École polytechnique de l'université d'Orléans

TRAITEMENT DES IMAGES

Christophe LÉGER

2005 – 2006

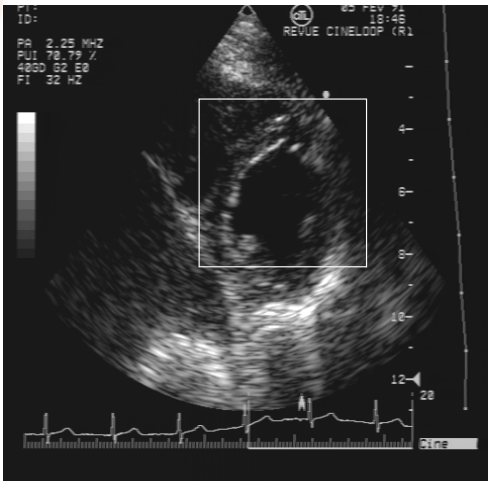
Polytech' Orléans

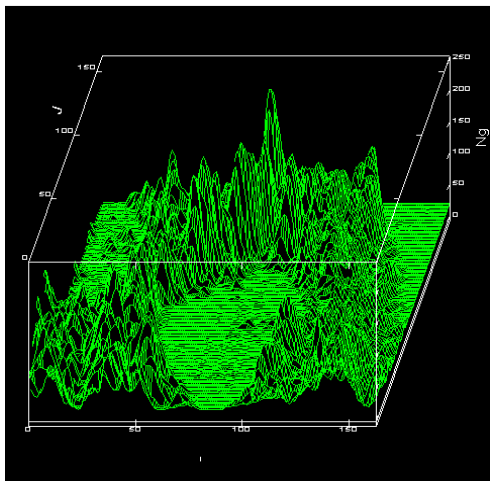
Traitements point à point

Traitement des images
Polytech'Orléans, 2005 – 2006
Christophe LÉGER

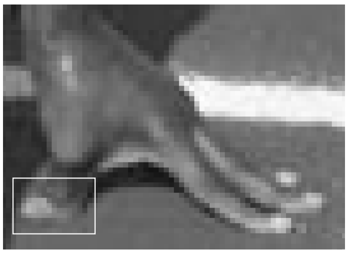
École polytechnique de l'université d'Orléans

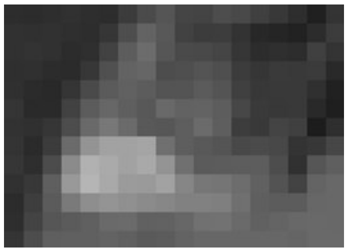
Représentation des images numériques





Représentation des images numériques





Représentation des images numériques

254	64	127	14	14	254	64	127	14	14	254	64	127	14	14	254
147	64	127	14	14	254	64	127	14	14	254	64	127	14	14	254
54	64	127	14	14	254	64	127	14	14	254	64	127	14	14	254
24	64	127	14	14	254	64	127	14	14	254	64	127	14	14	254
25	64	127	14	14	254	64	127	14	14	254	64	127	14	14	254
25	64	127	14	14	254	64	17	14	14	254	64	127	14	14	254
15	64	127	14	14	4	64	127	14	14	254	64	127	14	14	254
254	64	127	14	14	254	64	12	14	14	254	64	127	14	14	254
254	64	127	14	14	254	64	127	14	14	24	64	127	14	14	254
254	64	127	14	14	254	64	127	14	14	254	64	127	14	14	254
254	64	127	14	14	254	64	127	14	14	254	64	127	14	14	254
254	64	127	14	14	254	64	127	14	14	254	64	127	14	14	254
254	64	127	14	14	254	64	127	14	14	254	64	127	14	14	254
254	64	127	14	14	254	64	127	14	14	254	64	127	14	14	254
254	64	127	14	14	254	64	127	14	14	254	64	127	14	14	254
254	64	127	14	14	254	64	127	14	14	254	64	127	14	14	254

- Tableaux 2D

  `pIma[i][j] = Pix;`

- Vectorisation en tableaux 1D

  `pIma[i*nCol+j] = Pix;`

- Classe CImage

  `m_pIma(i,j) = Pix;`

- Classe CImageCoul

  ```
  m_pImaR(i,j) = PixR;
  m_pImaV(i,j) = PixV;
  m_pImaB(i,j) = PixB;
  ```

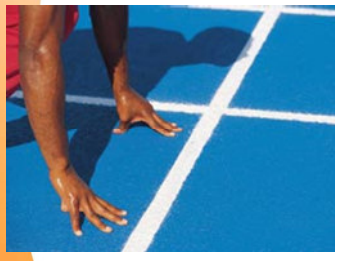
Représentation des images numériques

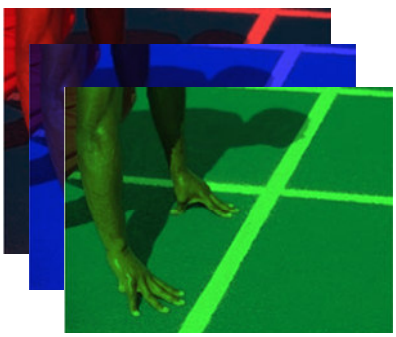
- 1 octet (8 bits) ⇒ [char] unsigned char
- 2 octets (16 bits) ⇒ short, unsigned short, ~~int, unsigned~~

```
#define TIMA unsigned char
class CImage
{
  private:
        int     m_nLig, m_nCol;
        TIMA*   m_pIma;
  public:
        CImage (int, int, TIMA = (TIMA) 0);
        ~CImage () { if (m_pIma != NULL) delete[] m_pIma;}
};
CImage::CImage (int Lig, int Col, TIMA pix)
{
  m_nLig = Lig; m_nCol = Col;
  m_pIma = new TIMA [m_nLig * m_nCol];
  for (int i=0;i<m_nLig;++i)
    for (int j=0;j<m_nCol;++j)
        m_pIma[i*m_nCol+j] = Pix;
}
```


Représentation des images numériques

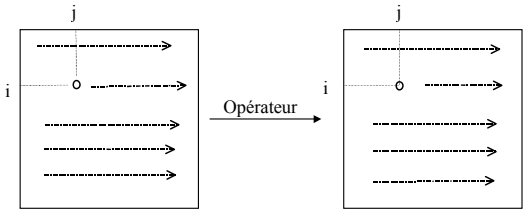
- Chaque pixel est composé de rouge, vert et bleu
- Chaque triplet RVB est codé sur 256 bits ⇒ 16 10^6 couleurs
- Gris = (R + V + B) / 3
- Gris = 0.299 × R + 0.587 × V + 0.114 × B (luminosité≈vert)





Traitements point à point

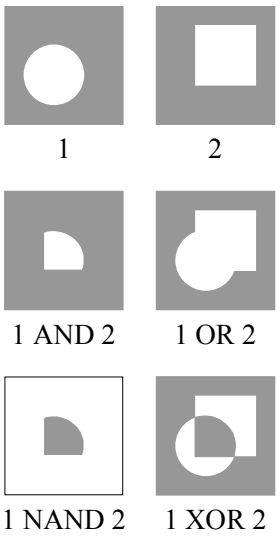
Chaque pixel de l'image initiale est traité indépendamment de ses voisins



- Opérations logiques
- Opérations arithmétiques
- Fonction de transfert
- Histogrammes
- Seuillage

1 – Opérations logiques

- Mise en œuvre
  - Opérations logiques sur des images (et, ou, ou exclusif, …) binaires (codées sur 1 seul bit)
- Applications
  - Sélection d'objets dans une zone (et logique entre l'image des objets et la zone)
  - Mise en évidence de l'intersection d'objets dans plusieurs images
  - Mise en évidence du déplacement d'un objet
  - Outil de base de la morphologie mathématique
- Extension aux images à niveaux de gris
  - Opération logique sur chaque bit (codage habituel ⇒ non conservation des propriétés de proximité)



2 – Opérations arithmétiques

Addition

- Mise en œuvre

  Chaque pixel de l'image initiale est additionné avec le pixel de même position dans une autre image pour donner un pixel résultat à la même position

- Problèmes

  Dynamique double ⇒ normalisation (division par 2 du résultat)

- Application

  Moyennage ⇒ réduction du bruit d'un facteur $\sigma/\sqrt{N}$

Soustraction

- Mise en œuvre

  idem addition

- Problèmes
  - Dynamique double
  - Normalisation difficile, dépendant des images soustraites : (images proches ⇒ * n, différentes ⇒ / 2)
- Applications
  - Suppression de tendance
  - Recalage par différence minimum
  - Compression (codage delta)

2 – Opérations arithmétiques

- Autres opérations linéaires
  - $I_R = I_{I1} * I_{I2}$
  - $I_R = I_{I1} / I_{I2}$

  Signification physique ?
- Autres opérations logiques
  - $I_R = Max \{I_{I1}, I_{I2}\}$
  - $I_R = Min \{I_{I1}, I_{I2}\}$

- Opérations non linéaires
  - $I_R = ABS \{I_I\}$
  - $I_R = Neg \{I_I\}$
  - $I_R = I_I^2$
  - $I_R = \sqrt{I_I}$
  - $I_R = exp \{I_I\}$
  - $I_R = log \{I_I\}$
  - …

2 – Moyennage

Soit une image réelle $I_{MES}(i,j)$ contenant l'information image $I_V(i,j)$ plus un terme de bruit b (i,j). On peut écrire :

$$I_{MES}(i,j) = I_V(i,j) + b(i,j)$$

où $I_V(i,j)$ est l'image non dégradée,
et b (i,j) est un bruit supposé blanc indépendant de l'image (E (b) = 0 et Var (b) = σ^2)

Par accumulation on obtient une image moyenne de toutes les N images acquises :

$$\overline{I_{MES}(i,j)} = \frac{1}{N}\sum_{n=0}^{N-1} I_{MES_n}(i,j)$$

$\overline{I_{MES}(i,j)}$ peut être considéré comme un estimateur de $I_V(i,j)$. On peut alors calculer la moyenne $E\left(\overline{I_{MES}(i,j)}\right)$ et l'écart type $Var\left(\overline{I_{MES}(i,j)}\right)$ de cet estimateur.

Moyenne de $\overline{I_{MES}(i,j)}$:

$$E\left(\overline{I_{MES}(i,j)}\right) = E\left(\frac{1}{N}\sum_{n=0}^{N-1} I_{MES_n}(i,j)\right)$$

$$E\left(\overline{I_{MES}(i,j)}\right) = \frac{1}{N}E\left(\sum_{n=0}^{N-1} I_V(i,j)\right) + \frac{1}{N}E\left(\sum_{n=0}^{N-1} b(i,j)\right)$$

Comme $I_V(i,j)$ est une grandeur déterministe et que la moyenne du bruit est nulle, on a :

$$\text{Moyenne de } \overline{I_{MES}(i,j)} = I_V(i,j)$$

Écart type de $\overline{I_{MES}(i,j)}$

$$Var\left(\overline{I_{MES}(i,j)}\right) = Var\left(\frac{1}{N}\sum_{n=0}^{N-1} I_{MES_n}(i,j)\right)$$

$$Var\left(\overline{I_{MES}(i,j)}\right) = \frac{1}{N^2}Var\left(\sum_{n=0}^{N-1} I_V(i,j)\right) + \frac{1}{N^2}Var\left(\sum_{n=0}^{N-1} b_n(i,j)\right)$$

On peut séparer les 2 termes de la somme car le bruit et l'image non dégradée sont indépendants. De plus la variance de l'image est nulle car c'est une grandeur déterministe. On obtient alors :

$$Var\left(\overline{I_{MES}(i,j)}\right) = \frac{1}{N^2}Var\left(\sum_{n=0}^{N-1} b_n(i,j)\right) = \frac{1}{N^2}N\sigma^2 = \frac{\sigma^2}{N}$$

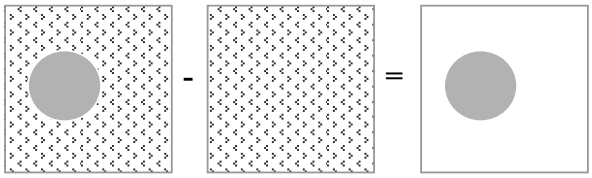
d'où

$$\text{Ecart type de } \left(\overline{I_{MES}(i,j)}\right) = \frac{\sigma}{\sqrt{N}}$$

Le moyennage réduit l'influence du bruit d'un facteur égal à la racine carrée du nombre N d'accumulations.

2 – Suppression de tendance

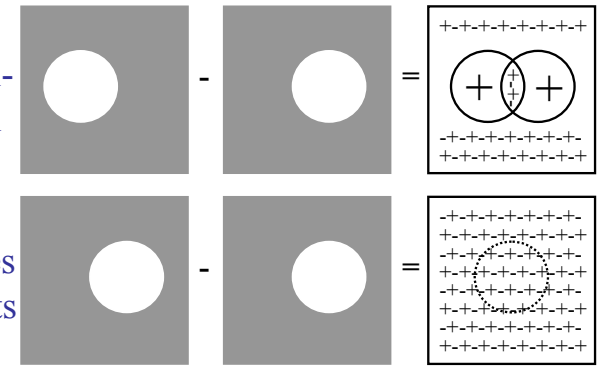
- La suppression d'une tendance ou d'un fond est réalisée en soustrayant de l'image mixte une image qui contient la tendance ou le fond



- La mise en œuvre est souvent difficile car elle nécessite que la valeur moyenne des niveaux de gris de l'image reste constante au cours du temps

2 – Recalage par soustraction

- Le recalage de deux images ayant une partie commune peut être réalisé par comparaison, en recherchant la position qui donne une différence minimum
- Limitations :
  - bruit+dérive de luminosité moyenne ⇒ réduction des performances de la méthode
  - Dans ce cas, on étudie les signes des pixels résultats et leur disposition dans l'image



Traitements point à point 13

2 – Codage Delta

- Compression de données simple à réaliser dans le cas de séquences d'images
- Plutôt que de transmettre toutes les images d'une séquence, on transmet uniquement leur différence
- La différence est souvent codée sur moins de bits si les images successives se ressemblent

Traitements point à point 14

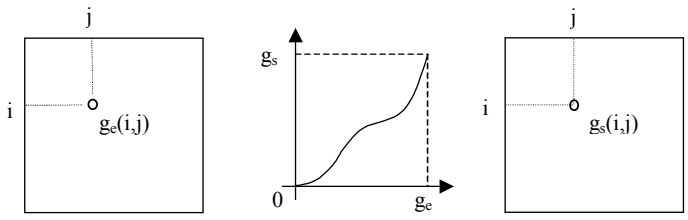
2 – Autres opérations arithmétiques

Multiplication et division

- Intérêt
  - Très peu, sinon pour normaliser lorsqu'un des deux opérandes est constant
- Mise en œuvre
  - Chaque pixel de l'image initiale est multiplié / divisé par le pixel de même position dans une autre image et donne un pixel résultat à la même position
- Problèmes
  - Temps de calcul beaucoup plus long que pour une addition / soustraction
  - Pour la division, résultat non entier, quel que soit le type des opérandes
- Solutions
  - Troncature pour les normalisations d'addition ou de soustraction
  - Division par 2 réalisée par décalage à droite, opération rapide pour un µp

3 – Fonctions de transfert : principe

- Une fonction est utilisée pour faire correspondre à un niveau de gris une nouvelle valeur de niveau de gris unique



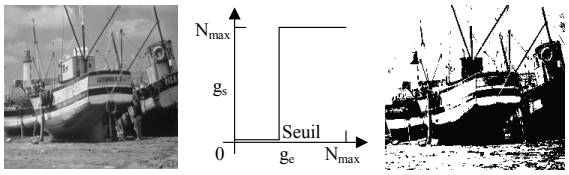
Fonction de transfert, $g_s(i,j)=f(g_e(i,j))$

3 – Fonctions de transfert : principe

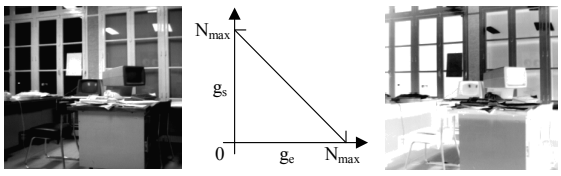
- Les fonctions de transfert ont pour effet de modifier l'information contenue dans l'image :
  - soit pour qu'elle puisse être utilisable par un traitement ultérieur (pré-traitement)
  - soit pour améliorer l'aspect visuel de l'image
- Attention : les fonctions de transfert conduisent souvent à une perte d'information (dégradation de l'image)

3 – Fonctions de transfert : applications

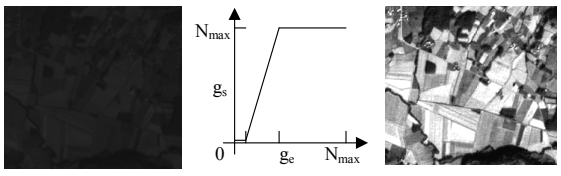
- Binarisation d'images
  - passage d'une image à 256 niveaux de gris à une image à deux niveaux de gris



- Inversion d'images
  - effet « négatif » : les niveaux de gris les plus fort deviennent les plus faibles, et inversement

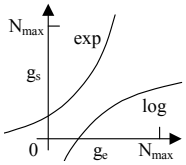


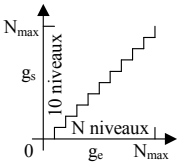
- Amplification de contraste
  - amélioration du confort visuel d'une image qui a peu de contraste

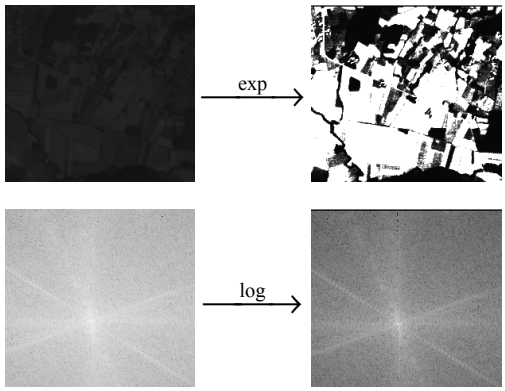


3 – Fonctions de transfert : applications

- Modification de la quantification
  - n'a de sens que pour réduire le nombre de niveaux de quantification
- Fonctions non linéaires
  - correction des non linéarités d'un capteur (log, exp, …)
  - représentation d'une image avec une grande dynamique

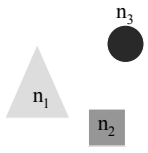


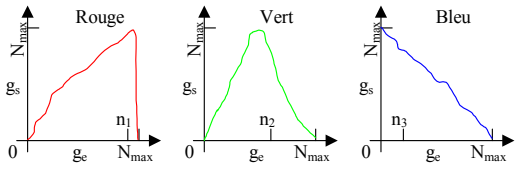


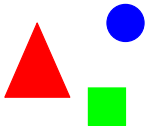


3 – Fonctions de transfert : applications

- Multifonctions, représentation en fausses couleurs
  - faire ressortir des zones de niveaux de gris proches grâce à des couleurs différentes
  - Chaud (rouge) / Froid (bleu)
  - Sommets (rouge) / plaines (vert)

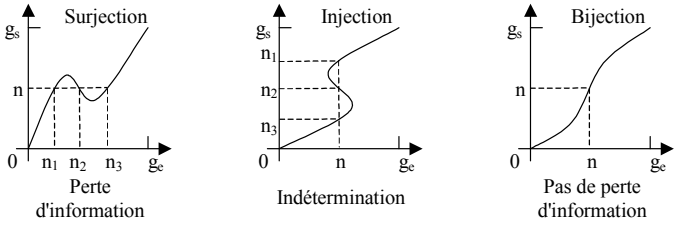






3 – Fonctions de transfert : problèmes

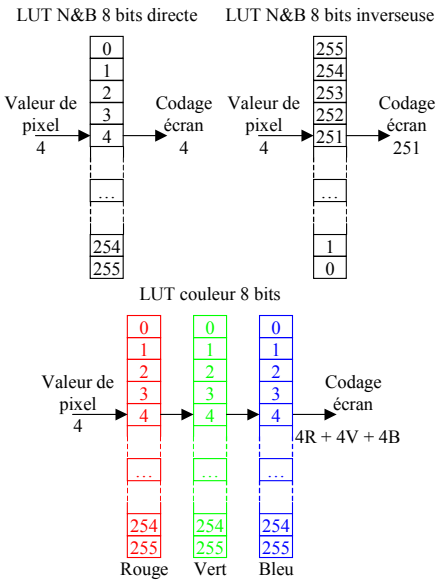
- Les opérations ne sont réversibles que si les fonctions de transfert sont bijectives :
  - surjection ⇒ perte d'information
  - injection ⇒ indétermination
  - bijection = pas de perte d'information



3 – Fonctions de transfert : mise en œuvre

- Plutôt que de modifier tous les pixels de l'image, on utilise un circuit électronique spécialisé (Look Up Table, LUT, ou table de transfert)
- Une zone mémoire (tableau) à N entrées (N=256 pour une LUT 8 bits) contient les valeurs de codage (éclairement) des pixels à l'écran
- A un pixel de l'image (entrée de la LUT), correspond un codage écran

⇒ N modifications dans la LUT plutôt que N×M modifications dans l'image



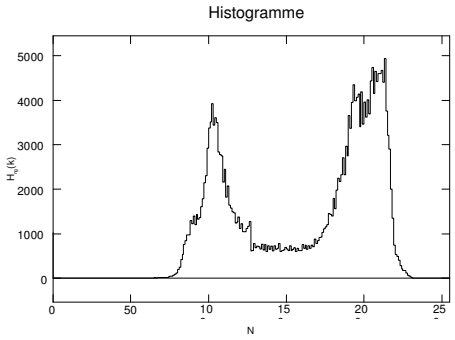
4 – Histogrammes : principe

L'histogramme d'une image est une fonction qui représente le nombre de pixels de l'image ayant un niveau de gris donné

$$H_{NG}(k) = \sum_{i}^{N} \sum_{j}^{M} \left(Im[i,j] / Im[i,j] = k \right)$$

- $H_{NG}(k)$ est le nombre de pixels de l'image au niveau k
- En sommant les $H_{NG}(k)$, on retrouve le nombre total de pixels de l'image :

$$\sum_{k=0}^{NG_{max}} H_{NG}(k) = N \times M$$



4 – Histogrammes : principe

- Assimilation des histogrammes à une probabilité

$$P_{NG}(k) = \frac{1}{N \times M} H_{NG}(k)$$

$P_{NG}(k)$ est la probabilité pour qu'un pixel Im[i,j] de l'image soit au niveau k

- La somme des probabilités est égale à 1 :

$$\sum_{k=0}^{NG_{max}} P_{NG}(k) = 1$$

4 – Histogrammes : programmation

```
#include <iostream.h>

TIMA& CImage::operator()(int i, int j)
{
  if ( (0<i) || (i>=m_nLig) ||
       (0<j) || (j>=m_nCol) )
    cerr << "Dépassement de tableau";
  else
    return m_iIma(i*m_nCol+j);
}
```

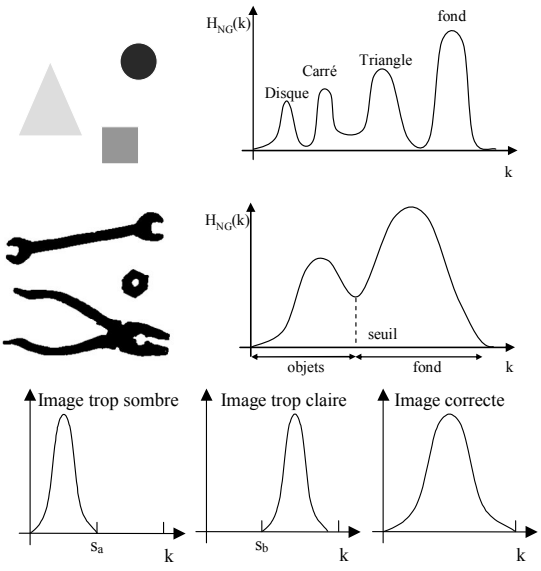
```
void Histo(const CImage& iIma)
{
  int i, j;
  int n;
  double* pHis;

  n = sizeof (TIMA) * 8;
  pHis = new double [n];
  for (i=0;i<n;++i) pHis[i] = 0;

  for (i=0;i<iIma.Lig();++i)
    for (j=0;j<iIma.Col();++j)
        pHis[ iIma(i,j) ]++;
  ...
  delete[] pHis;
 }
```

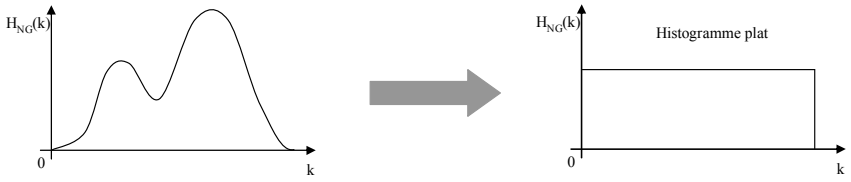

4 – Histogrammes : applications

- Détermination de la composition d'une scène par l'étude des maxima locaux (modes de l'histogramme)
- Utilisation pour déterminer un seuil et séparer les objets d'une scène (segmentation)
- Réglage dynamique d'une acquisition (gain et offset de la carte, ouverture de l'objectif)



4 – Histogrammes : égalisation

- L'objectif est de modifier les valeurs des pixels Im[i,j] de l'image de telle sorte que l'histogramme de l'image modifiée soit plat, c'est à dire que toutes les valeurs soient équiprobables

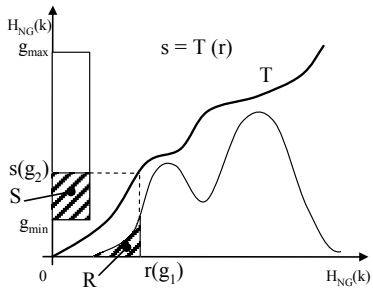


- Il faut appliquer une transformation aux pixels de l'image pour que son histogramme soit uniforme

Traitements point à point 27

4 – Histogrammes : égalisation

- Approche intuitive :
  - Comment faire progresser $r(g_1)$ et $s(g_2)$ pour que les deux surfaces S et R soient égales (S=R) ?
  - Si la valeur de $r(g_1)$ est faible, il faut que la pente de la transformation T soit grande



  - La transformation T doit respecter l'ordre des niveaux de gris : un pixel ayant un niveau de gris supérieur à celui d'un autre pixel ne peut se retrouver à un niveau de gris inférieur après transformation

Traitements point à point 28

4 – Histogrammes : égalisation

$$\int_{g_{min}}^{g_2} K \cdot dg_2' = \int_{0}^{g_1} r(g_1') \cdot dg_1', \; avec \; K = \frac{1}{g_{max} - g_{min}}$$

$$\Rightarrow K \cdot (g_2 - g_{min}) = Fr(g_1), \; fonction \; de \; répartition \; de \; r$$

$$\Rightarrow g_2 = \frac{Fr(r)}{K} + g_{min}$$

$$\Rightarrow T(r) = \frac{Fr(r)}{K} + g_{min} \qquad avec \; K = \frac{1}{g_{max} - g_{min}}$$

4 – Histogrammes : modification

- Le même principe peut être utilisé pour obtenir un histogramme de forme spécifiée
- Soit on détermine directement la formule de passage entre les deux histogrammes par égalisation de surface
- Soit on utilise une méthode indirecte qui passe par une égalisation d 'histogramme :

$$r(g_1) \xrightarrow{T_1} f(g_2) \xrightarrow{T_2} s(g_3)$$

histo. initial — *histo. égalisé* — *autre forme*

- T_2 est la fonction inverse de celle qu 'il faudrait appliquer pour passer de $s(g_3)$ à un histogramme égalisé

4 – Histogrammes : modification

- Calcul direct d'un histogramme de type exponentiel :

$$s(g2) = \alpha \cdot e^{-\alpha(g_2 - g_{2\,Min})}, \text{ avec } g_{2\,Min} \le g_2$$

- Relation intégrale d'égalité de surfaces :

$$\int_{g_{2\,Min}}^{g_2} \alpha \cdot e^{-\alpha(g_2' - g_{2\,Min})} \cdot dg_2' = \int_0^{g_1} r(g_1') \cdot dg_1'$$

- Par intégration, on obtient :

$$\left[-e^{-\alpha(g_2' - g_{2\,Min})} \right]_{g_{2\,Min}}^{g_2} = Fr(g_1)$$

- d 'où :

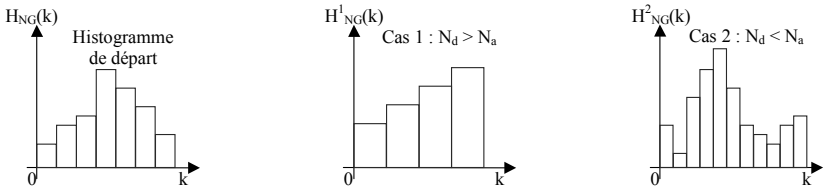
$$g_2 = g_{2\,Min} - \frac{1}{\alpha} \cdot \ln[1 - Fr(g_1)]$$

4 – Histogrammes : modification

	Histogramme final	Fonction de transfert
Uniforme	$P_s(S) = \frac{1}{g_{Max} - g_{Min}} \quad (g_{Min} \le g \le g_{Max})$	$T = (g_{Max} - g_{Min}) \cdot Pr(r) + g_{Min}$
Exponentiel	$P_s(S) = \alpha e^{-\alpha(s - g_{Min})} \quad (g_{Min} \le g)$	$T = g_{Min} - \frac{1}{\alpha} \ln[1 - Pr(r)]$
Rayleigh	$P_s(S) = \frac{s - g_{Min}}{\alpha^2} \alpha e^{\frac{-(s - g_{Min})^2}{2\alpha^2}} \quad (g_{Min} \le g)$	$T = g_{Min} + \left[2\alpha^2 \ln \frac{1}{1 - Pr(r)} \right]^{\frac{1}{2}}$
Hyperbolique racine cubique	$P_s(S) = \frac{1}{3} \frac{s^{-\frac{2}{3}}}{g_{Max}^{\frac{1}{3}} - g_{Min}^{\frac{1}{3}}}$	$T = \left[\left(g_{Max}^{\frac{1}{3}} - g_{Min}^{\frac{1}{3}} \right) \cdot Pr(r) + g_{Min}^{\frac{1}{3}} \right]^3$
Hyperbolique logarithmique	$P_s(S) = \frac{1}{s(\ln(g_{Max}) - \ln(g_{Min}))}$	$T = g_{Min} \left[\frac{g_{Max}}{g_{Min}} \right]^{Pr(r)}$

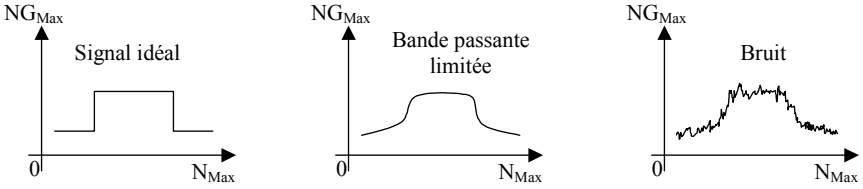
4 – Histogrammes : modification

- Soient N_d et N_a les nombres de niveaux de gris de départ et d'arrivée :
  - Si $N_d > N_a$: l'égalisation se fait par accumulation de niveaux de gris de départ dans un seul d 'arrivée ⇒ perte d'information, sensible dans les zones de niveaux de gris de faible occurence
  - Cas $N_d < N_a$: l'égalisation se fait par répartition des états les plus fréquents sur plusieurs niveaux de l'histogramme d'arrivée ⇒ pas de perte d'information, critères de voisinage



5 – Seuillage : principe

- Le seuillage est utilisé pour séparer une image en zones d'intérêts différents (fond / classe d'objets)
- Problème : il faut trouver la valeur du seuil, malgré une limite entre fond et objets difficile à trouver en raison du bruit d'acquisition et de la bande passante limitée



5 – Seuillage : principe

- Définition du seuillage
  - Soit un pixel g[i,j] de l'image de départ
  - Sa valeur est modifiée selon la relation suivante :

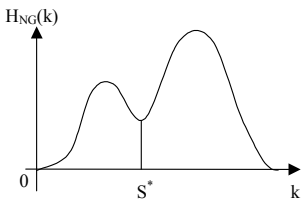
$$g[i,j] = \begin{cases} b_0 & \text{si } g[i,j] < S \\ b_1 & \text{si } g[i,j] \geq S \end{cases} \qquad \text{avec } 0 \leq S < g_{max}$$

  - On choisit généralement $b_0 = 0$ et $b_1 = g_{max}$
- Le problème est d'estimer S (c'est à dire trouver S^* qui correspond au seuil optimal)

5 – Seuillage : méthode de la vallée

- Quand l'histogramme présente deux modes « aisément » discernables, on estime la valeur du seuil de la manière suivante :

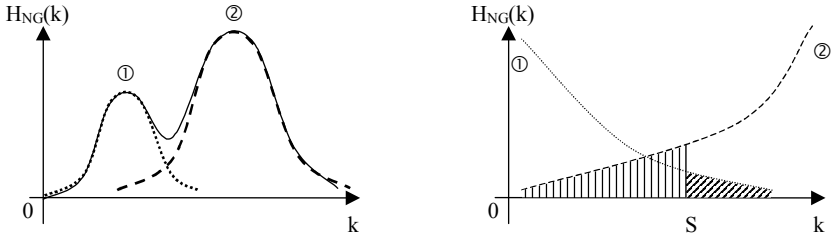
S^* = min (entre les deux modes)



- Pour améliorer la précision sur S^*, il est souvent nécessaire de lisser l'histogramme

5 – Seuillage : méthode du minimum d'erreur

- Chaque mode de l'histogramme est modélisé par une densité de probabilité gaussienne
- Le seuil recherché correspond alors au point d'intersection des deux gaussiennes, qui correspond à l'aire « résiduelle » minimum



5 – Seuillage : méthode du minimum d'erreur

Hypothèse : les événements ① et ② sont disjoints

On a :

p (g) = p_1 . p (g/1) + p_2 . p (g/2) = probabilité pour qu'un pixel ait un niveau g

p (g/1) = probabilité pour qu'un pixel ait un niveau de gris g sachant qu'il appartient à l'objet ① dans l'image

p_1 = probabilité a priori d'avoir un point appartenant à l'objet ① dans l'image

$$p(g/1) = \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{(g-\mu_1)^2}{2\sigma_1^2}\right), \text{ et } p_1 = \int_{-\infty}^{+\infty} p(g/1)\cdot dg$$

De même, pour p (g/2) et p_2, on a :

$$p(g/2) = \frac{1}{\sqrt{2\pi}\sigma_2} \exp\left(-\frac{(g-\mu_2)^2}{2\sigma_2^2}\right), \text{ et } p_2 = \int_{-\infty}^{+\infty} p(g/2)\cdot dg$$

5 – Seuillage : méthode du minimum d'erreur

- La somme des deux aires hachurées est donnée par :

$$A = \int_{-\infty}^{S} p_2 \cdot p(g/2) \cdot dg + \int_{S}^{+\infty} p_1 \cdot p(g/1) \cdot dg$$

- L'annulation de la dérivée de cette somme donne S^* :

$$\frac{\partial A}{\partial g} = 0 \Rightarrow \frac{\partial}{\partial g}\int_{-\infty}^{S} p_2 \cdot p(g/2) \cdot dg + \frac{\partial}{\partial g}\int_{S}^{+\infty} p_1 \cdot p(g/1) \cdot dg = 0$$

$$\Rightarrow p_2 \cdot p(g/2) = p_1 \cdot p(g/1) \qquad \left[\text{car } \frac{\partial}{\partial g}\int_{a}^{b} f(g) \cdot dg = f(b) - f(a) \right]$$

$$\Rightarrow \frac{p_2}{\sigma_2} \exp\left(-\frac{(S^* - \mu_2)^2}{2\sigma_2^2} \right) = \frac{p_1}{\sigma_1} \exp\left(-\frac{(S^* - \mu_1)^2}{2\sigma_1^2} \right)$$

- Il faut résoudre cette équation du second degré en S^*, après avoir estimé p_1, p_2, σ_1, σ_2, μ_1, et μ_2. Pour cela, on utilise une méthode moindres carrés ou itérative

Traitements point à point

Efficacité ≠ Rendu

Attention à la perte d'information

- addition, soustraction : résultats sur 9 bits ramené sur 8
- multiplication, division, …
- fonctions de transfert :
  - surjection ⇒ perte d'information
  - injection ⇒ indétermination
  - bijection = pas de perte d'information
- seuillage, fenêtrage, quantification, amplification de contraste ⇒ perte d'information

Traitement point à point

- Exercices :
  1. Doubler l'intensité d'une image (au moindre coût)
     (Note : 256 niveaux de gris, 0 = noir, 255 = blanc)
  2. Seuiller au niveau de gris x $\begin{cases} \geq x \Rightarrow 255 \\ < x \Rightarrow 0 \end{cases}$
  3. Rehausser entre x et y $\begin{cases} < x \text{ ou } > y \Rightarrow \text{pas de changement} \\ \geq x \text{ et } \leq y \Rightarrow 255 \end{cases}$
  4. Soustraire k de tous les niveaux (au moindre coût)
  5. Améliorer le contraste $\begin{cases} \text{niveau le plus bas } x \Rightarrow x = 0 \\ \text{niveau le plus haut } y \Rightarrow y = 255 \end{cases}$

Polytech' Orléans

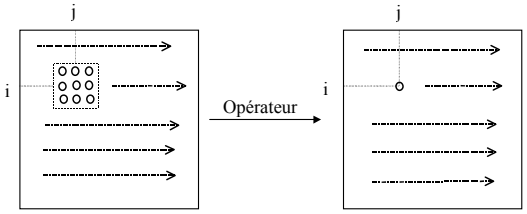
Traitements locaux

Traitement des images
Polytech'Orléans, 2005 – 2006
Christophe LÉGER

Traitements locaux

Chaque pixel de l'image initiale est traité relativement à ses voisins

- Les images possédant une certaine redondance spatiale, des pixels voisins ont généralement des caractéristiques identiques
- Il faut donc opérer des transformations qui, pour chaque pixel, tiennent compte de son voisinage
- Traitements locaux = Traitements de voisinage



- Filtrage linéaire passe-bas
  - Réduction du bruit
- Filtrage linéaire passe-haut
  - Segmentation
- Filtrage non linéaire

1 – Filtrage linéaire : principe

- Soit une image I(i,j), 0≤i<N, 0≤j<M, et un filtre 2D H(k,l), -K/2≤k≤K/2, -L/2≤l≤L/2. On définit le filtrage de l'image I(i,j) par H(k,l) par :

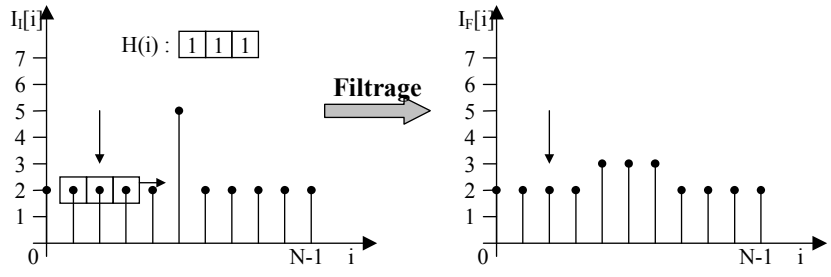
$$I_F(i,j)=\frac{1}{C}\sum_{k=-K/2}^{K/2}\sum_{l=-L/2}^{L/2}H(k,l)\cdot I(i+k,j+l)$$

- C est un coefficient de normalisation
- L'équation précédente peut être vue comme une convolution discrète où H(k,l) est la réponse impulsionnelle du filtre, qui le caractérise complètement :

$$I_F(i,j)=H(k,l)*I(i,j)$$

1 – Filtrage linéaire : mise en œuvre

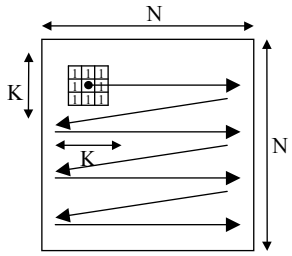
Exemple de filtrage 1D (moyenne sur 3 échantillons consécutifs), pour i variant de 0 à N-1 :



$$H[i]=[1,1,1] \qquad I_F[i]=\frac{1}{3}\left[1\times I_I[i-1]+1\times I_I[i]+1\times I_I[i+1]\right]$$

1 – Filtrage linéaire : mise en œuvre

Exemple de filtrage 2D (moyenne 3 × 3), pour i et j variant de 0 à N-1 (N × N est la dimension de l'image) :



$$I_F[i,j]=\frac{1}{9}[\ 1\times I(i-1,j-1)\ +1\times I(i-1,j)\ +1\times I(i-1,j+1)\ +1\times I(i,j-1)\ +1\times I(i,j)\ +1\times I(i,j+1)\ +1\times I(i+1,j-1)\ +1\times I(i+1,j)\ +1\times I(i+1,j+1)]$$

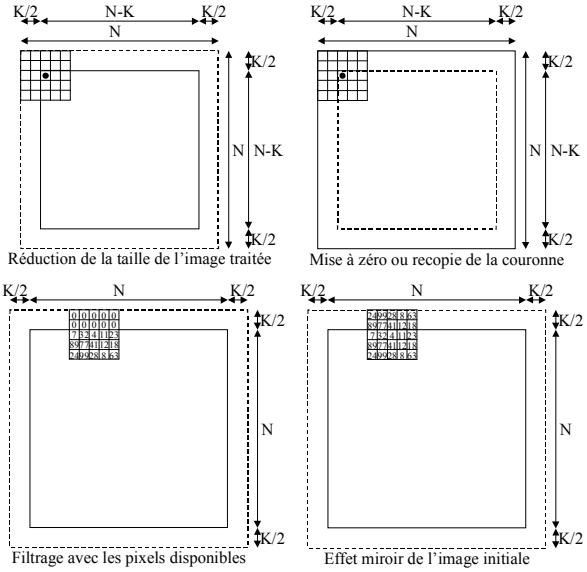
1 – Filtrage linéaire : programmation

```
CImage Filtre (const CImage& Ima, const CFiltre& Fil)
{
  TFIL  pf;
  CImage        Imaf (Ima.Lig(), Ima.Col());

  for (int i=Fil.Lig()/2;i<Ima.Lig()-Fil.Lig()/2;++i)
    for (int j=Fil.Col()/2;j<Ima.Col()-Fil.Col()/2;++j)
    {
        pf = (TFIL) 0;
        for (int p=0;p<Fil.Lig();++p)
          for (int q=0;q<Fil.Col();++q)
            pf += Fil[p*Fil.Col()+q]
                * Ima[(i-Fil.Lig()/2+p)*Ima.Col())+(j-Fil.Col()+q)];
        Imaf[i*Ima.Col()+j] = round (pf / (double) Fil.Cdn);
    }
  return Imaf
}
```

1 – Filtrage linéaire : effets de bord

- Réduction de la taille de l'image traitée
- Mise à zéro des pixels de la couronne
- Recopie des pixels de l'image initiale sur la couronne
- Pour la couronne, filtrage avec les pixels disponibles
- Effet miroir de l'image initiale
- Pas de solution « idéale »



1 – Filtrage linéaire : remarques

Temps de calcul

- Soit un filtre de taille K×K et une image de taille N×N
- Nombre de multiplications $K^2 \times N^2$
- Nombre d'additions $(K^2-1) \times N^2$
- Exemple : N = 256, K = 11 ⇒ ≈7 10^6 multiplications et additions par image + divisions de normalisation

⇒Les filtres doivent être de taille réduite

⇒Les images et les filtres doivent avoir des valeurs entières (calculs plus rapides)

Coefficient de normalisation

- Le choix d'un coefficient de normalisation dépend de la structure du filtre
- Si tous les coefficients sont de même signe :

  C = Somme des coefficients
- Si la somme des coefficients est nulle :

  C = 1
- C doit être choisi pour fournir une dynamique de l'image finale proche de la dynamique maximum

1 – Filtrage linéaire : aspects fréquentiels

On peut considérer l'opération de filtrage comme une convolution :

$$I_F(i,j) = H(k,l) * I(i,j)$$

- $I_F(i,j)$, image filtrée, est la sortie d'un Système Linéaire Invariant
- $H(k,l)$ est la réponse impulsionnelle du filtre, ou fonction de transfert (valeurs des coefficients)
- $I(i,j)$, image initiale, est l'entrée du Système Linéaire Invariant

On a alors (propriétés de la transformée de Fourier) :

$$TF[I_F(i,j)] = TF[H(k,l) * I(i,j)] = TF[H(k,l)] \cdot TF[I(i,j)]$$

- TF[H(k,l)] est la réponse en fréquence de H(k,l). La réponse en fréquence du filtre est la transformée de Fourier de la réponse impulsionnelle, ou de la suite de coefficients
- L'étude de la TF des coefficients caractérise le filtrage réalisé

1 – Filtrage linéaire : propriétés

- Au niveau de l'image
  - Si $I = I_1 + I_2$
    pour $I_F = H * I$,
    on a alors $I_F = I_{F1} + I_{F2}$
    avec $I_{F1} = H * I_1$
    et $I_{F2} = H * I_2$
  - Si $I = \lambda . I_1$
    pour $I_F = H * I$,
    on a alors $I_F = \lambda . I_{F1}$
    avec $I_{F1} = H * I_1$
- Au niveau du filtre
  - Si $H = H_1 + H_2$ et $I_F = H * I$
    on a alors $I_F = I_{F1} + I_{F2}$
    si $I_{F1} = H_1 * I$ et $I_{F2} = H_2 * I$
  - Si $H = \lambda . H_1$ et $I_F = H * I$
    alors $I_F = \lambda . I_{F1}$

1 – Filtrage linéaire : composition de filtres passe-bas

- Remarque :
  - plus la dimension d'un filtre est grande,
  - plus puissants sont ses effets (donc plus l'image filtrée est floue),
  - mais plus le temps de traitement est long
- Suggestion :
  - Décomposer un filtre de grande dimension en une cascade de filtres de dimensions plus petites

- Application :
  - La composition de filtres est réalisée par la convolution des coefficients des filtres :

    Si $I_F = H_2 * I_2$ et $I_2 = H_1 * I_1$

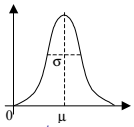
    on a $I_F = (H_1 * H_2) * I_1$
  - Ceci permet d'obtenir des filtres de dimension plus grande à partir de filtres de base
  - La mise en œuvre peut être faite de deux manières :
    - Soit par calcul du filtre global
    - Soit par filtrage successif

1 – Filtrage linéaire : composition de filtres

- Exemple du filtre gaussien 1D

Soit un ensemble de filtres gaussiens G_i :

$$G_i(x,\sigma_i) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{x^2}{2\sigma_i^2}}$$



Convoluons tous ces filtres et calculons leur transformée de Fourier :

$$\Im\left[\prod_{i=0}^{N_x-1} G_i(x,\sigma_i)\right] = \prod_{i=0}^{N_x-1} \Im[G_i(x,\sigma_i)]$$

où $\prod_{i=0}^{N-1} f_i = f_0 * f_1 * \cdots * f_{N-1}$

$$\Im\left[\prod_{i=0}^{N_x-1} G_i(x,\sigma_i)\right] = \prod_{i=0}^{N_x-1} e^{-\frac{(2\cdot\pi\cdot f)^2\sigma_i^2}{2}}$$

$$\Im\left[\prod_{i=0}^{N_x-1} G_i(x,\sigma_i)\right] = e^{-\frac{(2\cdot\pi\cdot f)^2\sum_{i=0}^{N-1}\sigma_i^2}{2}}$$

$$\prod_{i=0}^{N_x-1} G_i(x,\sigma_i) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} = G(x,\sigma)$$

avec $\sigma^2 = \sum_{i=0}^{N-1}\sigma_i^2$

Si $\sigma_i = \sigma_0$, $\sigma = \sigma_0.\sqrt{N}$, où N est le nombre de filtres cascadés

σ fixe K : si $\sigma\searrow$, $K\searrow$, $\Rightarrow$ temps de calcul$\searrow$

(4 $\sigma \rightarrow$ 95%, 6 $\sigma \rightarrow$ 99%)

1 – Filtrage linéaire : composition de filtres

Composition de filtres de même forme

- Exemple de calcul de composition de deux filtres 1D à 3 coefficients

```
      1 1 1
*     1 1 1
-------------
      1 1 1
    1 1 1
  1 1 1
-------------
  1 2 3 2 1
```

C'est une convolution en considérant les filtres bordés par des zéros

On réalise une multiplication coefficient à coefficient puis un décalage

Les coefficients intermédiaires de même position sont additionnés

1 – Filtrage linéaire : composition de filtres

- Exemple de calcul de composition de deux filtres 2D à 3×3 coefficients

$$\begin{matrix}1&1&1\\1&1&1\\1&1&1\end{matrix} * \begin{matrix}1&1&1\\1&1&1\\1&1&1\end{matrix} \rightarrow \begin{matrix}1&2&3&2&1\\2&4&6&4&2\\3&6&9&6&3\\2&4&6&4&2\\1&2&3&2&1\end{matrix}$$

  - La même opération qu'en 1D est effectuée, mais en réalisant un décalage dans deux directions
  - La somme des coefficients est faite pour chaque position
  - Gain en temps de calcul

    Ex : filtre 5×5 issu de la composition de deux filtres 3×3
    - Filtre 5×5 : 25 opérations de base sur chaque pixel de l'image
    - Deux filtres 3×3 : $2 \times 9 = 18$ opérations de base sur chaque pixel

1 – Filtrage linéaire : composition de filtres

Composition des filtres de formes différentes

- La composition de deux filtres 1D respectivement horizontal et vertical permet de construire un filtre 2D

$$\begin{matrix}0&1&0\\0&0&0\\0&1&0\end{matrix} * \begin{matrix}0&0&0\\1&1&1\\0&0&0\end{matrix} \rightarrow \begin{matrix}1&1&1\\0&0&0\\1&1&1\end{matrix}$$

  - Cette composition permet de réaliser très simplement des filtres 2D aux comportements horizontal et vertical différents
  - Gain en temps de calcul

    Ex : filtre 5×5 issu de la composition de deux filtres 1×5
    - Filtre 5×5 : 25 opérations de base sur chaque pixel de l'image
    - Deux filtres 1×5 et 5×1 : $2 \times 5 = 10$ opérations de base par pixel

1 – Filtrage linéaire : filtres à noyau séparable

- Démonstration à partir de l'exemple du filtre gaussien $G(x,y,\sigma)$:

$$I_F(x,y) = I(x,y) * G(x,y,\sigma)$$

$$\Im[I_F(x,y)] = \Im[I(x,y) * G(x,y,\sigma)] = \Im[I(x,y)] \cdot \Im[G(x,y,\sigma)]$$

$$\Im[I_F(x,y)] = \Im[I(x,y)] \cdot \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} G(x,y,\sigma) \cdot e^{-i(2\pi f_x x + 2\pi f_y y)} \cdot dx \cdot dy$$

$$\Im[I_F(x,y)] = \Im[I(x,y)] \cdot \int_{-\infty}^{+\infty} G(x,\sigma) \cdot e^{-i2\pi f_x x} \cdot dx \cdot \int_{-\infty}^{+\infty} G(y,\sigma) \cdot e^{-i2\pi f_y y} \cdot dy$$

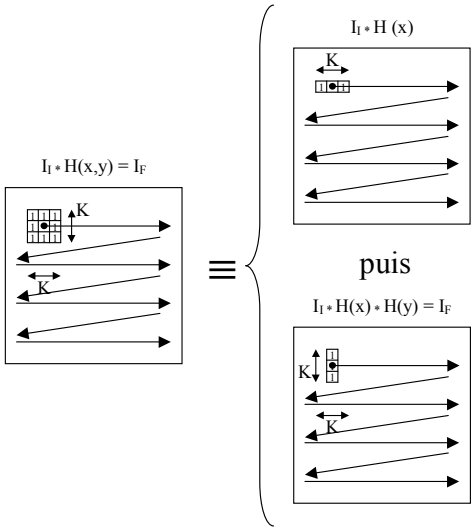
$$\Im[I_F(x,y)] = \Im[I(x,y)] \cdot \Im[G(x,\sigma)] \cdot \Im[G(y,\sigma)]$$

$$I_F(x,y) = I(x,y) * G(x,\sigma) * G(y,\sigma)$$

- La fonction $G(x,y,\sigma)$ est à noyau séparable. Le filtrage est donc réalisable soit par convolution avec un filtre $K \times K$ (K^2 opérations), soit par convolution avec deux filtres K ($2 \times K$ opérations)

1 – Filtrage linéaire : filtres à noyau séparable

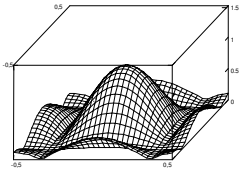
- La convolution d'une image avec un filtre 2D à noyau séparable est équivalente à la convolution avec deux filtres 1D
- Comparaison des temps de calcul
  - Filtre 2D :
    - $N^2 \times K^2$ multiplications
    - $N^2 \times (K^2 - 1)$ additions
  - Succession de 2 filtres 1D :
    - $N^2 \times 2 \times K$ multiplications
    - $N^2 \times 2 \times (K - 1)$ additions



2 – Filtrage passe-bas

- Les images sont souvent altérées par du bruit impulsionnel, brusque variation d'un pixel par rapport à ses voisins
- Le filtrage passe-bas, en réduisant les hautes fréquences, limite ces brusques variations
- Filtres rectangulaires
  - Inconvénient : phase non nulle
  - Plus le filtre a de coefficients, plus le filtrage est important, mais plus l'image filtrée perd de sa netteté (suppression des hautes fréquences, et donc des brusques transitions)

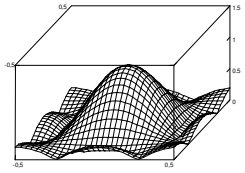
$$H_1 = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



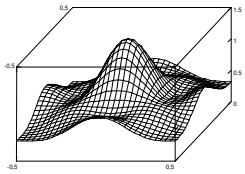
2 – Filtrage passe-bas

- Filtres à phase nulle
  - H_2 donne plus de poids au pixel central, ce qui le rend moins actif que H_1
  - H_3 privilégie les directions x et y
- Synthèse de filtre
  - Remez, Fourier, … à partir de gabarits
  - Inconvénient : souvent beaucoup de coefficients

$$H_2 = \frac{1}{10}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



$$H_3 = \frac{1}{42}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 4 & 1 & 1 \\ 1 & 4 & 6 & 4 & 1 \\ 1 & 1 & 4 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$



2 – Filtrage passe-bas

- Réponses fréquentielles connues
  - Hanning

$$H(i,j) = 0{,}5 + 0{,}5 \cdot \cos\left(\frac{2\pi i}{K}\right) \cdot \cos\left(\frac{2\pi j}{K}\right)$$

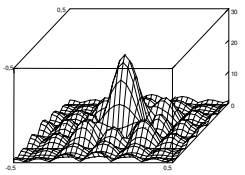
  - Hamming $\quad (-K/2 < i, j < +K/2)$

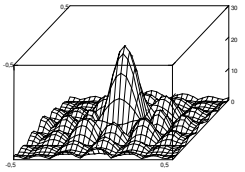
$$H(i,j) = 0{,}54 + 0{,}46 \cdot \cos\left(\frac{2\pi i}{K}\right) \cdot \cos\left(\frac{2\pi j}{K}\right)$$

  - Butterworth d'ordre n

(D_0 : fréquence de coupure)

$$H(u,v) = \frac{1}{1 + \left[\sqrt{u^2 + v^2}/D_0\right]^{2n}}$$



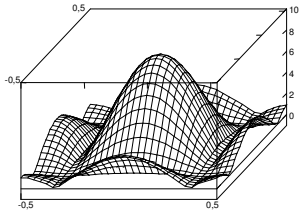


2 – Filtrage passe-bas

Exemples de filtres passe-bas

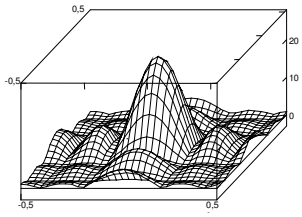
1	1	1
1	1	1
1	1	1

Filtre moyenneur 3 × 3
C=1/9



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

Filtre moyenneur 5 × 5
C=1/25

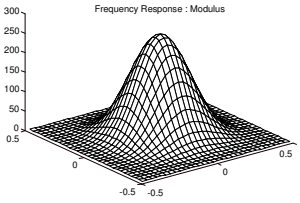


2 – Filtrage passe-bas : filtre Gaussien

- Un seul lobe dans le domaine temporel
- Un seul lobe dans le domaine fréquentiel
- Symétrie relativement à la rotation
- Largeur paramétrée par un seul paramètre σ
- Filtre séparable (1 filtrage 2D est équivalent à deux filtrages 1D dans des directions opposées) ⇒ gain énorme sur le temps de calcul

$$1D : G(i) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{i^2}{2\sigma^2}}$$

$$2D : G(i,j) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(i^2+j^2)}{2\sigma^2}}$$



1	2	3	2	1		
2	4	6	4	2		
3	6	7	6	3		
2	4	6	4	2		
1	2	3	2	1		

Filtre Gaussien 5 × 5
C=1/79

1	4	7	10	7	4	1
4	12	26	33	26	12	4
7	26	55	71	55	26	7
10	33	71	91	71	33	10
7	26	55	71	55	26	7
4	12	26	33	26	12	4
1	4	7	10	7	4	1

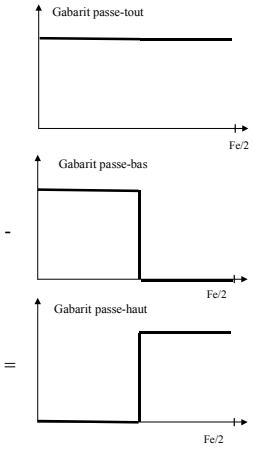
Filtre Gaussien 7 × 7
C=1/1181

3 – Autres gabarits : filtres passe-haut

- Conservent les hautes fréquences d'une image
⇒ Détection des transitions dans l'image
⇒ Extraction de contours des objets d'une scène
- TRÈS SENSIBLES AU BRUIT
- Coeff. passe-haut = 1 – Coeff. Passe-bas (1 étant le filtre passe-tout avec le même gain que le passe-bas)
- Exemple de calcul :

$$[0 \quad 0 \quad 1 \quad 0 \quad 0]\times 16 - [1 \quad 4 \quad 6 \quad 4 \quad 1] \rightarrow [-1 \quad -4 \quad 10 \quad -4 \quad -1]$$

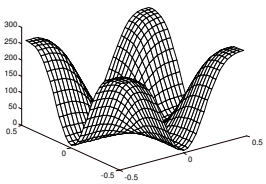
Passe-haut = passe-tout - passe-bas



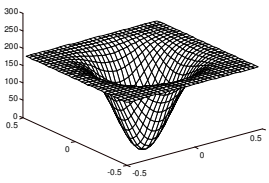
3 – Autres gabarits : filtres passe-haut

- La somme des coefficients des filtres passe haut est nulle (annulation de la courbe de réponse en fréquence à la fréquence 0)
- La mise en œuvre de filtres passe-haut 2D peut s'envisager par la composition de deux filtres 1D. Mais à cause de l'annulation aux fréquences 0 en f_x et f_y, on n'obtient pas une symétrie de révolution
- Pour obtenir cette symétrie, il faut réaliser des opérations d'addition / soustraction sur le filtre 2D
- Exemple : Filtre [1 4 6 4 1] composé en 2D puis soustraction de 220 au point central

$$H_1 = \begin{bmatrix} 1 & 4 & -10 & 4 & 1 \\ 4 & 16 & -40 & 16 & 4 \\ -10 & -40 & 100 & -40 & -10 \\ 4 & 16 & -40 & 16 & 4 \\ 1 & 4 & -10 & 4 & 1 \end{bmatrix}$$



$$H_2 = \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 24 & 36 & 24 & 4 \\ 6 & 36 & -184 & 36 & 6 \\ 4 & 24 & 36 & 24 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$



3 – Filtres High-boost

- Un filtrage passe-haut peut être obtenu en soustrayant d'une image originale cette image filtrée passe bas :

  Passe-haut = Originale – Passe-bas

- Les filtres high-boost (ou high-frequency-emphasis) sont obtenus en multipliant l'image originale par un coefficient :

  High-boost = A * Originale – Passe-bas

  High-boost = (A–1) * Originale + Passe-haut
  - Si A = 1 : filtrage passe-haut standard
  - Si A > 1 : une partie de l'image originale est ajoutée à l'image filtrée passe-haut. A contrôle le renforcement de contour.
- Application : publication et impression

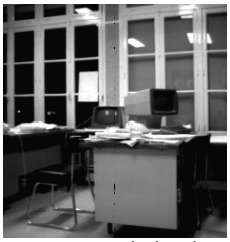
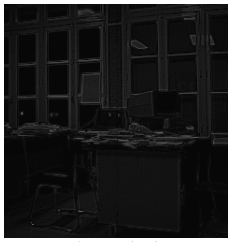
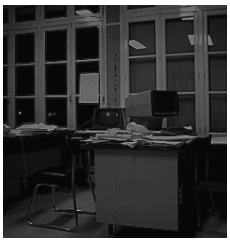
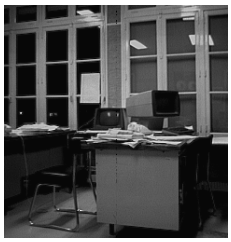


Image originale


A = 1,1

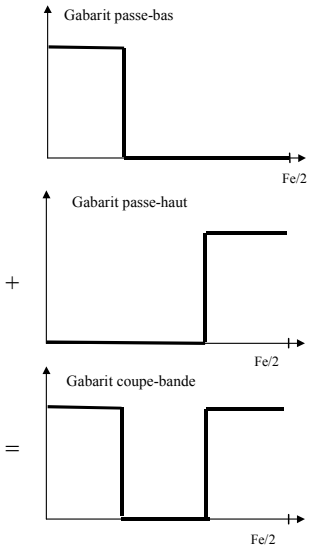

A = 1,4


A = 1,7

3 – Autres gabarits : filtres coupe-bandes

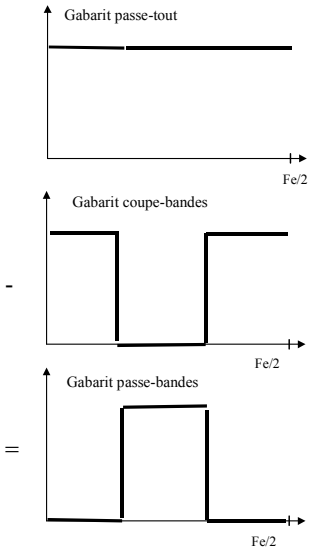
- Les filtres coupe-bandes peuvent être obtenus en composant filtres passe-bas et passe-haut :

  Coupe bande = passe-bas + passe-haut
- Attention : la fréquence de coupure du passe-bas doit être inférieure à celle du passe-haut
- La normalisation doit toujours s'effectuer relativement à la valeur du module du filtre dans la bande passante
- Pour obtenir de vrais coupes-bande, il faut équilibrer les gains du passe-bas et du passe-haut ⇒ les deux filtres doivent avoir la même somme de la valeur absolue de leurs coefficients
- Pour avoir une atténuation maximum (transmittance nulle) dans la bande coupée, il faut soustraire une composante continue en fréquence pour compenser le gain résiduel dû à l'addition des deux filtres



3 – Autres gabarits : filtres passe-bandes

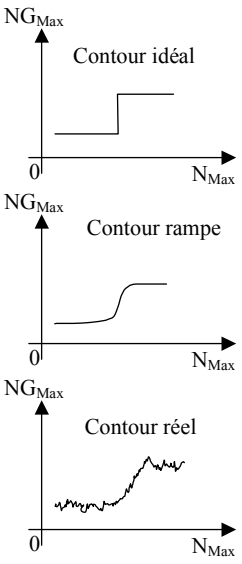
- Les filtres passe-bandes peuvent être obtenus de deux manières :
  - en composant filtres passe-bas et passe-haut
    Passe bande = 1 – [passe-bas + passe-haut]
  - à partir de filtres coupe-bandes
    Passe bande = 1 – coupe-bande
- Cette méthode de synthèse (comme la précédente) présente certaines limites, dues en particulier aux formes que l'on peut obtenir à partir des passe-bas et passe-haut, et à la troncature des coefficients
- Des méthodes de synthèse classiques permettent d'obtenir des résultats plus efficaces, surtout du point de vue de la symétrie de la courbe de réponse en fréquences (voir cours sur les traitements globaux)



Traitements locaux 27

4 – Opérateurs dérivées

- Un contour est caractérisé par une brusque variation des caractéristiques locales de l'image (valeur moyenne de niveau de gris, variance locale, etc. ⇒ limite de région)
- Un domaine important du traitement d'image consiste à séparer automatiquement les images en régions de même propriétés, caractérisées par leur frontière. On parle de segmentation (découpage) par extraction de contour
- La détection de contours consiste à faire apparaître les contours des objets d'une image. Elle permet de passer d'une image à niveaux de gris à une image de contours (rendue binaire après seuillage)
- Le suivi de contours permet de relier ensemble des points de contours pour obtenir des contours fins et continus.



Traitements locaux 28

4 – Opérateurs dérivées

- Beaucoup de méthodes pour obtenir des contours sont basées sur des mesures de dérivées
- Celles-ci permettent de mettre en évidence les brusques variations de niveaux de gris
- Pour réduire le nombre de coefficients, on privilégie une approche intuitive basée sur le postulat :

  Dérivée = différenciation = différence
- L'opération de dérivée peut alors s'écrire (en 1D) :

  $F'(i) = F(i) - F(i-1)$
- On en déduit la forme d'un filtre dérivateur :

-1	1

4 – Opérateurs dérivées

- Formule de Taylor

$$f(a) = f(x) + (a-x)f'(x) + \frac{(a-x)^2}{2!}f''(x) + \frac{(a-x)^3}{3!}f'''(x) + \cdots$$

- En posant a = x+h et a = x–h

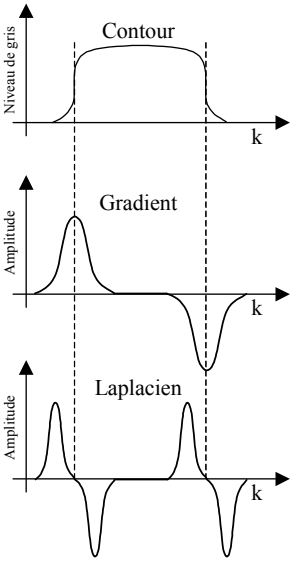
$$f(x+h) = f(x) + h f'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f'''(x) + \cdots \qquad (1)$$

$$f(x-h) = f(x) - h f'(x) + \frac{h^2}{2!}f''(x) - \frac{h^3}{3!}f'''(x) + \cdots \qquad (2)$$

- (1) $\Rightarrow f'(x) \approx \frac{-f(x)+f(x+h)}{h}$
- (1) – (2) $\Rightarrow f'(x) \approx \frac{-f(x-h)+f(x+h)}{2h}$
- (1) + (2) $\Rightarrow f''(x) \approx \frac{f(x-h)-2f(x)+f(x+h)}{h^2}$

4 – Opérateurs dérivées

- Dérivation en présence d'un contour :
  - En dérivant une fois, on obtient un gradient
  - En dérivant deux fois on obtient un Laplacien
- Les opérateurs dérivées sont orientés (direction de la différenciation)



H_x | -1 | 1 |

H_y | -1 |
| 1 |

Roberts, 1965

4 – Opérateurs dérivées filtrées 1D

- L'accentuation des hautes fréquences apportée par la différentiation introduit une amplification du bruit au détriment de l'information utile
- On préfère alors utiliser des versions dites « dérivée filtrée » pour réaliser une détection de contours
- Le filtre H_1=[-1,0,1] en est la version la plus simple
- Plus généralement, on obtient ces filtres par convolution du filtre dérivée par un filtre lisseur :

[-1,1] * [filtre lisseur]

- Ex : [-1,1] * [1,2,1] → [-1,-1,1,1]

4 – Opérateurs dérivées filtrées 1D

- Pour conserver un nombre de coefficients impair on choisit : [-1,1] * [1,3,3,1] → [-1,-2,0,2,1] qui peut aussi être obtenu par : [-1,0,1] * [1,2,1]
- Ces filtres dérivateurs sont antisymétriques, la somme de leurs coefficients est donc nulle, et le coefficient central (filtres à nombre de coefficients impair) est nul
- Ces filtres peuvent être considérés comme des passe-hauts ou passe-bandes, mais ils s'en distinguent par leur phase égale à π/2 qui impose l'antisymétrie des coefficients

4 – Opérateurs dérivées filtrées 2D

- En 2D, la dérivée donne un résultat vectoriel : le gradient. On dispose ainsi de deux informations : l 'amplitude et la direction
- On détermine les dérivées horizontale $A_x(i,j)$ et verticale $A_y(i,j)$, puis on déduit l'amplitude A(i,j) et la direction D(i,j) du gradient en chaque point (i,j) de l'image par :

$$A(i,j) = \sqrt{A_x(i,j)^2 + A_y(i,j)^2} \qquad D(i,j) = \arctan\left(\frac{A_x(i,j)}{A_y(i,j)}\right)$$

- Pour réduire le temps de calcul, on approche souvent l 'amplitude A(i,j) du gradient par :

$$A(i,j) \approx |A_x(i,j)| + |A_y(i,j)| \approx Max(A_x(i,j), A_y(i,j))$$

4 – Opérateurs dérivées filtrées 2D

- Le résultat d'une dérivée 2D se présente couramment sous la forme de deux images : la projection du gradient sur l'axe X et la projection sur l'axe Y (une représentation module angle peut être aussi très utile)
- Comme en 1D, les masques couramment utilisés en 2D réalisent une dérivée lissée plutôt qu'une dérivée simple
- Les plus utilisés sont les masques de Sobel ou Prewit

H_x

-1	0	1
-1	0	1
-1	0	1

H_y

-1	-1	-1
0	0	0
1	1	1

Sobel

H_x

-1	0	1
-2	0	2
-1	0	1

H_y

-1	-2	-1
0	0	0
1	2	1

Prewitt

4 – Opérateurs dérivées filtrées 2D

- Orientation des opérateurs dérivée
  - pour pallier l'inconvénient de l'orientation des opérateurs dérivée, Kirsh a développé des masques optimaux, en associant un masque dans chaque direction détectable dans le voisinage d'un point
  - Ces opérateurs de Kirsh ne sont pas à noyau séparable

$$H_1 = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \quad H_2 = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$$

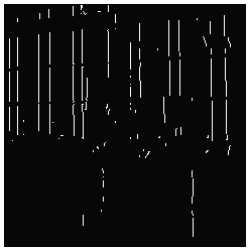
$$H_3 = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \quad H_4 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$$

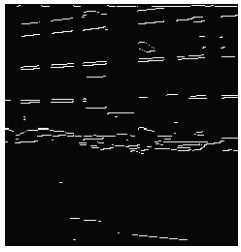
$$H_5 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \quad H_6 = \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}$$

$$H_7 = \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} \quad H_8 = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

4 – Opérateurs dérivées : exemple


Image originale

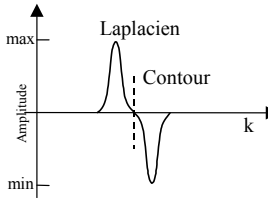

Gradient en X


Gradient en Y


Amplitude du gradient

4 – Opérateurs dérivées : Laplacien

- Laplacien = dérivée seconde
- Laplacien = scalaire ⇒ résultat indépendant de la direction des contours
- Opérateurs très sensibles au bruit
- Points de contours = valeurs nulles bordées de deux extrema



- Exemples de filtres Laplacien :

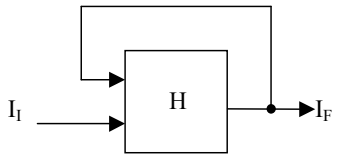
$$H_1 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$H_2 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$H_3 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

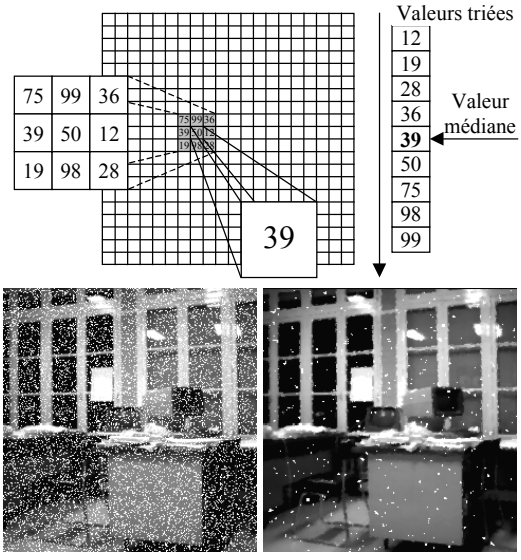
5 – Filtres linéaires récursifs

- Principe
  - $I_F = H * (I_I + I_F)$
- Avantages
  - Nombre réduit de coefficients comparativement à l'efficacité du filtrage ⇒ réduction du nombre de calculs
  - Réponse impulsionnelle infinie RII ⇒ équivalence avec des filtres non récursifs mais avec un nombre de coefficients infini
  - Utilisation pour certains filtres dits « optimaux »
- Mise en œuvre
  - Utilisation des résultats des calculs précédents pour déterminer les valeurs des nouveaux échantillons filtrés
- Application
  - Intégration : FI(i) = FI(i-1) – X(i). Attention : si la valeur moyenne de la fonction est ≠ 0, le résultat croit indéfiniment ⇒ risque d'instabilité



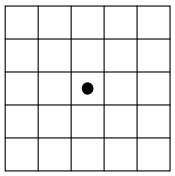
6 – Filtrage non linéaire : filtrage médian

- Ce filtrage s'oppose au filtrage linéaire car il n'est pas le résultat d'une combinaison linéaire de pixels
- Les pixels du masque sont classés par ordre croissant, puis la valeur médiane est affectée au pixel courant
- Plus le masque est grand, plus le filtrage paraît efficace, mais plus il déforme l'image
- Efficace sur du bruit impulsionnel

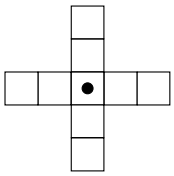


6 – Filtrage non linéaire : filtrage médian

- Le filtrage médian permet de supprimer des points qui sont illogiques dans une progression de niveaux de gris (pics parasites)
- Ce filtrage est utilisé pour améliorer la visualisation des images
- On peut mettre en place une implémentation récursive pour les filtres médians (plus efficace dans le cas d'une série de points alternés)
- Une pondération des points peut être effectuée avant le classement, (conservation des sommets larges)
- Les paramètres des filtres médians sont la taille du voisinage et le nombre de passages effectués sur l'image
- Exemple de masques utilisés pour le filtrage médian :



Masque carré



Masque en croix (moins de perturbations que le masque en carré)

7 – Méthodes locales séquentielles

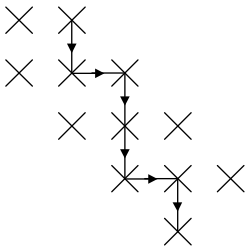
- Les méthodes locales séquentielles réalisent des filtrages sur des zones de traitement limitées à un voisinage d'intérêt dans l'image (l'image n'est pas traitée systématiquement sur toute son étendue)
- Exemple de l'extraction de contour
  - Il est intéressant de ne traiter que le voisinage du contour
  - S'il n'existe aucune connaissance *a priori*, sur la position du contour, la zone de traitement (i. e. la position du filtre appliqué) évolue en fonction des points déjà traités ⇒ suivi de contour
  - L'algorithme fonctionne en trois phases
    - Une phase d'initialisation : pour trouver un point de départ du contour
    - Une phase de suivi de contour
    - Une phase d'arrêt

7 – Méthodes locales séquentielles

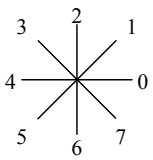
- Phase d'initialisation
  - Balayage systématique de l'image jusqu'à obtenir un résultat qui caractérise un point de départ
- Phase de suivi
  - Recherche dans le voisinage du point courant le point de contour suivant. Pour cela, il faut choisir deux paramètres :
    - Critère de choix du point suivant. Ex : valeur du gradient, ou direction du gradient, ...
    - Limitation de la zone de recherche. Ex : concavité maximum du contour, ou orientation globale connue, ...
- Phase d'arrêt
  - Détection d'un bouclage (passage deux fois sur le même point), ou arrivée sur un bord

7 – Méthodes locales séquentielles

- Avantages
  - rapidité (en N alors que la détection est en N^2)
  - unicité du contour
  - connexité du contour
  - informations *a priori* incluses dans la méthode



- Inconvénients
  - résultats aléatoires : bouclage, perte du contour, ...
  - problème du démarrage
  - irrégularité de l'algorithme
- Les contours sont codés suivant le codage de Freeman
  - Codage des directions
    - Exemple : à quel contour correspond le code 1 0 1 0 7 0 7 4 4 4 4 4 4 4 ?



Codage des directions

Polytech' Orléans

Morphologie mathématique

Traitement des images
Polytech'Orléans, 2005 – 2006
Christophe LÉGER

École polytechnique de l'université d'Orléans

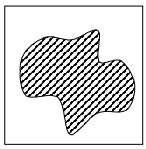
UNIVERSITÉ D'ORLÉANS

Morphologie mathématique

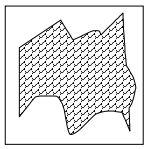
- La morphologie mathématique consiste à comparer les objets d'une image relativement à un autre objet de forme connue : l'élément structurant
- En quelque sorte, chaque élément structurant fait apparaître l'objet sous un aspect différent
- Fondamentalement, la morphologie mathématique utilise des opérations ensemblistes pour transformer les images

1.1 – Transformations ensemblistes

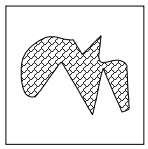
- Soient deux ensembles X et Y, on dispose des opérations classiques suivantes :



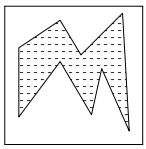
Ensemble X



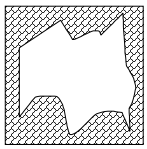
Union $X \cup Y$



Intersection $X \cap Y$



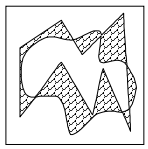
Ensemble Y



Complémentation

$((X \cup Y)^C)_Z = (X \cup Y)^C \cap Z$

$(x \in (X \cup Y)^C \Leftrightarrow x \notin (X \cup Y))$

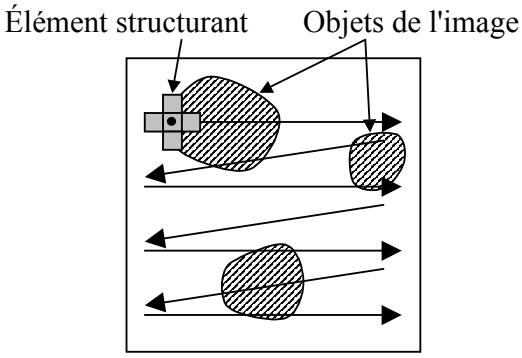


Différence symétrique

$X / Y = X \cup Y - X \cap Y$

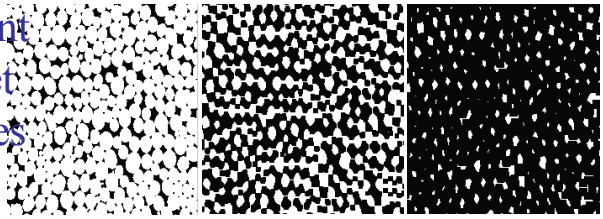
1.2 – Transformations en tout ou rien

- Soit un élément B de géométrie connue (muni d'une origine) appelé élément structurant
- B est déplacé de manière à ce que son origine passe par tous les pixels de l'image
- Pour chaque pixel, on pose une question relative à l'union, l'intersection, ou l'inclusion de B
- La réponse est positive ou négative, d'où le nom de transformation en tout ou rien



1.2 – Transformations en tout ou rien

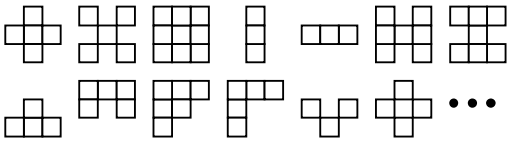
- La morphologie mathématique consiste à comparer les objets de l'image que l'on souhaite analyser à un élément structurant
- Chaque élément structurant fait alors apparaître l'objet sous des formes différentes
- L'ensemble des points correspondant à des réponses positives ou négatives forme un nouvel ensemble qui constitue l'image transformée



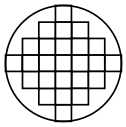
1.3 – Choix d'un élément structurant

- Critères de sélection d'un élément structurant :
  - forme, direction, taille (⇒ efficacité)

  Tous ces critères dépendent des objets à analyser
- Exemples :



- Problèmes
  - géométrie de l'échantillonnage (quantification induite par la distance inter-pixel)



  - Choix d'une origine

1.4 – Caractéristiques des transformations tout ou rien

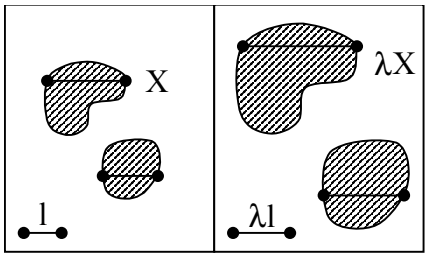
- Invariance par translation
  - Soit X_k le translaté de X selon le vecteur k
  - T est invariante par translation si et seulement si :

$$T(X_k) = [T(X)]_k$$

- Compatibilité avec les homothéties
  - Soit λX un ensemble homothétique de X et T_λ une transformation dépendant de λ
  - T est compatible avec les homothéties si et seulement si :

$$T_\lambda(\lambda X) = \lambda\, T(X)$$

- Ex : extraction des cordes d'un ensemble X. Il est équivalent d'extraire les cordes d'unité sur X, ou d'extraire les cordes d'unité λ sur λX en les réduisant d'un rapport 1/ λ

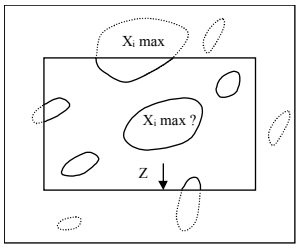


1.4 – Caractéristiques des transformations tout ou rien

- Connaissance locale uniforme
  - Le critère de connaissance locale uniforme est utilisé lorsque le champ de mesure contient une partie de l'ensemble X (structures observées à travers le champ d'un oculaire)
  - Soit Z l'ensemble définissant le champ de mesure où X est connu (on a $X \cap Z$). T vérifie le principe de connaissance locale unifor-me si, quel que soit l'ensemble Z' borné à l'intérieur duquel X est connu, il existe un ensemble Z, dépendant de Z' et de T mais pas de X, à l'intérieur duquel on connaît T(X) de telle sorte que l'on ait :

$$T(X \cap Z) \cap Z' = T(X) \cap Z'$$

- Parmi toutes les transformations, cette condition de connaissance locale uniforme élimine toutes les transformations concernant des maxima de X (longueur, surface, etc.)



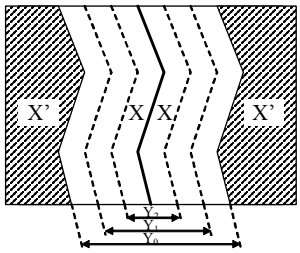
Exemple du non respect de la connaissance locale uniforme

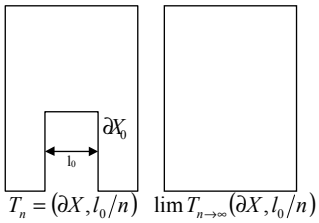
1.4 – Caractéristiques des transformations tout ou rien

- Continuité monotone
  - Soient deux ensembles X et X'. Une transformation est dite continue lorsque, en se rapprochant de plus en plus de X, le transformé de X' (T(X')) se rapproche également de plus en plus du transformé de X (T(X)):

$$X' \to X \Leftrightarrow T(X') \to T(X)$$

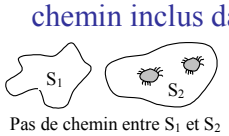
  - Ex 1 : X et X' représentent des grains idéaux et expérimentaux ($X' \subset X$). Soit Y le joint entre ces grains $Y = (X_{exp} \cup X'_{exp})^C$ et T la transf. qui réduit l'épaisseur de Y. T est définie par $Y_i = T(Y_{i-1})$. On a $Y_i \subset Y_{i-1}$. En ré-appliquant T à Y_i on a $Y_{i+1} = T(Y_i)$, et par conséquent $T(Y_i) \subset T(Y_{i-1})$. T est croissante.
  - Ex 2 : La concavité de X a une épaisseur l_0. Soit une transformation T(∂X,n) telle que l'épaisseur de la concavité devienne $l = l_0/n$. Lorsque $n \to \infty$, la concavité disparaît. T n'est donc pas continue.



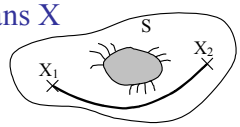


1.5 – Rappels de topologie

- Connexité
  - On dit qu'un ensemble X est connexe si, à toute paire de points X_1 et X_2 appartenant à X, on peut faire correspondre au moins un chemin inclus dans X

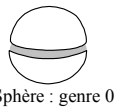


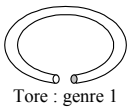
Pas de chemin entre S_1 et S_2



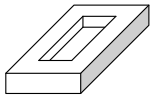
Chemin X_1 X_2 sur S

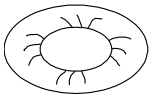
- Genre d'une surface
  - On appelle genre d'une surface g(s) le nombre maximal de coupures que l'on peut faire dans une surface de telle façon que la surface obtenue soit encore connexe


Sphère : genre 0


Tore : genre 1

- Surfaces homéomorphes
  - Deux surfaces S_1 et S_2 sont homéomorphes si, en déformant de manière continue l'une des surfaces, on peut tendre vers la forme de la seconde surface





1.5 – Rappels

- Nombre de connexité $N_c(x)$

Déf1 : $N_c(X) = \sum_{i=0}^{N-1} [1 - g_i(S_i)]$

  - N : nombre total de surfaces S_i délimitant l'ensemble X
  - g_i : genre des surfaces S_i

Déf2 : $N_c(X) = n - h$

  - n : nombre de sous-ensembles connexes
  - h : nombre de « trous » dans ces sous-ensembles

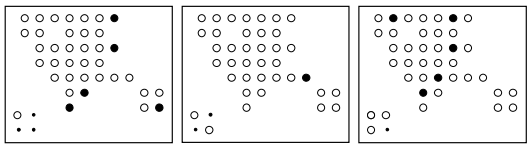
Ex : Sphère : g=0, N_n=1
Torre : g=1, N_n = 0

- Estimation du nombre de connexité des images numériques
  - Pour un graphe hexagonal :

$$N_c(X) = N\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} - N\begin{pmatrix} \bullet & 1 \\ 1 & 0 \end{pmatrix}$$

  - Pour un graphe carré :

$$N_c(X) = N\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + N\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - N\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$



Avec ●=1 et ○=0, $N_c(X) = 5 + 1 - 5 = 1$

1.5 – Propriétés des transformations tout ou rien

- Propriétés algébriques
  - T est une transformation croissante si et seulement si :

$$X \subset Y \Rightarrow T(X) \subset T(Y)$$

  - T est une transformation extensive si et seulement si :

$$X \subset T(X)$$

Inversement, T est anti-extensive si et seulement si :

$$T(X) \subset X$$

  - T est une transformation idempotente si et seulement si :

$$T\,[T(X)] = T(X)$$

Ex : Tamisage d'une poudre

- Propriétés topologiques
  - Transformation homotopique : une transformation est dite homotopique si elle ne modifie pas le nombre de connexité. On a alors :

$$N_c[T(X)] = N_c[X]$$

  - Préservation du nombre de connexité : une transformation préserve la connexité si, X étant connexe, T(X) est connexe

2.1 – Érosion sur images binaires

- Soient un ensemble X et un élément structurant B_x centré en x. B_x est déplacé de telle sorte que son centre occupe successivement toutes les positions x de l'espace.
- On note $E^B(X)$ l'ensemble Y érodé de X tel que :

  $E^B(X) = Y = \{ x : B_x \subset X\}$
- On note aussi :

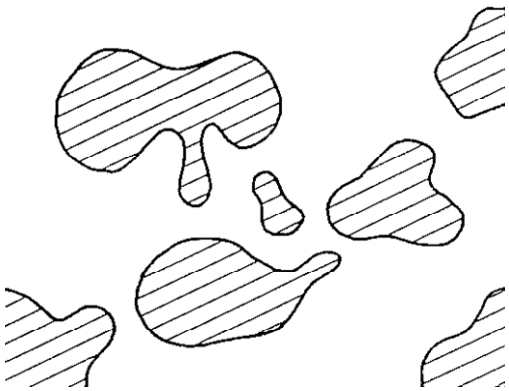
  $Y = X \ominus B^T$

  où B^T est le transposé de B, c'est à dire son symétrique par rapport à son origine

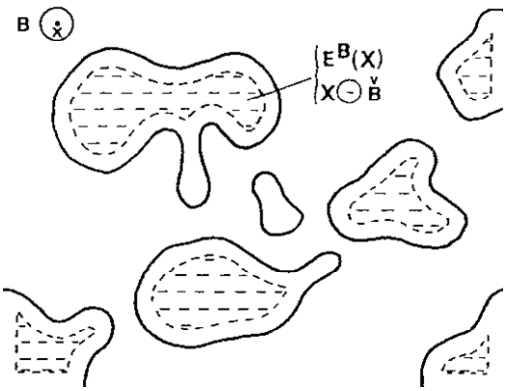
- Algorithme

  Pour tous les points de l'image $I_I(i,j)$
  Calculer le ET des voisins dans B
  Si (ET=1) alors image $I_F(i,j)=I_I(i,j)$
  sinon image $I_F(i,j)=0$
  fin Si
  fin Pour
- Applications
  - Comptage d'éléments (cellules) : érodé ultime

2.1 – Exemple d'érosion



Ensemble d'objets X



Erosion de X par un élément structurant circulaire

2.1 – Érosion sur images à niveaux de gris

- L'érosion d'images à niveaux de gris est définie de la manière suivante :

$E^B(f(x)) = f_I(x) = \inf \{ f(x) : x \in B_x \}$

- Ainsi, pour construire la fonction érodée par un élément structurant plan, il suffit d'attribuer à chaque point du domaine B_x la valeur inférieure que prend f(x) dans ce domaine
- On note aussi :

$$f_I(x) = f(x) \ominus B^T$$

où, comme précédemment, B^T est le transposé de B

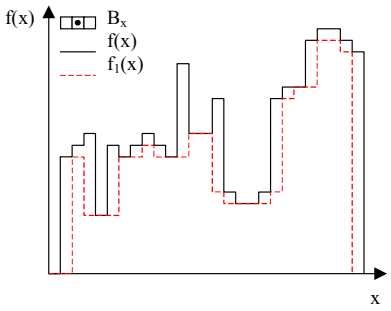
- Algorithme

Pour tous les points de l'image $I_I(i,j)$

Rechercher le minimum dans le voisinage de X centré en (i,j)

Affecter cette valeur à image $I_F(i,j)$

fin Pour



2.2 – Dilatation sur images binaires

- Soient un ensemble X et un élément structurant B_x centré en x. B_x est déplacé de telle sorte que son centre occupe successivement toutes les positions x de l'espace.
- De la même manière que pour l'érosion, on note $D^B(X)$ l'ensemble Y dilaté de X tel que :

$D^B(X) = Y = \{ x : B_x \cap X \neq \varnothing \}$

- On note aussi :

$$Y = X \oplus B^T$$

où B^T est le transposé de B, ou son symétrique par rapport à son origine

- Algorithme

Pour tous les points de l'image $I_I(i,j)$

Calculer le OU des voisins dans B

Si (OU=1)alors image $I_F(i,j)=I_I(i,j)$

sinon image $I_F(i,j)=0$

fin Si

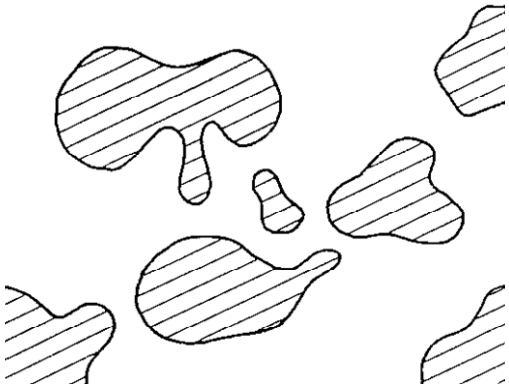
fin Pour

- Remarque
  - Erosion et dilatation ne sont pas indépendantes :

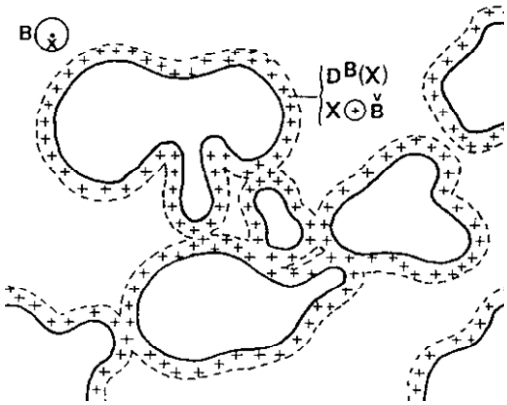
$X \oplus B^T = (X^C \ominus B_T)^C$

- Applications

2.2 – Exemple de dilatation



Ensemble d'objets X



Dilatation de X par un élément structurant circulaire

2.2 – Dilatation sur images à niveaux de gris

- La dilatation d'image à niveaux de gris est définie de la manière suivante :

$D^B(f(x)) = f_1(x) = \sup \{ f(x) : x \in B_x \}$

- Ainsi, pour construire la fonction dilatée par un élément structurant plan, il suffit d'attribuer à chaque point du domaine B_x la valeur supérieure que prend f(x) dans ce domaine
- On note aussi :

$$f_1(x) = f(x) \oplus B^T$$

où, comme précédemment, B^T est le transposé de B

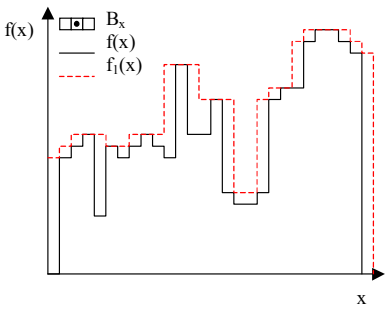
- Algorithme

Pour tous les points de l'image $I_I(i,j)$

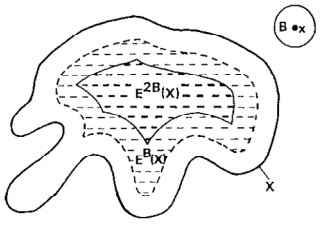
Rechercher le maximum dans le voisinage de X centré en (i,j)

Affecter cette valeur à image $I_F(i,j)$

fin Pour

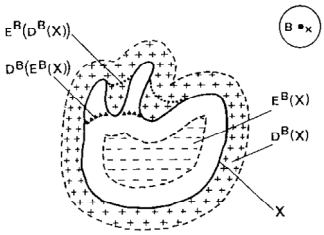


2.3 – Propriétés de l'érosion et de la dilatation

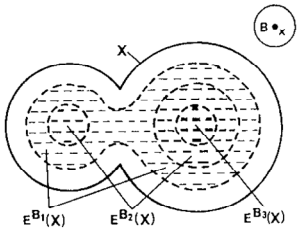


Propriété d'inclusion et d'itérativité de l'inclusion pour l'érosion :

$B \subset B' \Rightarrow E^{B'}(X) \subset E^{B}(X)$

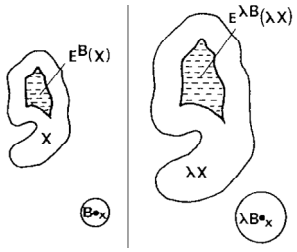


Exemple montrant qu'un ensemble érodé puis dilaté est inclus dans l'ensemble dilaté puis érodé par le même B_x

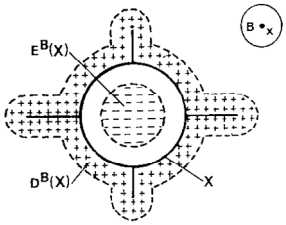


Exemple montrant que l'érosion n'est pas une transformation homotopique

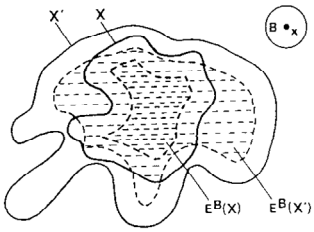
2.3 – Propriétés de l'érosion et de la dilatation



Propriétés d'homothétie pour l'érosion et la dilatation



Condition de continuité pour l'érosion et la dilatation (érosion discontinue, dilatation continue)

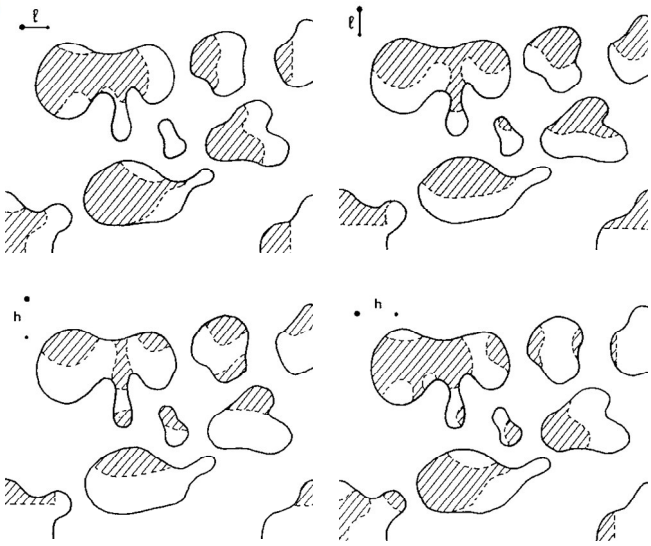


Propriété de croissance dans le cas de l'érosion :

si $X \subset X'$

alors $E^{B}(X) \subset E^{B}(X')$

2.4 – Influence de l'élément structurant



Érosion d'un ensemble d'objets X par un segment de droite, dans deux directions orthogonales

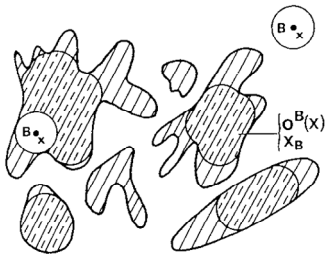
Érosion d'un ensemble d'objets X par un bi-point, dans deux directions différentes

3.1 – Ouverture

- Définition
  - On définit l'ouverture comme une érosion suivie d'une dilatation

    $O^B(X) = D^{BT}[E^B(X)]$

    ou $O^B(X) = (X \ominus B^T) \oplus B$
- Applications : suppression de détails sans déformation
  - Coupe les isthmes
  - Supprime îles et caps étroits
  - Arase les pics
- Propriétés de l'ouverture
  - $O^B(X)$ est croissante, idem-potente et anti-extensive



Ouverture d'un ensemble d'objets X par un élément structurant circulaire

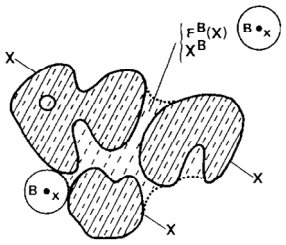
3.2 – Fermeture

- Définition
  - On définit la fermeture comme une dilatation suivie d'une érosion

$$F^B(X) = E^{BT}[D^B(X)]$$

ou $F^B(X) = (X \oplus B^T) \ominus B$

- Applications : suppression de détails sans déformation
  - Bouche les canaux étroits
  - Supprime lacs et golfes étroits
  - Ferme les vallées



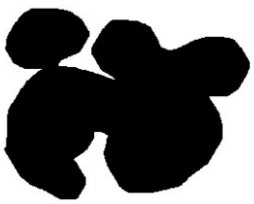
Fermeture d'un ensemble d'objets X par un élément structurant circulaire

3.3 – Comparaison des érosion / dilatation / ouverture / fermeture

Image originale

Érosion

Ouverture



Dilatation

Fermeture

4 – Filtres morphologiques

- Il est possible de combiner ouverture et fermeture successivement :

  OF, FO } idempotence

  FOF, OFO } construction de filtres médians

- Extraction des pics

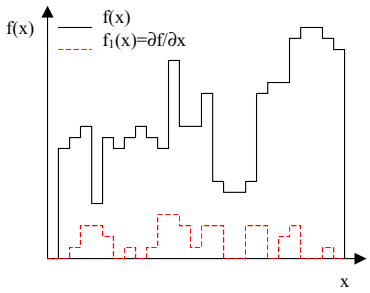
$$Tf(x) = f(x) - O^{\lambda B} f(x)$$

- Extraction des vallées

$$Tf(x) = F^{\lambda B} f(x) - f(x)$$

- Gradients morphologiques

$$g(x) = \frac{D^{\lambda B} f(x) - E^{\lambda B} f(x)}{2}$$



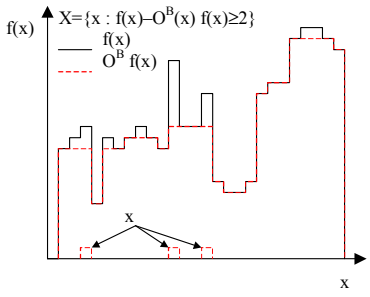
4 – Filtres morphologiques

- La transformation « chapeau haut de forme » extrait les sommets puis réalise un seuillage

$$X = \{x : f(x) - O^{\lambda B} f(x) \geq t\}$$

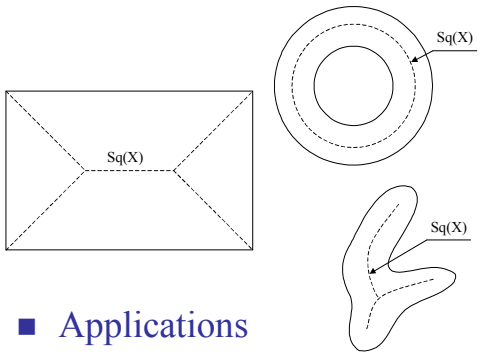
- Avantage :
  - Insensibilité aux faibles variations de niveaux de gris
- Applications : extraction
  - Chromatine des noyaux
  - Fissures de sol, etc.

- Ne sont conservées dans l'image que les portions « entrant dans le chapeau » et dépassant son sommet



5 – Squelettes

- Une représentation simplifiée de la forme d'un objet peut être faite à l'aide d'une forme stylisée qui rappelle celle de l'objet
- Lorsque cette forme stylisée représente l'ossature sur laquelle l'objet est construit, on parle de squelette
- La morphologie mathématique permet de déterminer automatiquement ce genre de forme à partir de l'objet binarisé

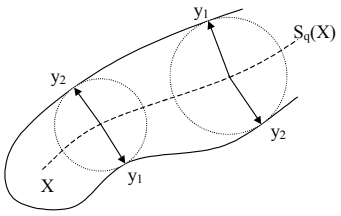


- Applications (empruntées à la géographie)
  - Ligne de partage des eaux
  - Bassins versants
  - Lignes de crête
  - Sommets-rivières

5.1 – Squelettes internes : définitions

- Soient un ensemble X et sa frontière ∂x. Un point s de X appartiendra au squelette de X, noté $S_q(X)$, si la distance euclidienne de s à ∂x est atteinte en au moins deux points distincts de X :

  $s \in S_q(X) \Leftrightarrow \exists y_1, y_2 \in \partial x,\ y_1 \neq y_2,$ tel que $d(s,\partial x)=d(s,y_1)=d(s,y_2)$
- $S_q(X)$ peut être défini aussi comme étant l'ensemble des centres des boules maximales B contenues dans X

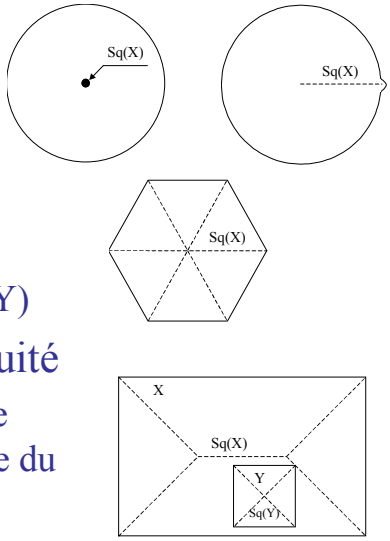


- Définition par analogie avec les feux de prairie
- Le squelette peut aussi s'exprimer en terme d'érosion et d'ouverture

$$Sq(X) = \bigcup_{\lambda>0} \bigcap_{\mu>0} \left[E^{\lambda B}(X) / O^{\mu B}(E^{\lambda B}(X)) \right]$$

5.3 – Squelettes : propriétés

- La squelettisation est une transformation
  - Anti-extensive : $Sq(X) \subset X$
  - Idempotente : $Sq\,(Sq\,(X)) = Sq\,(X)$
  - Ni croissante, ni décroissante :
    $X \subset Y$ n'implique pas $Sq(X) \subset Sq(Y)$
- Elle possède une mauvaise continuité
  - Une modification minime de l'image entraîne une modification importante du squelette



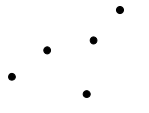
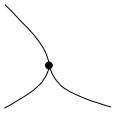
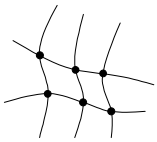
5.4 – Squelettes : points singuliers

- Mauvaises propriétés du squelette

⇒ nécessité de réaliser des post-traitements afin de
  - détecter les points particuliers du squelette
  - décider si les segments associés sont significatifs
- Cette opération est appelée ébarbulage
- Il est nécessaire de déterminer les points singuliers du squelette (points extrêmes, triples, multiples ou isolés)
- Une distance faible entre deux points singuliers indique un segment peu significatif de la forme globale de l'objet, qui doit être éliminé

5.4 – Squelettes : points singuliers

- Détection des points singuliers :

Points isolés	Points extrêmes	Points triples	Points multiples
			
$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} \bullet & 0 & 0 \\ 1 & 1 & 0 \\ \bullet & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} \bullet & 0 & \bullet \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$
	et les rotations	et les rotations	et d'autres configs

- La réduction d'information réalisée par les squelettes, tout en conservant les éléments essentiels des objets, compense largement les mauvaises propriétés de cette transformation

5.5 Squelettes par zone d'influence

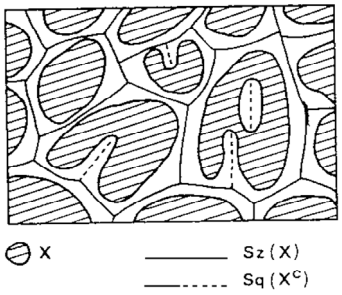
- Soit un ensemble X constitué de particules
- A chaque particule X_i, on peut associer une zone d'influence Y_i telle que tous les points y formant Y_i soient plus proches des X_i que n'importe que autre X_j, $j \neq i$. On a :

$$Sz(X) = \bigcup_i (Y_i)^C$$

avec

$$Y_i = \bigcap_i [y : d(y, x_i) < d(y, x_j), j \neq i]$$

- La squelette par zone d'influence (skiz) Sz(X) partage l'image en séparant chaque sous-ensemble non connexe par une frontière ($d(y,x_i) = d(y,x_j)$)



- Sz(X) est un sous espace du squelette complémentaire :

$Sz(X) \subset Sq(X^C)$

Polytech' Orléans

Segmentation : approches contours / régions

Traitement des images
Polytech'Orléans, 2005 – 2006
Christophe LÉGER

École polytechnique de l'université d'Orléans

UNIVERSITÉ D'ORLÉANS

Segmentation

- La segmentation des images consiste à regrouper les pixels de ces images qui partagent une même propriété pour former des régions connexes
- Deux approches :
  - Approche contour : les régions sont délimitées par les contours des objets qu'elles représentent (intuition, séparation)
  - Approche région : les régions sont déterminées en fonction de leurs propriétés intrinsèques (agrégation)

1 – Approche contours

- Principe (4 étapes)
  1. Mise en évidence des contour : différenciation de l'image
  2. Réduction de l'épaisseur des contours : l'épaisseur des contours doit être d'un pixel
  3. Binarisation des contours : au choix, réduction puis binarisation, ou inversement
  4. Description des contours : organisation des contours en structures simples telles que segments de droite, arcs de cercle, etc., fermeture des contours.

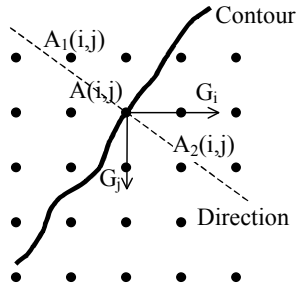
1.1 – Mise en évidence des contours

- Opérateurs de différenciation
  - Filtres passe-haut
  - Dérivées, Laplaciens
  - Morphologie mathématique, gradient morphologique
  - Réduction du bruit par filtrage passe-bas 
- Contours ⇒ séparation (segmentation) des divers éléments de l'image en régions connexes supposées de même propriétés

1.2 – Réduction des contours

Méthode de suppression des points non maximaux (Canny) pour « amincir » les lignes de contour

1. Soient A(i,j) et D(i,j) les tableaux des amplitudes et directions des gradients (on dit aussi cartes des amplitudes et directions)
2. Pour chaque point A(i,j), on détermine les deux points adjacents $A_1(i,j)$ et $A_2(i,j)$ qui sont dans la direction du gradient
3. Si A(i,j) est supérieur à la fois à $A_1(i,j)$ et $A_2(i,j)$, alors A(i,j) est conservé. Sinon, A(i,j) est annulé



Pour les images binaires, squelettisation

1.3 – Amplitudes et directions du gradient

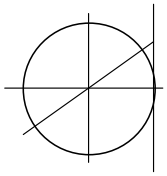
- Amplitude du gradient
  - $A(i,j) = [A_i(i,j)^2+A_j(i,j)^2]^{½} \approx |A_i(i,j)| + |A_j(i,j)| \approx \text{Max}\,[A_i(i,j), A_j(i,j)]$
- Direction du gradient
  - $D(i,j) = \arctan(A_x(i,j)/A_y(i,j))$
  - En C/C++ :

    $-\pi/2 \le D[i,j] = \text{atan}\,(Ai[i,j]/Aj[i,j]) \le +\pi/2$

    $-\pi \le D[i,j] = \text{atan2}\,(Ai[i,j],Aj[i,j]) \le +\pi$
  - Formules de changement d'échelle

    $-\pi \le D[i,j] \le +\pi \rightarrow 0 \le (D[i,j] - \pi)/(2\cdot\pi)*255 \le 255$

    $0 \le D[i,j] \le 255 \rightarrow -\pi \le D[i,j]/255*(2\cdot\pi)-\pi \le +\pi$



1.3 – Binarisation des contours

- La binarisation des contours permet d'éliminer et/ou conserver certains contours
- La segmentation par extraction de contours est efficace sur des images contrastées
- Dans le cas contraire, il peut y avoir perte d'une partie du contour lors de la binarisation

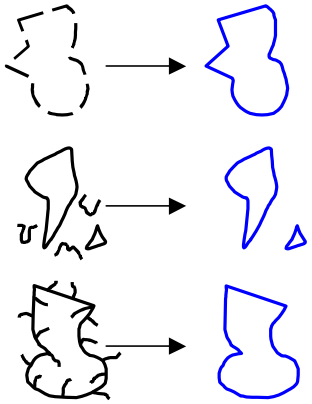
⇒ Seuillage adaptatif : le choix du seuil est fonction de la direction du gradient pour pallier les distorsions d'amplitude

1.3 – Seuillage adaptatif

- Principe
  - Choix d'un seuil initial
  - Détermination de la réponse de l'opérateur de dérivation
  - Multiplication du seuil initial par un coefficient pondérateur calculé à partir de la réponse de l'opérateur de dérivation suivant la direction trouvée
  - Seuillage avec la valeur pondérée du seuil initial

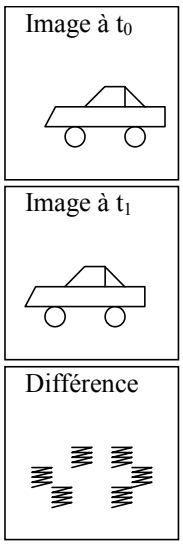
1.4 – Description des contours

- Post-traitements sur les contours
  - Fermeture des contours par extrapolation
  - Suppression des contours non fermés
  - Suppression des branches pendantes des contours fermés
- Codage
- Reconnaissance



1.5 – Objets en mouvement

- Il est possible de suivre le contour d'objets en mouvement dans une séquence d'images
- Principe
  - La différence entre des images acquises à des instants proches fait apparaître les contours perpendiculaires au déplacement (ainsi que direction, sens et vitesse du mouvement)
- Inconvénients
  - Sensibilité au type de mouvement
  - Sensibilité à la cadence d'acquisition et à la vitesse du déplacement
- Avantage
  - Méthodes très utilisée lorsque les paramètres précédents sont contrôlés



2 – Approche régions

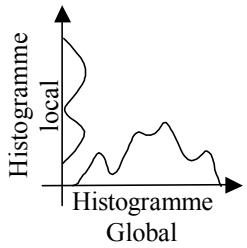
- Contrairement à l'extraction des contours qui s'intéresse aux bords des régions, la segmentation en régions homogènes vise à segmenter l'image en se basant sur des propriétés intrinsèques des régions

2.1 – Segmentation par seuillage

- Principe
  - Comparer la valeur de chaque pixel de l'image à un ou plusieurs seuils de manière à attribuer chaque pixel à une certaine classe
- Problème
  - Trouver les L-1 seuils S_l qui permettent de classer les pixels de l'image en L classes disjointes
- Remarque
  - La segmentation par seuil segmente les images indépendamment de la position des pixels (classement des pixels indépendamment de leur position)
- Trois approches : globale, locale, hybride

2.1 – Segmentation par seuillage

- Approche globale
  - Les seuils utilisés ne dépendent que d'une mesure globale calculée sur toute l'image ⇒ Histogramme 1D sur l'image
- Approche locale
  - Les seuils dépendent d'une mesure locale sur une portion de l'image ⇒ Histogramme 1D sur des portions de l'image
- Approche hybride
  - Les seuils dépendent d'une mesure à la fois globale et locale ⇒ Histogramme 2D



2.2 – Segmentation par division

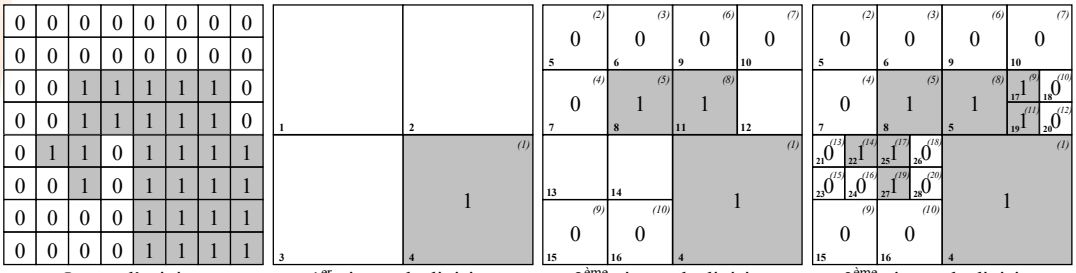
- Segmentation par division [Split]
- Principe
  - Définition d'un critère d'homogénéité
  - Test de la validité du critère sur l'image
  - Si le critère est valide, l'image est segmentée [arrêt de la méthode]
  - Sinon, l'image est découpée en zones plus petites et la méthode est réappliquée sur chacune des zones

2.2 – Segmentation par division

- Paramètres de la segmentation par division
  - Critère d'homogénéité : forme *a priori* de l'histogramme, *extrema* de l'image (valeurs minimum et maximum), valeurs identiques, variance limitée, approximation par un polynôme de degré N, etc.

    $\Rightarrow$ définition d'erreurs, de seuils ou d'intervalles de validité
  - Matière : niveaux de gris, couleurs, textures, etc.
  - Décomposition de l'image : division en 4, en 6, en polygones, etc.

2.2 – Segmentation par division



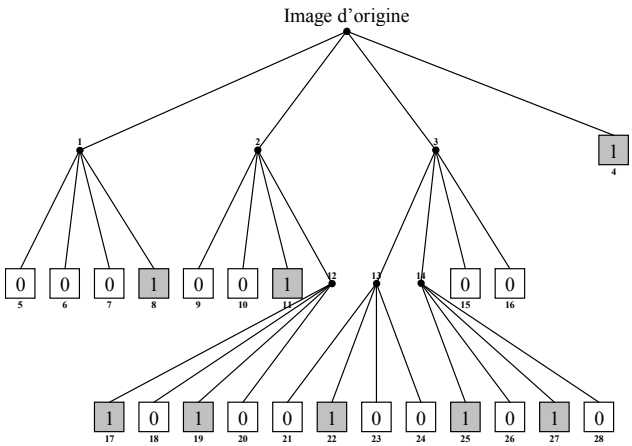
- Tant que tous les pixels ne sont pas identiques dans la région, celle-ci est divisée en 4
- Le numéro de division apparaît en bas à gauche
- Le numéro de région apparaît en haut à droite
- Quel est le critère de division ? Est-il unique ? Que penser de la division obtenue ?

2.2 – Mise en évidence des régions

- La segmentation en régions ne doit pas être systématiquement associée à une représentation en image (informations sur la structure de l'image)
- Néanmoins, il est courant de visualiser des images de régions. Il faut pour cela :
  - Affecter une couleur à chacune des régions
  - Attribuer des niveaux très différents à deux régions adjacentes (pour pouvoir visualiser cette différence)
  - Prendre en compte des nombres de régions supérieurs à 256
  - ⇒ Assigner aux régions des niveaux aléatoires entre 0 et 255

2.2 – Représentation par arbre

- ⇒ Structure hiérarchisée des régions
- Une région qui satisfait au critère forme un nœud terminal, ou feuille de l'arbre
- Une région qui ne satisfait pas au critère forme un nœud d'où partent 4 branches (division par 4) correspondant aux 4 zones sur lesquelles l'algorithme est ré-appliqué



2.2 – Exemple de critère d'homogénéité

Image d'origine

Division récursive d'une image en quadrants

Image reconstruite à l'aide de polynômes 2D de degré 0 (8317 régions)

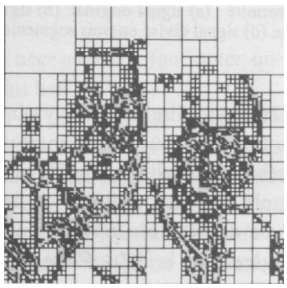
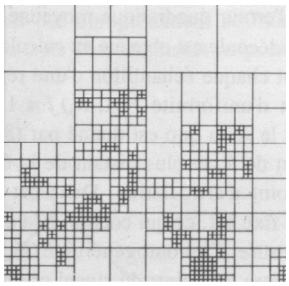


Image reconstruite à l'aide de polynômes 2D de degré 3 (583 régions)



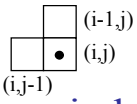
2.3 – Segmentation par rassemblement

- Segmentation par rassemblement [Merge]
- Principe
  - On explore l'image à partir de petites régions
  - On fait croître celles-ci si elles satisfont à un critère d'homogénéité ou de regroupement
- Paramètres
  - Choix du critère d'homogénéité : différence de niveau de gris moyen, valeurs similaires, etc.
  - Critère d'arrêt

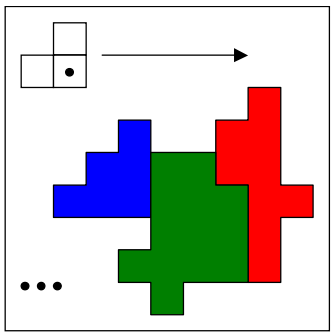
2.3 – Segmentation par rassemblement

Exemple 1 : méthode locale itérative : coloration de taches [blob coloring]

- Soit la fenêtre suivante :



- Chaque couleur représente un index qui caractérise les régions, déterminées selon l'algorithme suivant
- Inconvénients :
  - Segmentation uniligne.
  - Parcours déterminé *a priori*



2.3 – Segmentation par rassemblement

- Algorithme de coloration de taches

Pour chaque pixel I(i,j) *Faire*
  Si Critère (i,j) = Critère (i-1,j)
  Alors Couleur (i,j) ← Couleur (i-1,j)
  Sinon *Si* Critère (i,j) = Critère (i,j-1)
    Alors Couleur (i,j) ← Couleur (i,j-1)
    Sinon Couleur (i,j) ← NouvelleCouleur
  Si (Critère (i,j) = Critère (i-1,j)) ET (Critère (i,j) = Critère (i,j-1))
  Alors Fusionner les régions en donnant la même couleur aux pixels I(i,j), I(i-1,j) et I(i,j-1)
FinPour

2.3 – Segmentation par rassemblement

Exemple 2 : méthode locale récursive

- Principe : on fait croître une région avant de passer à la suivante, sans parcours particulier déterminé *a priori* (méthode par agrégation libre de pixels)
  - Germe [seed] : 
  - Croissance suivant un critère de similarité
  - Critère d'arrêt : convexité maximum, etc.
- Inconvénients
  - Méthode récursive ⇒ risques de débordements (pile)
  - Influence de la position initiale du germe

2.3 – Segmentation par rassemblement

- Algorithme de la méthode locale récursive

```
Pour chaque pixel I(i,j) Faire
    Si I(i,j) n'a pas déjà été traité
    Alors Sauvegarder (i,j), Croissance (i,j), Incrémenter Région
FinPour
Croissance (i,j)
    Pour tout Pixel(k,l) adjacent à I(i,j)      // Pour tous les 8-pixels
        Si (Pixel(k,l) pas déjà traité) ET
                (Critère (Pixel(k,l)) = Critère (I(i,j))
        Alors Croissance (k,l)
    FinPour
```

2.4 – Division et rassemblement

- Constat
  - La segmentation par division fournit une structure hiérarchisée qui permet d'établir des relations de proximité entre les régions, mais qui peut fractionner une même région en plusieurs ensembles distincts
  - La segmentation par fusion produit un nombre minimal de régions connexes, mais fournit celles-ci dans une structure horizontale qui n'indique pas de relation de proximité

2.4 – Division et rassemblement

- Proposition
  - **Rassembler**, à partir de la division grossière obtenue par **division**, les différents blocs adjacents de l'image

  ⇒Algorithme de division et rassemblement, aussi appelé algorithme Split and Merge

2.5 – Mesures d'inhomogénéité

- Inhomogénéité régionale

$$E(R_l) = \frac{1}{N_l} \sum_{n=0}^{N_l-1} [I(n) - \mu(R_l)]^2$$

  - N_l : nombre de pixels dans la région R_l (N_l=Card(R_l))
  - I(n) : valeur des pixels n ($0 \le n < N_l$) de la région R_l
  - μ (R_l) : valeur moyenne des pixels de la région R_l

$$\mu(R_l) = \frac{1}{N_l} \sum_{\substack{n=0 \\ n \in R_l}}^{N_l-1} I(n)$$

2.5 – Prédicats d'uniformité

- Prédicat d'uniformité

$$P[A, R_l] = \begin{cases} \text{Vrai} & \text{si } E(R_l) < T_l \\ \text{Faux} & \text{sinon} \end{cases}$$

  - $P[A,R_l]$: prédicat d'uniformité de la région R_l [la région R_l est considérée uniforme si le prédicat d'uniformité est vrai]
  - $E(R_l)$: valeur d'inhomogénéité de la région R_l
  - T_l : seuil de tolérance sur l'inhomogénéité de la région R_l

2.5 – Mesures d'inhomogénéité

- Inhomogénéité interrégionale

$$E(R_l, R_m) = \frac{1}{N_l + N_m} \sum_{n=0}^{N_l-1+N_m-1} [I(n) - \mu(R_l, R_m)]^2$$

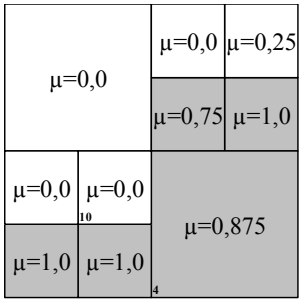
  - N_l, N_m : nombre de pixels dans les régions R_l et R_m
  - $I(n)$: valeur des pixels n ($0 \le n \le N_l-1+N_m-1$) des régions R_l et R_m
  - $\mu(R_l,R_m)$: moyenne des pixels de la région $R_l \cup R_m$

$$\mu(R_l, R_m) = \frac{1}{N_l + N_m} \sum_{\substack{n=0 \\ n \in R_l \cup R_m}}^{N_l-1+N_m-1} I(n)$$

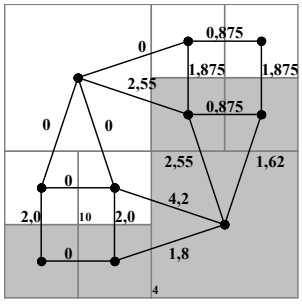
2.5 – Graphes de contiguïtés

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1
0	0	0	0	0	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	0	1	1	1
0	0	0	0	0	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

Image d'origine



Moyenne des régions



Graphe des contiguïtés

- L'image d'origine est segmentée en 10 régions
- Sur le graphe de contiguïtés des régions, la valeur $E(R_i,R_j)$ est indiquée pour chaque lien
- $\mu(R_i)$ représente la moyenne des pixels dans chaque région
- Exemple de calcul : $E(R_4,R_{10})$

$\mu(R_4,R_{10}) = 14/20$

$E(R_4,R_{10}) = 6(0-\mu)^2+14(1-\mu)^2 = 4,2$

2.5 – Critères de fin de rassemblement

- Critère de fin de rassemblement des régions

$$SEQ(L) = \sum_{i=0}^{L-1} \sum_{\substack{n=0 \\ n \in R_i}}^{N_i-1} [I(n) - \mu(R_1)]^2$$

  - SEQ(L) : somme des erreurs quadratiques
  - L : nombre de nœuds du graphe ≡ nombre de régions de l'image à l'itération considérée
  - I(n) : valeur du pixel au point n
  - $\mu(R_l)$: moyenne des pixels dans la région R_l
- Critère d'arrêt
  - Si SEQ(L) > T_2, seuil fixé *a priori*

2.5 – Critères de fin de rassemblement

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1
0	0	0	0	0	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	0	1	1	1
0	0	0	0	0	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

Image d'origine

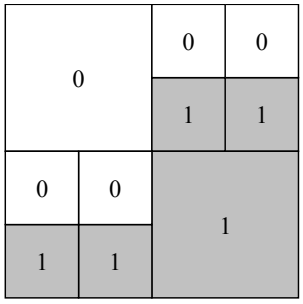


Image divisée (T_l=0,2)

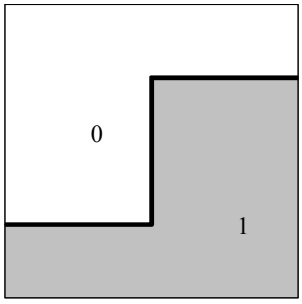


Image rassemblée (T_2=5)

- L'image est tout d'abord divisée en choisissant T_l=0,2 comme valeur de seuil de tolérance de l'inhomogénéité régionale
- Les différentes régions sont ensuite rassemblées en choisissant T_2=5 comme valeur de seuil appliqué au critère de fin de rassemblement

2.5 – Rapport de vraisemblance

- Soient deux régions R1 et R2 de N1 et N2 pixels
- Hypothèse 1 : les deux régions appartiennent au même objet ⇒ les intensités des pixels suivent une même distribution Gaussienne (μ0,σ0)
- Hypothèse 2 : les deux régions appartiennent à deux objets différents ⇒ chaque région possède une distribution Gaussienne différente (μ1,σ1) et (μ2,σ2)
- Avec $p[I(n)] = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{[I(n)-\mu]^2}{2\sigma^2}}$ on a $\hat{\mu} = \frac{1}{N}\sum_{n=0}^{N-1} I(n)$ $\hat{\sigma} = \frac{1}{N}\sum_{n=0}^{N-1}[I(n)-\hat{\mu}]^2$

2.5 – Rapport de vraisemblance

Avec l'hypothèse H_1, la densité de probabilité conjointe est :

$$p[I(0), I(1), \ldots, I(N_1 - 1 + N_2 - 1) | H_1] = \prod_{n=0}^{N_1-1+N_2-1} p[I(n)|H_1]$$

$$= \prod_{n=0}^{N_1-1+N_2-1} \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{[I(n)-\mu_0]^2}{2\sigma_0^2}} = \frac{1}{\left(\sqrt{2\pi}\sigma_0\right)^{N_1+N_2}} e^{-\frac{\sum_{i=0}^{N_1+N_2-2}[I(n)-\mu_0]^2}{2\sigma_0^2}}$$

$$\boxed{= \frac{1}{\left(\sqrt{2\pi}\sigma_0\right)^{N_1+N_2}} e^{-\frac{N_1+N_2}{2}}}$$

2.5 – Rapport de vraisemblance

Avec l'hypothèse H_2, la densité de probabilité conjointe est :

$$p\left[I(0),I(1),\ldots,I(N_1-1+N_2-1)\middle|H_2\right]=\prod_{n=0}^{N_1-1+N_2-1}p\left[I(n)\middle|H_2\right]$$

$$=\prod_{n=0}^{N_1-1}\frac{1}{\sqrt{2\pi}\sigma_1}e^{-\frac{[I(n)-\mu_1]^2}{2\sigma_1^2}}\cdot\prod_{n=0}^{N_2-1}\frac{1}{\sqrt{2\pi}\sigma_2}e^{-\frac{[I(n)-\mu_2]^2}{2\sigma_2^2}}$$

$$=\frac{1}{\left(\sqrt{2\pi}\sigma_1\right)^{N_1}}e^{-\frac{N_1}{2}}\cdot\frac{1}{\left(\sqrt{2\pi}\sigma_2\right)^{N_2}}e^{-\frac{N_2}{2}}$$

2.5 – Rapport de vraisemblance

Le rapport de vraisemblance L est défini par :

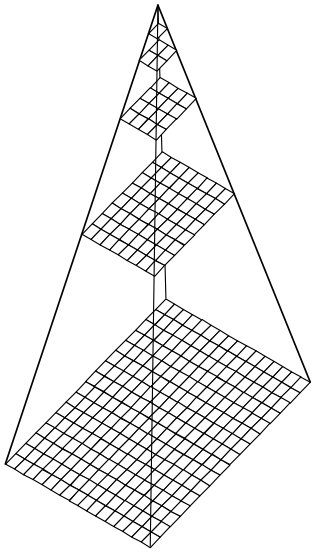
$$L=\frac{p\left[I(0),I(1),\ldots,I(N_1-1+N_2-1)\middle|H_1\right]}{p\left[I(0),I(1),\ldots,I(N_1-1+N_2-1)\middle|H_2\right]}=\frac{\sigma_0^{N_1+N_2}}{\sigma_1^{N_1}\cdot\sigma_2^{N_2}}$$

σ_0, σ_1, σ_2 sont estimés par la relation $\hat{\sigma}=\frac{1}{N}\sum_{n=0}^{N-1}[I(n)-\hat{\mu}]^2$

- Si L>seuil, les deux régions sont distinctes
- A l'inverse, si L≤seuil, les deux régions doivent être regroupées

2.6 – Segmentation multirésolution

- Un grand nombre de propriétés d'images peut être calculé à l'aide d'opérateurs appliqués localement
- Ex : statistique de niveau de gris, propriétés texturales, discontinuités locales
- Propriétés texturales:
  - Résolution ⇒ sélection des textures
  - Micro-textures ⇒ segments isolés
- Parade : multi-résolution pyramidale



Polytech' Orléans

Traitements globaux

Traitement des images
Polytech'Orléans, 2005 – 2006
Christophe LÉGER

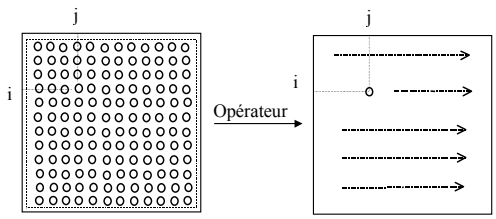
École polytechnique de l'université d'Orléans

UNIVERSITE D'ORLEANS

Traitements globaux

École polytechnique de l'université d'Orléans

- Tous les pixels de l'image initiale interviennent pour déterminer la valeur de chaque pixel de l'image résultat
- Chaque pixel de l'image transformée est une combinaison linéaire de tous les pixels de l'image initiale



- Transformations unitaires
  - conservent toute l'information de l'image, mais en changent la représentation
- Transformat. géométriques
  - modifient la position des pixels sans en changer la valeur

1 – Transformations unitaires

- On désigne par le terme « transformation unitaire » toute transformation qui représente une même information sous une base différente
- Les transformations unitaires conservent toute l'information des images
- La transformée de Fourier est un exemple de transformation unitaire

1.1 – Transformée de Fourier : définitions

- Cas continu

$$\Im\{f(x,y)\} = F(u,v) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\, e^{-i2\pi(ux+vy)}\, dx\, dy$$

$$f(x,y) = \Im^{-1}\{F(u,v)\} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} F(u,v)\, e^{+i2\pi(ux+vy)}\, du\, dv$$

- Cas discret

$$\Im\{f(m,n)\} = F(u,v) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)\, e^{-i2\pi\left(\frac{um}{M}+\frac{vn}{N}\right)}$$

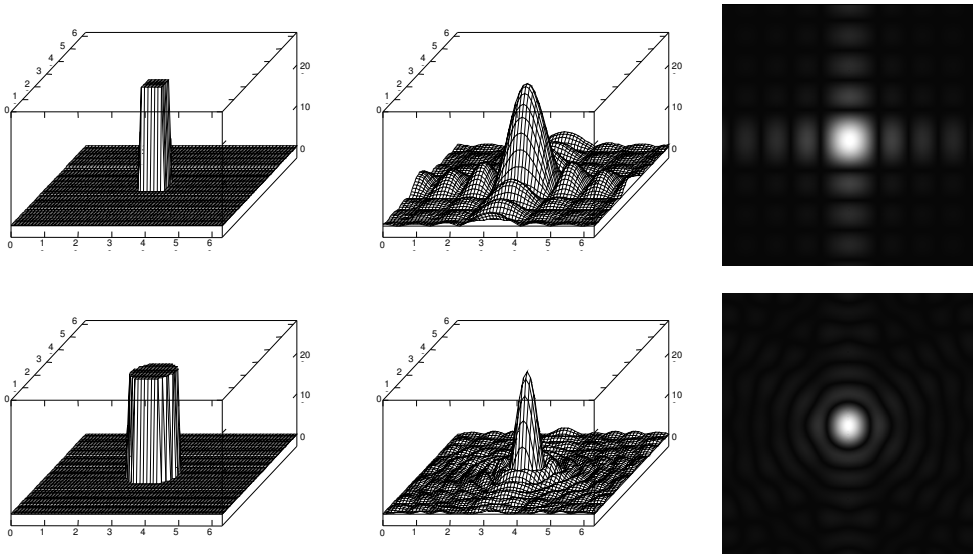
$$f(m,n) = \Im^{-1}\{F(u,v)\} = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} F(u,v)\, e^{+i2\pi\left(\frac{um}{M}+\frac{vn}{N}\right)}$$

- Cas M=N

$$\Im\{f(m,n)\} = F(u,v) = \frac{1}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m,n)\, e^{-\frac{i2\pi}{N}(um+vn)}$$

$$f(m,n) = \Im^{-1}\{F(u,v)\} = \frac{1}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} F(u,v)\, e^{+\frac{i2\pi}{N}(um+vn)}$$

1.1 – Transformée de Fourier : exemples



1.1 – Transformée de Fourier : propriétés

- Linéarité : $\Im\{f_1(n,m)+f_2(n,m)\}=\Im\{f_1(n,m)\}+\Im\{f_2(n,m)\}$

  et $\alpha\Im\{f_1(n,m)\}=\Im\{\alpha f_1(n,m)\}$

- Translation : $\Im\{f(m-m_0,n-n_0)\}=\Im\{f(m,n)\}\cdot e^{+i2\pi\left(m_0\frac{u}{M}+n_0\frac{v}{N}\right)}$
  - Le module de la TF n'est influencé que par le contenu de l'image, et non par sa position
- Rotation :
  - Après passage en coordonnées polaires, en posant $n = \rho \cos\theta$, $m = \rho \sin\theta$, $u = \omega \cos\varphi$, $v = \omega \sin\varphi$, f (n,m) et F(u,v) deviennent $f(\rho, \theta)$ et $F(\omega, \varphi)$: $\Im\{f(\rho,\theta+\theta_0)\}=F(\omega,\varphi+\theta_0)$
  - La rotation de f(n,m) d'un angle θ_0 entraîne la rotation de F(u,v) d'un même angle

1.1 – Transformée de Fourier : propriétés (suite)

- Homothétie : $\Im\{f(a\cdot n, b\cdot m)\} = \frac{1}{a\cdot b} F\left(\frac{u}{a}, \frac{v}{b}\right)$
  - Ceci signifie que lorsqu'une forme « s'élargit » dans une image, la transformée de Fourier se « rétrécit »
- Périodicité : $F(u,v) = F(u+M,v) = F(u,v+N) = F(u+M,v+N)$
- Symétrie conjuguée : $F(u,v) = F^*(-u,-v)$ et $|F(u,v)| = |F(-u,-v)|$
- Laplacien : $\Im\{\nabla^2 f(n,m)\} \Leftrightarrow -(2\pi)^2 (u^2+v^2) F(u,v)$

  avec $\nabla^2 f(m,n) = \frac{\partial^2 f}{\partial m^2} + \frac{\partial^2 f}{\partial n^2}$
- Valeur moyenne : $\overline{f(m,n)} = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n) = \frac{1}{MN} F\left(\frac{M}{2}, \frac{N}{2}\right)$

1.1 – Transformée de Fourier : propriétés (fin)

- Séparabilité :

$$F(u,v) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n) e^{-i2\pi\frac{um}{M}} e^{-i2\pi\frac{vn}{N}}$$

$$F(u,v) = \frac{1}{M}\sum_{m=0}^{M-1} e^{-i2\pi\frac{um}{M}} \frac{1}{N}\sum_{n=0}^{N-1} f(m,n) e^{-i2\pi\frac{vn}{N}}$$

$$F(u,v) = \frac{1}{M}\sum_{m=0}^{M-1} F(m,v) e^{-i2\pi\frac{um}{M}} \quad \text{avec} \quad F(m,v) = \frac{1}{N}\sum_{n=0}^{N-1} f(m,n) e^{-i2\pi\frac{vn}{N}}$$

  - Ceci signifie que le calcul peut être effectué en deux étapes : calcul sur l'indice des lignes puis sur celui des colonnes

1.1 – Transformée de Fourier : mise en œuvre

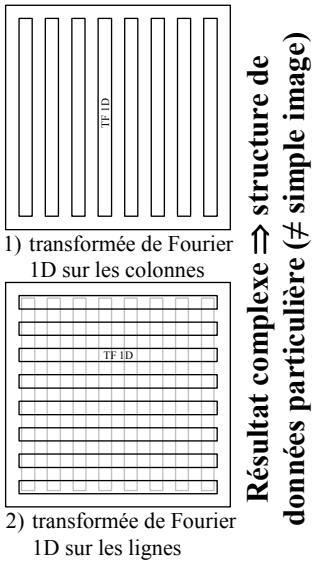
- La transformée de Fourier d'une fonction 2D F(u,v) est obtenue par :
  1) Transformée de Fourier 1D sur les lignes de l'image
  2) Transformée de Fourier 1D du résultat précédent sur les colonnes

OU 1) Transformée de Fourier 1D sur les colonnes de l'image,
2) Transformée de Fourier 1D du résultat précédent sur les lignes

⇒ Réduction de la complexité de l'algorithme de calcul :

- Calcul classique 2D : complexité en N^4
- Noyau séparable $2 \times N \times 1D$: complexité en N^3
- Utilisation d'un algorithme FFT : complexité en $2 \times N^2 \times \log_2 N$ mais images de dimensions puissances entières de 2

1.1 – Transformée de Fourier : mise en œuvre



1.1 – Transformée de Fourier : représentation

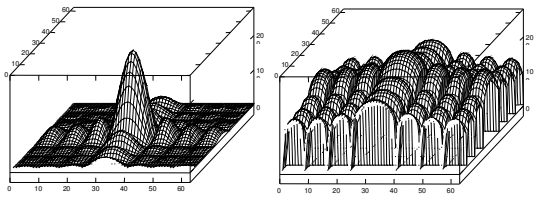
- Échelle de représentation
  - Amplitude importante en F(0,0)

$$F(0,0) = N^2 \langle F(m,n) \rangle$$

  - Décroissance rapide ⇒ Rep log

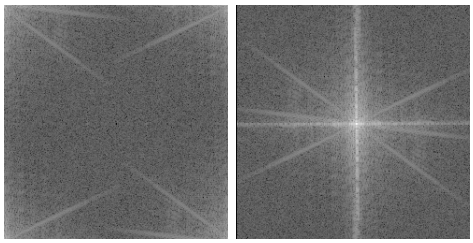
$$D(u,v) = K \log\left(1 + |F(u,v)|\right)$$

$$K = \frac{255}{\log\left(1 + |\max(F(u,v))|\right)}$$



- Position des fréquences
  - On place généralement la fréquence (0,0) au centre
  - On utilise la périodicité de la transformée

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



1.1 – Transformée de Fourier : difficultés

- Aliasing (repliement)
  - fe/2 + ε ⇒ fe/2 – ε
- Ondulations (riple)
  - Durée finie d'échantillonnage ⇒ convolution avec sinus cardinal
- Étalement des fréquences (leakage)
  - Durée finie d'échantillonnage ⇒ convolution avec sinus cardinal

1.2 – Filtrage par transformée de Fourier

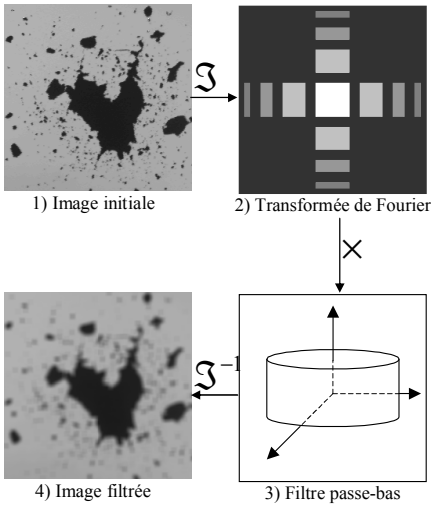
- Filtrage par TF
  - ⇔ au filtrage linéaire par convolution
  - ➢ critères de sélection : rapidité de calcul, sélectivité, synthèse aisée
- Mise en œuvre
  - ➢ Soit une image I et un filtre F :

$$\Im\{I * F\} = \Im\{I\} \times \Im\{F\}$$

  - ➢ Cela revient à :

$$I * F = \Im^{-1}\{\Im\{I\} \times \Im\{F\}\}$$

  - ➢ Le calcul se réduit à deux TF et une multiplication



1.2 – Filtrage par transformée de Fourier

- Exemples de filtres
  - ➢ disques concentriques
  - ➢ anneaux
  - ➢ couronnes
- Temps de calcul
  - ➢ Soit N taille image et M taille filtre
  - ➢ Filtrage par convolution : $M^2 \times N^2$
  - ➢ Filtre à noyau séparable : $2 \times M \times N^2$
  - ➢ Filtrage par FFT : $2 \times N^2 \times \log_2 N$

N	64	128	256	512	1024
M	3	3	4	4	4
M (noy. sép.)	6	7	8	9	10

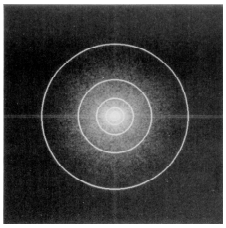
M : taille du filtre à partir de laquelle il vaut mieux utiliser un filtrage FFT

- Sélection des fréquences
  - ➢ Méthodes de synthèse avec gabarit
- Attention
  - ➢ Effets de bord si discontinuités (phénomène de Gibbs)
  - ➢ Etalement (leakage) et repliement (aliasing) des fréquences

⇒ Utilisation de fenêtres de pondération pour limiter cet inconvénient

1.2 – Filtrage par transformée de Fourier

Filtrage passe-bas idéal



Spectre de l'image d'origine. Les cercles représentent respectivement 90, 93, 95, 99 et 99,5 % de la puissance totale de l'image

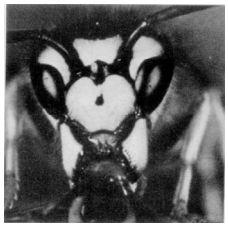
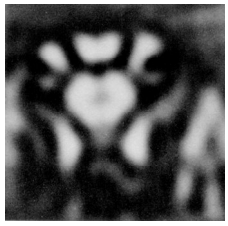
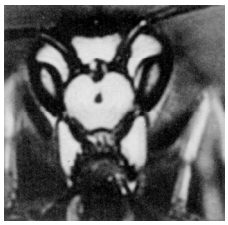


Image d'origine

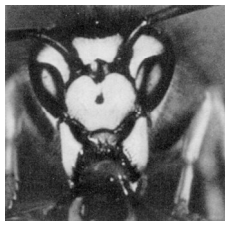


Filtrage passe-bas avec 90 % de la puissance

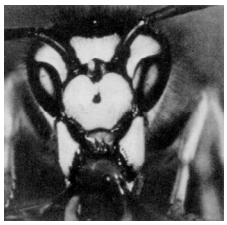
Filtrage passe-bas avec 93 % de la puissance



Filtrage passe-bas avec 95 % de la puissance



Filtrage passe-bas avec 99 % de la puissance

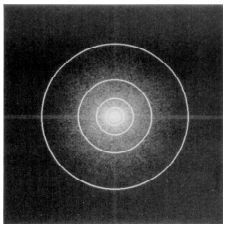


Filtrage passe-bas avec 99,5 % de la puissance

Traitements globaux 15

1.2 – Filtrage par transformée de Fourier

Filtrage passe-bas de Butterworth



Spectre de l'image d'origine. Les cercles représentent respectivement 90, 93, 95, 99 et 99,5 % de la puissance totale de l'image

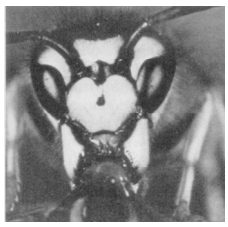
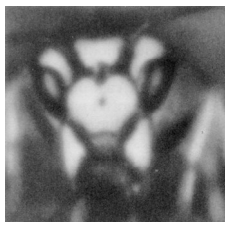
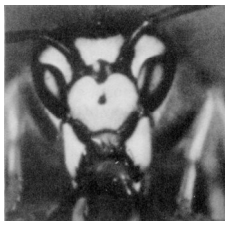


Image d'origine

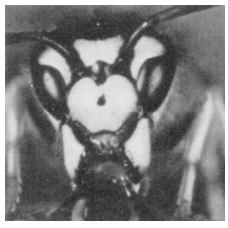


Filtrage passe-bas avec 90 % de la puissance

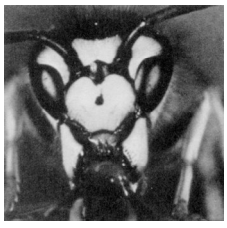
Filtrage passe-bas avec 93 % de la puissance



Filtrage passe-bas avec 95 % de la puissance



Filtrage passe-bas avec 99 % de la puissance



Filtrage passe-bas avec 99,5 % de la puissance

Traitements globaux 16

1.3 – Déconvolution par transformée de Fourier

- Soit une image $I_F(x,y)$ connue mais dégradée par une déformation linéaire (flou [mauvais réglage de la caméra], bougé [acquisition])
- $I_F(x,y)$ peut être écrite sous la forme d'une convolution :

$$I_F(x,y) = h(x,y) * I_I(x,y)$$

- Ce qui s'écrit encore, après passage dans le domaine fréquentiel :

$$\Im\{I_F(x,y)\} = H(u,v) \times \Im\{I_I(x,y)\}$$

- En caractérisant la déformation (sa réponse impulsionnelle h ou sa fonction de transfert H par modélisation ou par étalonnage), on peut théoriquement remonter à l'image initiale inconnue :

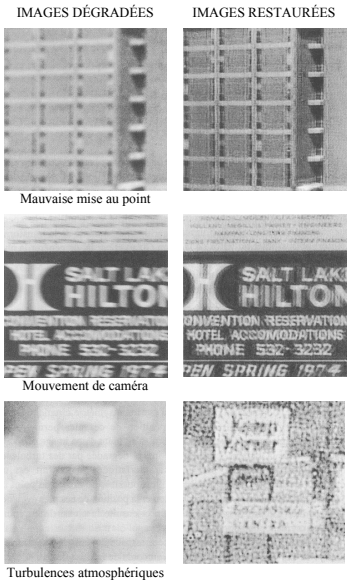
$$I_I(x,y) = \Im^{-1}\left\{\frac{\Im\{I_F(x,y)\}}{H(u,v)}\right\}$$

1.3 – Déconvolution par transformée de Fourier

- Caractérisation de la fonction de transfert
  - acquisition d'une image impulsionnelle (point lumineux) dans les mêmes conditions que pour l'image dégradée
- Attention
  - H(u,v) peut être nulle ⇒ division par zéro
  - TF [$I_F(x,y)$] et H(u,v) peuvent être nulles simultanément ⇒ indétermination
  - En présence de bruit, on obtient :

$$I_I(u,v) = \frac{I_F(u,v)}{H(u,v)} - \frac{B(u,v)}{H(u,v)}$$

  - et B(u,v) / H(u,v) peut être très grand...



1.4 – Autres applications de la transformée de Fourier

- Compression d'images
  - L'énergie de l'image est compressée dans les basses fréquences
  - Si ces composantes sont seulement conservées, le contenu global de l'image l'est aussi, et la perte d'information est faible
  - En pratique, ce sont d'autres techniques sont utilisées pour la compression
- Caractérisation
  - Comparaison d'images ayant subit des déplacements
  - Les propriétés d'invariance du module en translation et celles en rotation permettent de retrouver des critères communs à des images
  - Ces éléments sont appelés invariants de l'image

1.5 – Autres transformations unitaires

- Transformations à noyau séparable
  - Transformation directe

$$T(u,v) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)\, g(m,n,u,v)$$

    - $g(m,n,u,v)$ est appelé noyau direct
    - Le noyau est dit séparable si : $g(m,n,u,v) = g_1(m,u)\cdot g_2(n,v)$
    - Le noyau est dit symétrique si : $g_1(m,u) = g_2(n,v)$
  - Transformation inverse

$$f(m,n) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} T(u,v)\, h(m,n,u,v)$$

    - $h(m,n,u,v)$ est appelé noyau inverse

Questions:
- La TF est-elle séparable ?
- La TF est-elle symétrique ?

1.5 – Autres transformations unitaires

- Remarque : la transformée de Fourier est un cas particulier des transformations à noyau séparable :

$$g(m,n,u,v) = \frac{1}{MN} e^{-i2\pi\left(u\frac{m}{M}+v\frac{n}{N}\right)} \qquad h(m,n,u,v) = e^{+i2\pi\left(u\frac{m}{M}+v\frac{n}{N}\right)}$$

- Toute transformation séparable peut être calculée en deux temps :
  - sur chaque ligne, puis sur chaque colonne ; ou inversement
- Représentation matricielle
  - On peut écrire : $T = AFA$ ou $F = BTB$ avec $B = A^{-1}$
  - où F est une image N×N, A une matrice de transformation N×N aux éléments $a_{ij}=g_1(i,j)$, T la matrice N×N résultat transformée

1.6 – Transformée en cosinus

- Discrete Cosinue Transform (DCT) :

$$g(m,n,0,0) = \frac{1}{N} \qquad g(m,n,u,v) = \frac{1}{2N^3} \cdot \cos\left[(2m+1)u\pi\right] \cos\left[(2n+1)v\pi\right]$$

  - Le noyau de la transformée inverse h(m,n,u,v) est le même
- Propriétés :
  - La DCT est réelle, est à noyau séparable, a deux fois plus de fonctions que la TF, et peut s'obtenir à partir de la TF :

$$DCT(u) = \sqrt{\frac{2}{M}} \mathrm{Re}\left\{ e^{-i\pi\frac{u}{2M}} \sum_{m=0}^{2M-1} f(m) \cdot e^{-i\pi\frac{um}{M}} \right\}$$

- Applications :
  - Compression d'images (algorithme JPEG) : compactage de l'énergie pour des images corrélées

1.7 – Transformée de Walsh

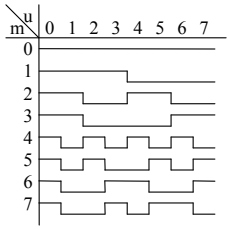
- La transformée de Walsh s'écrit en utilisant les noyaux :

$$g(m,n,u,v) = h(m,n,u,v) = \frac{1}{N}\prod_{i=0}^{k-1}(-1)^{[b_i(m)\cdot b_{k-1-i}(u)+b_i(n)\cdot b_{k-1-i}(v)]}$$

où $b_k(p)$ est le $k^{ème}$ bit de la représentation binaire de p

Exemple : n=3, p=6 (110 binaire) $b_0(p)=0$, $b_1(p)=1$ et $b_2(p)=1$

- Remarques
  - Il existe un algorithme rapide basé sur celui de la FFT
  - Décomposition en fonctions de base valant ±1
- Applications
  - Codage (transformée sans produit), compression



Valeurs du noyau 1D de la transformée de Walsh pour N=8

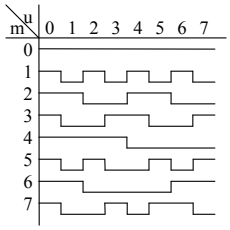
1.8 – Transformée de Hadamard

- La transf. de Hadamard s'obtient à partir des noyaux :

$$g(m,n,u,v) = h(m,n,u,v) = \frac{1}{N}(-1)^{\sum_{i=0}^{n-1}[b_i(m)\cdot b_i(u)+b_i(n)\cdot b_i(v)]}$$

où $b_k(p)$ est le $k^{ème}$ bit de la représentation binaire de p

- Remarques
  - Comme pour la transformée de Walsh, les noyaux directs et inverses sont identiques
  - ⇒ Algorithme identique pour les transformées directe et inverse
  - Pour $N=2^n$, il y a équivalence entre la transformée de Walsh et de Hadamard. Seul l'ordre des lignes et des colonnes est échangé. On parle de transformée de Walsh-Hadamard



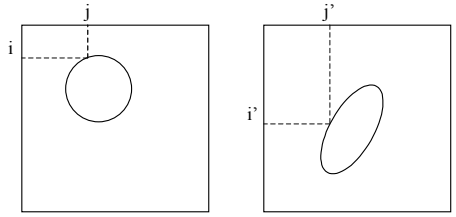
Valeurs du noyau 1D de la transformée de Hadamard pour N=8

1.10 – Autres transformations

- Pour mémoire :
  - Transformée de Haar
  - Transformée de Slant
  - Transformée de Hotteling
  - Transformée de Karhunen-Laève
  - Transformée SVD
  - Transformé de Radon
  - Transformée en ondelettes
  - et bien d'autres encore...

2.1 – Transformations géométriques

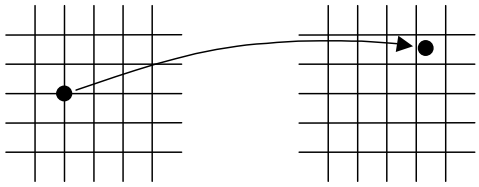
- Une transformation géométrique pure ne change pas les valeurs de niveaux de gris, mais seulement leur position
- C'est donc une transformation qui modifie les indices des pixels
- les transformations géométriques sont aussi appelées transformations élastiques



$$I(i', j') = I(i, j) \text{ avec } \begin{cases} i' = r(i, j) \\ j' = s(i, j) \end{cases}$$

2.2 – Transformations géométriques : mise en œuvre

- En pratique, l'utilisation d'une transformation continue fournit des coordonnées (i',j') qui ne correspondent pas à la grille d'échantillonnage



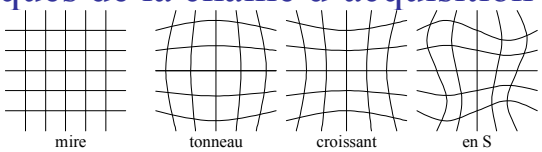
⇒ Algorithme en deux étapes

① Transformation spatiale, qui définit le réarrangement des pixels de l 'image

② interpolation (si l'équation de la déformation est connue) ou utilisation d'une table de correspondance position à position

2.3 – Transformations géométriques

- Exemple : $\begin{cases} r(i,j) = i/2 \\ s(i,j) = j/2 \end{cases}$
  - réduction de dimension (/2) dans chaque dimension spatiale
- Application 1 : Corrections des déformations géométriques de la chaîne d'acquisition

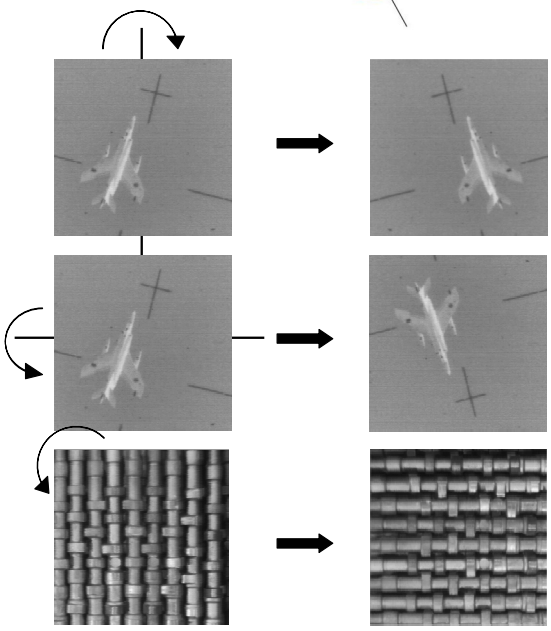


⇒ définition de points invariants (3) pour donner un référentiel

- Application 2 : Mise en correspondance
  - cartographie aérienne, images médicales

2.3 – Rotations, translations, …

- Retournement vertical
- Retournement horizontal
- Transposition
- Décalage à gauche
- Décalage à droite
- Pivot à gauche
- Pivot à droite
- Rotation
- …

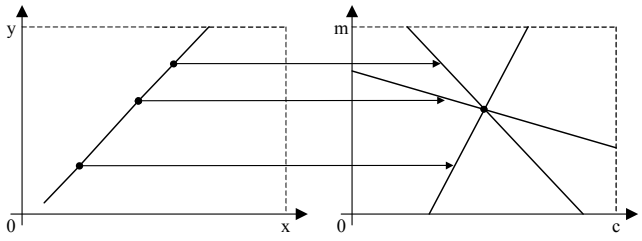


3 – Transformée de Hough

- La transformée de Hough est utilisée pour mettre en évidence des points alignés dans des images
- Principe : changer l'espace de représentation

Droite d'équation
$y = m . x + c$

Les variables sont x et y



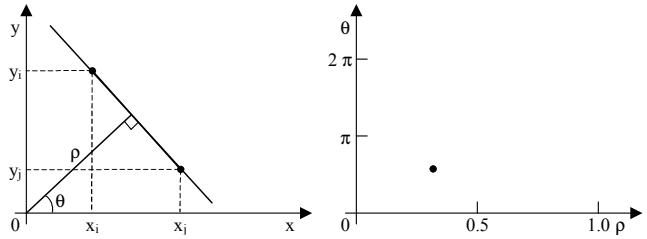
Droite d'équation
$c = y – m . x$

Les variables sont m et c. x et y fixent l'ordonnée à l'origine et la pente des droites

- Une droite est représentée par un point, un point par une droite
- La transformation de Hough est aussi appelée transformation ligne vers point

3 – Transformée de Hough

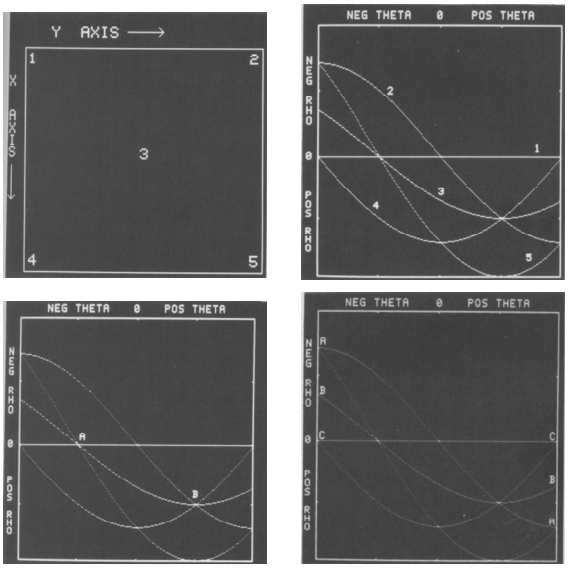
- Pour éviter les problèmes de pente infinie, une ligne de l'image est décrite par l'équation : $\rho = x \cos\theta + y \sin\theta$, où ρ est la distance de la ligne à l'origine et θ l'angle de cette droite par rapport à l'axe x



- La transformée de Hough de cette ligne devient un point de coordonnées (ρ,θ) dans le domaine polaire
- Tous les points (x,y) de la ligne donnent des sinusoïdes qui se croisent au point de coordonnées (ρ,θ)

3 – Transformée de Hough

- Exemple : voir ci-contre
- Complément :
  - La transformée de Hough peut s'appliquer à des formes géométriques autres que des droites
  - Il faut connaître une représentation paramétrique de la forme et pouvoir réaliser le changement d'espace de représentation.
  - Mais cette technique est souvent difficile à mettre en œuvre du fait d'un trop grand nombre de paramètres de forme à prendre en compte



Bibliographie

- Traitement de l'image sur micro-ordinateur, *Jean Jacques Toumazet*, Sybex
- Géométrie discrète en analyse d'images, *Jean-Marc Chassery, Annick Montanvert*, Hermès
- Analyse d'images : filtrage et segmentation, *Jean-Pierre Coquerez, Sylvie Philipp*, Masson
- Précis d'analyse d'images, *Michel Coster, Jean-Louis Chermant*, Presses du CNRS
- Morphologie mathématique, *Michel Schmitt, Juliette Mattioli*, Masson
- Traitement numérique des images, *Mura Kunt*, Presses Polytechniques et Universitaires Romandes
- Machine Vision, *Ramesh Jain, Rangachar Kasturi, Brian Schunk*, Mc Graw-Hill
- Two-dimensionnal imaging, *Ronald Bracewell*, Prentice Hall
- Digital Image Processing, *Rafael Gonzalez, Richard Woods*, Addison Wesley

Coordonnées

Christophe LEGER
Université d'Orléans
Polytech'Orléans/LESI
12 rue de Blois – BP 6744
45067 Orléans cedex 02 – FRANCE
Tel : (33) (0)2 38 49 45 63
Fax : (33) (0)2 38 41 72 45

Christophe.Leger@univ-orleans.fr

http://www.univ-orleans.fr/polytech

http:// www.univ-orleans.fr/lesi/personnel/leger