

# Traitement des images (Partie 1: pré-traitement)

Patrick Hébert & Denis Laurendeau (Dernière révision : mai 2017)

*Références utiles:*

Sonka et al : 3.1, 3.2 jusqu'à 3.2.5, 3.3, 5.3.1,  
13.1 à 13.4

- Le traitement des images se place aux *premières étapes* du processus de vision artificielle
- Les opérations sur les images effectuées à cette étape se font au *niveau des pixels*. Aucune interprétation de haut niveau du contenu de l'image n'est tentée.

# Remarques préliminaires

Le traitement des images dans un contexte de vision artificielle est utile pour:

- 1. Restaurer** le contenu (e.g. atténuer les effets du bruit)
- 2. Rehausser** certains éléments dans les images (e.g. mettre en évidence les contours (discontinuités d'illuminance))
- 3. Compresser** le contenu des images en supprimant les informations redondantes (moins important pour le cours de vision)

Les concepts précédents (restauration, rehaussement, compression) sont assimilés à des opérations de *filtrage*

Les **objectifs** du cours sur le traitement des images (partie 1) sont de:

1. Présenter quelques notions sur les différents types de **bruit** pouvant influencer le contenu d'une image
2. Présenter des approches de **filtrage linéaire**
  1. filtre moyenneur uniforme
  2. filtre gaussien (et conception d'un filtre gaussien)
3. Présenter des approches de **filtrage non-linéaire**
  1. Filtrage médian
  2. Filtrage bilatéral
  3. Morphologie mathématique
4. Présenter des outils pour la détection de caractéristiques

# Le *bruit* dans les images

- Le cours d'introduction a présenté brièvement quelles sont les sources de bruit lors de l'acquisition d'une image par une caméra réelle
- Dans ce qui suit, nous allons présenter les approches permettant de réduire les effets du bruit dans les images



# Qu'est ce que le bruit?

De manière générale, le bruit est tout ce qu'il n'est pas possible de décrire ou modéliser de manière déterministe.

On doit par conséquent utiliser des modèles stochastiques pour le caractériser.

Modélisation stochastique simple pour le bruit en un pixel donné:

- Pour une scène statique (i.e. sans mouvement et changement d'éclairage):
  - faire l'acquisition de plusieurs images et examiner la distribution de l'illuminance en fonction du temps

Deux paramètres fondamentaux permettant de décrire cette distribution temporelle de l'illuminance sont les deux premiers moments statistiques:

1. La moyenne de l'illuminance pour l'ensemble des images (échantillons)
2. l'écart-type de l'illuminance pour l'ensemble des images (échantillons)

# Pour un pixel (i,j) dans l'image

Moyenne (premier moment)

$$\bar{E}(i,j) = \frac{1}{N} \sum_{k=1}^N E_k(i,j) \quad (1)$$

Écart-type (second moment)

$$\sigma(i,j) = \left( \frac{1}{N-1} \sum_{k=1}^N (E_k(i,j) - \bar{E}(i,j))^2 \right)^{\frac{1}{2}} \quad (2)$$



On prend  $1/(N-1)$  pour avoir une estimation non biaisée quand le nombre d'échantillons est petit ( $<75$ )

# Principaux *types* de bruit

Les principaux types de bruit que nous considèrerons dans le cadre du cours sont:

1. Le bruit additif
2. le bruit multiplicatif (impulsionnel)
3. le bruit de quantification
4. le bruit d'échantillonnage

# Bruit additif

On fait généralement l'hypothèse que le bruit est *additif* (le bruit s'additionne à un signal pur):

$$\hat{E}(i, j) = E(i, j) + n(i, j) \quad (3)$$

The diagram illustrates the additive noise model equation  $\hat{E}(i, j) = E(i, j) + n(i, j)$ . Three arrows point from labels below to the terms in the equation: a horizontal arrow from "signal bruité au pixel (i,j)" points to  $\hat{E}(i, j)$ ; a vertical arrow from "signal pur au pixel (i,j)" points to  $E(i, j)$ ; and a horizontal arrow from "bruit additif au pixel (i,j)" points to  $n(i, j)$ .



L'importance du bruit est caractérisée par le ***rapport signal à bruit*** (Signal-to-Noise Ratio (SNR)):

$$SNR = \frac{\sigma_{signal}}{\sigma_{bruit}} \quad (4)$$

Cette quantité est souvent exprimée en ***décibels***:

$$SNR_{db} = 10 \log_{10} \left( \frac{\sigma_{signal}}{\sigma_{bruit}} \right) \quad (5)$$

# Bruit multiplicatif

On peut aussi rencontrer du bruit ***multiplicatif***:

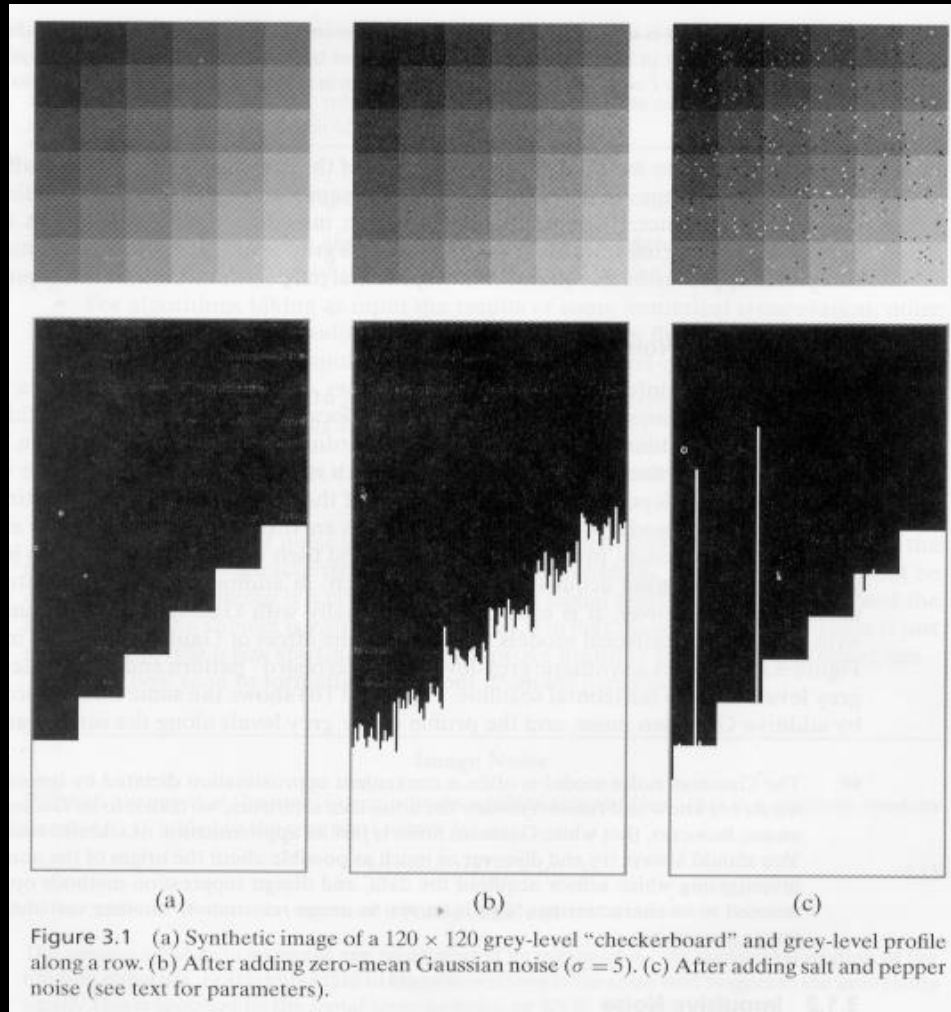
$$\hat{E}(i, j) = E(i, j) \cdot n(i, j) \quad (6)$$

Ce type de bruit, aussi appelé bruit ***impulsionnel***, provoque l'apparition de valeurs isolées très différentes des valeurs aux pixels voisins

Ce type de bruit peut provoquer une ***saturation*** d'illuminance à certains pixels de l'image

# Exemples de bruit

Signal sans bruit →



← Bruit impulsionnel (multiplicatif)

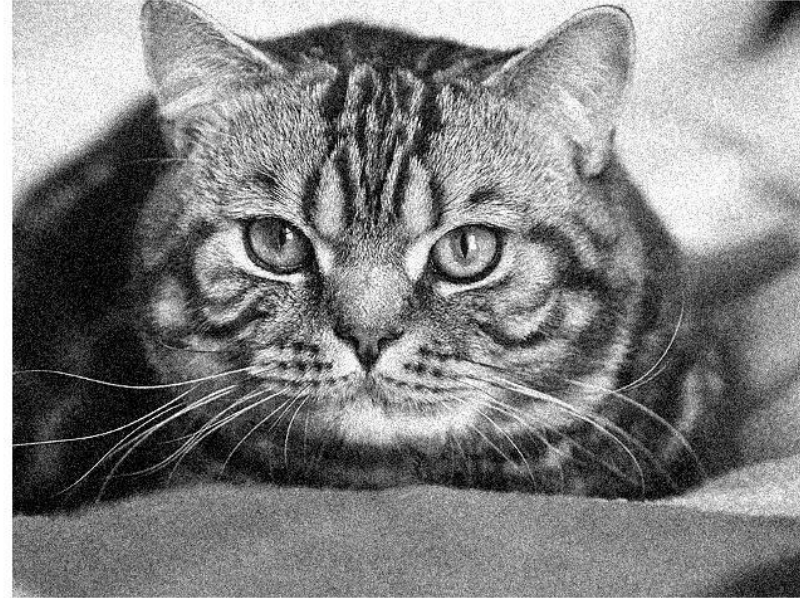
↑  
Bruit gaussien (additif)

# Bruit gaussien

image noir et blanc



image noir et blanc avec bruit gaussien ajoute



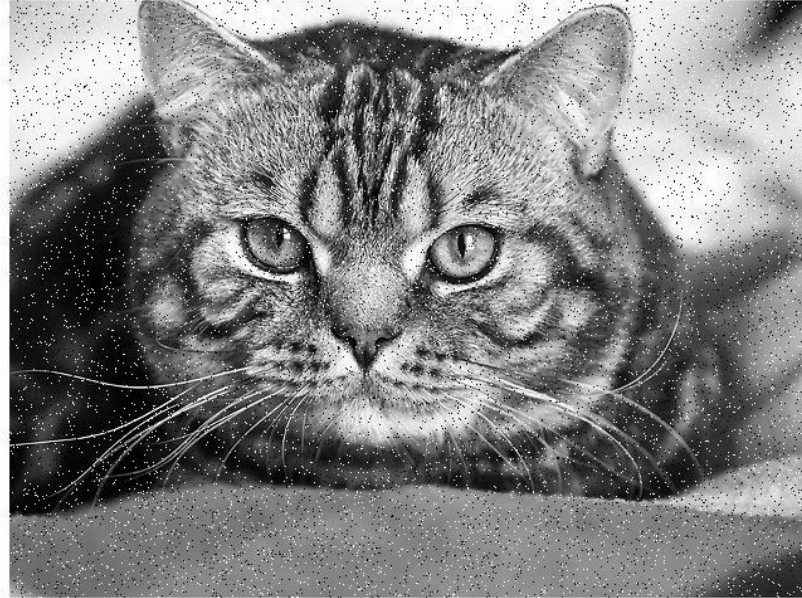


# Bruit impulsif

image noir et blanc



image avec bruit impulsif ajoute



# Bruit de quantification

Ce type de bruit est causé par la *résolution* avec laquelle l'illuminance est échantillonnée en chaque pixel:

- la plupart des cartes d'échantillonnage utilisent une plage de quantification (résolution en amplitude) de **8 bits** (i.e. l'illuminance en un pixel est répartie sur 256 valeurs)
- ...mais on peut aussi avoir des cartes (plus chères) offrant des plages d'échantillonnage de 10 ou 12 bits



# Bruit d'échantillonnage

Ce type de bruit est causé par la *résolution spatiale* avec laquelle le champ de vision observé par la caméra est échantillonné:

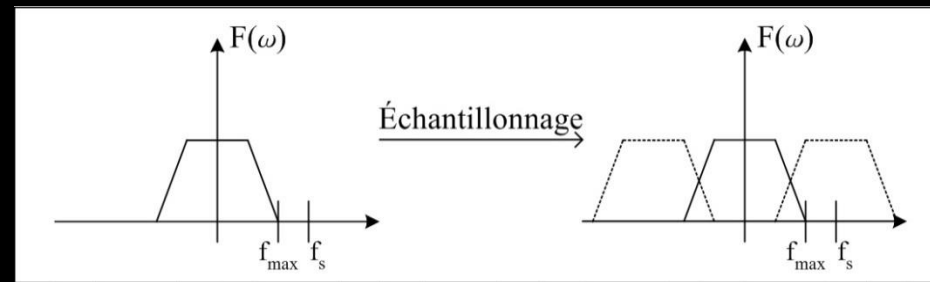
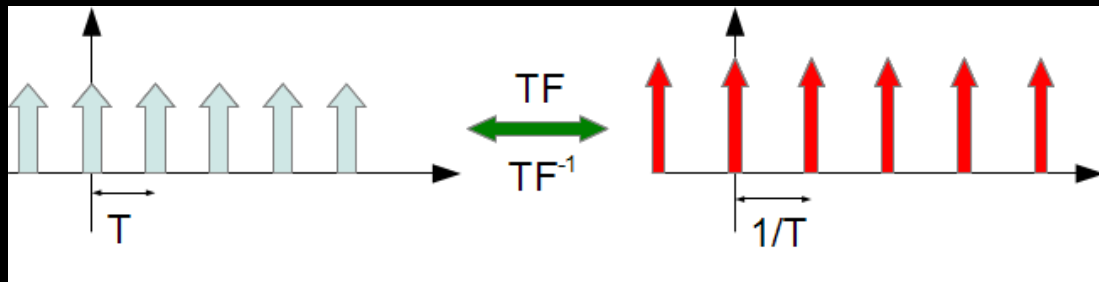
- exemple classique: 640 colonnes x 480 lignes
- 1024 x 768
- etc.

Ce bruit d'échantillonnage spatial a les effets suivants:

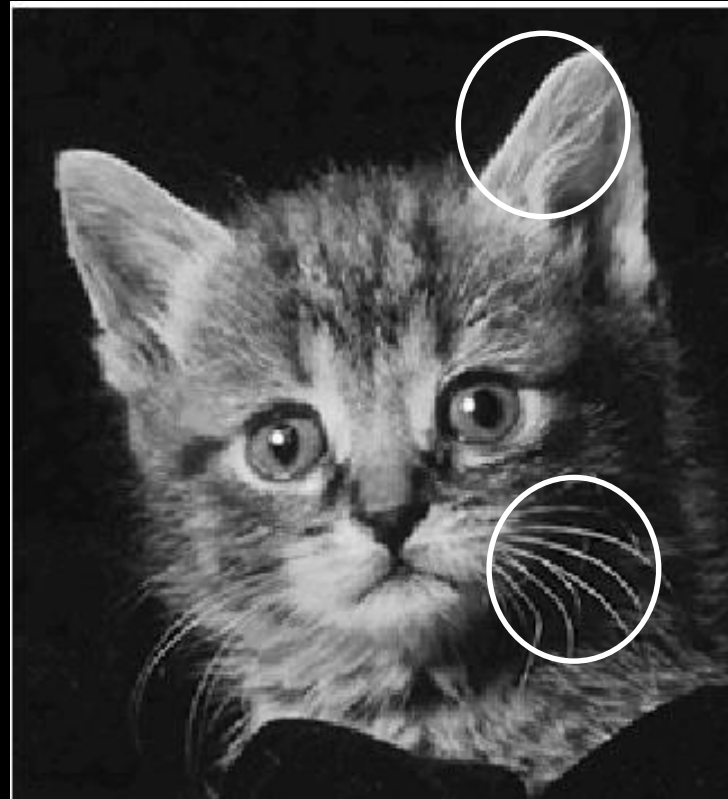
- affecte la netteté des variations rapides (contours) d'illuminance dans l'image
- amène des effets de *recouvrement spectral* (aliasing) lorsque le *critère d'échantillonnage de Nyquist* n'est pas respectée

# Critère d'échantillonnage de Nyquist (pour éviter le recouvrement spectral):

$$f_{\max} < \frac{f_{\text{ech}}}{2} \quad (7)$$



# Exemple de l'effet du recouvrement spectral causé par l'échantillonnage spatial



# Filtrage pour limiter les effets indésirables du bruit

Le filtrage a pour but de *réduire* les effets du bruit *sans affecter trop le signal*

- On peut considérer deux types de filtres:
  - les filtres linéaires
  - les filtres non-linéaires

# Filtrage *linéaire* - introduction



Un filtre *linéaire* possède les propriétés suivantes:

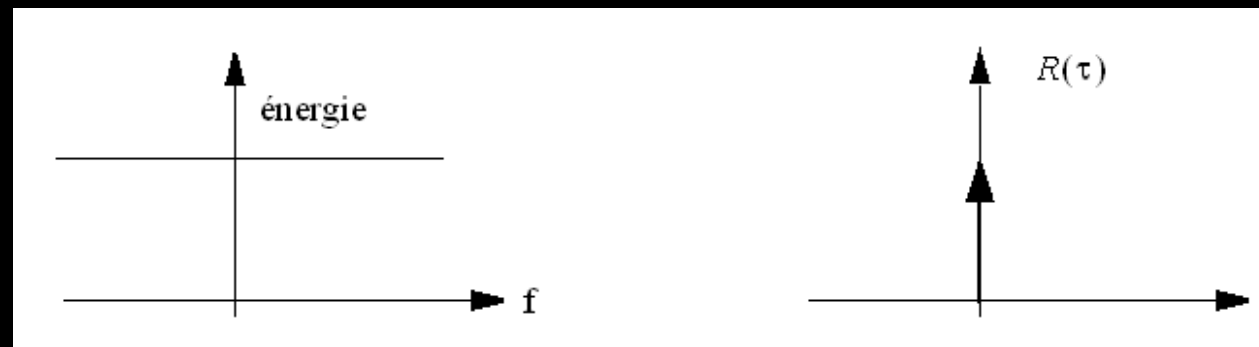
- *invariance à la position*: le résultat du filtrage ne dépend que du *voisinage* d'un pixel et non de la position absolue du pixel dans l'image
- respecte les principes de *mise à l'échelle* et de *superposition* (le résultat du filtrage de la somme de deux images est le même que celui de la somme des deux images filtrées):

$$f(ax + by) = af(x) + bf(y) \quad (8)$$

Pour le bruit en un pixel de l'image, on fait souvent l'hypothèse de **bruit blanc gaussien** de moyenne nulle

Bruit blanc: indépendant d'un pixel à l'autre  
(espace) et d'une image à l'autre  
(temps)

Gaussien: distribution d'amplitude gaussienne



L'hypothèse fondamentale derrière le filtrage linéaire est que la *moyenne* de plusieurs échantillons devrait *réduire le bruit* (i.e. l'écart-type du signal résultat du moyennage de  $N$  échantillons devrait être plus faible que celui de la distribution de laquelle proviennent ceux-ci).

Supposons  $N$  échantillons  $x_i$  tirés d'une même distribution statistique de moyenne  $\mu$  et de variance  $\sigma^2$ .

La moyenne  $\bar{x}$  des  $N$  échantillons est donnée par:

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_N}{N} = \frac{1}{N}(x_1 + x_2 + \dots + x_N) \quad (9)$$

Comme les  $N$  échantillons sont des variables aléatoires identiquement distribuées (parce que tirées de la même population) on a que :

$$E(x_i) = \mu \quad (10)$$

$$Var(x_i) = \sigma^2 \quad (11)$$

Regardons maintenant quelle est la moyenne  $E(\bar{x})$  et la variance  $Var(\bar{x})$  de  $\bar{x}$

En utilisant les identités suivantes:

$$E(nx) = nE(x) \quad (12)$$

$$Var(nx) = n^2Var(x) \quad (13)$$

et en utilisant la partie de droite de (9) :

$$E(\bar{x}) = E\left(\frac{1}{N}(x_1 + x_2 + \dots + x_N)\right) = \frac{1}{N}E(x_1 + x_2 + \dots + x_N) \quad (14)$$

$$Var(\bar{x}) = Var\left(\frac{1}{N}(x_1 + x_2 + x_3 + \dots + x_N)\right) = \frac{1}{N^2}Var(x_1 + x_2 + x_3 + \dots + x_N) \quad (15)$$

De plus, on sait que:

$$E(x_1 + x_2 + x_3 + \cdots x_N) = E(x_1) + E(x_2) + E(x_3) + \cdots E(x_N) \quad (15)$$

et

$$Var(x_1 + x_2 + x_3 + \cdots x_N) = Var(x_1) + Var(x_2) + Var(x_3) + \cdots Var(x_N) \quad (16)$$

Avec (1) et (11) :

$$E(x_1 + x_2 + x_3 + \cdots x_N) = E(x_1) + E(x_2) + E(x_3) + \cdots + E(x_N) = \mu + \mu + \mu + \cdots + \mu = n\mu \quad (17)$$

$$Var(x_1 + x_2 + x_3 + \cdots x_N) = Var(x_1) + Var(x_2) + Var(x_3) + \cdots Var(x_N) = \sigma^2 + \sigma^2 + \sigma^2 + \cdots + \sigma^2 = n\sigma^2 \quad (18)$$

Finalemment, en combinant (12) et (13) et (17) et (18) on obtient le résultat important suivant:

$$E(\bar{x}) = n\left(\frac{1}{N}\mu\right) = \mu \quad (19)$$

$$Var(\bar{x}) = n\left(\frac{1}{N^2}\sigma^2\right) = \frac{1}{N}\sigma^2 \quad (20)$$

qui dit qu'en moyennant N valeurs d'illuminance en un pixel, la moyenne demeure la même, mais la variance est réduite de 1/N, déduisant ainsi l'importance du bruit.

Filtrage *linéaire* – convolution dans  
le domaine spatial (i.e. l'image) et  
filtre moyennneur simple



En traitement de signal classique, on effectue souvent le filtrage d'un signal bruité dans le domaine *fréquentiel*. Selon cette procédure, on multiplie la transformée de Fourier  $S(\omega)$  du signal temporel  $s(t)$  par le filtre  $F(\omega)$ :

$$S_{\text{filtre}}(\omega) = F(\omega)S(\omega) \quad (20)$$

En vision artificielle, l'utilisation de la transformée de Fourier est peu répandue car il est plus naturel de travailler *directement dans l'image*.

Or, on sait qu'une *multiplication* dans le domaine des fréquences est équivalente à une *convolution* dans le domaine spatial, c'est-à-dire:

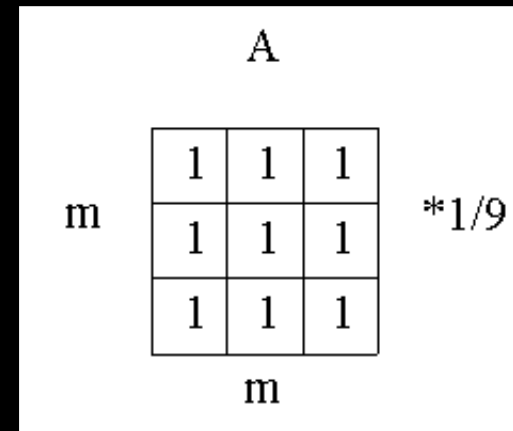
$$f(t) \bullet s(t) \Leftrightarrow F(\omega)S(\omega) \quad (21)$$

Dans le cadre du cours, le **filtrage linéaire** se fera donc par la **convolution** d'un filtre exprimé dans le domaine spatial avec l'image. On appelle ce filtre un opérateur de convolution.

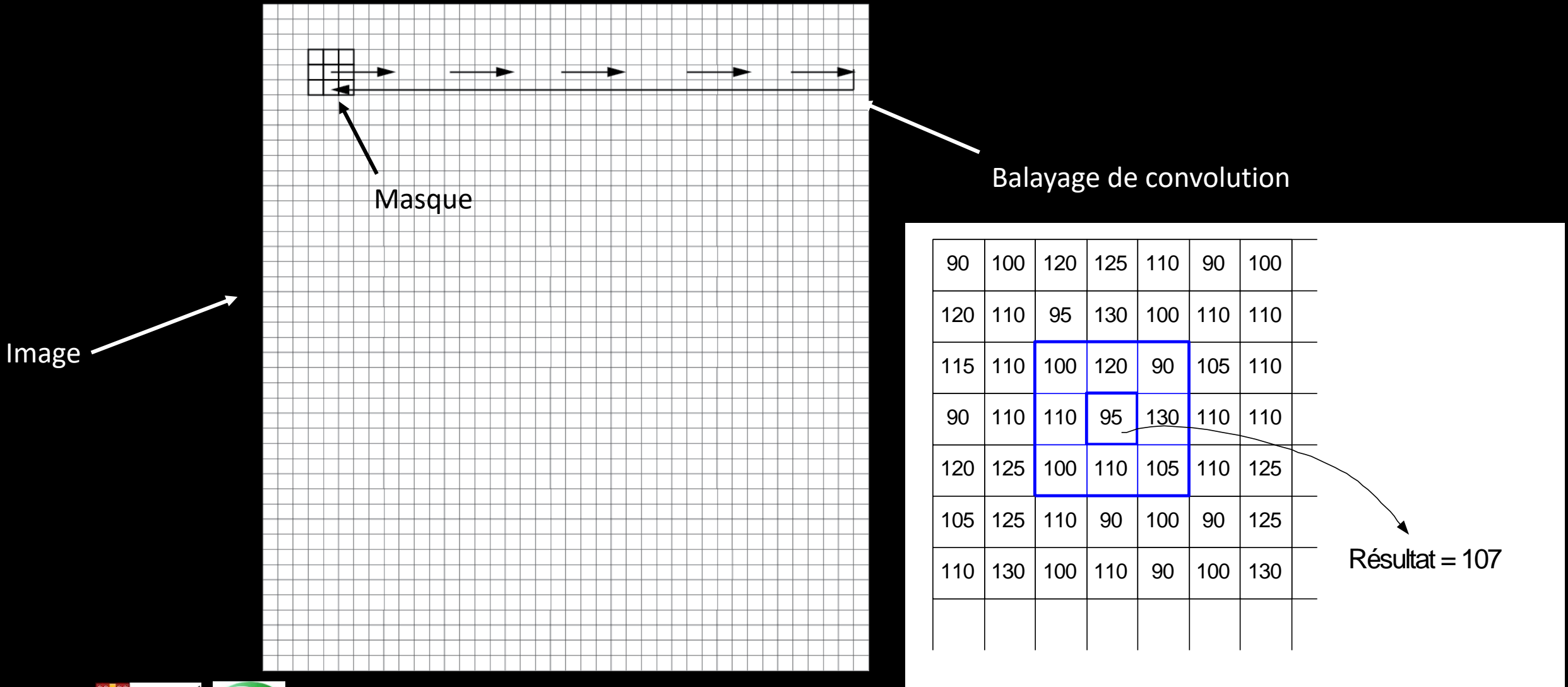
Cet opérateur de convolution prend la forme d'un **masque** ou **noyau** ("kernel") de convolution.

L'opération de filtrage consiste à **convoluer** ce masque avec l'image.

Exemple de masque de convolution: le filtre moyenneur



La convolution consiste à balayer l'image avec le masque et tel qu'illustré ci-dessous



L'équation de convolution pour le filtrage linéaire d'une image  $E(x,y)$  avec un filtre de noyau  $A(h,k)$  est la suivante pour chaque pixel d'illuminance  $E(i,j)$ :

$$E_{\text{filtre}}(i,j) = A(h,k) \cdot E(i,j) = \frac{1}{m^2} \sum_{h=-\frac{m}{2}}^{\frac{m}{2}} \sum_{k=-\frac{m}{2}}^{\frac{m}{2}} A(h,k)E(i-h,j-k) \quad (22)$$

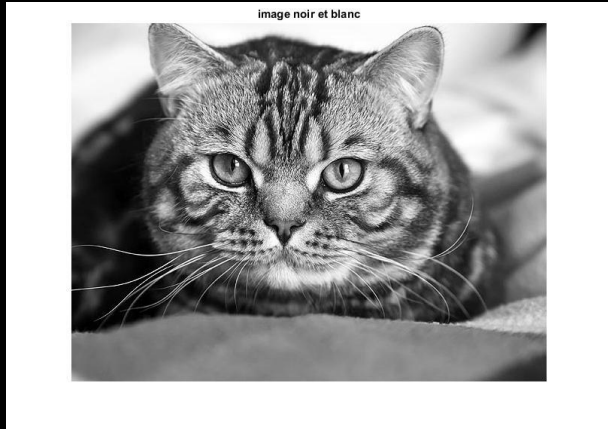
Pour chaque pixel  $(i,j)$ , les valeurs d'illuminance  $E(i,j)$  des pixels couverts par le masque  $A(h,k)$  sont multipliées par les valeurs du masque et additionnées pour produire la moyenne en multipliant par la taille du masque ( $1/m^2$ )

Comme nous l'avons vu à l'équation (20), le moyennage permet de réduire les effets du bruit.

En observant l'équation (22), on constate que:

1. il est plus pratique d'utiliser des masques pour lesquels  $m$  est *impair*
2. plus  $m$  augmente, plus le moyennage s'effectue sur une grande région autour du pixel  $(i,j)$  et plus le filtrage est important
3. le filtrage permet d'éliminer les "hautes" fréquences dans l'image et limite donc les fluctuations rapides de l'illuminance

# Exemple de filtrage avec un masque comme celui de la p. 42



image



image avec bruit blanc gaussien

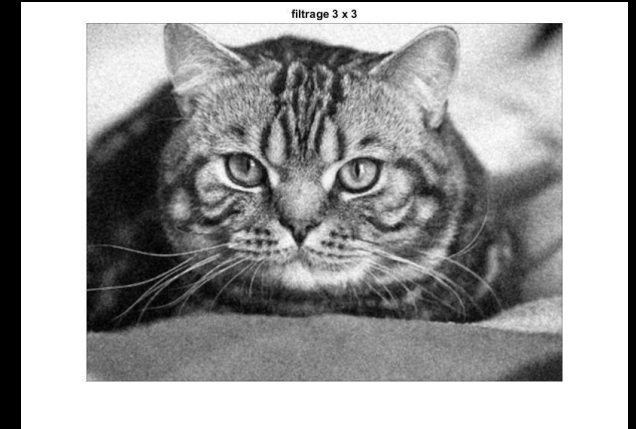


image avec filtrage moyennneur 3 x 3



image avec filtrage moyennneur 5 x 5

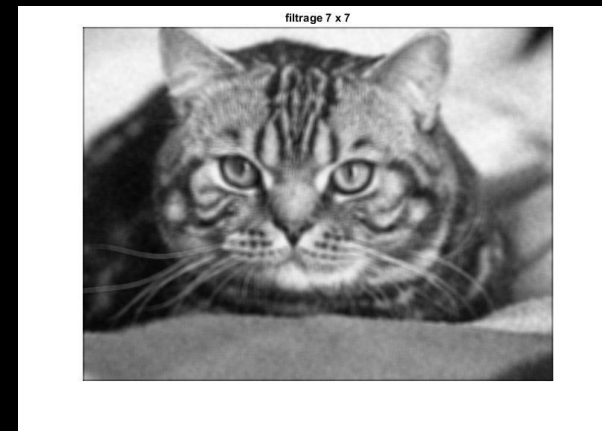


image avec filtrage moyennneur 7 x 7

On constate que si le bruit est réduit, l'image devient de plus en plus *floue* quand  $m$  augmente

image noir et blanc



image avec bruit

filtrage 7 x 7

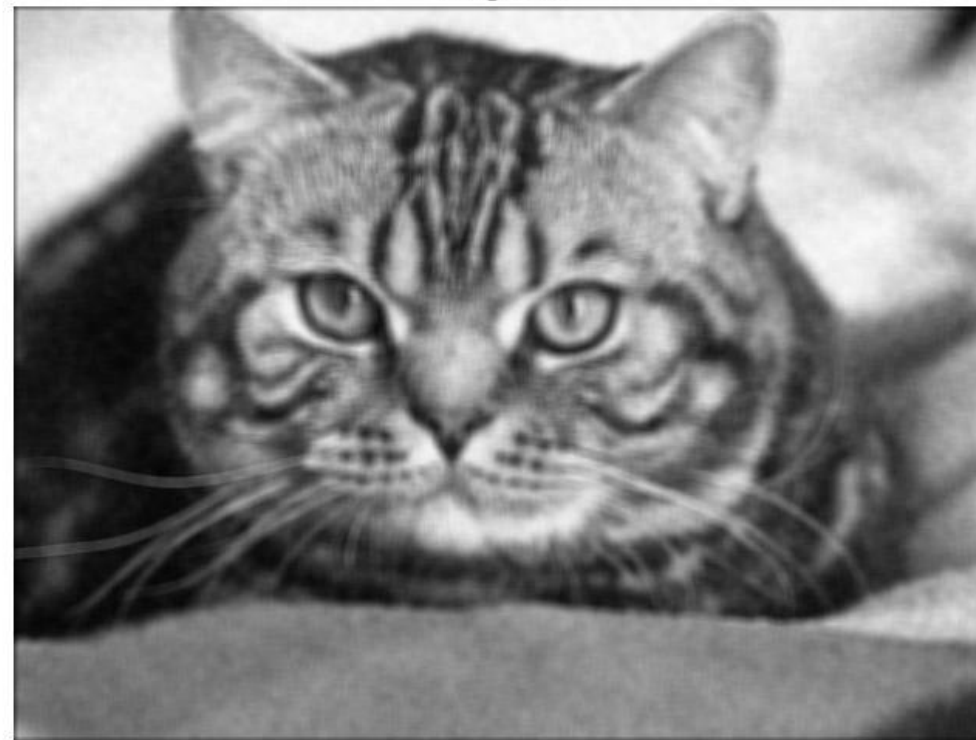


image avec filtrage moyennneur 7 x 7

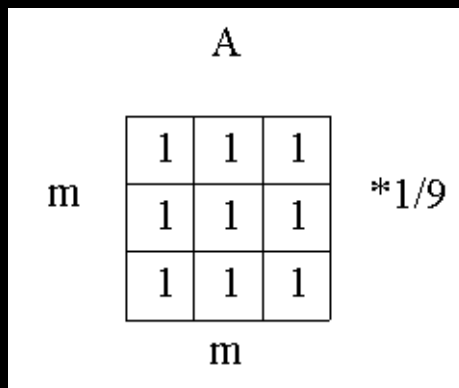
## Remarques sur la convolution (et le filtrage linéaire)

La convolution est une opération linéaire. On peut donc écrire que:

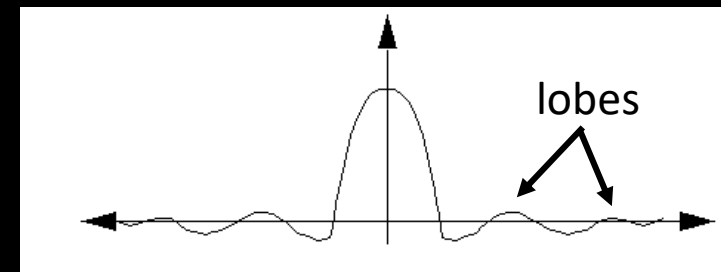
commutativité  $A \cdot E = E \cdot A$  (23)

associativité  $A \cdot (B \cdot E) = (A \cdot B) \cdot E$  (24)

Le filtre moyenneur uniforme ***atténue les hautes fréquences***, mais provoque aussi du ***recouvrement spectral***.



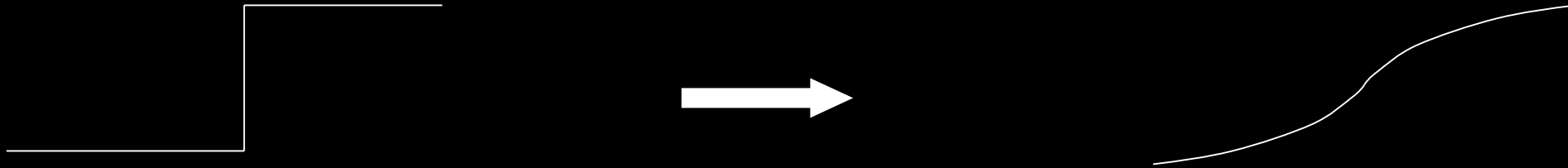
transformée de Fourier



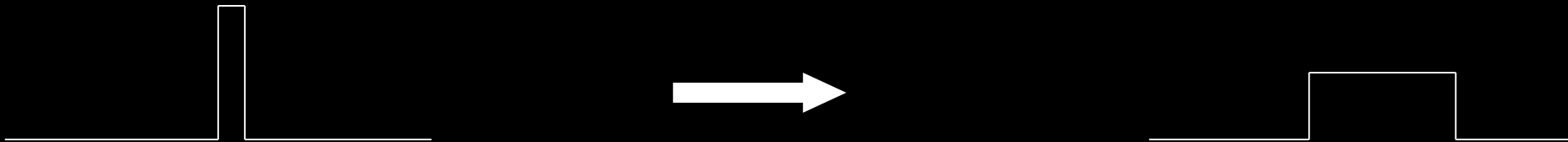


## ***Inconvénient 1*** du filtre moyeneur uniforme:

Les ***transitions abruptes*** d'illuminance sont ***adoucies***



Le ***bruit impulsionnel*** est "***écrasé***" (diffusé sur les pixels voisins) mais n'est ***pas éliminé***



Tel que mentionné précédemment, le filtre cause du ***recouvrement spectral***, ce qui ***laisse du bruit*** dans l'image

**Inconvénient 2** du filtre moyenneur uniforme (et de toutes les opérations de convolution similaires):

Que faire avec les **bords** de l'image?

Quelques approches sont proposées:

- les ignorer → image résultat plus petite
- on peut ajouter des valeurs constantes sur les bords mais ...
- on peut considérer l'image comme étant périodique mais ...
- on peut considérer le bord de l'image comme un miroir mais ...

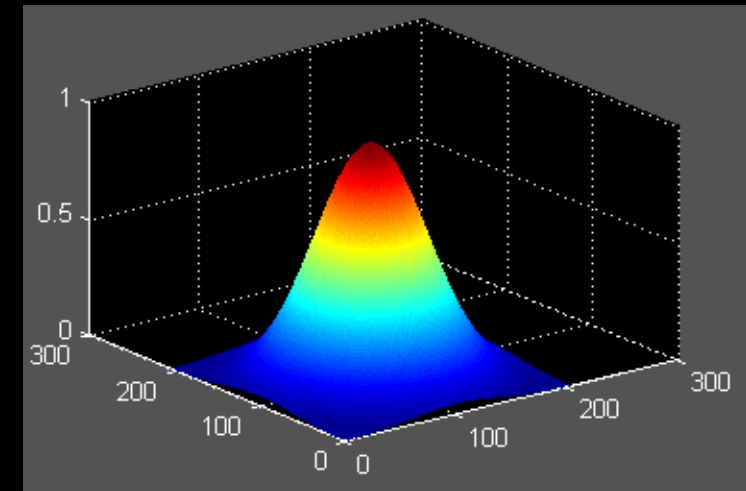
?	?	?					
?	95	130	120	125	110	90	100
?	110	105	95	130	100	110	110
	115	110	100	110	125	105	110
	90	110	120	110	90	110	110
	120	125	90	100	110	110	125
	105	125	110	90	100	90	125
	110	130	100	110	90	100	130

# Filtrage *linéaire* – filtre gaussien

Comme son nom le dit, le *filtre gaussien* possède un *noyau de convolution* de forme *gaussienne*

La forme du filtre est obtenu de l'équation d'une gaussienne:

$$G_{\sigma}(x, y) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \quad (25) \quad \longrightarrow$$

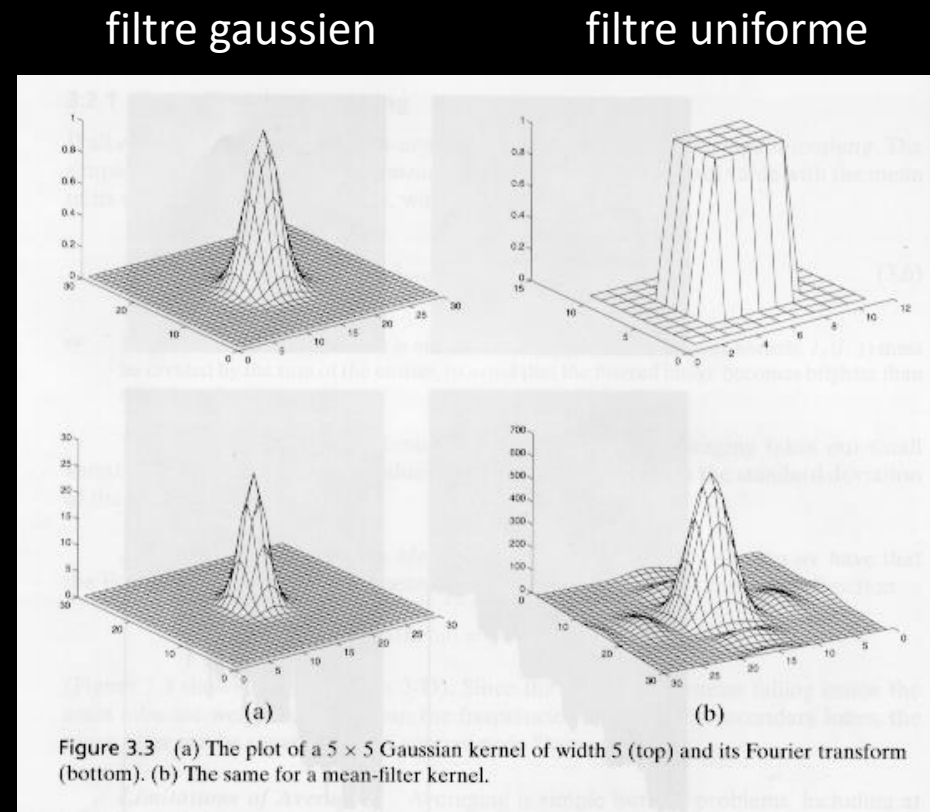


## Avantage 1 du filtre gaussien

- Contrairement au filtre moyeneur uniforme, il ne cause pas de recouvrement spectral car sa **transformée de Fourier** est une **gaussienne** sans lobes secondaires

Noyau de convolution

Trans. de Fourier



\*tiré de Trucco et Verri

## Avantage 2 du filtre gaussien

- le filtre gaussien est *séparable*:
  - on peut filtrer les *lignes* avec un noyau 1 ligne x 5 colonnes de profil gaussien (par exemple)
  - on filtre le résultat (les *colonnes*) avec un noyau de 5 lignes x 1 colonne de profil gaussien
- la propriété d'être séparable permet d'accélérer les calculs par rapport à la convolution classique en 2D. Par exemple, pour un filtre 5 x 5:
  - une convolution 2D classique requiert 25 multiplications et 24 additions soit 49 opérations par pixel
  - en exploitant la séparabilité du filtre, on réduit la complexité algorithmique à 2 x 5 multiplications et 2 x 4 additions, soit 18 opérations par pixel

## Désavantage du filtre gaussien

- Les multiplications utilisent des *poids non unitaires*, ce qui augmente la complexité algorithmique par rapport au filtre moyenneur uniforme (pour lequel on peut se contenter de faire seulement les additions dans l'opération de convolution)

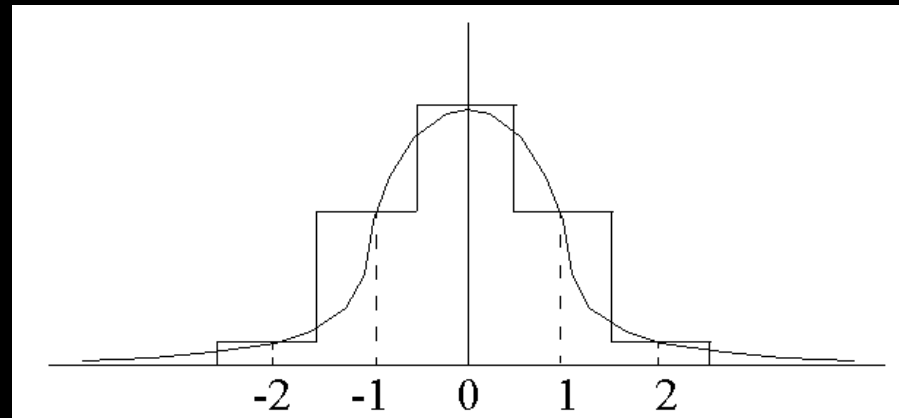
Ce qui suit illustre ceci

# Conception du filtre gaussien



1. Choisir une taille impaire pour le masque (i.e. largeur avec un nombre impair d'éléments)
  - en règle générale, pour que le masque couvre environ 98.76% de l'aire sous la courbe de la gaussienne, on choisit  $m = 5 * \sigma$
2. On échantillonne la gaussienne en  $m$  endroits (par exemple pour  $m = 5$ )

$$G = e^{\frac{-x^2}{2\sigma^2}}$$



### 3. Pour augmenter l'efficacité computationnelle:

- on fixe le plus petit poids à 1
- on applique le facteur d'échelle aux autres poids
- on arrondit la valeur des poids à l'entier le plus près
- une fois la convolution effectuée en un pixel donné, on divise la somme des multiplications par la somme des poids pour conserver un gain de 1

exemple pour  $\sigma = 1$  et  $m = 3$

x	G	poids
0	1	7
1	0,6	4
2	0,14	1

# Exemples pour 3 et 5 échantillons de la gaussienne

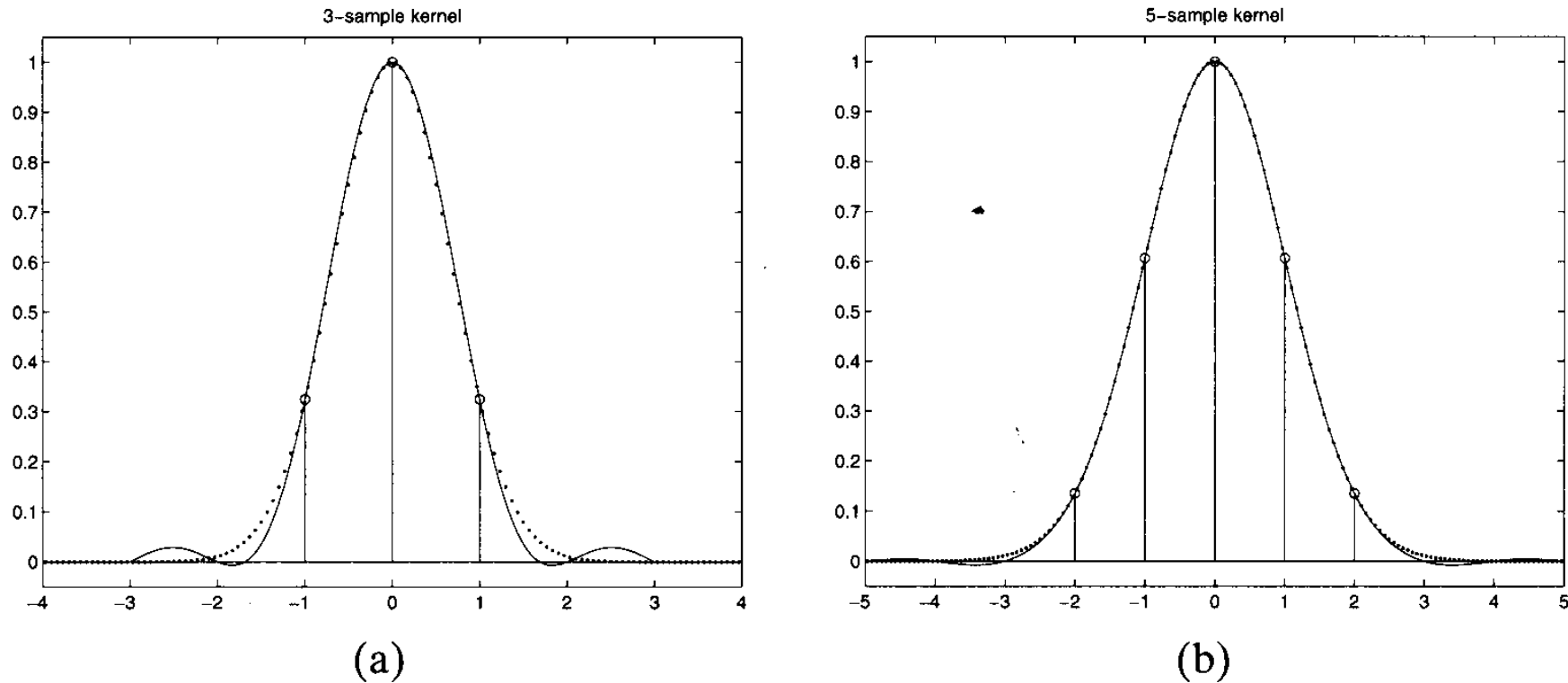


Figure 3.6 Continuous Gaussian kernels (dotted), sampled real kernels, and continuous kernels reconstructed from samples (solid), for  $\sigma = 0.6$  ( $w = 3$ ) (a) and  $\sigma = 1$  ( $w = 5$ ) (b) respectively.

\*tiré de Trucco et Verri

Stratégie pour les filtres gaussiens *larges* (i.e.  $m$  grand):

Au lieu de convoluer un filtre gaussien très *large*, on peut convoluer un *masque 3 x 3 plusieurs fois* (en exploitant également la séparabilité à chaque passe)

# Filtrage non-linéaire

Dépendant du filtre utilisé, le filtrage *non linéaire*:

- permet de réduire le bruit *impulsionnel* (multiplicatif), ce que le filtre moyenneur (ou même gaussien) n'arrive pas à faire
- préserve mieux les *discontinuités* (sans les adoucir)
- est généralement plus *coûteux* en temps calcul que le filtre linéaire moyenneur
- ne *respecte pas* les propriétés de commutativité et d'associativité du filtre linéaire

## Exemples de filtres non-linéaires:

- filtre *médian*
- filtre *bilatéral*
- opérateurs *morphologiques*

# Le filtre médian



Le filtre médian est bien adapté au filtrage du bruit *impulsionnel*

Le filtre médian utilise aussi un noyau sur lequel on effectue les opérations suivantes:

1. **trier** les valeurs d'illuminance des pixels couverts par le masque
2. extraire la **médiane** des données triées
3. **remplacer** la valeur du pixel central par la médiane

64 64 64 64 **64** 64 255 255 255  
↑  
médiane

← tri

64	64	64	64	64	64	64	64
64	64	64	64	64	64	64	64
64	64	64	64	64	255	255	64
64	64	64	255	255	64	64	64
64	64	64	64	255	64	64	255
64	64	64	64	64	255	64	128
64	64	64	64	64	64	128	128
64	64	64	64	255	128	128	128

# Exemple

image noir et blanc



image avec bruit impulsionnel ajoute

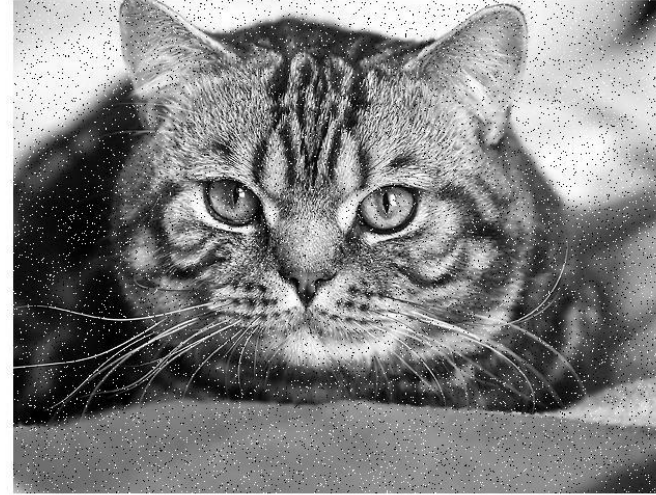


image avec filtre moyennneur 3 x 3

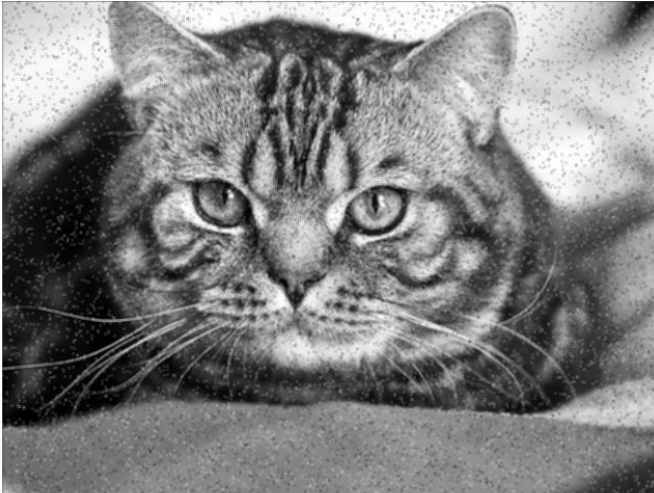
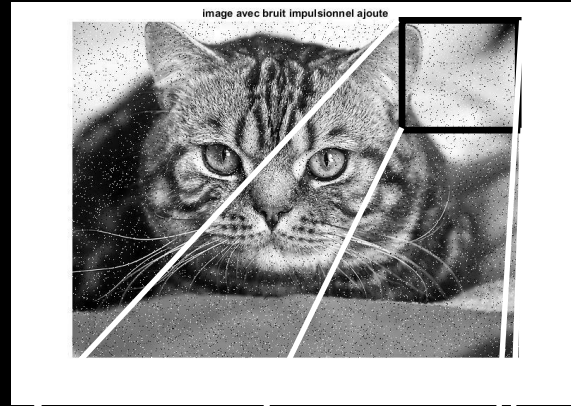


image avec filtrage median



# Exemple suite



Image



Filtre moyennneur

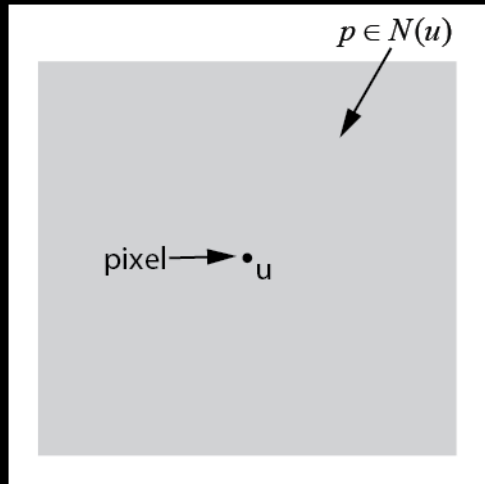


Filtrage médian

# Le filtre filtre bilatéral

- L'idée derrière le filtrage gaussien (ou le simple filtre moyenneur) est que des pixels voisins dans l'espace image ont des illuminances voisines.
- L'hypothèse est donc faite que l'illuminance varie lentement et qu'un moyennage spatial permet de réduire le bruit.
- Si l'hypothèse de variation lente de l'illuminance n'est pas respectée (comme sur les arêtes des images correspondant aux changements rapides de l'illuminance), le moyennage adoucit ces discontinuités, ce qui n'est pas idéal
- Le filtre **bilatéral** tente de compenser ces défaillances en repensant la notion de **proximité** entre les pixels sans avoir recours à un processus itératif

Reprenons l'expression du filtre passe bas (Eq. (22) implanté via une convolution et reformulons son expression comme suit:



$$E(u) = \frac{\sum_{p \in N(u)} W_c(\|p - u\|) E(p)}{\sum_{p \in N(u)} W_c(\|p - u\|)} \quad (26)$$

$$K = \sum_{p \in N(u)} W_c(\|p - u\|) \quad (27)$$

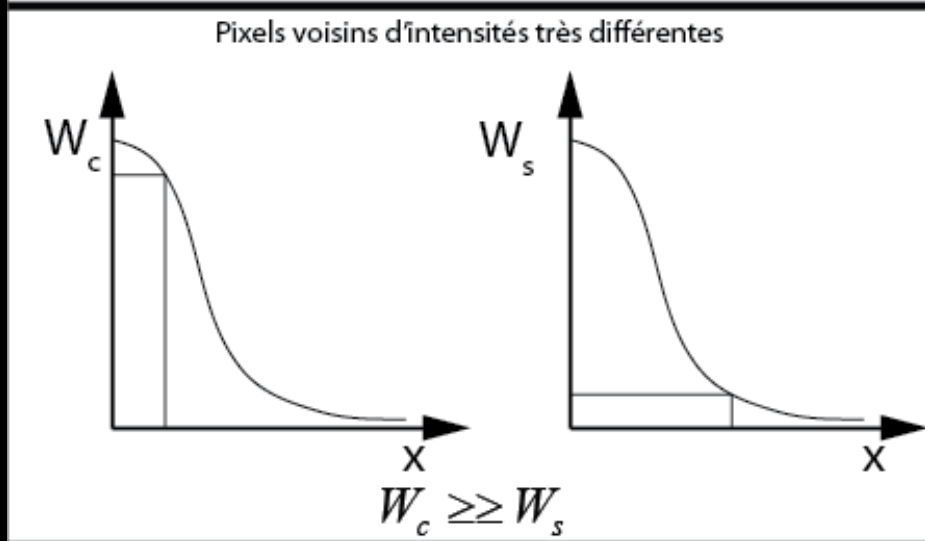
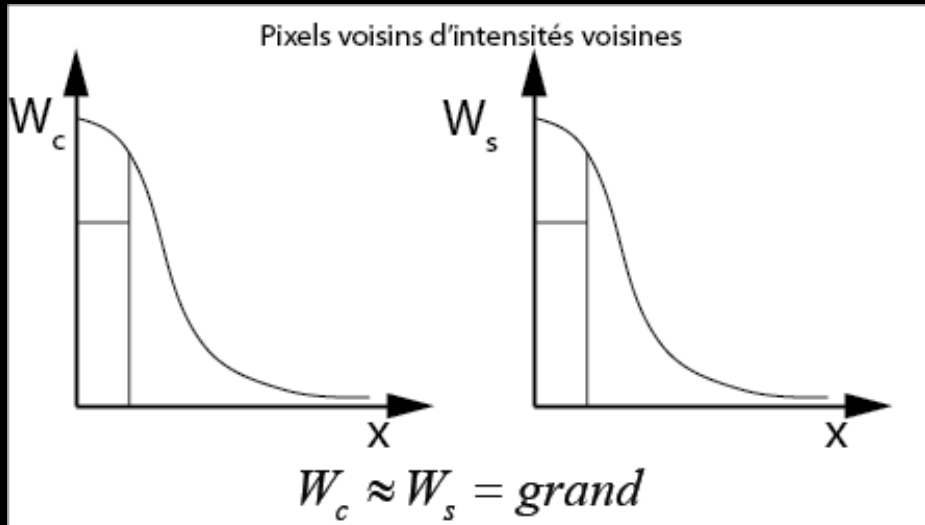
Les  $W_c(p)$  sont les poids des éléments du noyau de convolution à une distance  $\|p - u\|$  du pixel central  $u$ .

- L'idée à la base du *filtre bilatéral* est *d'ajouter des poids* à la convolution qui tiennent compte de la *distance* entre les pixels dans *l'espace des illuminances*
- Ainsi, deux pixels voisins dans l'espace image mais d'illuminance très différentes (comme sur une discontinuité) ne seront pas moyennés car le poids associé à leur distance dans l'espace des illuminances sera faible.
- En général, on choisit les poids selon une distribution gaussienne.

$$W_c = e^{\frac{-x^2}{2\sigma_c^2}} \quad (28)$$

$$W_s = e^{\frac{-x^2}{2\sigma_s^2}} \quad (29)$$





La convolution est maintenant formulée comme suit:

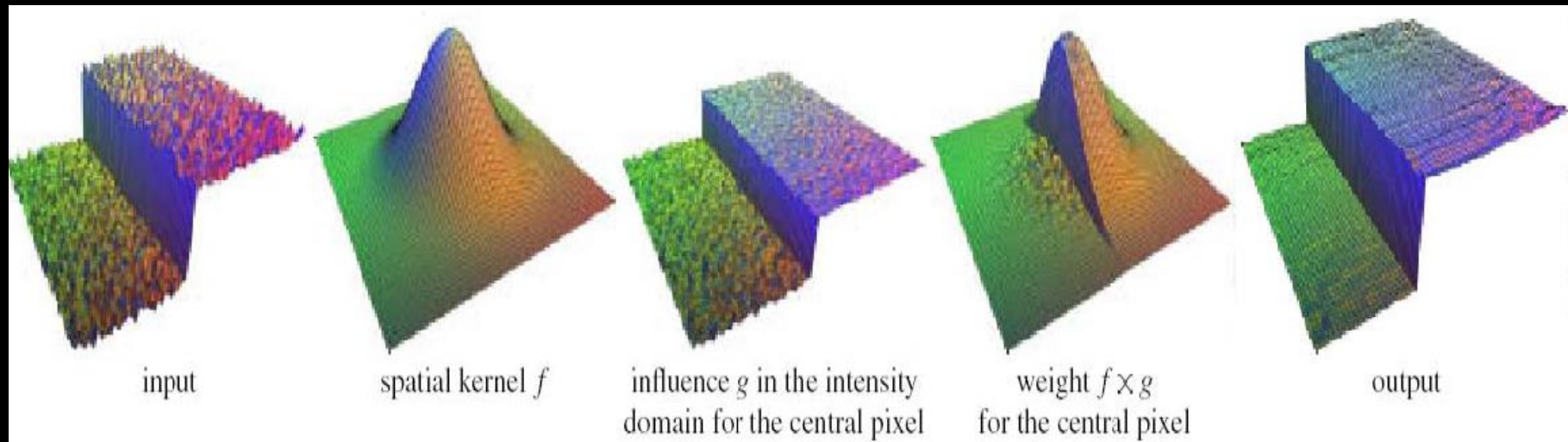
$$E_{\text{filtre}}(u) = K^{-1} \sum_{p \in N(u)} W_c(\|p - u\|) W_s(\|E(p) - E(u)\|) E(p) \quad (30)$$

avec

$$K = \sum_{p \in N(u)} W_c(\|p - u\|) W_s(\|E(p) - E(u)\|) \quad (31)$$



# Résultat de l'application du filtre bilatéral



signal bruité avec une discontinuité d'illuminance

noyau "spatial"

influence de l'illuminance sur le noyau spatial"

la discontinuité est préservée

$$W_c$$

$$W_c \cdot W_s$$

## Exemple 1



Tomasi et Manduchi, ICCV 98

## Exemple 2



## Exemple 3

Le principe fonctionne avec les images en couleurs



Weiss, Siggraph 2006

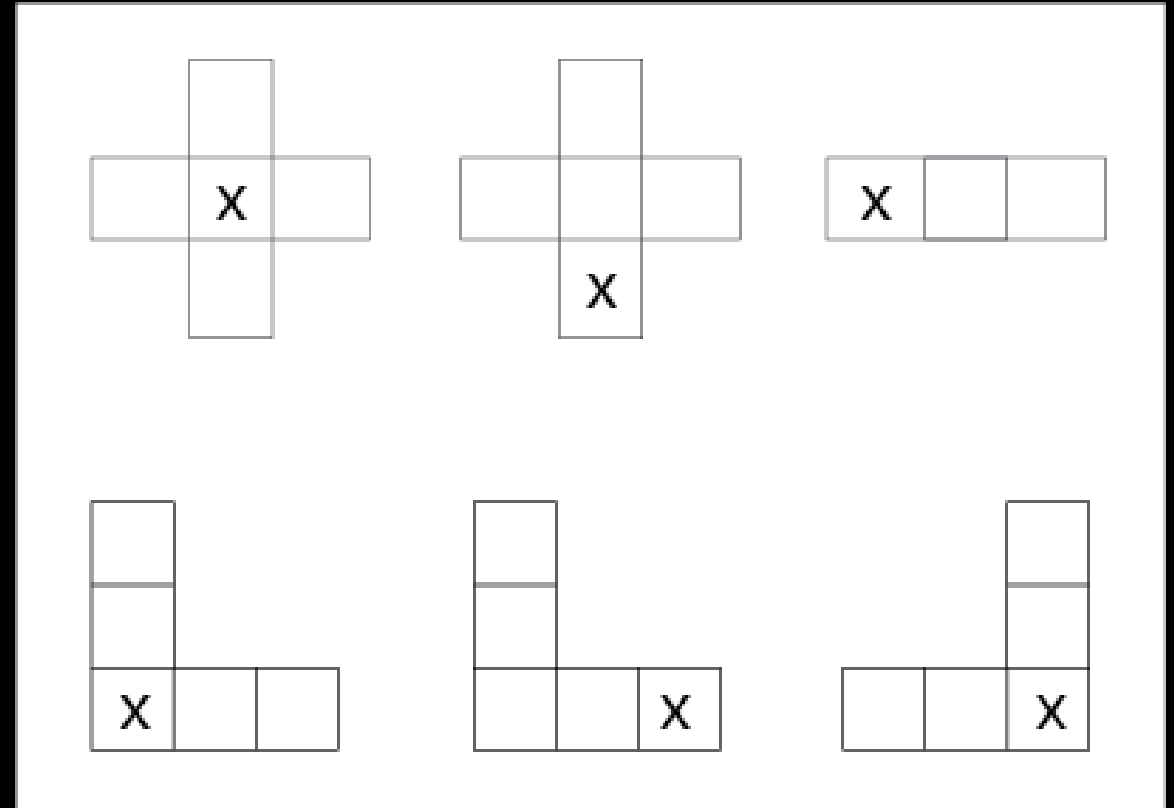
Effet du choix de  $\sigma_s$

# Le filtrage avec les opérateurs *morphologiques* sur des images *binaires*

On compte plusieurs types de filtres non-linéaires dits *morphologiques*.

Ces filtres effectuent des opérations *min-max* appliquées sur un *noyau* de *forme variable* "s" appelé *élément structurant*

Exemples d'éléments structurants



Les opérations min-max peuvent être appliquées en séquence sur les images avec un élément structurant donné.

- Érosion (min): si un pixel prend la valeur 0 sur un des éléments du masque, le pixel “central” du masque est mis à 0
- Dilatation (max): si un pixel prend la valeur 1 sur un des éléments du masque, le pixel “central” du masque est mis à 1
- Ouverture ( $f \circ s$ ): érosion suivie d’une dilatation – élimine les “grenailles” dans l’image binaire
- Fermeture ( $f \bullet s$ ): dilatation suivie d’une érosion – bouche les trous dans une image binaire

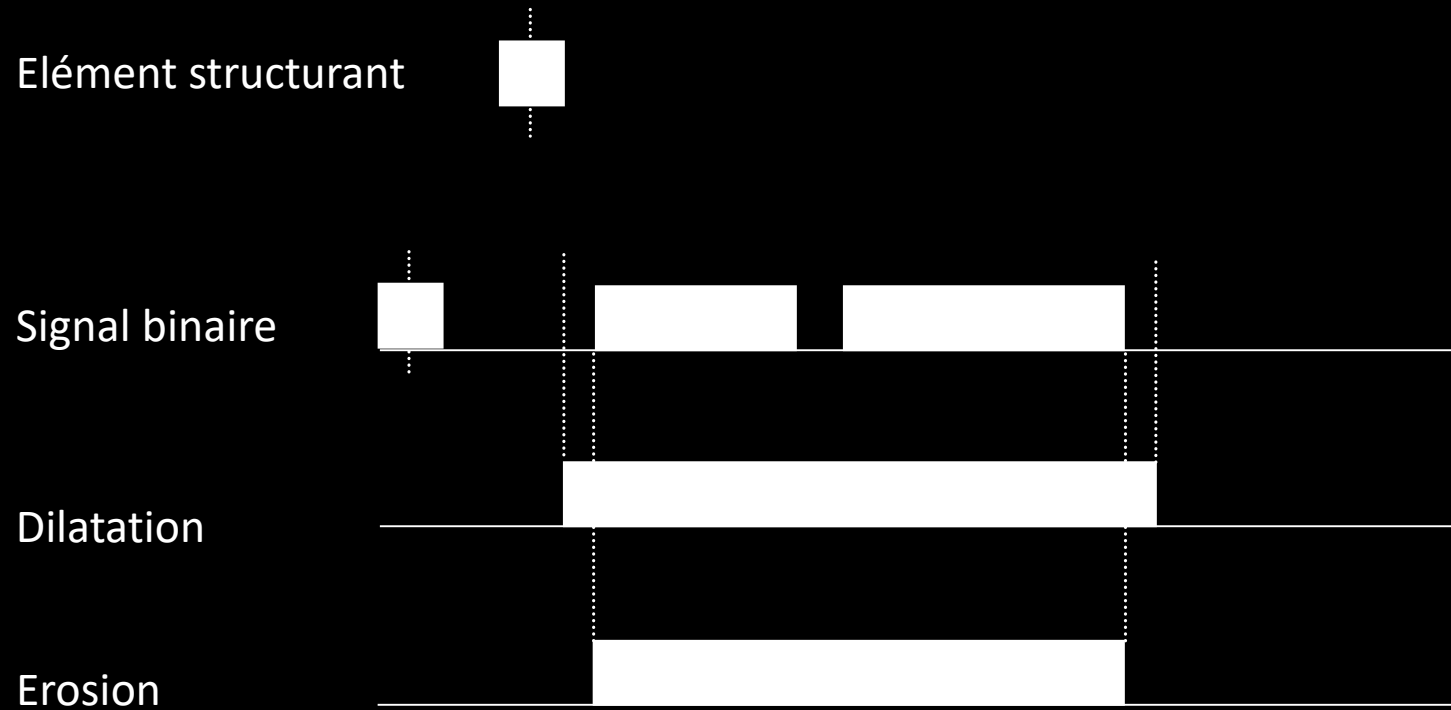
- pour les images à niveau de gris:  
opérateur chapeau haut-de-forme (top hat operator)

$$f - (f \circ s)$$

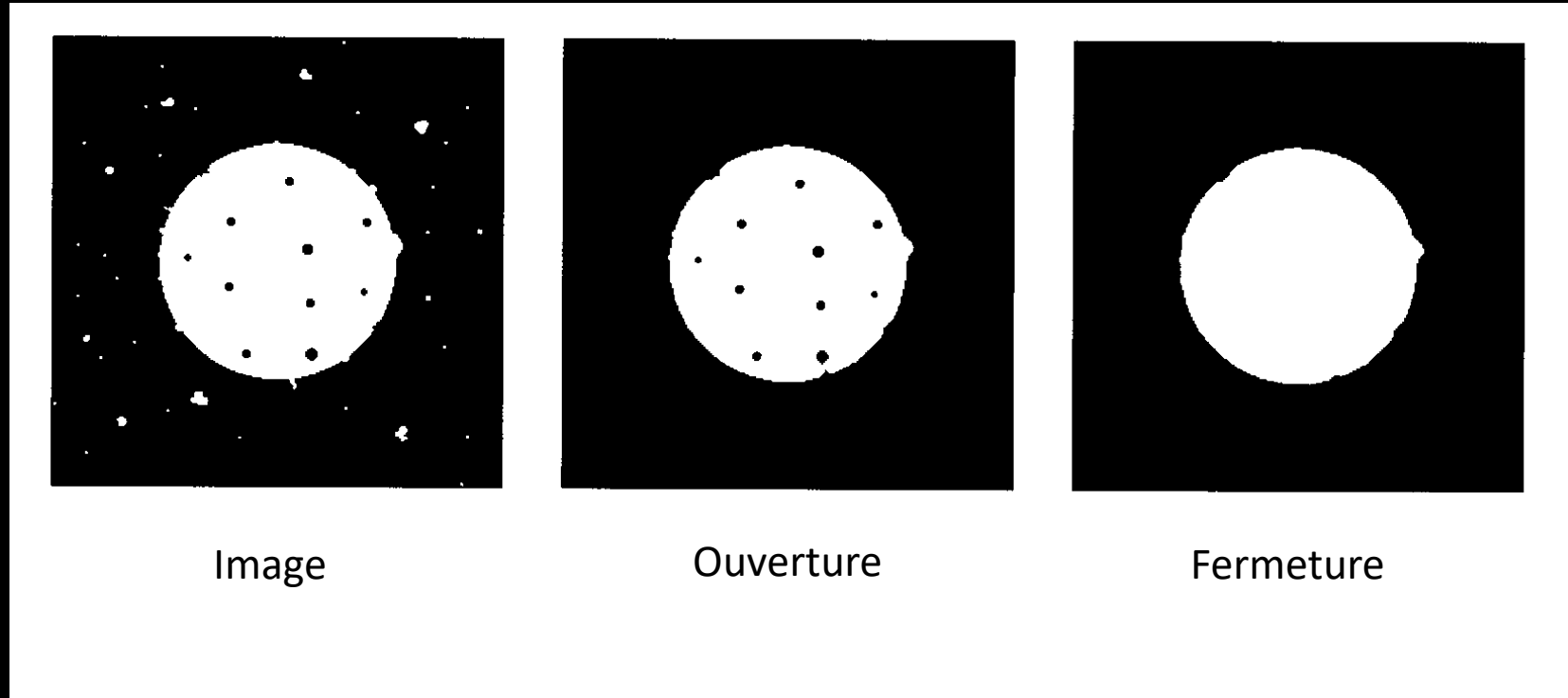
image de laquelle on soustrait son ouverture – détecte les pics et les crêtes



# Exemple simple de fermeture en 1D

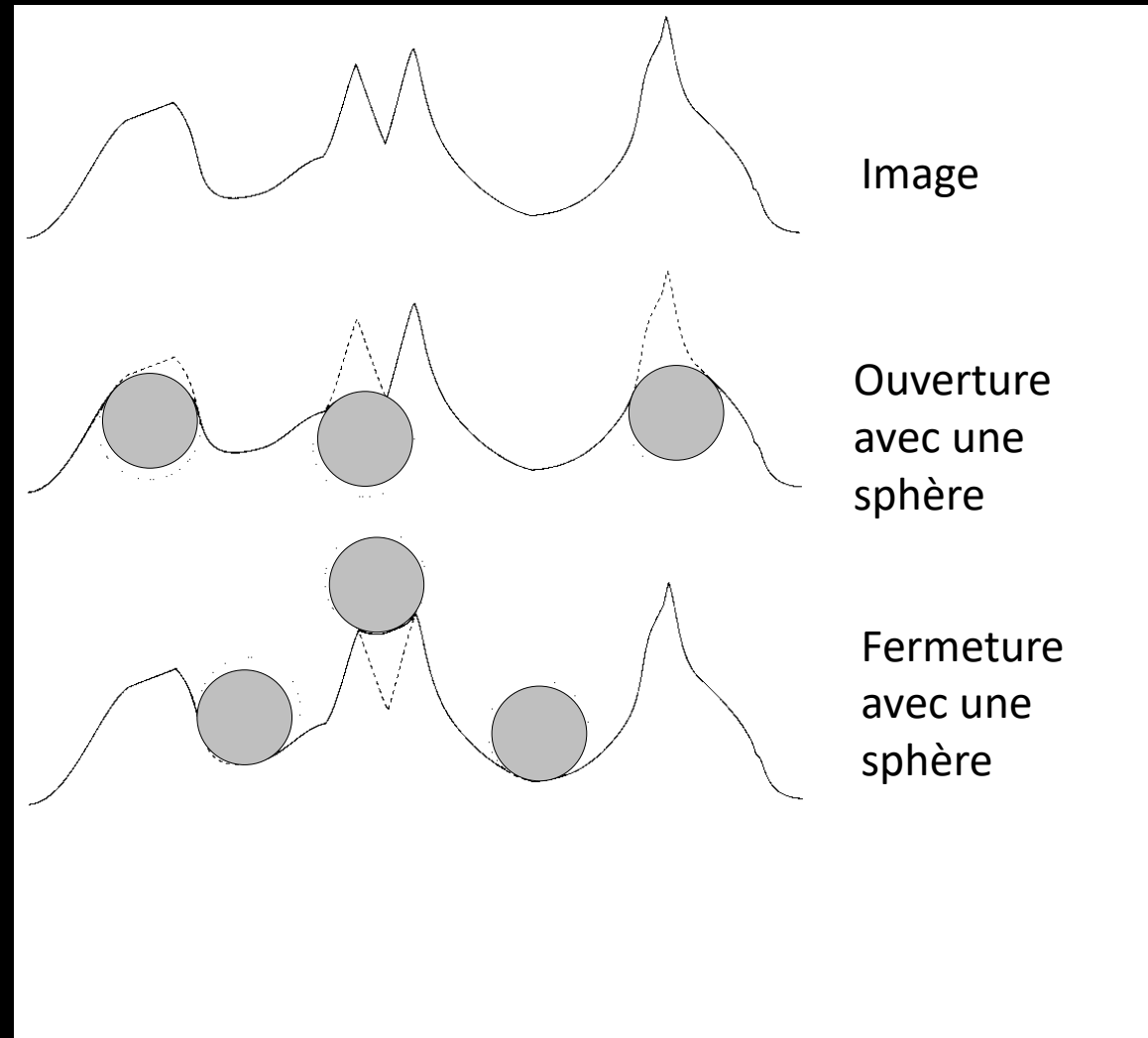


## Exemple sur une image binaire

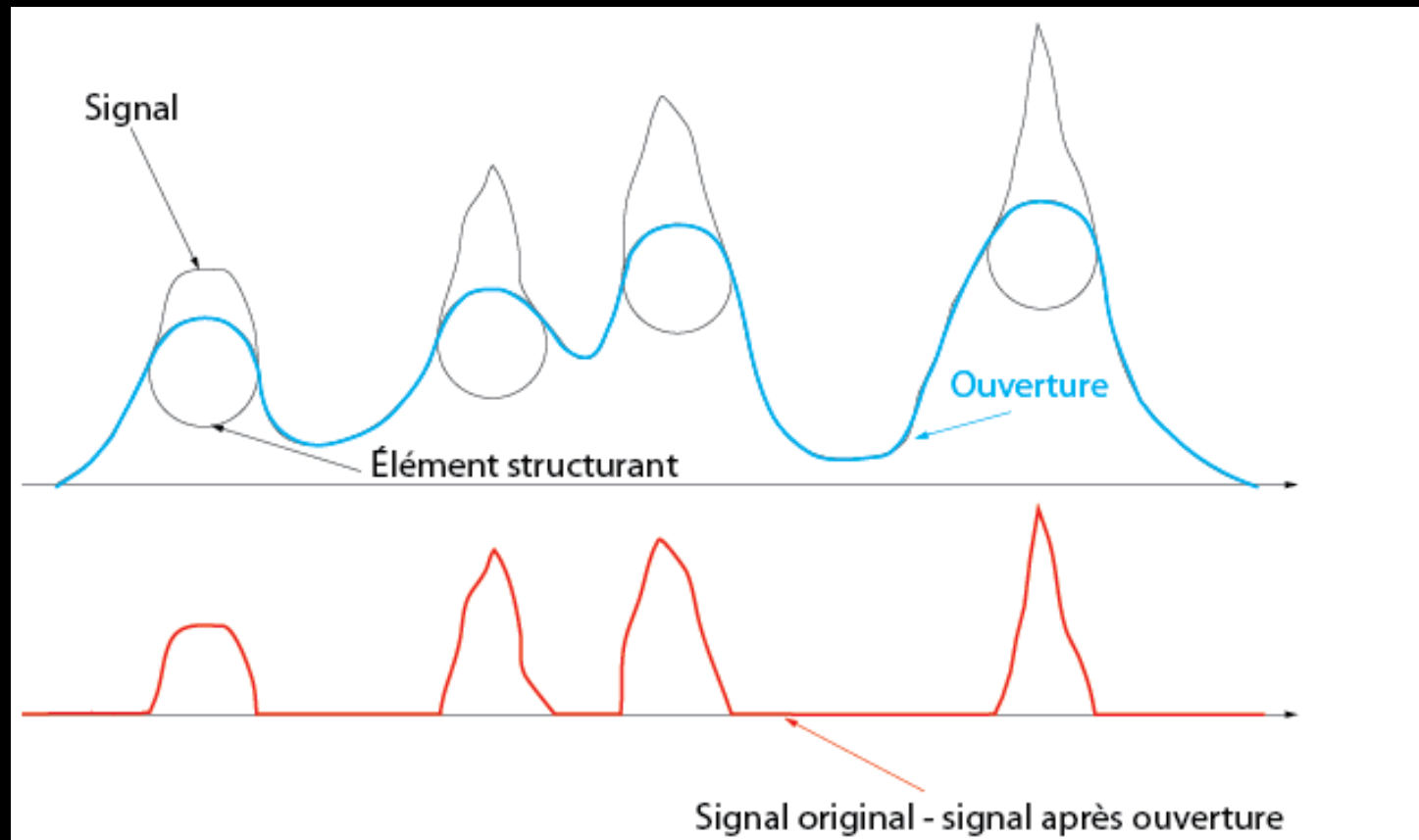


\*tirée de Efford

# Exemple sur une image à niveaux de gris



# Exemple simple d'opérateur "Top-Hat" en 1 D pour détecter les maximums dans un signal



## Références utiles

- C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images", *Proceedings of the 1998 IEEE International Conference on Computer Vision*, Bombay, India.
- B. Weiss, "Fast Median and Bilateral Filtering", *Siggraph 2006*, Boston, USA.