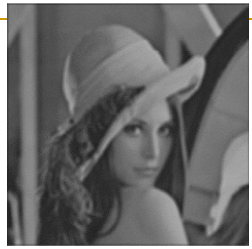
Traitement d'Image :

Filtrage d'image

$$G_\sigma = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\|x\|^2}{2\sigma^2}\right)$$



Plan

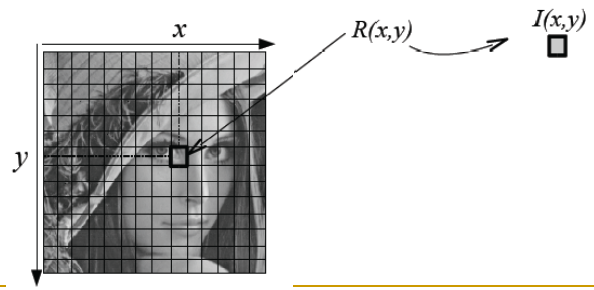
- 1. Rappel
- 2. Problématique du filtrage
- 3. Filtres de lissage
- 4. Filtres dérivateurs
- 5. Approche différentielle
- 6. Restauration : initiation

Plan

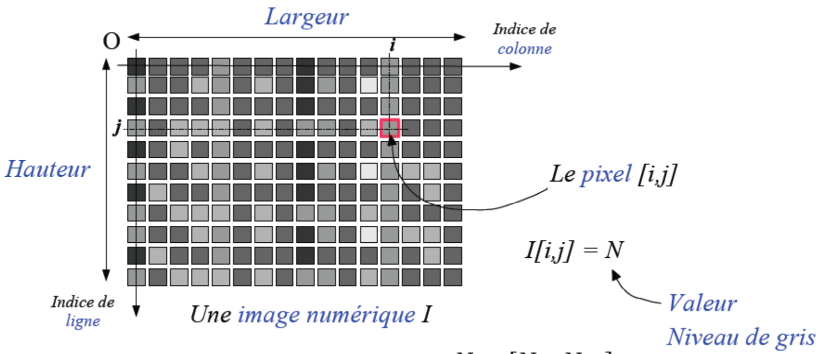
- 1. **Rappel**
- 2. Problématique du filtrage
- 3. Filtres de lissage
- 4. Filtres dérivateurs
- 5. Approche différentielle
- 6. Restauration : initiation

1. Rappel

- L'***échantillonnage*** est le procédé de discrétisation spatiale d'une image consistant à associer à chaque zone rectangulaire *R(x,y)* d'une image continue une unique valeur *I(x,y)*.



1. Rappel



$N \in [N_{min}, N_{max}]$

$(N_{max} - N_{min})$ = *nombre de niveaux de gris*

$Log_2(N_{max} - N_{min})$ = *dynamique*

Plan

- 1. Rappel
- 2. **Problématique du filtrage**
- 3. Filtres de lissage
- 4. Filtres dérivateurs
- 5. Approche différentielle
- 6. Restauration initiation

2. Filtrage d'image

- Le filtrage d'image a pour but :
  - D'atténuer l'effet du bruit d'acquisition sur une image.
  - Extraire des caractéristiques de l'image (contours).
  - Améliorer le contraste d'une image.

2. Filtrage d'image

- En termes de filtrage, on distinguera 2 grands types de filtre :
  - Les filtres de lissage (atténuation du bruit)
  - Les filtres dérivateurs (extraction des contours, rehaussement de contraste)

Plan

- 1. Rappel
- 2. Problématique du filtrage
- 3. **Filtres de lissage**
- 4. Filtres dérivateurs
- 5. Approche différentielle
- 6. Restauration : initiation

3. Filtre de lissage

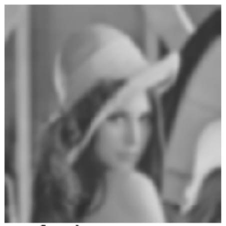
- On s'intéresse ici aux techniques d'*amélioration* des images numériques, pour augmenter la qualité de leur rendu visuel, ou pour faciliter leur analyse.
- On cherche donc à atténuer, sinon supprimer une certaine *dégradation*.
- On distinguera ici :
  - les dégradations liées au *bruit* : $g(x) = f(x)+b(x)$ ou $g(x) = f(x)b(x)$ liées au capteur, à la quantification, à la transmission... On les traite en tirant parti des informations locales par le *filtrage*.
  - les dégradations *convolutives* : $g(x) = f(x)*b(x)$ liées à un mouvement du capteur ou un défaut de mise au point. On les traite en inversant un opérateur linéaire, donc supposé connu : ce sont les techniques dites de *restauration (voir section 6)*.

3. Filtre de lissage

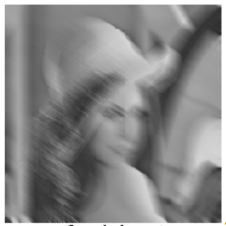
- Exemples :

bruit additif

bruit multiplicatif



flou de mise au point



flou de bougé

3. Filtre de lissage

- ***Les filtres de lissage*** sont des opérateurs qui *éliminent* des éléments *pertubateurs / non significatifs* dans les images numériques,
  - soit pour *améliorer* leur visualisation,
  - soit pour les *simplifier* en but d'un traitement postérieur.

Bruit gaussien

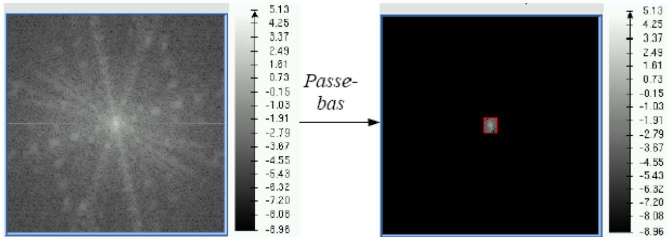
Bruit impulsionnel

3. Filtre de lissage

- *On distingue 3 grands types de filtres de lissage*
  - *(1) Filtrage dans le domaine de Fourier*
  - *(2) Filtrage par convolution*
  - *(3) Filtres non linéaires*

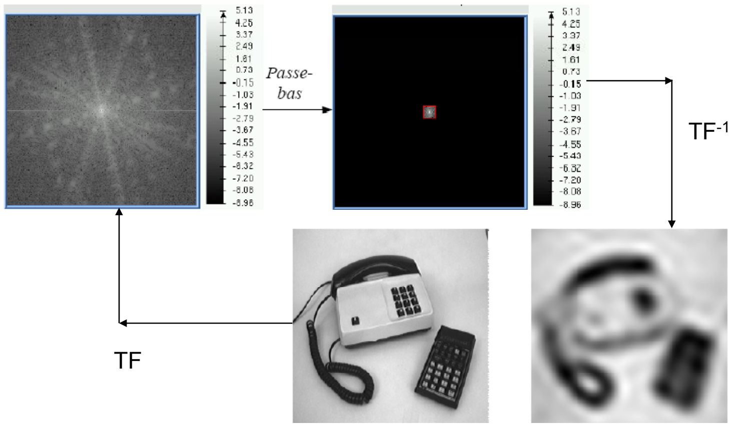
3. Filtre de lissage

- **Filtrage passe-bas dans le domaine de Fourier**
  - Le filtrage passe-bas est la multiplication dans le domaine fréquentiel par une *fonction porte* (fonction indicatrice d'un intervalle $[-u_{max}, u_{max}] \times [-v_{max}, v_{max}]$).



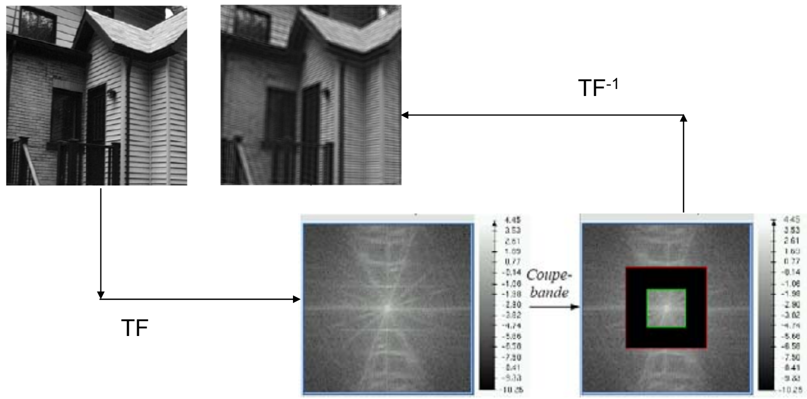
3. Filtre de lissage

- **Filtrage passe-bas dans le domaine de Fourier**



3. Filtre de lissage

- **Filtrage coupe-bande dans le domaine de Fourier**



3. Filtre de lissage

- **Filtrage par convolution**
  - La **multiplication** dans le domaine fréquentiel correspond à la **convolution** dans le domaine spatial.
  - Un grand nombre de filtres de lissage peut être obtenu à partir de noyaux de convolution symétriques et normalisés (de somme égale à 1).

3. Filtre de lissage

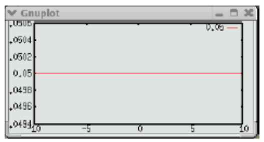
- **Filtrage par convolution**

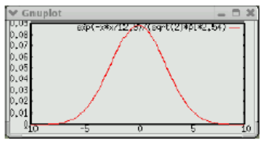
Les trois types de filtres les plus utilisés sont :

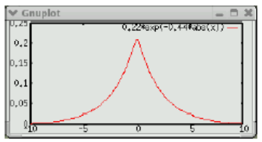
- Les filtres moyenneurs $h(x,y)=\frac{1}{\lambda^2}$
- Les filtres gaussiens $h(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{(x^2+y^2)}{2\sigma^2}}$
- Les filtres exponentiels $h(x,y)=\frac{\gamma^2}{4}e^{-\gamma(|x|+|y|)}$

3. Filtre de lissage

- **Filtrage par convolution : réponse impulsionnelle**







$$\frac{1}{25}\begin{bmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\end{bmatrix}$$

Filtre moyenneur (5x5)

$$\frac{1}{864}\begin{bmatrix}11&23&29&23&11\\23&48&62&48&23\\29&62&80&62&29\\23&48&62&48&23\\11&23&29&23&11\end{bmatrix}$$

Filtre gaussien (σ = 1,41)

$$\frac{1}{80}\begin{bmatrix}1&1&3&1&1\\1&3&7&3&1\\3&7&16&7&3\\1&3&7&3&1\\1&1&3&1&1\end{bmatrix}$$

Filtre exponentiel (γ = 0,8)

3. Filtre de lissage

- **Filtrage par convolution : exemples**

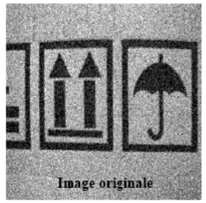
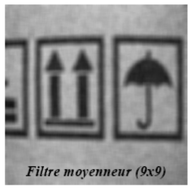
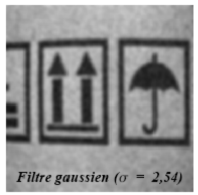


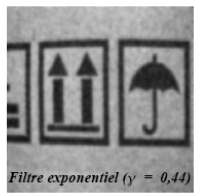
Image originale



Filtre moyenneur (9x9)



Filtre gaussien (σ = 2,54)



Filtre exponentiel (γ = 0,44)

3. Filtre de lissage

- **Implantation des filtres de lissage linéaires**
  - En traitement d'images, les volumes de données traités sont bien sûr très importants.
  - La prise en compte du temps de calcul reste un élément majeur dans les algorithmes en dépit des progrès technologiques exponentiels des microprocesseurs.
  - L'implantation des filtres linéaires, en particulier ceux dont le support est grand, voire infini, est un problème incontournable.

3. Filtre de lissage

- **Implantation des filtres de lissage linéaires**
  - Les méthodes les plus courantes sont :
    - Multiplication dans le domaine de Fourier (La convolution devient une multiplication)
    - Convolution par noyau tronqué
    - Noyaux séparables
    - Implantation récursive des filtres RII

3. Filtre de lissage

- **Inconvénients des filtres de lissage linéaires**
  - Par convolution : effet de bord
  - Dégradation des contours de l'image
  - Inefficace sur les bruits de type impulsionnel (« salt and pepper »)

3. Filtre de lissage

- **Filtres non linéaires**
  - Deux aspects du lissage sont concernés par le filtrage non linéaire :
    - *Le bruit impulsionnel* : les filtres linéaires éliminent mal les valeurs aberrantes.
    - *L'intégrité des frontières* : on souhaiterait éliminer le bruit sans rendre flou les frontières des objets.

3. Filtre de lissage

- **Filtres non linéaires**
  - Les principaux filtres non linéaires sont :
    - Les filtres d'ordre (médian en particulier)
    - Le filtre de Nagao
    - Les filtres morphologiques

3. Filtre de lissage

- **Filtres d'ordre**
  - Les filtres d'ordres procèdent en remplaçant les valeurs de chaque pixel par la valeur qui occupe *un certain rang* lorsqu'on trie les valeurs observées dans *un certain voisinage* du pixel.

3. Filtre de lissage

- **Filtres d'ordre : principe**
  - Les valeurs dans le voisinage de *(x,y)* :

$$V(x,y) = [a_1, a_2, ..., a_n]$$

  - Permutation des valeurs dans l'ordre croissant

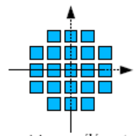
$$V_p(x,y) = [b_1, b_2, ..., b_k]$$

  - Le filtre d'ordre de rang *k* est alors défini par :

$$\rho_k(x,y) = b_k$$

3. Filtre de lissage

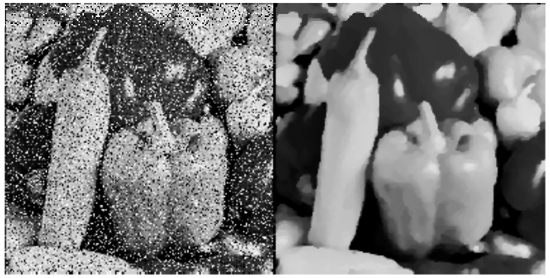
- **Filtres d'ordre : exemples**
  - Pour k=N/2, on parle de filtre médian
  - Pour k=1, on parle d'érosion morphologique
  - Pour k=N, on parle de dilatation morphologique
- L'élément structurant de voisinage n'a pas nécessité à être carré



voisinage : élément structurant

3. Filtre de lissage

- **Filtres d'ordre : filtre médian**

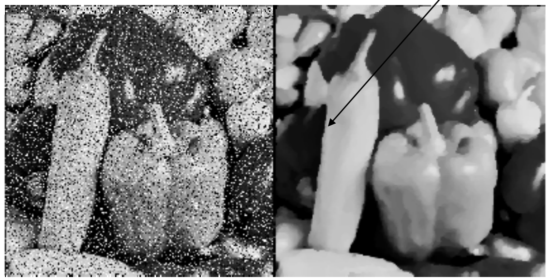


ex : bruit impulsionnel traité par un filtre médian

3. Filtre de lissage

- **Filtres d'ordre : filtre médian**

Effet de bloc



ex : bruit impulsionnel traité par un filtre médian

3. Filtre de lissage

- **Filtres d'ordre : érosion et dilatation**

Erosion Dilatation

3. Filtre de lissage

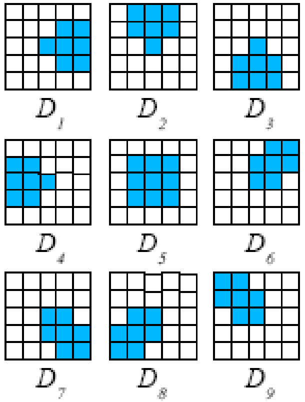
- **Filtre de Nagao**
  - On trouve dans la littérature certains filtres ***combinant*** filtrage linéaire et filtre d'ordre.
  - Le filtre de Nagao en est un.

3. Filtre de lissage

- **Filtre de Nagao**
  - Le filtre de Nagao examine la fenêtre 5x5 centrée sur chaque pixel.
  - 9 domaines sont définis dans cette fenêtre (voir figure suivante).
  - On calcule pour chaque domaine D_i la moyenne μ_i et la variance v_i.
  - Le résultat de l'opérateur est la moyenne du domaine qui présente la plus faible variance.

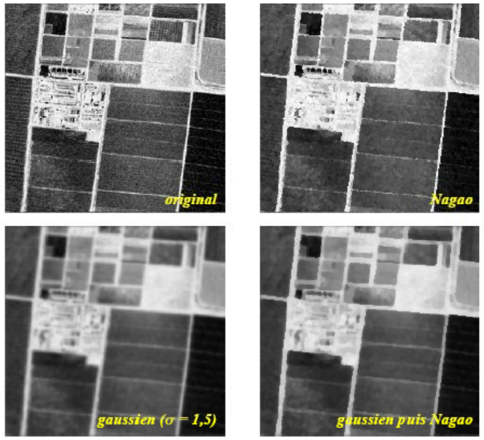
3. Filtre de lissage

- **Filtre de Nagao**
  - **Les 9 fenêtres :**



3. Filtre de lissage

- **Filtre de Nagao :**



3. Filtre de lissage

- **Filtres morphologiques**
  - Il existe deux grands types de filtres morphologiques :
    - Le filtre d'ouverture
    - Le filtre de fermeture

3. Filtre de lissage

- <u>**Ouverture et fermeture**</u>
  - L'*ouverture* morphologique s'obtient par la composition d'une érosion suivie d'une dilatation.
  - La *fermeture* morphologique est l'opération duale de l'ouverture : Elle est égale à la composition d'une dilatation suivie d'une érosion.

3. Filtre de lissage

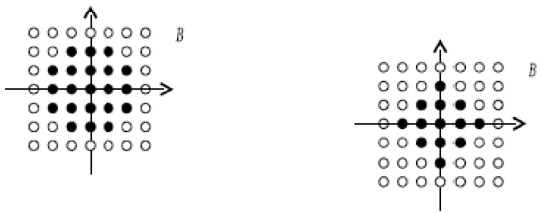
- <u>**Ouverture et fermeture**</u>
  - Visuellement :
    - L'ouverture élimine les petites composantes, et ouvre les petits isthmes.
    - La fermeture bouche les petites trous, et ferme les petits détroits.

3. Filtre de lissage

- <u>**Ouverture et fermeture ?**</u>

3. Filtre de lissage

- <u>**Ouverture et fermeture ?**</u>
  - Le choix du voisinage est important

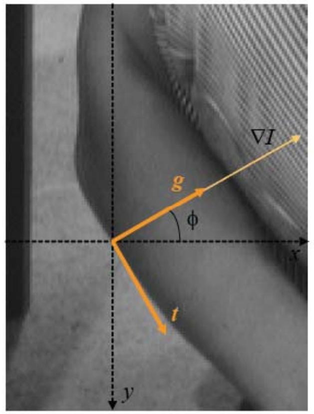


Plan

- 1. Rappel
- 2. Problématique du filtrage
- 3. Filtres de lissage
- 4. **Filtres dérivateurs**
- 5. Approche différentielle
- 6. Restauration initiation

4. Filtres dérivateurs

- **Rappel :**
  - **Le modèle différentiel**
  - Au premier ordre, on peut ainsi associer à chaque point (*x,y*) un repère propre (*t,g*), où le vecteur *t* donne la direction de l'*isophote* (ligne de variation minimale) et *g* la direction orthogonale, celle du *gradient*.



4. Filtres dérivateurs

- Les *variations locales d'intensité* constituent une source primordiale d'information en traitement d'images.
- Elles sont mesurées par le *gradient*, fonction *vectorielle* des pixels *[i,j]* :

$$\nabla f[i,j] = \left(\frac{\partial f}{\partial x}[i,j], \frac{\partial f}{\partial y}[i,j] \right)$$

4. Filtres dérivateurs

- D'autres grandeurs différentielles sont utilisées en traitement d'images, comme le *laplacien*, fonction *scalaire* de *[i,j] :*

$$\Delta f[i,j] = \frac{\partial^2 f}{\partial x^2}[i,j] + \frac{\partial^2 f}{\partial y^2}[i,j]$$

- ou encore le *hessien*, fonction *matricielle* de *[i,j]* :

$$H_f[i,j] = \begin{pmatrix} \frac{\partial^2 f}{\partial x^2}[i,j] & \frac{\partial^2 f}{\partial x \partial y}[i,j] \\ \frac{\partial^2 f}{\partial x \partial y}[i,j] & \frac{\partial^2 f}{\partial y^2}[i,j] \end{pmatrix}$$

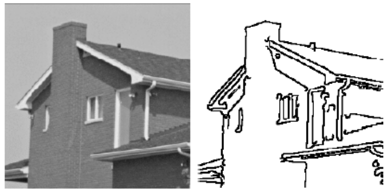
4. Filtres dérivateurs

- Le problème du calcul des filtres dérivateurs dans les images numériques est l'*approximation* de ces grandeurs différentielles dans notre espace discret

- On s'intéresse aussi à leur *utilisation* dans le cadre du rehaussement, et de la détection de contours,...

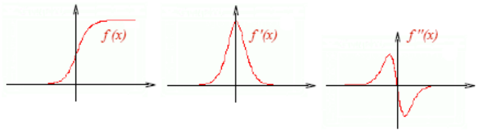
4. Filtres dérivateurs

- **Exemples :**



les contours (image de droite) constituent une simplification de l'image utile dans de nombreuses applications. Dans les approches linéaires, ils sont en général obtenus à partir des maxima locaux de la dérivée première, ou des passages par zéro de la dérivée seconde .

réhaussement de contraste obtenu par combinaison linéaire avec le laplacien.



4. Filtres dérivateurs

- **Exemples :**
  - L'implantation de ces filtres peut se faire comme précédemment soit dans le domaine fréquentiel (filtrage passe-haut par exemple)
  - Soit par convolution directe.

4. Filtres dérivateurs

- **Implantation par convolution**
  - Dérivées directionnelles d'ordre 1
    - Les approximations les plus simples des dérivées directionnelles se font par différences finies calculées par convolution avec des noyaux très simples :

$$\frac{\partial f}{\partial x} \longrightarrow [-1,1] \qquad \frac{\partial f}{\partial y} \longrightarrow \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

4. Filtres dérivateurs

- **Implantation par convolution**
  - Dérivées directionnelles d'ordre 1
    - On utilise plus souvent :

$$\frac{\partial f}{\partial x} \longrightarrow [-1,0,1] \qquad \frac{\partial f}{\partial y} \longrightarrow \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

Ces noyaux permettent en effet d'obtenir des contours plus épais mais bien centrés

4. Filtres dérivateurs

- **Implantation par convolution**
  - Dérivées directionnelles d'ordre 1
    - Ces opérations étant très sensibles au bruit, on les combine en général avec un filtre lisseur dans la direction orthogonale à celle de dérivation
    - par exemple par le noyau suivant (ou sa transposée) :

$$[1,2,1]$$

4. Filtres dérivateurs

- **Implantation par convolution**
  - Dérivées directionnelles d'ordre 1
    - Le calcul des dérivées directionnelles en x et en y revient finalement à la convolution avec les noyaux suivants :

$$f_x(i,j) = (f * h_x)[i,j] \qquad f_y(i,j) = (f * h_y)[i,j]$$

avec

$$h_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \qquad h_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

- Ce filtre est connu sous le nom de **filtre de Sobel**

4. Filtres dérivateurs

- **Implantation par convolution**
  - Dérivées directionnelles d'ordre 1
    - On peut ensuite calculer la norme du gradient :

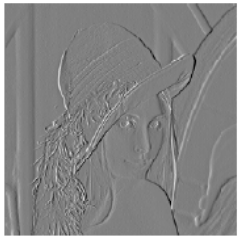
$$\|\nabla f(i,j)\|_2 = \sqrt{f_x(i,j)^2 + f_y(i,j)^2}$$
$$\|\nabla f(i,j)\|_1 = \left| f_x(i,j) + f_y(i,j) \right|$$

- Et son orientation : $\arg(\nabla f(i,j)) = \arctan\left(\frac{f_y(i,j)}{f_x(i,j)}\right)$

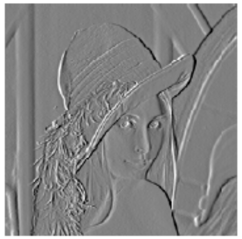
4. Filtres dérivateurs

- **Implantation par convolution**
  - Dérivées directionnelles d'ordre 1

Original



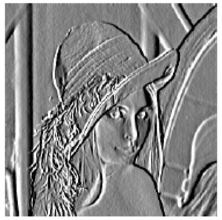
Noyau [-1 1]



Noyau [-1 0 1]

4. Filtres dérivateurs

- **Implantation par convolution**
  - Dérivées directionnelles d'ordre 1



Gradient horizontal (Sobel)

Gradient vertical (Sobel)

Module du gradient de Sobel

4. Filtres dérivateurs

- **Implantation par convolution**
  - Dérivées directionnelles d'ordre 2
    - On utilise le plus souvent :

$$\frac{\partial^2 f}{\partial x^2} \longrightarrow [1,-2,1] \qquad \frac{\partial^2 f}{\partial y^2} \longrightarrow \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

4. Filtres dérivateurs

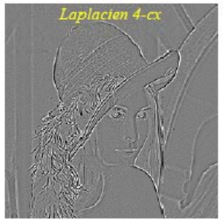
- **Implantation par convolution**
  - Dérivées directionnelles d'ordre 2
    - Le Laplacien peut donc être estimé par le noyau suivant

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$ 4-connexité

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$ 8-connexité

4. Filtres dérivateurs

- **Implantation par convolution**
  - Dérivées directionnelles d'ordre 2



4. Filtres dérivateurs

- **Rehaussement de contraste**
  - L'opération de rehaussement de contraste à base de filtre dérivateur s'obtient de la manière suivante :

$$R_f(i,j) = f(i,j) - \gamma.\Delta f(i,j)$$

  - En soustrayant le Laplacien à l'image originale, on augmente le contraste.

4. Filtres dérivateurs

- **Rehaussement de contraste**

Plan

- 1. Historique du TI
- 2. Images numériques
- 3. Outils fondamentaux
- 4. Filtrage d'image
- 5. **Approche variationnelle**
- 6. Restauration : initiation

5. Approche variationnelle
5.1 Introduction

- L'approche variationnelle du filtrage d'image est (d'un certain point de vue) une réécriture de la problématique sous forme d'Equations aux Dérivées Partielles (EDP).

- ***Origine*** **:**
  - Analogie entre l'atteinte de l'équilibre thermique d'un ensemble de particules en thermodynamique (diffusion thermique) et la régularisation de la luminance des images bruitées.

5. Approche variationnelle
5.1 Introduction

- **Propriété importante :**
  - *Sous certaines conditions, il y a équivalence entre diffuser itérativement une image au moyen de l'EDP de la chaleur et la filtrer par convolution avec un masque gaussien d'écart-type σ.*

5. Approche variationnelle
5.1 Introduction

- **Propriété importante :**
  - Soit f_0 une image bruitée et t le paramètre temporel de diffusion, l'EDP de la chaleur appliquée à f_0 s'écrit alors :

$$\begin{cases} f(x,y,0) = f_0(x,y) \\ \dfrac{\partial f(x,y,t)}{\partial t} = \Delta f(x,y) = div(\nabla f(x,y)) \end{cases}$$

5. Approche variationnelle
5.1 Introduction

- **Propriété importante :**
  - ***L'équivalence avec le filtrage gaussien d'écart-type σ est vérifiée lorsque*** $\sigma = \sqrt{2t}$

$$\begin{cases} f(x,y,0) = f_0(x,y) \\ \dfrac{\partial f(x,y,t)}{\partial t} = \Delta f(x,y) = div(\nabla f(x,y)) \end{cases}$$

5. Approche variationnelle

5.1 Introduction

- **Propriété importante :**
  - **Piste pour la preuve :**
    - Déterminer les solutions du type : $f(x, t) = f_1(x,y) \, . \, f_2(t)$
    - En déduire la ***solution fondamentale*** de l'équation de la chaleur
    - En déduire la solution générale donnée par :

$$f(x,y,t) = \frac{1}{2\sqrt{\pi . t}} \int_{-\infty}^{+\infty} e^{-\frac{(x-X)^2 + (y-Y)^2}{4t}} . f_0(X,Y) dX dY$$

5. Approche variationnelle

5.1 Introduction

- **Propriété importante :**
  - **Piste pour la preuve :**

$$f(x,y,t) = \frac{1}{2\sqrt{\pi . t}} \int_{-\infty}^{+\infty} e^{-\frac{(x-X)^2 + (y-Y)^2}{4t}} . f_0(X,Y) dX dY$$

- Ceci est par définition la convolution de f_0 par une gaussienne d'écart-type $\sigma = \sqrt{2t}$
- Pour plus de détails, me demander le pdf.

5. Approche variationnelle

5.2 Filtrage anisotrope

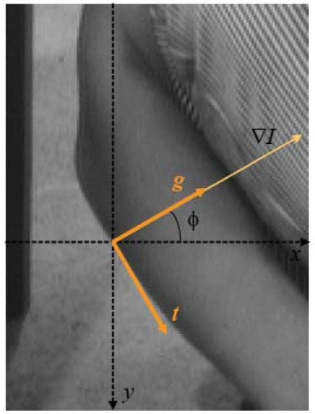
- L'équation de la chaleur est donc une autre façon de concevoir le filtrage gaussien.
- Cependant, un tel filtrage comme nous l'avons vu est isotrope et ne permet donc pas de préserver les zones sensibles de l'image (les contours ou transitions).

5. Approche variationnelle

5.2 Filtrage anisotrope

- Afin de lever cette limitation une écriture plus générale de l'équation de la chaleur est proposée :

$$\frac{\partial f}{\partial t} = c_t . f_{tt} + c_g . f_{gg}$$



5. Approche variationnelle

5.2 Filtrage anisotrope

- Une telle écriture permet de pondérer différemment les directions de diffusion (isophote et gradient) !!!

- On parle d'***anisotropie***

Pondération du gradient

$$\frac{\partial f}{\partial t} = c_t . f_{tt} + c_g . f_{gg}$$

Pondération de l'isophote

5. Approche variationnelle

5.2 Filtrage anisotrope

- Dans le cas où les deux coefficients de pondération sont égaux à 1, on retrouve l'équation de la chaleur

$$\frac{\partial f}{\partial t} = f_{tt} + f_{gg} = \Delta f$$

5. Approche variationnelle

5.3 Perona-Malik

- L'EDP proposée par Perona-Malik s'inspire de l'EDP de la chaleur, mais les auteurs y intégre une **non-linéarité** sous forme d'une fonction ***g*** à valeur positive, monotone et décroissante :

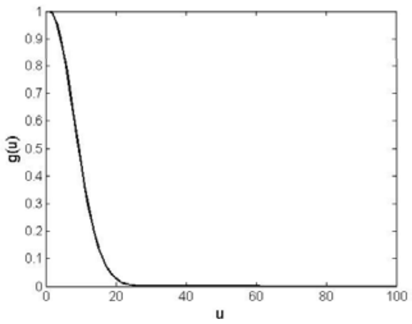
$$\frac{\partial f(x,y,t)}{\partial t} = div\left(g\left(\left|\nabla f(x,y)\right|\right)\nabla f(x,y)\right)$$

5. Approche variationnelle

5.3 Perona-Malik

- La fonction *g* la plus couramment utilisée est du type :

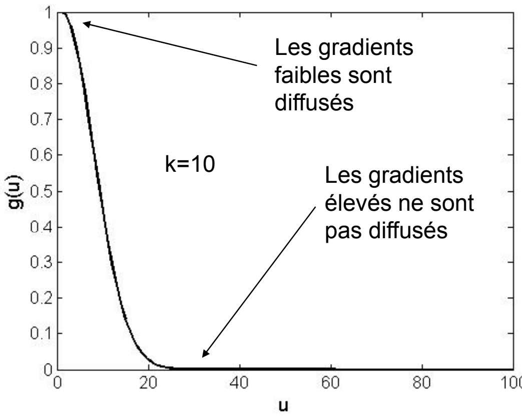
$$g(u) = e^{-\frac{u^2}{k^2}}$$



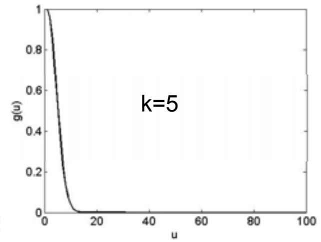
5. Approche variationnelle
5.3 Perona-Malik

- La fonction *g* permet de pondérer la diffusion en fonction de l'appartenance ou non du pixel courant à un contour :
  - Si le pixel appartient à un contour (gradient local élevé) alors *g(.)* renvoie une valeur proche de 0 : il n'y a pas diffusion
  - Si le pixel n'appartient pas à un contour (gradient local faible) alors *g(.)* renvoie cette fois ci une valeur importante non nulle : il y a diffusion

5. Approche variationnelle
5.3 Perona-Malik



La valeur de k permet de régler la sélectivité de g(.)



5. Approche variationnelle
5.3 Perona-Malik

5. Approche variationnelle
5.3 Perona-Malik

- L'équation de Perona-Malik peut s'écrire sous la forme généralisée décrite précédemment en prenant :

$$c_t = g\left(\left|\nabla f\right|\right)$$
$$c_g = g'\left(\left|\nabla f\right|\right).\left|\nabla f\right| + g\left(\left|\nabla f\right|\right)$$

- **Problème : c_g peut être négatif. La méthode est alors instable.**

5. Approche variationnelle
5.3 Perona-Malik

- Plusieurs améliorations ont été apportées par la suite. Les deux principales étant :
  - Lissage du gradient par convolution avec une gaussienne avant le calcul de la fonction *g* : **approche de *Catté et al***
  - Remplacement de la fonction *g* par une matrice *D*, le tenseur de structure associée à l'image (emprunt à la mécanique) : ***approche matricielle de Weickert***

Plan

- 1. Historique du TI
- 2. Images numériques
- 3. Outils fondamentaux
- 4. Filtrage d'image
- 5. Approche variationnelle
- 6. **Restauration : initiation**

6. Restauration : initiation

- **Problématique :**
  - On se place dans le cadre où l'image est dégradée de manière *convolutive* : $i(x,y) = b(x,y) * h(x,y)$.
  - Ces dégradation sont liées à un mouvement du capteur ou un défaut de mise au point.
  - On les traite en inversant un opérateur linéaire, donc supposé connu.

6. Restauration : initiation

- **Problématique :**
  - Dans le domaine spectrale : I(u,v)=B(u,v).H(u,v)
  - On cherche donc une fonction de transfert H'(u,v) telle que :
    $$o(x,y) = i(x,y) * h'(x,y)$$
  - Avec $$o(x,y) \approx b(x,y)$$
  - Soit $$O(u,v) = I(u,v).H'(u,v) = B(u,v).H(u,v).H'(u,v)$$

6. Restauration : initiation

- **Problématique :**
  - On en conclut donc que *H(u,v).H'(u,v)=1*
  - Et que *h'(x,y)* est telle que :

$$H'(u,v) = \frac{1}{H(u,v)}$$

  - **Problèmes :** Solution instable si *H(u,v)=0*

6. Restauration : initiation

- **Solution n°1**
  - On construit *H'(u,v)* de la manière suivante :

$$H'(u,v) = \frac{1}{H(u,v)+C}$$

  - Avec *C* une constante positive telle que *H(u,v) >> C*
  - Si *H(u,v)* tend vers 0 alors *H'(u,v)* tend vers 1/C (saturation)

6. Restauration : initiation

- **Solution n°1**
  - Résultat

  - **Problème :** Artefacts HF très visibles (amplification en 1/C)

6. Restauration : initiation

- **Solution n°2**
  - On construit *H'(u,v)* de la manière suivante :

$$H'(u,v) = \frac{H(u,v)}{H^2(u,v)+C}$$

  - Avec *C* une constante positive telle que $H^2(u,v) >> C$
  - Si *H(u,v)* tend vers 0 alors *H'(u,v)* tend vers 0

6. Restauration : initiation

- **Solution n°2**
  - Résultat


  - Proche du filtrage optimal

6. Restauration : initiation

- **Filtrage optimal : Filtre de Wiener**
  - On construit *H'(u,v)* de façon à ce que :

$$H'(u,v) = \frac{H(u,v)}{H^2(u,v) + \frac{\Phi_{nn}}{\Phi_{bb}}}$$

  - Φ_{ab} (u,v) étant la TF de la corrélation entre a(x,y) et b(x,y)
  - *n(x,y)* représentant un bruit additif détériorant l'image

6. Restauration : initiation

- **Filtrage optimal : Filtre de Wiener**
  - Rappel : fonction de corrélation entre a(x,y) et b(x,y)

$$\phi_{ab}(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} a(\xi - x, \eta - y) b(\xi, \eta) d\xi d\eta$$

  - Φ_{aa} *(x,y)* : autocorrélation

6. Restauration : initiation

- **Filtrage optimal : Filtre de Wiener**
  - **Construction de *H'(u,v)* :** on cherche à minimiser une fonction énergétique caractérisant la qualité de restauration :

$$E = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (o(x,y) - d(x,y))^2 dx dy$$

  - *Avec d(x,y)* l'image désirée et *o(x,y)=i(x,y)*h'(x,y)*

6. Restauration : initiation

- **Filtrage optimal : Filtre de Wiener**
  - Ainsi :

$$E = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(o^2(x,y) - 2o(x,y)d(x,y) + d^2(x,y))dxdy$$

  - Or par définition

$$\begin{aligned} o^2(x,y) &= \textstyle\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} i(x-\xi, y-\eta)h'(\xi,\eta)d\xi d\eta \times \\ &\quad \textstyle\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} i(x-\alpha, y-\beta)h'(\alpha,\beta)d\alpha d\beta \end{aligned}$$

6. Restauration : initiation

- **Filtrage optimal : Filtre de Wiener**
  - Et donc :

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} o^2(x,y)dx\,dy = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \phi_{ii}(\xi-\alpha, \eta-\beta)h'(\xi,\eta)h'(\alpha,\beta)d\xi d\eta d\alpha d\beta$$

  - De plus

$$o(x,y)d(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} i(x-\xi, y-\eta)h'(\xi,\eta)d(x,y)d\xi d\eta$$

6. Restauration : initiation

- **Filtrage optimal : Filtre de Wiener**
  - Donc :

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} o(x,y)d(x,y)dxdy = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \phi_{id}(\xi,\eta)h'(\xi,\eta)d\xi d\eta$$

  - Or

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} d^2(x,y)dxdy = \phi_{dd}(0,0)$$

6. Restauration : initiation

- **Filtrage optimal : Filtre de Wiener**
  - Enfin :

$$\begin{aligned} E &= \textstyle\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \phi_{ii}(\xi-\alpha, \eta-\beta)h'(\xi,\eta)h'(\alpha,\beta)d\xi d\eta d\alpha d\beta \\ &\quad \textstyle -2\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \phi_{id}(\xi,\eta)h'(\xi,\eta)d\xi d\eta + \phi_{dd}(0,0) \end{aligned}$$

  - Il s’agit du critère à minimiser pour déterminer *h’(x,y)*

6. Restauration : initiation

- **Filtrage optimal : Filtre de Wiener**
  - Solution (Euler-Lagrange):

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\phi_{ii}(\xi-\alpha,\eta-\beta)h'(\alpha,\beta)d\alpha d\beta-\phi_{id}(\xi,\eta)=0$$

  - Soit :

$\phi_{id}=\phi_{ii}*h'$ ou $\boxed{\Phi_{id}=H'\Phi_{ii}}$ Domaine fréquentielle

6. Restauration : initiation

- **Filtrage optimal : Filtre de Wiener**
  - Solution : Cas du bruit additif

$$i(x,y)=b(x,y)+n(x,y).$$ Avec *d(x,y)* =*b(x,y)*

  - Donc

$$\Phi_{id}=\Phi_{bb}+\Phi_{nb}$$
$$\Phi_{ii}=\Phi_{bb}+\Phi_{bn}+\Phi_{nb}+\Phi_{nn}$$

6. Restauration : initiation

- **Filtrage optimal : Filtre de Wiener**
  - Hypothèse : bruit non corrélé aux données

$$H'=\frac{\Phi_{id}}{\Phi_{ii}}=\frac{\Phi_{bb}}{\Phi_{bb}+\Phi_{nn}}=\frac{1}{1+\Phi_{nn}/\Phi_{bb}}.$$

6. Restauration : initiation

- **Filtrage optimal : Filtre de Wiener**
  - Solution : Cas de détériorations convolutive et additive

$$i=b*h+n$$ Avec *d(x,y)* =*b(x,y)*

  - Donc

$$\Phi_{id}-H\Phi_{bb}+\Phi_{nb}$$
$$\Phi_{ii}=H^2\Phi_{bb}+H(\Phi_{nb}+\Phi_{bn})+\Phi_{nn}$$

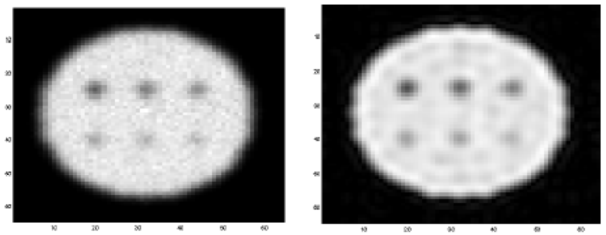
6. Restauration : initiation

- **Filtrage optimal : Filtre de Wiener**
  - Même hypothèse sur le bruit

$$H' = \frac{\Phi_{id}}{\Phi_{ii}} = \frac{H\Phi_{bb}}{H^2\Phi_{bb} + \Phi_{nn}}$$

6. Restauration : initiation

- **Filtrage optimal : Filtre de Wiener**
  - Résultat



Bibliographie

- J.P. Cocquerez et S. Philipp « *Analyse d'images : filtrage et segmentation* » Masson 1995.
- R.C. Gonzalez et Woods « *Digital Image Processing - 2d edition* » Addison Wesley 2002.
- A. Rosenfeld et A.C. Kak « *Digital picture processing* » Academic Press - London 1982.
- H. Maître (ss la direction de) « *Le traitement des images* » Hermes Lavoisier IC2 2003.
- J.R. Parker « *Algorithms for Image Processing and Computer Vision* » Wiley & Sons 1997.
- S. Bres, J.M. Jolion, F. Lebourgeois « *Traitement et analyse des images numériques»* Hermes Lavoisier 2003
- I.T. Young, J.J. Gerbrands et L.J. Van Vliet « *Fundamentals of Image Processing* » Université de Delft. http://www.ph.tn.tudelft.nl/~lucas/publications/1995/FIP95TYJGLV/FIP2.2.pdf)
- D. Lingrand « *Introduction au Traitement d'images* » Vuibert 2004.