

$$p(z)=\begin{cases}p_{\mathrm{a}}, & z=a\\ p_{\mathrm{b}}, & z=b\end{cases}$$



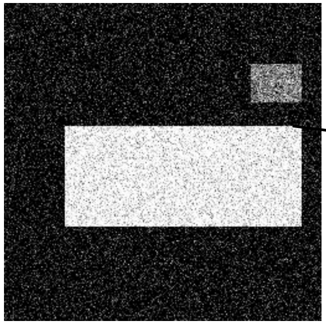
Densité de probabilité
bruit impulsionnel
ou ***poivre&sel***

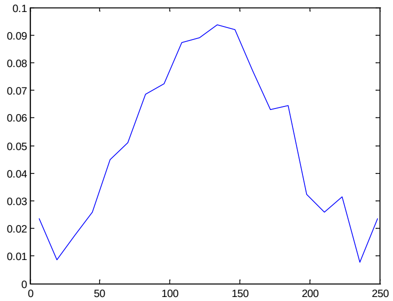
Estimation des paramètres du bruit

Pour estimer les paramètres du bruit additif présent on peut:

→ utiliser les données matérielles du système d'acquisition si existantes

→ rechercher une région homogène dans l'image, suffisamment grande pour en extraire un histogramme



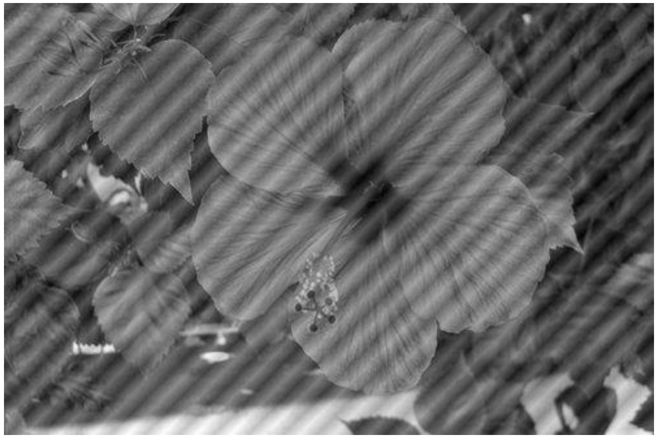


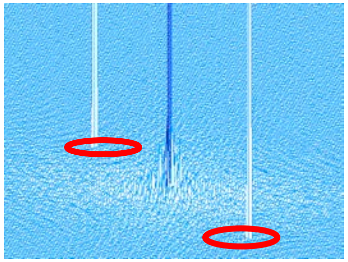
Histogramme **normalisé** de la sous-région

$$\bar{z} = \sum_{i=0}^{255} z_i p(z_i), \quad \sigma^2 = \sum_{i=0}^{255} (z_i - \bar{z})^2 p(z_i)$$

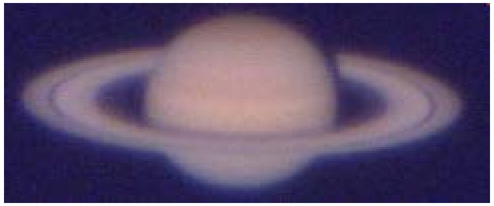
Image uniquement bruitée

Si bruit périodique spatial: l'analyse de *N(u,v)* permettra de détecter des pics anormaux et un filtre local (chapitre 5) permet souvent de supprimer ce bruit

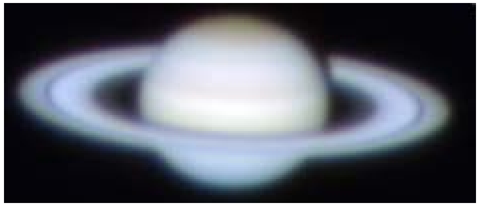




Si on dispose de plusieurs images de la scène: une très bonne solution consiste à faire la **moyenne des images** (ex: en astronomie avec une caméra vidéo).



x 2000 →



moyenne de 2000 images

Image uniquement bruitée

$$g(x,y) = f(x,y) + \eta(x,y)$$

Pour les autres cas: les filtres spatiaux vus au chapitre 4 sont les mieux adaptés. En considérant S_{nm} un voisinage rectangulaire de taille *[n,m]* centré en chaque point de l'image, on peut utiliser un filtre:

→ **moyenneur**

filtre arithématique
(filtre moyenneur classique)

$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(i,j)\in S_{nm}} g(i,j)$$

++ simple
-- génère du flou

Image uniquement bruitée

$$g(x, y) = f(x, y) + \eta(x, y)$$

On peut aussi appliquer d'autres filtres, sans convolution :

filtre géométrique

$$\hat{f}(x, y) = \left[\prod_{(i,j)\in S_{nm}} g(i, j) \right]^{1/nm}$$

un peu moins de flou que le filtre arithmétique

filtre harmonique

$$\hat{f}(x, y) = \frac{mn}{\sum_{(i,j)\in S_{nm}} 1/g(i, j)}$$

++ bruit sel (impulsion positive)
-- bruit poivre

filtre contraharmonique d'ordre *Q*

$$\hat{f}(x, y) = \frac{\sum_{(i,j)\in S_{nm}} g(i, j)^{Q+1}}{\sum_{(i,j)\in S_{nm}} g(i, j)^{Q}}$$

Q>0 → bon pour bruit poivre
Q<0 → bon pour bruit sel

Image uniquement bruitée

→ **Gaussien**: convolution avec un masque Gaussien de taille [n,m]

filtre Gaussien

$$\hat{f}(x,y) = g(x,y) * e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)}$$

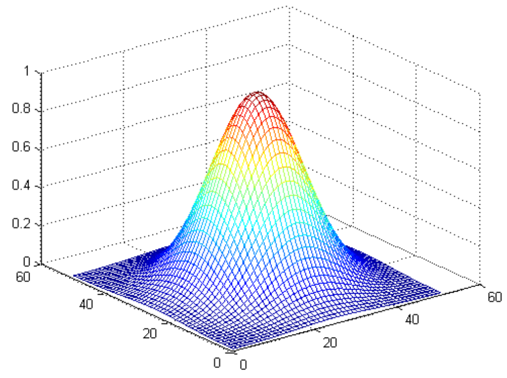


Image uniquement bruitée

→ **filtres d'ordre statistique**

filtre Médian

$$\hat{f}(x,y) = \underset{(i,j)\in S_{nm}}{median}\{g(i,j)\}$$

++++ *bruit poivre et/ou sel*

filtre Min

$$\hat{f}(x,y) = \min_{(i,j)\in S_{nm}}\{g(i,j)\}$$

++ *bruit sel et pour respecter les zones sombres*

filtre Max

$$\hat{f}(x,y) = \max_{(i,j)\in S_{nm}}\{g(i,j)\}$$

++ *bruit poivre et pour respecter les zones claires*

filtre dichotomique

$$\hat{f}(x,y) = \frac{1}{2}\left[\min_{(i,j)\in S_{nm}}\{g(i,j)\} + \max_{(i,j)\in S_{nm}}\{g(i,j)\}\right]$$

++ *bruit Gaussien*

Plan

- Du 1D vers le 2D
- Histogramme d’une image
- Transformation d’histogramme
- Filtres passe-bas
- Filtres passe-haut
- Filtres morphologiques
- Filtrage en présence de bruit

Plan

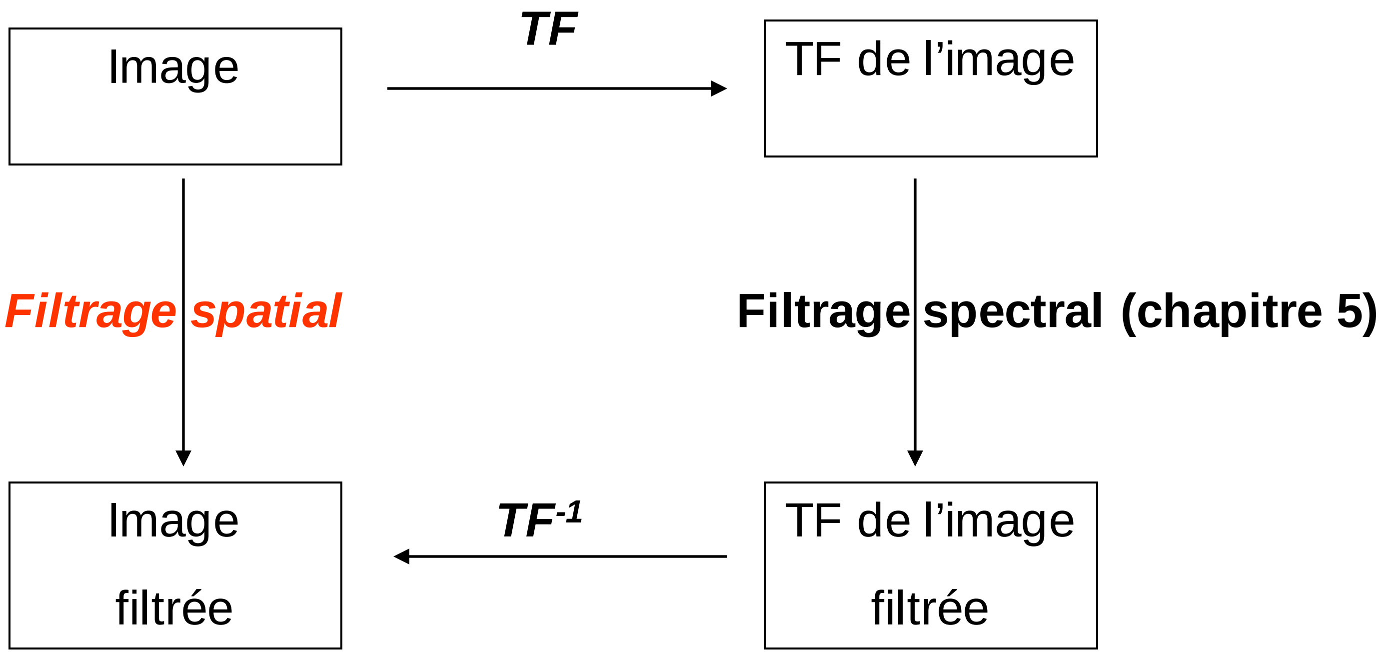


Image
TF
TF de l'image
Filtrage spatial
Filtrage spectral (chapitre 5)
Image
filtrée
TF-1
TF de l'image
filtrée

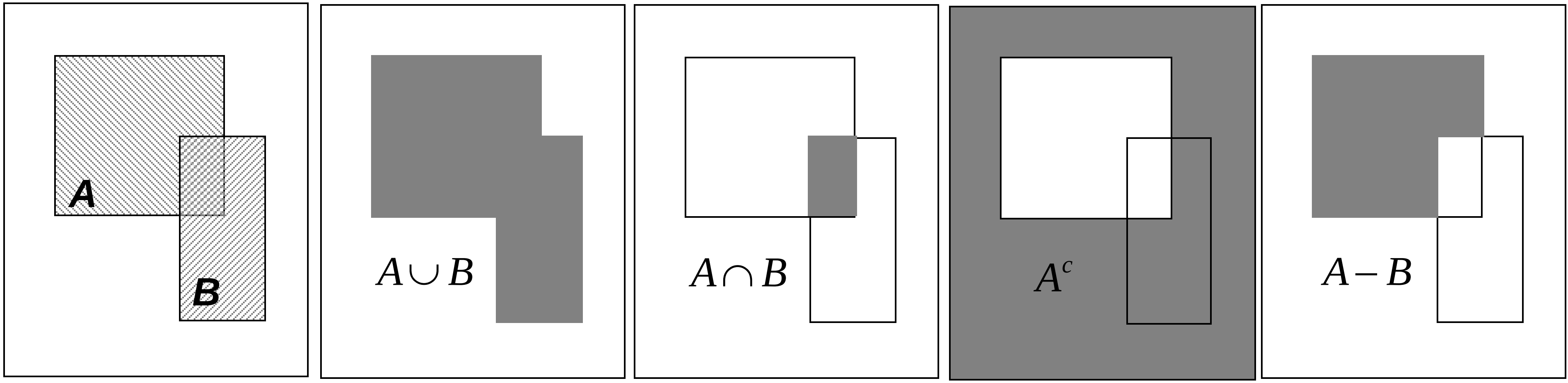
Filtrage morphologique

Principe

Le traitement d'images basé sur la morphologie mathématiques utilise la **théorie des ensembles**

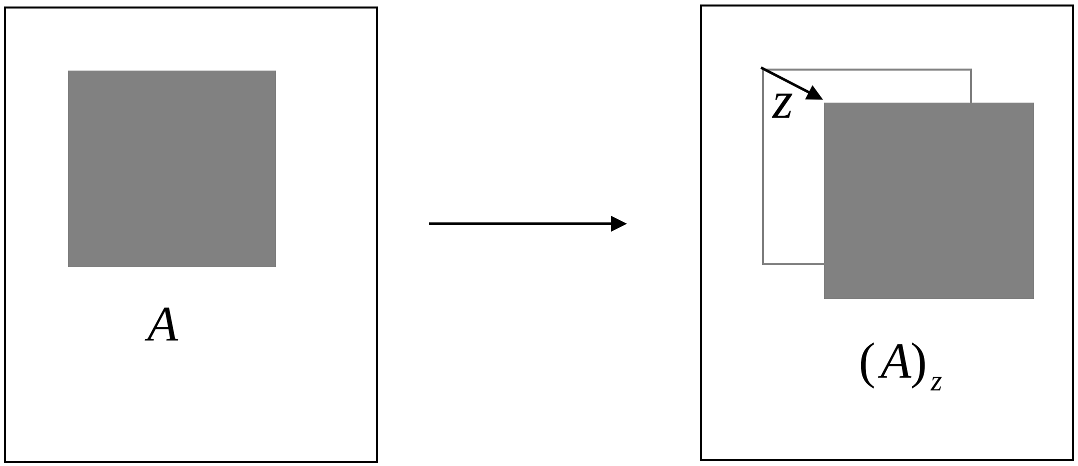
La morphologie mathématique a été inventée au Centre de Morphologie Mathématique en France depuis 1964 par Matheron et Serra, et continue d'être développée.

En imagerie, on parle d'ensemble de coordonnées de points:



union *intersection* *complémentaire* *différence*

Translation



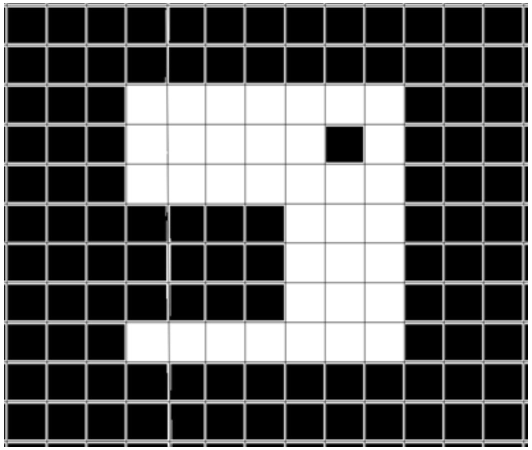
Erosion

Définition

L'érosion de l'ensemble A par l'ensemble B est défini par

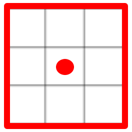
$$A \Theta B = \{ z \mid (B)_z \subseteq A \}$$

Principe:



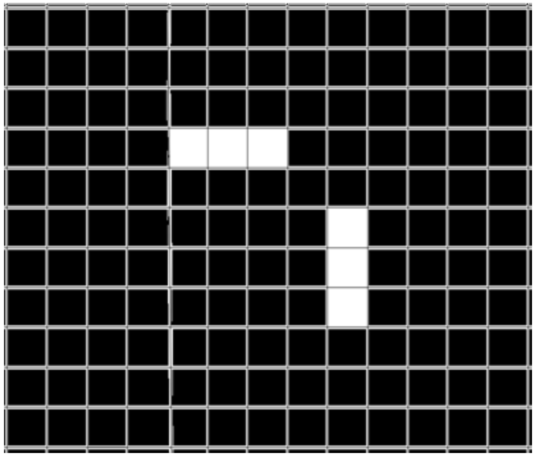
A

Θ



B

$$z \mid (B)_z \subseteq A \rightarrow Vrai / Faux$$



$A \Theta B$

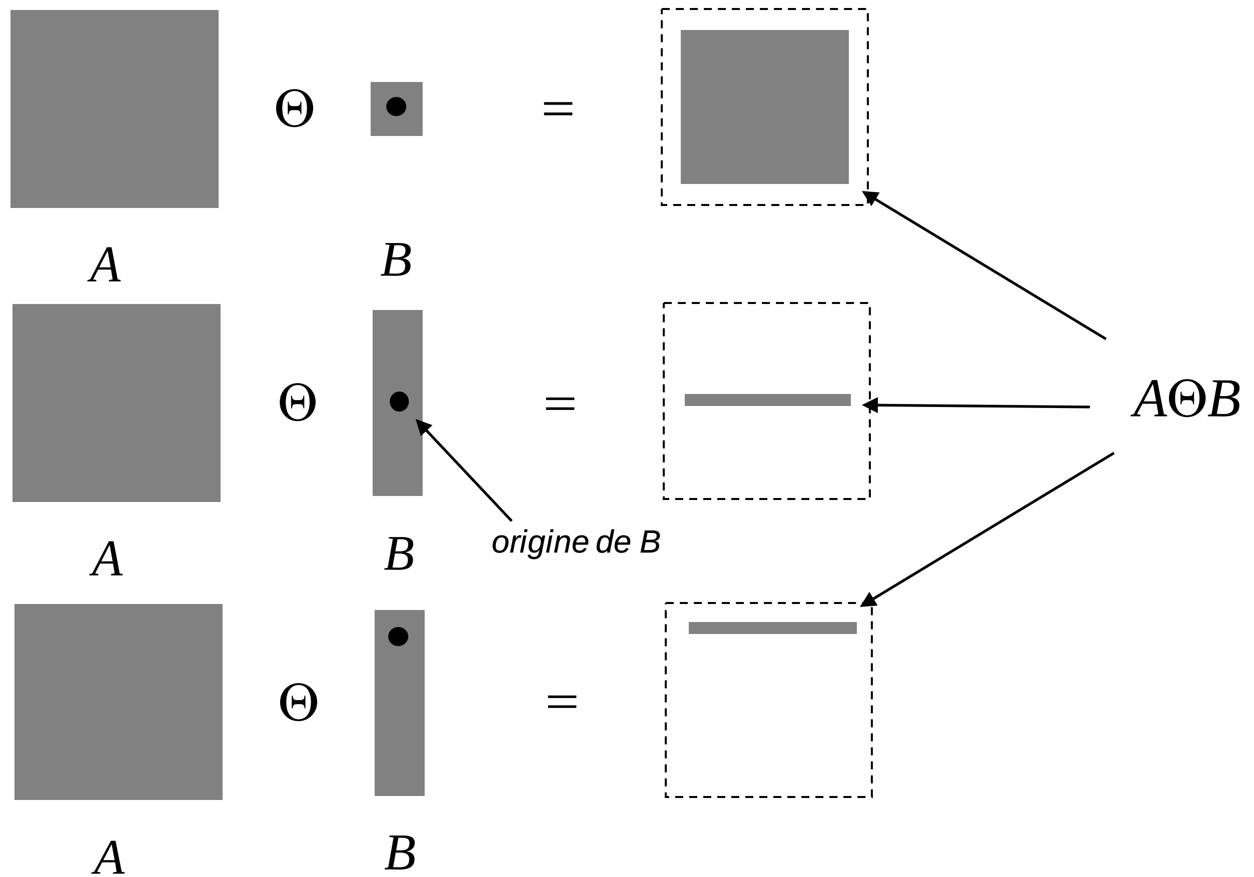
Erosion

Définition

L'érosion de l'ensemble A par l'ensemble B est défini par

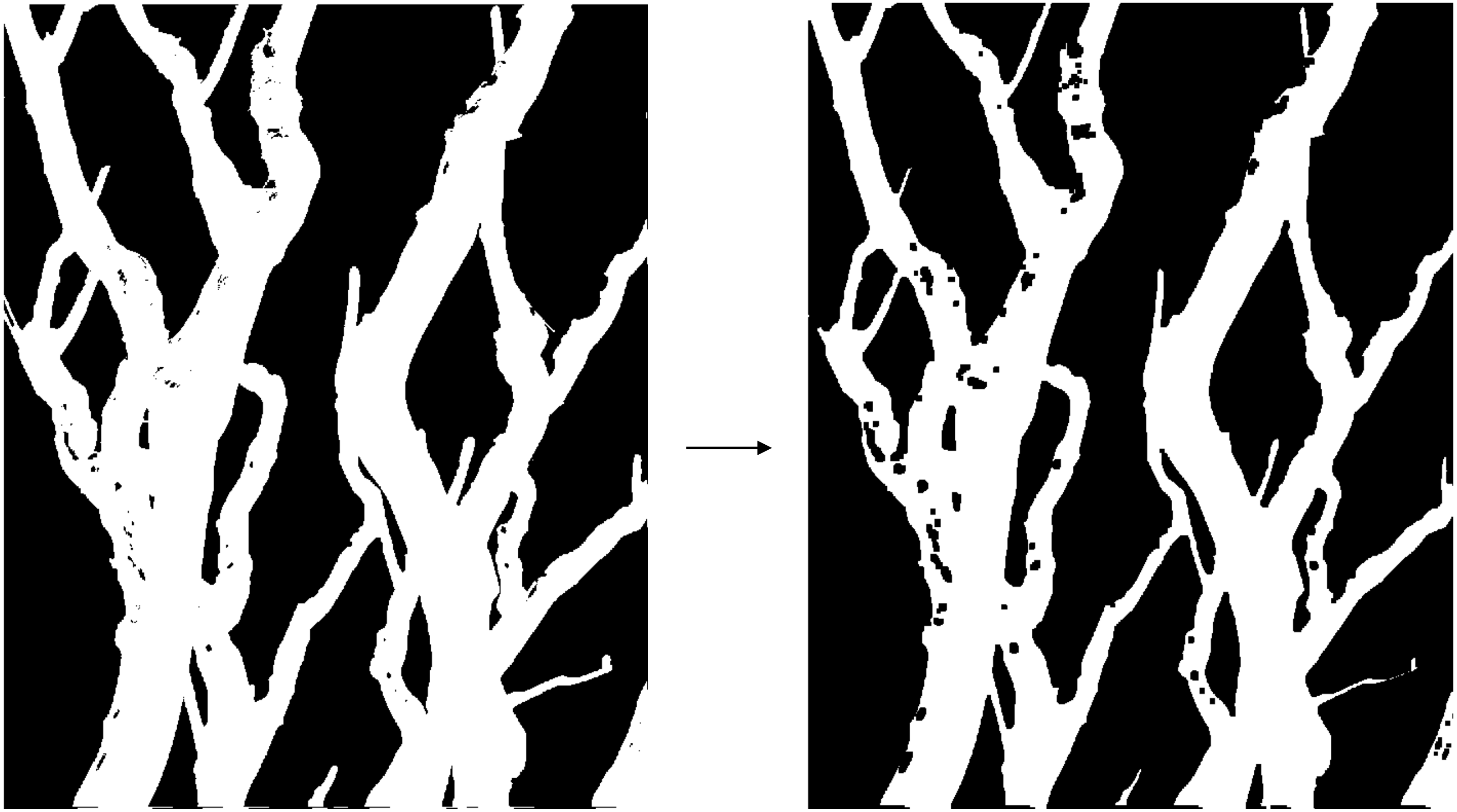
$$A \Theta B = \{ z \mid (B)_z \subseteq A \}$$

ex:



Erosion

ex: érosion avec un carré de taille 7*7



Erosion

<u>ex:</u> image originale – image érodée = contours !

—

=

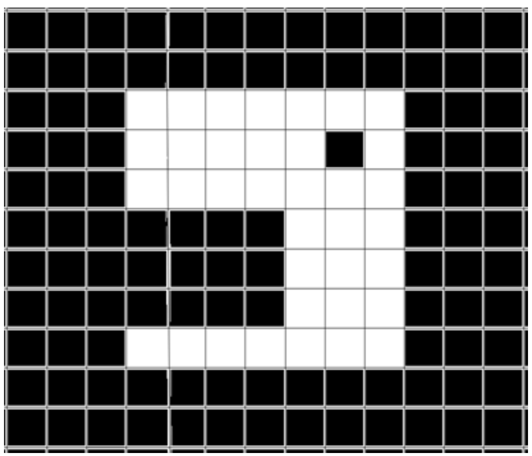
Dilatation

Définition

La dilatation de l'ensemble A par l'ensemble B est défini par

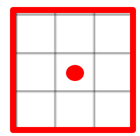
$$A \oplus B = \{ z = a + b \mid a \in A, b \in B \}$$

<u>Principe:</u>



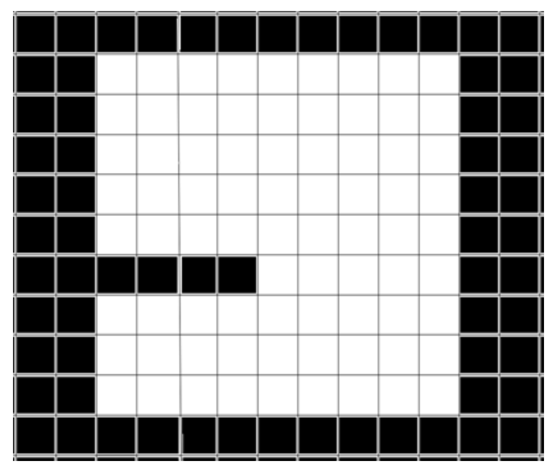
A

$\oplus$



B

$$z = a + b \mid a \in A, b \in B \rightarrow \mathit{Vrai}\ \mathit{Faux}$$



$A \oplus B$

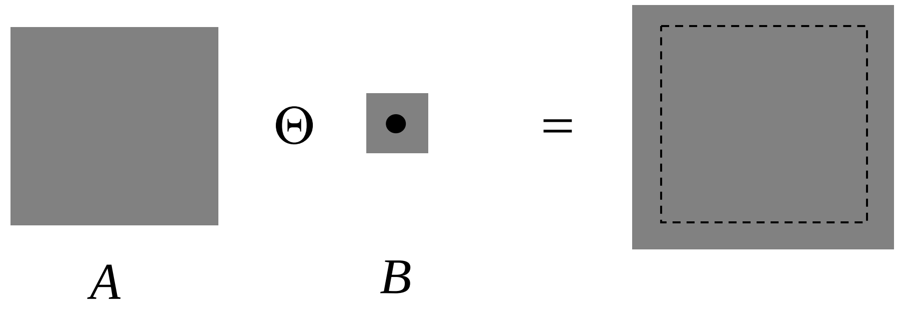
Dilatation

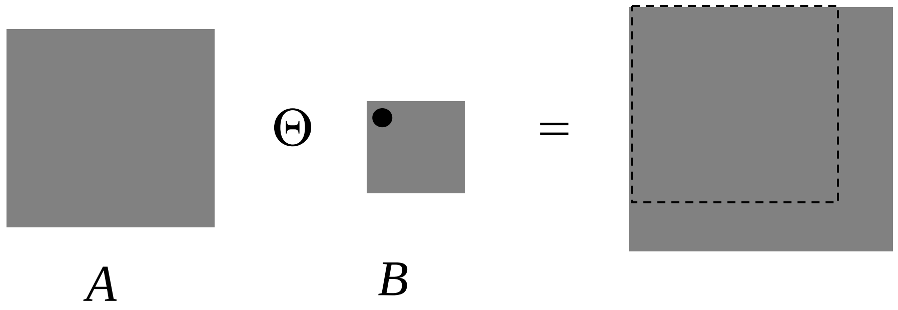
Définition

La dilatation de l'ensemble A par l'ensemble B est défini par

$$A \oplus B = \{z = a + b \mid a \in A, b \in B\}$$

ex:





Dilatation

ex: dilatation avec un carré de taille 11*11

Ouverture / Fermeture

Définition

L'ouverture de l'ensemble A par l'ensemble B est définie par

$$A \circ B = (A \Theta B) \oplus B$$

→érosion suivie d'une dilatation

La fermeture de l'ensemble A par l'ensemble B est définie par

$$A \bullet B = (A \oplus B) \Theta B$$

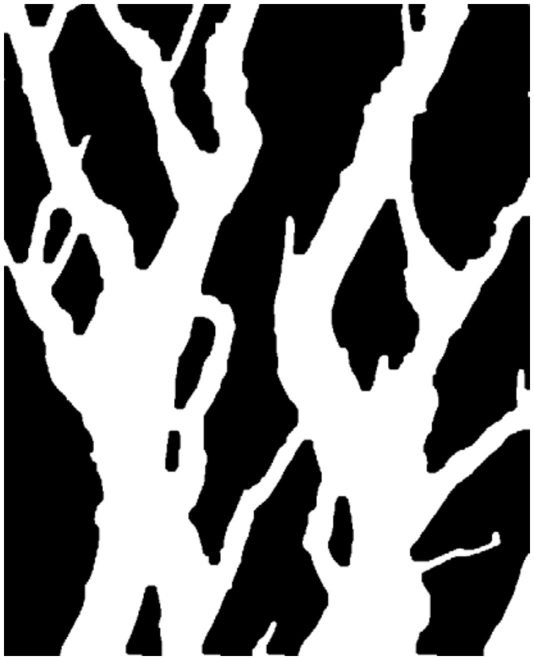
→ dilatation suivie d'une érosion

Ouverture / Fermeture

ex: élément structurant: cercle de rayon 7

image originale

ouverture



fermeture

Ouverture / Fermeture

ex: élément structurant: cercle de rayon 15

image originale

ouverture+fermeture

→ peut servir à éliminer le bruit

Morphologie et images en niveaux de gris

Principe

On applique les mêmes principes mais aux surfaces 3D représentées par les images en niveau de gris

érosion

dilatation

fermeture

ouverture

Morphologie et images en niveaux de gris

image originale

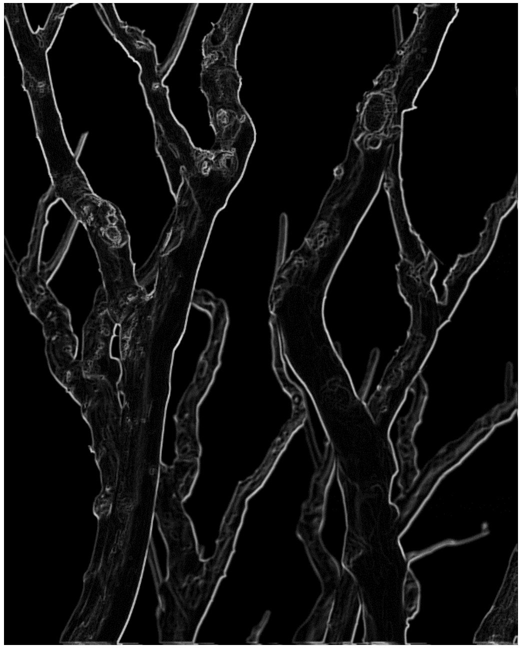


image originale – image érodée

Morphologie et images en niveaux de gris

image originale

image ouverte puis fermée

Fonctions Matlab

>>strel(*shape*,parametres)	construit un élément structurant disque, carré, arbitraire, etc...
>>imerode(I,se)	érosion (se = élément structurant)
>>imdilate(I,se)	dilatation
>>imopen(I,se)	ouverture
>>imclose(I,se)	fermeture