République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique
Université Ibn Khaldoun – Tiaret
Faculté des Mathématique et de l'Informatique
Département d'Informatique

TRAITEMENT ET ANALYSE DES IMAGES NUMERIQUES

COURS POUR :

MASTER GENIE INFORMATIQUE : 2019/2020

Par : Mr. KARIM MEZZOUG

karim.mezzoug@univ-tiaret.dz

PROGRAMME SEMESTRIEL DU MODULE TRAITEMENT ET ANALYSE D'IMAGES NUMERIQUES

Chapitre 1 : Introduction à l'image Numérique

Chapitre 2 : Amélioration et Restauration d'Images Numériques

Chapitre 3 : Détection de Contour et Segmentation d'Images Numérique

Chapitre 4 : Codage et Compression d'Images Numériques

Chapitre 1 : Introduction à l'image numérique

1- Historique :

Le traitement d'images commence à être étudié dans les années 1920 pour la transmission d'images par le câble sous-marin allant de New York à Londres. Harry G. Bartholomew et Maynard D. McFarlane effectuent la première numérisation d'image avec compression de données pour envoyer des fax de Londres à New York. Le temps de transfert passe ainsi de plus d'une semaine à moins de trois heures. Il n'y a pas vraiment d'évolution par la suite jusqu'à la période d'après-guerre.

Le traitement du signal prend de l'importance vers la fin de la Seconde Guerre mondiale avec l'arrivée du radar. La prospection pétrolière participe aussi beaucoup au développement des techniques de traitement du signal.

Le véritable essor du traitement d'images n'a lieu que dans les années 1960 quand les ordinateurs commencent à être suffisamment puissants pour travailler sur des images. Peu après, la redécouverte de la transformée de Fourier rapide (FFT) révolutionne le domaine, en rendant possible les manipulations du contenu fréquentiel des signaux sur ordinateur. Cependant, l'essentiel des recherches porte encore, à cette époque, sur l'amélioration des images et leur compression.

En 1980, David Marr formalise le premier la détection de contours de manière précise (D. Marr et E. Hildreth : *Theory of Edge Detection*, Proc. R. Soc. London, B 207, 187-217, 1980). Au cours des années 1980, un véritable engouement se fait jour pour le traitement de l'image et surtout pour la compréhension de l'image par des systèmes experts. Les ambitions étaient beaucoup trop grandes, l'échec fut d'autant plus cuisant.

Les années 1990 sont témoin de l'amélioration constante des opérateurs. La recherche médicale devient un très gros demandeur en traitement d'images pour améliorer les diagnostics faits à partir des nombreuses techniques d'imagerie médicale, la technique reine étant l'IRM. Les publicitaires, puis le grand public se familiarisent avec la retouche d'image grâce au logiciel Photoshop, et le traitement d'images dans un objectif esthétique se répand avec l'apparition d'autres logiciels dédiés (The Gimp, Paint Shop Pro). Enfin, la décennie s'achève sur l'engouement pour les ondelettes et les images multimodales.

2- Définition du Terme Scientifique :

Le traitement et l'analyse d'images numérique désigne une discipline de l'informatique et des mathématiques appliquées qui étudie les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information.

Il s'agit d'un sous-ensemble du traitement et analyse du signal dédié aux images et aux données dérivées comme la vidéo (par opposition aux parties du traitement du signal consacrées à d'autres types de données : son et autres signaux monodimensionnels notamment), tout en opérant dans le domaine numérique (par opposition aux techniques analogiques de traitement du signal, comme la photographie ou la télévision traditionnelles).

Dans le contexte de la vision artificielle, le traitement d'images se place après les étapes d'acquisition et de numérisation, assurant les transformations d'images et la partie de calcul permettant d'aller vers une interprétation des images traitées. Cette phase d'interprétation est d'ailleurs de plus en plus intégrée dans le traitement d'images, en faisant appel notamment à l'intelligence artificielle pour manipuler des connaissances, principalement sur les informations dont on dispose à propos de ce que représentent les images traitées (connaissance du *domaine*).

3- Les Différentes Transformations Possibles sur l'Image Numérique:

On peut distinguer quatre (04) types ou modes de transformations, qu'on peut effectuer à partir ou vers une image numérique :

A. **Le Traitement des Images** : image → image («image processing»).
B. **L'Analyse des Images** : image → mesures («image analysis») analyse sémantique:
C. **Image description de haut niveau** : images → Classes d'objet («image understanding or recognition»).
D. **Reconstruction d'Images** : Opérateurs ensemble d'informations → image

4- Exemples de Domaines d'Applications :

- **Contrôle de présence / absence**. Sur des chaînes de production, on vérifie en bout de chaîne avec une caméra vidéo la présence d'une pièce dans un ensemble plus complexe. Pour cela bien souvent il suffit de faire un simple seuillage dans une région spécifique.

- **Contrôle du niveau de maturation des fruits sur une chaîne de conditionnement**. Il s'agit de reconnaître à la couleur et à la texture du fruit son degré de maturité et donc la catégorie sous laquelle il sera emballé puis vendu.

- **Construction et correction de cartes géographiques d'après des images satellites ou des images aériennes**. On recale d'après des informations topographiques les images reçues, puis on les met sur la carte en correspondance avec les informations trouvées dans l'image : voies de communication, voies et plans d'eau, parcelles agricoles...

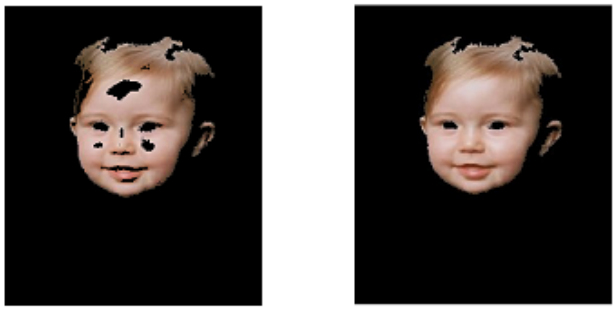
- **Surveillance et évaluation de la production agricole**. Il est possible de déterminer le degré de maturation des cultures, la quantité d'eau nécessaire pour l'irrigation, le rendement moyen... On peut ainsi établir des prévisions à large échelle de la récolte à venir.

- **Reconnaissance de l'écriture**. La reconnaissance de l'écriture manuscrite progresse de jour en jour. Elle est suffisamment opérationnelle pour que la majorité des adresses, même manuscrites, soient reconnues automatiquement sur le courrier postal.

- **Recherche d'image par le contenu**. L'objectif de cette technique est de rechercher, parmi une base de données d'images, les images similaires à une image exemple, ou ayant certaines caractéristiques, par exemple rechercher toutes les images comportant un vélo.

- **Analyse de la vidéo**. L'objectif de cette technique devenue une discipline depuis les années 2000 (lorsque la puissance des processeurs peu onéreux et en particulier des PC a permis des traitements puissants en temps réel) est d'interpréter les faits observés à l'image afin de signaler ou d'enregistrer des faits marquants. Le plus souvent, la caméra est fixe et observe les mouvements d'une scène. Les applications sont nombreuses : Protection des biens (détection d'intrusion, détection d'objet abandonné ou déposé...), Identification (biométrie faciale), Sécurité des personnes (détection de chutes de personnes, franchissement de rambardes, ...), Animations (planchers animés selon les mouvements des danseurs en boîte de nuit), Détection de feux (industriel, forêts, tunnels, ...), Surveillance de tunnels (comptage, mesure de vitesse, détection de fuites/anomalies dans les plafonds), surveillance de tuyaux et autres process industriels.

Exemple de résultats de segmentation avec tous les masques des régions en une seule image.



(a) (b)

(a) : détection de visage par la méthode de segmentation couleur
(b) : Traitement de l'image extraite.

5- L'Image Numérique :

Une image est avant tout un **signal 2D** (x,y), souvent, cette image représente une **réalité 3D** (x,y,z), d'un point de vue **mathématique** :

- Une image est une matrice de nombres, représentant un signal.
- Plusieurs outils permettent de manipuler ce signal.

D'un point de vue **humain** :

- Une image contient plusieurs informations sémantiques. Il faut interpréter le contenu au-delà de la valeur des nombres

Image naturelle → Plusieurs moyens d'acquisition → caméra, microscope, tomographie, infra-rouge, satellite, ...

Image artificielle → Plusieurs outils de représentation → synthèse d'images, réalité virtuelle, visualisation scientifique, ...

Image naturelle

Image artificielle

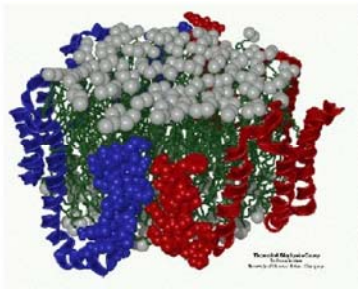


Image artificielle

En réalité il existe, (03) trois principaux types d'images :

Images en niveaux de gris
$I(x,y) \in [0..255]$

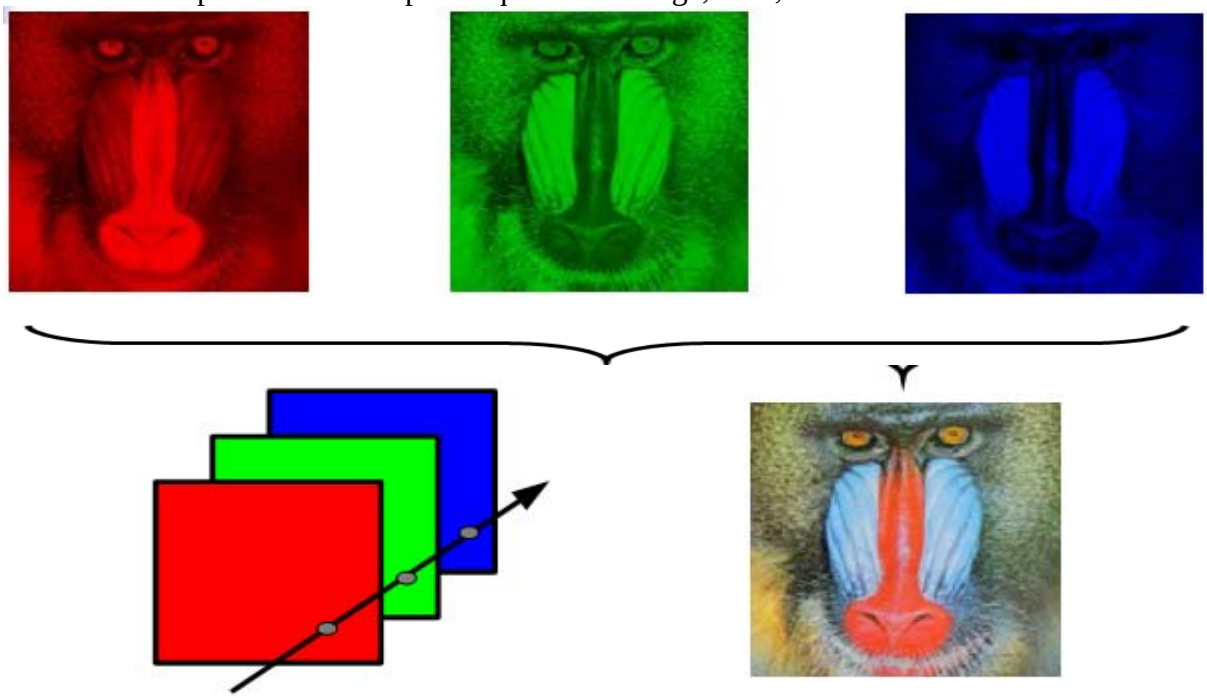
Images binaires
$I(x,y) \in \{0 , 1\}$

Images couleurs
$I_R(x,y)$; $I_G(x,y)$; $I_B(x,y)$

... et plus encore (image 3D, image réelle, ...)

A- Image couleur dans l'espace RVB :

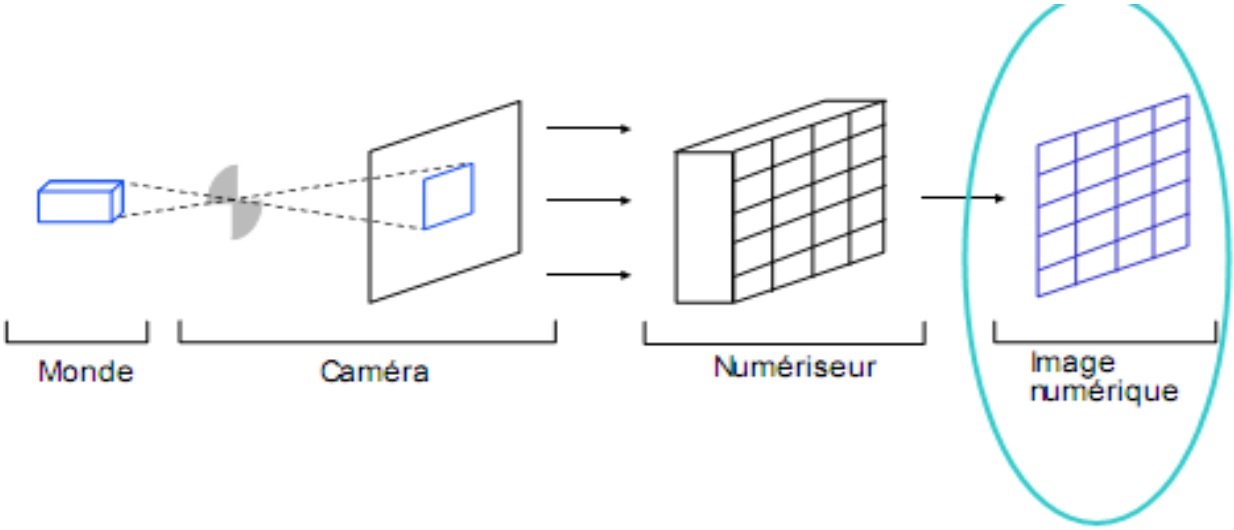
En peut représenter une image couleur numérique par addition de (03) trois couches des couleurs représentatifs du spectre qui sont : Rouge, Vert, Bleu.



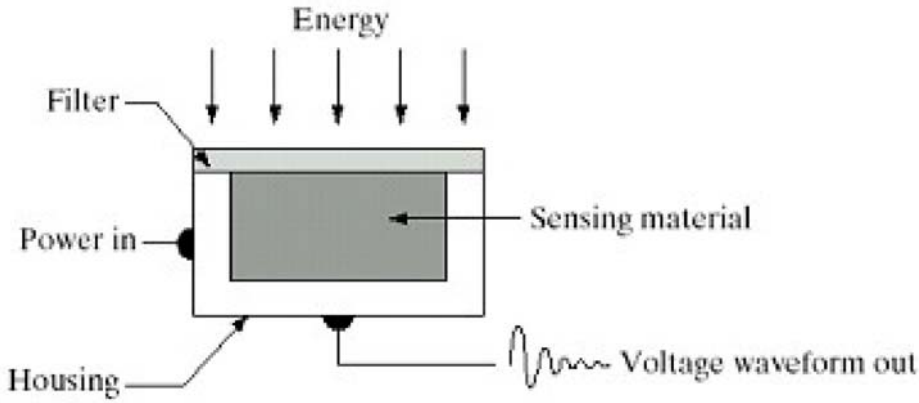
Mais il existe aussi d'autres codages de la couleur que RVB... ex : CMJ : Cyon, Magenta, Jaune

B- Acquisition des Images Numériques :

Le processus général pour l'acquisition d'une image numérique est illustré sur la figure suivante :

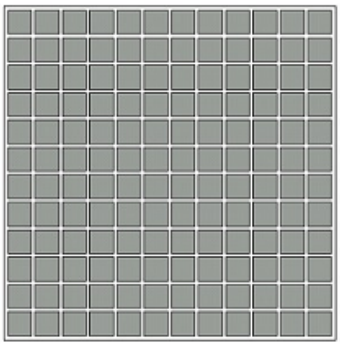


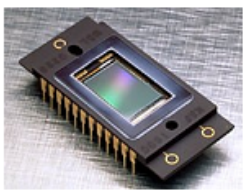
Le Capteur : Le Principe général (ex: photodiode) :



- ✓ L'énergie incidente est convertie en signal électrique.
- ✓ Sortie est proportionnelle à la lumière.
- ✓ Filtre : pour augmenter la sélectivité.

Exemple de Capteur 2D :





KAF-1600 - Kodak.

Capteur CCD : Caméras numériques CCD

- ✓ Matrices CCD (Charged Coupled Devices)
- ✓ Système d'acquisition numérique 2D le plus utilisé
- ✓ La réponse est proportionnelle à l'intégrale de l'énergie lumineuse qui atteint chaque élément
- ✓ Pour la couleur, on utilise trois capteurs par pixel réagissant à des longueurs d'ondes différentes (rouge, vert et bleu)

C- De l'analogique vers le Numérique :

- Les valeurs de *f(x,y)* sont la réponse du capteur au phénomène observé.
- Les valeurs de *f(x,y)* sont des valeurs de « voltage » continu.
- Les valeurs de *f(x,y)* doivent être converties vers le domaine numérique

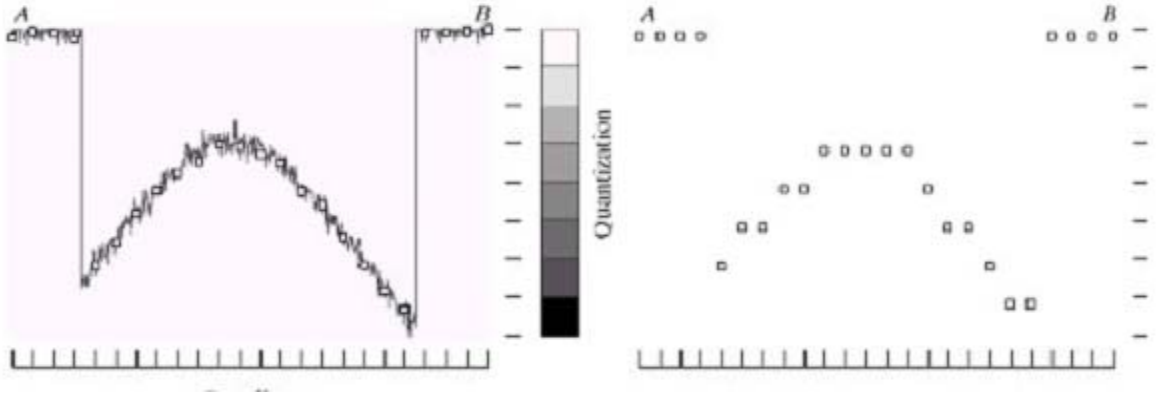
Conversion Analogique/Numérique (A/N)

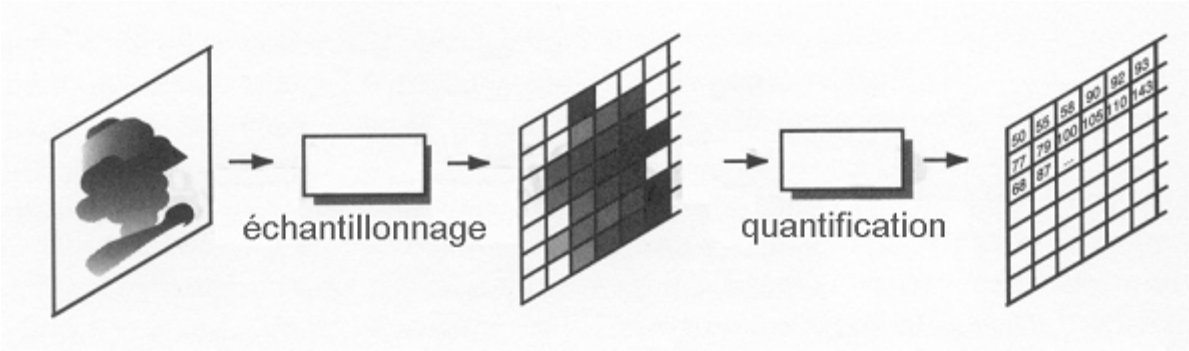
- Deux procédés sont impliqués pour numériser une image :

Numérisation = Échantillonnage + Quantification

D- Echantillonnage et quantification :

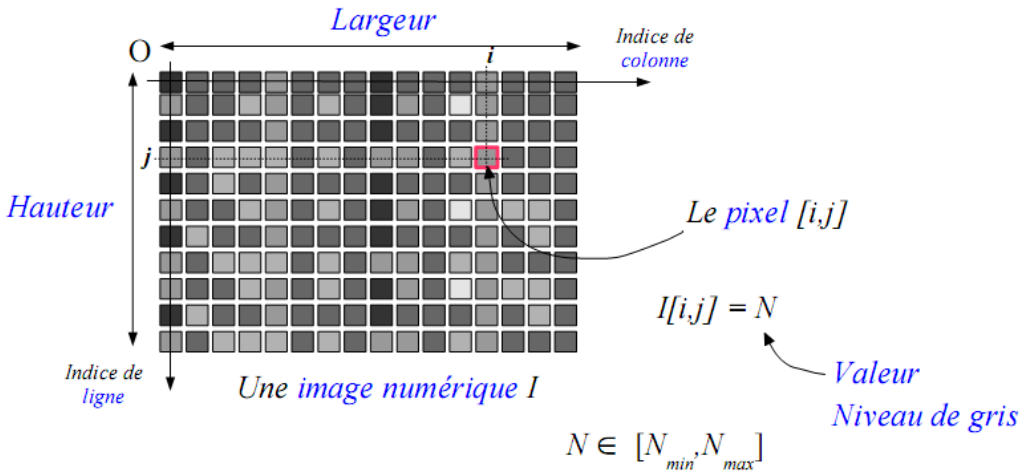
L'échantillonnage est limité par la capacité du capteur, donc le nombre de pixels disponible (ou autre limite imposée), tandis que la quantification est limitée par la quantité de tons (de gris) définie dans l'intervalle.





E- Représentation des images :

1. Une image ***I*** est une Matrice de dimension ***L*** x ***H***
2. Chaque élément à une valeur entière dans l'intervalle [***Nmin , Nmax***].
3. Le nombre de « **bits** » requis pour représenter les niveaux de gris dans l'intervalle « ***N*** » est « ***K*** »
4. La relation entre « ***K*** » et « ***N*** » est : $\mathbf{N = 2^K}$
5. Le nombre de bit pour représenter une image est donc : ***b = L . H . K***



On peut distinguer de mode de résolutions des images :

1. **Résolution spatiale** : est le plus petit détail discernable.
2. **Résolution tonale** (de tons de gris) : est le plus petit changement discernable.

Une image a donc une résolution spatiale de ***LxH*** pixel et une résolution de tons de gris de ***K*** bits ou de ***N*** niveaux ou tons.

Résolution spatiale : échantillonnage

Résolution tonale : quantification

F- Formats de fichiers d'images :

Il existe beaucoup de format de fichiers pour sauvegarder numériquement, les images :

- ✓ TIF, GIF, JPEG, PNG, PPM, PGM, BMP, ...

,et chaque format a ses particularités :

- ✓ Entête contenant les informations de l'image
- ✓ Pixels de l'image codés de différentes façons

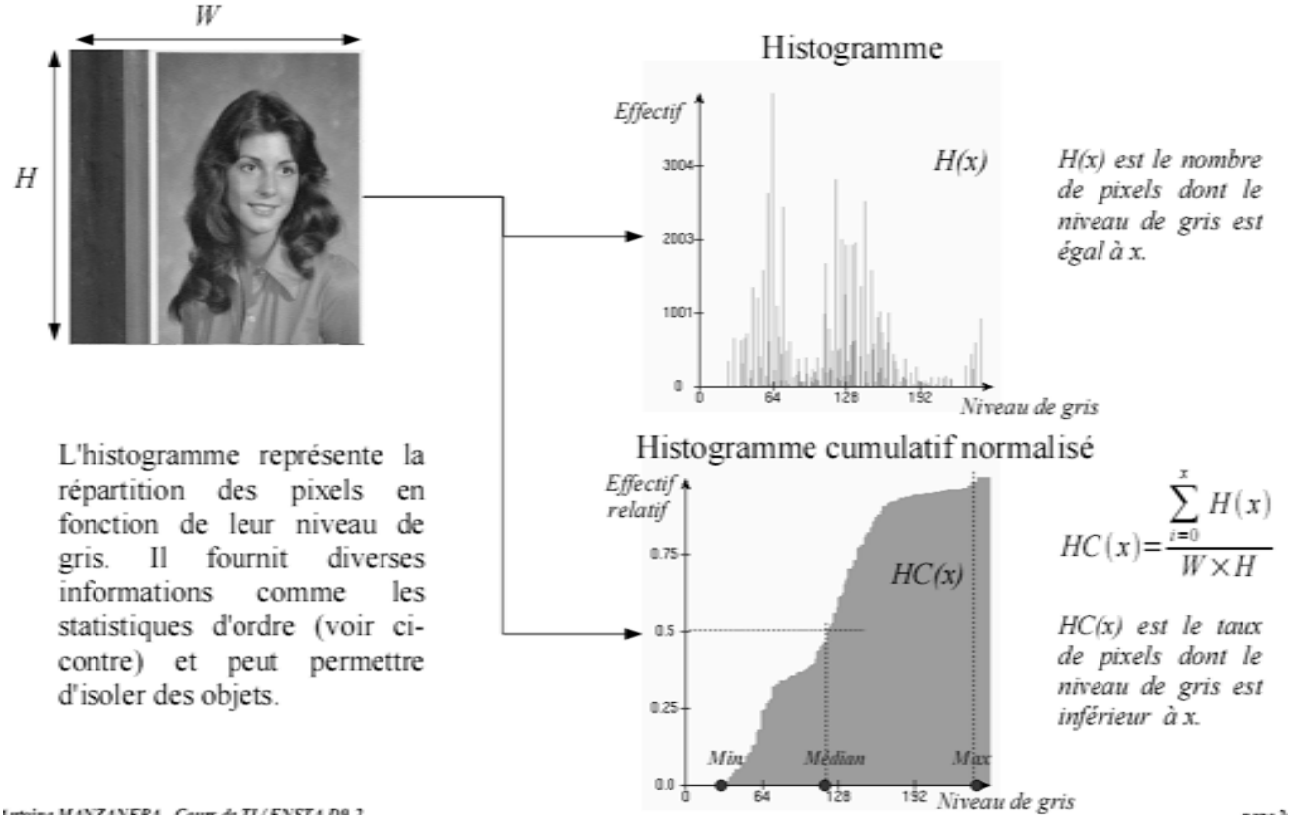
Les formats les plus simples sont les images sans compression d'où les pixels sont codés directement, les uns à la suite des autres. Dans les formats avec compression, les pixels sont compressés pour que le fichier soit plus petit sur le disque.

Compression sans perte : Fichier compressé sans modification de la valeur des pixels Exemple : PNG

Compression avec perte : Valeurs des pixels modifiées pour prendre encore moins de place après la compression. Exemple : JPEG qui est très bien pour visualiser les images (vision humaine) mais très mauvais pour le traitement d'images, car on perd beaucoup d'informations.

6- Outils Fondamentaux :

1. **L'histogramme** : outil de base pour l'étude des capteurs ou de la dynamique d'une scène, il est utilisé par certains opérateurs d'analyse. On retiendra cependant qu'il ne faut pas considérer l'histogramme comme une caractéristique fondamentale de l'image dans la mesure où on peut le transformer radicalement sans changer significativement l'image.



L'histogramme représente la répartition des pixels en fonction de leur niveau de gris. Il fournit diverses informations comme les statistiques d'ordre (voir ci-contre) et peut permettre d'isoler des objets.

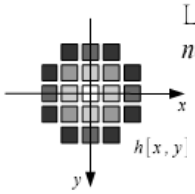
2. **La convolution** : C'est l'opérateur de base du traitement linéaire des images. Apparue très tôt dans les premiers systèmes d'analyse d'images sous forme empirique et justifiée par des considérations d'implantation, ce n'est que plus tard qu'on a fourni des justifications physiques et fait le lien théorique avec les filtres et le traitement du signal.

Soit I une image numérique.
Soit h une fonction de $[x_1,x_2]\times[y_1,y_2]$ à valeurs réelles.

La *convolution* de I par h est définie par :

$$(I*h)[x,y]=\sum_{i=x_1}^{x_2}\sum_{j=y_1}^{y_2} h[i,j]\cdot I[x-i,y-j]$$

La fonction h est dite *noyau de convolution*



Les nouvelles valeurs du pixel sont calculées par *produit scalaire* entre le noyau de convolution et le *voisinage* correspondant du pixel.

Propriétés de la convolution :

COMMUTATIVITÉ $h*g=g*h$

ASSOCIATIVITÉ $(h*g)*k=h*(g*k)=h*g*k$

DISTRIBUTIVITÉ / + $h*(g+k)=(h*g)+(h*k)$

3. **La transformée de Fourier** : Outil fondamental d'analyse en traitement du signal, le pendant (complémentaire) bidimensionnel de la TF et sa version discrète peut être appliqué avec profit aux images numériques. Si son utilisation en tant qu'outil analytique et algorithmique s'est estompée (moins utilisé) en traitement d'images au profit d'approches mieux adaptées à la localisation spatiale des fréquences (ondelettes), elle reste un outil théorique et pédagogique intéressant.

La transformée de Fourier permet la décomposition d'un signal f en *combinaison linéaire de sinusoïdes complexes*, dont les coefficients $F[u,v]$ dit *coefficients de Fourier*, fournissent des informations sur les *fréquences (u,v)* et permettent des manipulations dans le *domaine fréquentiel*.

Transformée de Fourier discrète bidimensionelle :

(x,y) sont les coordonnées du *domaine spatial*

Directe :

$$F[u,v]=\sum_{x=0}^{w-1}\sum_{y=0}^{h-1} f[x,y]\,e^{-2i\pi(ux+vy)/wh}$$

(u,v) sont les coordonnées du *domaine fréquentiel*

Inverse :

$$f[x,y]=\frac{1}{wh}\sum_{u=0}^{w-1}\sum_{v=0}^{h-1} F[u,v]\,e^{2i\pi(ux+vy)/wh}$$

ÉCRITURE SOUS FORME MODULE / PHASE $F[u,v]=\|F[u,v]\|e^{i\varphi[u,v]}$

PÉRIODICITÉ $F[u,v]=F[u+w,v+h]$

$$si \begin{cases} f[x,y] \xrightarrow{TF} F[u,v] \\ f_1[x,y] \xrightarrow{} F_1[u,v] \\ f_2[x,y] \xrightarrow{} F_2[u,v] \end{cases}$$

CORRESPONDANCE CONVOLUTION / PRODUIT

$$f_1[x,y] * f_2[x,y] \rightarrow F_1[u,v] \cdot F_2[u,v]$$
$$f_1[x,y] \cdot f_2[x,y] \rightarrow F_1[u,v] * F_2[u,v]$$

LINEARITÉ

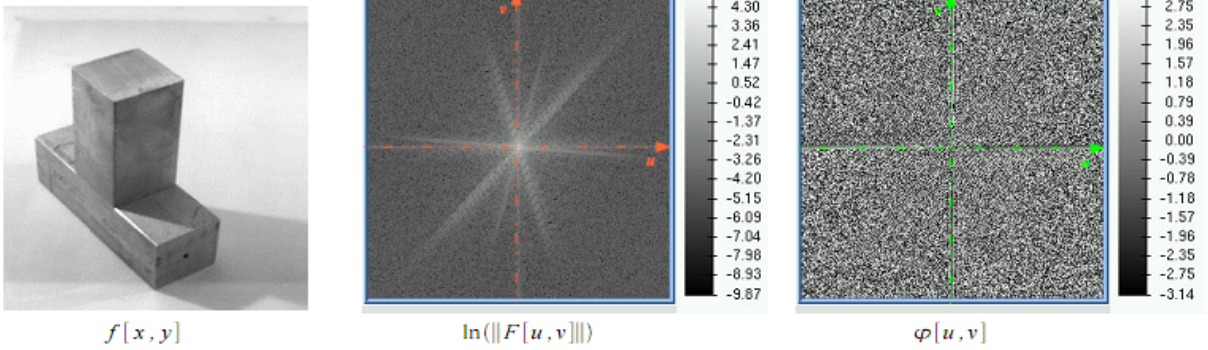
$$a \cdot f_1[x,y] + b \cdot f_2[x,y] \rightarrow a \cdot F_1[u,v] + b \cdot F_2[u,v]$$

TRANSLATIONS SPATIALES / FRÉQUENTIELLES

$$f[x-x',y-y'] \rightarrow F[u,v] \cdot e^{-2i\pi(ux'+vy')/wh}$$
$$f[x,y] \cdot e^{2i\pi(u'x+v'y)/wh} \rightarrow F[u-u',v-v']$$

THÉORÈME DE PARSEVAL

$$\sum_{x=0}^{w-1}\sum_{y=0}^{h-1} \|f[x,y]\|^2 = \frac{1}{wh}\sum_{u=0}^{w-1}\sum_{v=0}^{h-1} \|F[u,v]\|^2$$

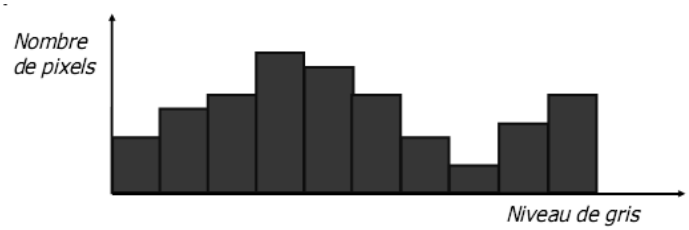


$f[x,y]$ $\ln(\|F[u,v]\|)$ $\varphi[u,v]$

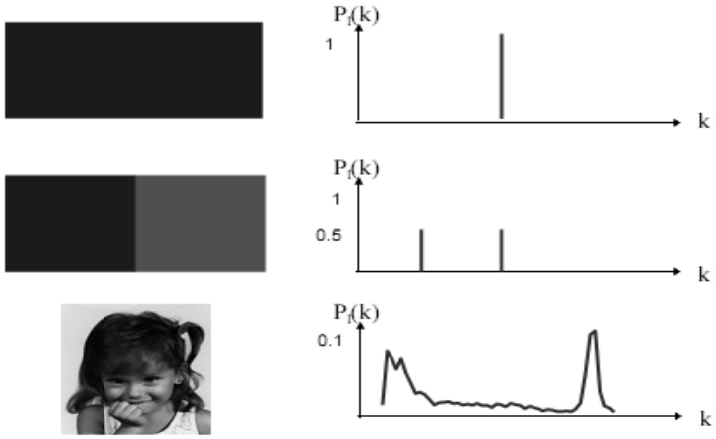
Chapitre 2 : Amélioration et Restauration des Images Numériques

I. Traitement et Analyse Spatiale :

1- Histogramme d'une Image Numérique :



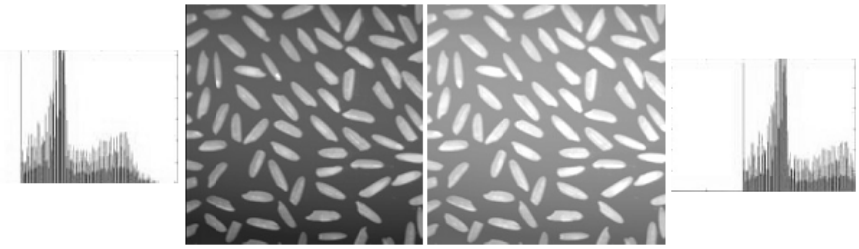
- L'histogramme représente la distribution des niveaux de gris (ou de couleurs) dans une image
- H(k) = nombre de pixels de l'image ayant la valeur k.



Dynamique d'une image = [valeur_min,valeur_max]

A- Luminance ou Brillance d'une Image Numérique :

- La luminance (ou brillance) est définie comme la moyenne de tous les pixels de l'image.
- Dans les deux images suivantes, seule la luminance est différente :



B- Contraste d'une Image Numérique :

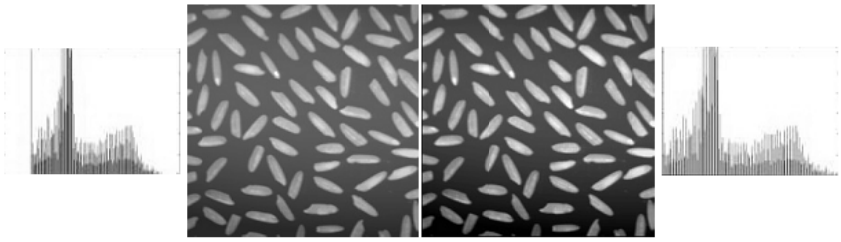
- Le contraste peut être défini de plusieurs façons :
  - Ecart-type des variations des niveaux de gris

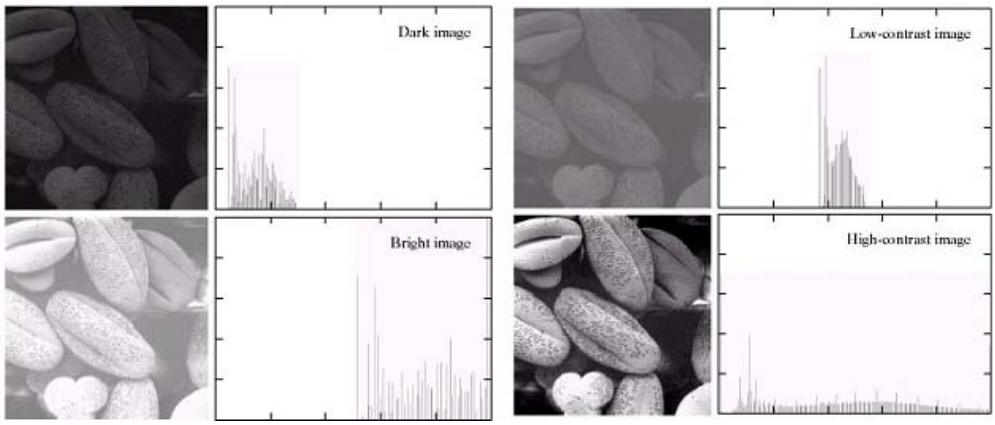
$$C = \sqrt{\frac{1}{M \times N} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (f(x,y) - Moy)^2}$$

  - Variation entre niveaux de gris min et max

$$C = \frac{\max[f(x,y)] - \min[f(x,y)]}{\max[f(x,y)] + \min[f(x,y)]}$$

Les deux images suivantes possèdent un contraste différent :

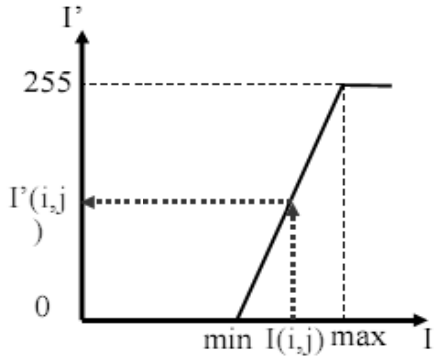




C- Amélioration du Contraste :

- Plusieurs méthodes possibles :
  - Transformation linéaire
  - Transformation linéaire avec saturation
  - Transformation linéaire par morceau
  - Transformation non-linéaire
  - Égalisation de l'histogramme

D-Transformation Linéaire (Approche Histogramme) :

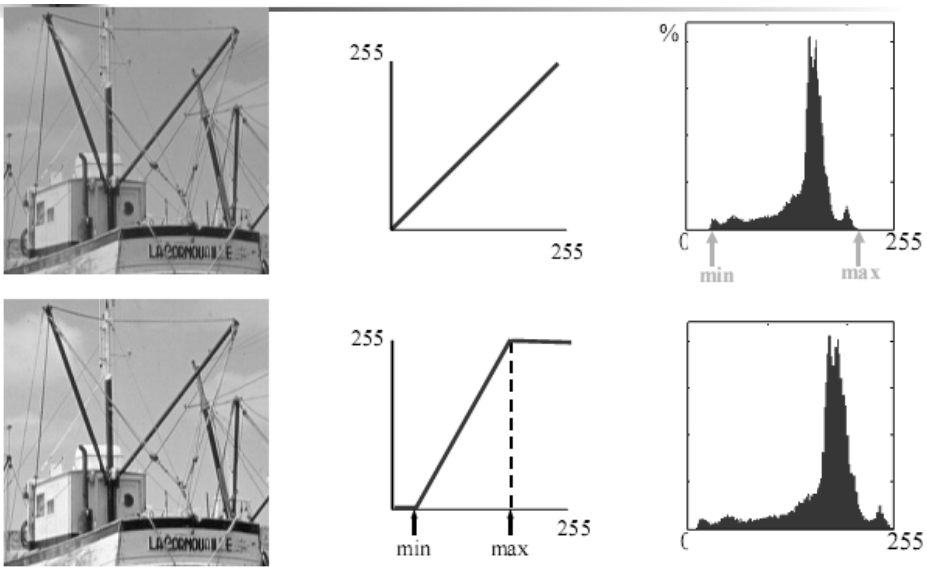


$$\frac{\max - \min}{I(i,j) - \min} = \frac{255 - 0}{I'(i,j) - 0}$$

Alors :

$$I'(i,j) = \frac{255}{\max - \min}\left(I(i,j) - \min\right)$$

$$I'(i,i) = \frac{255}{\max - \min}\left(I(i,j) - \min\right) \quad avec \frac{(I(i,j) - \min)}{\max - \min} \in [0,1]$$



Implémentation :

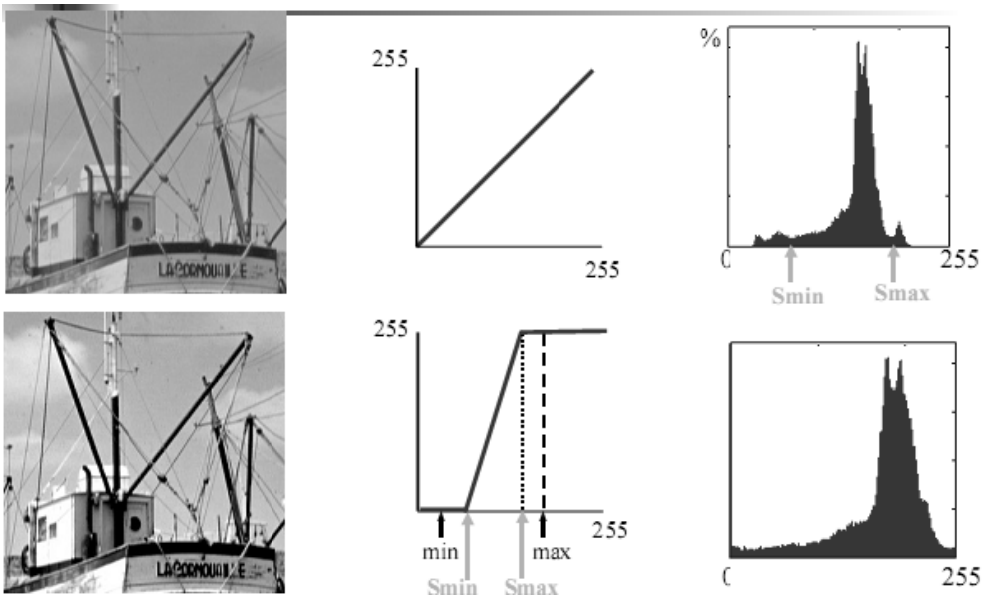
```
Pour i=1 à nblig
   Pour j=1 à nbcol
      I'(i,j) = 255*(I(i,j)-minI)
                /(maxI-minI);
```

Pas optimal

➔ **Utilisation d'une LUT (Look Up Table)**

```
/* Initialisation de la LUT */
Pour i=0 à 255
    LUT[i]=255*(i-minI)/(maxI-minI);

/* Initialisation de la LUT */
Pour i=1 à nblig
    Pour j=1 à nbcol
        I'(i,j) =LUT[I(i,j)];
```



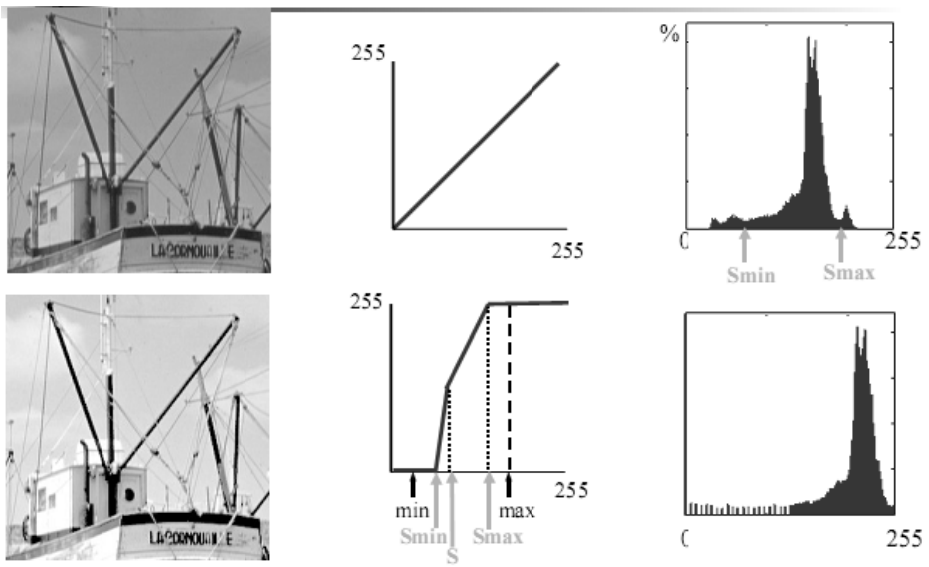
$$I'(i,j) = \frac{255}{S_{\max} - S_{\min}} \left(I(i,j) - S_{\min}\right)$$

$$I'(i,j) \leq 0 \Rightarrow I'(i,j) = 0$$

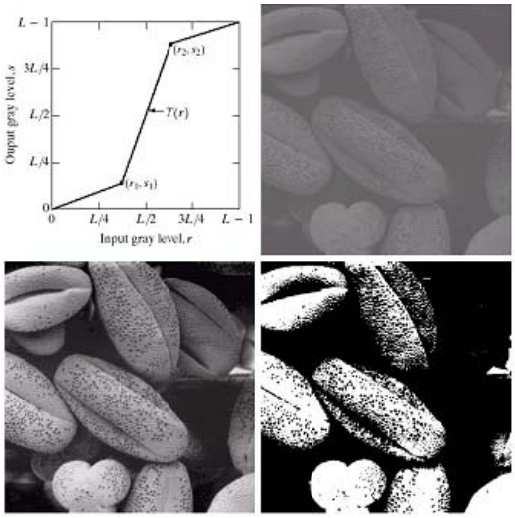
$$I'(i,j) \geq 255 \Rightarrow I'(i,j) = 255$$

$$\min(I(i,j)) \leq S_{\min} < S_{\max} \leq \max(I(i,j))$$

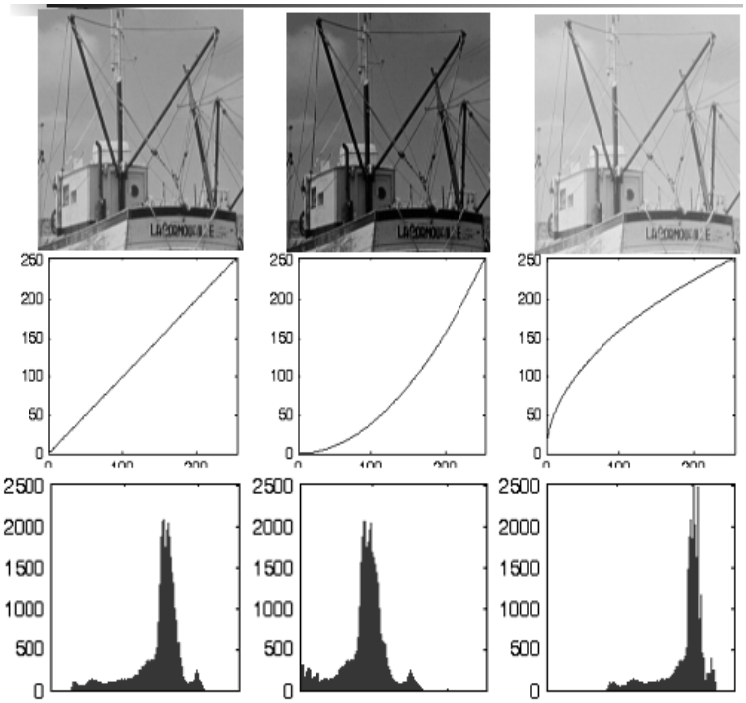
E- Transformation Linéaire par Morceaux:



Exemple d'une fonction ad-hoc de modification du contraste de l'image.



F- Transformation Non Linéaire (Approche Histogramme) :



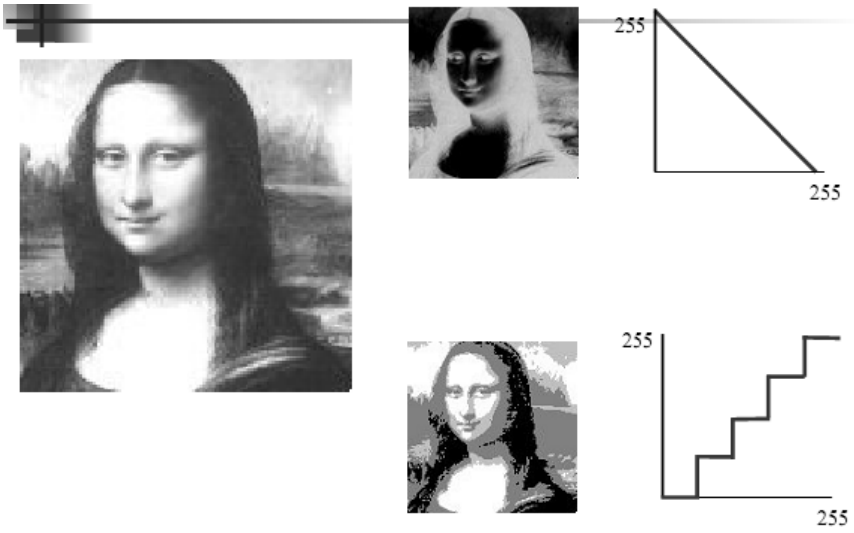
logarithmique : $g(x,y) = c.log(1 + f(x,y)), c > 0$

exponentielle gamma :

$g(x,y) = c.f(x,y)^{\gamma}, c > 0$

- $\gamma < 1$: étire les faibles valeurs, compresse les valeurs foncées
- $\gamma < 1$: compresse les faibles valeurs, étire les valeurs foncées

G-Autres Fonctions Possibles (Inversion, Mis à l'échelle):



H-Correction de la Dynamique de l'Image :

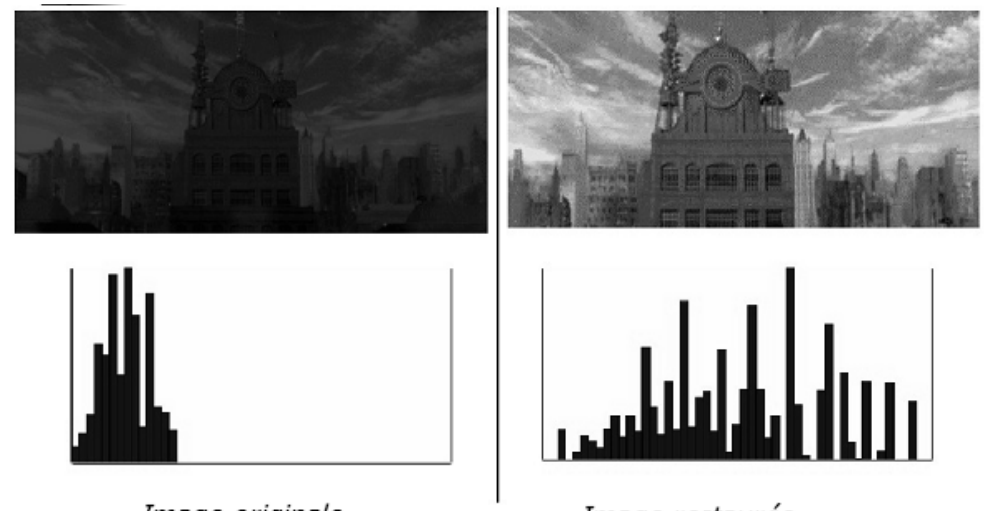
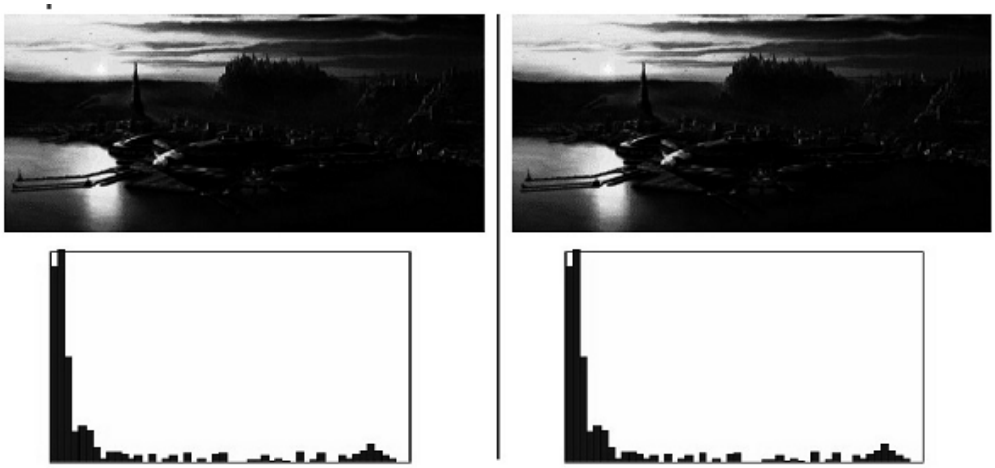


Image originale *Image restaurée*



Dans le cas où l'histogramme initial occupe toute la plage de dynamique, aucun changement n'est visible.

I- Egalisation de l'Histogramme :

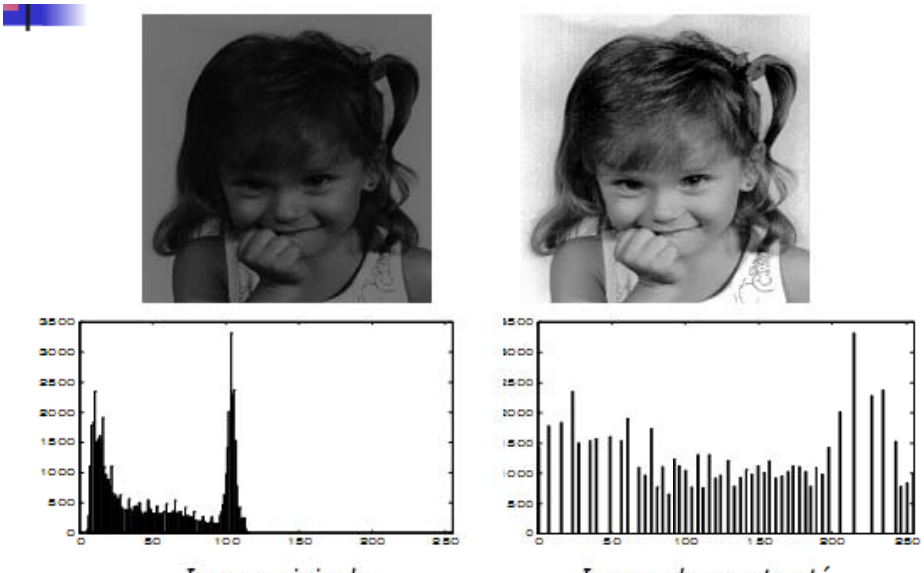
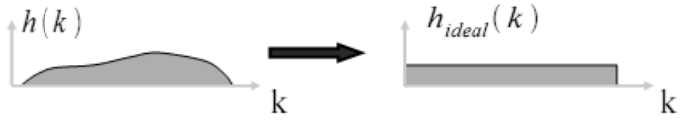


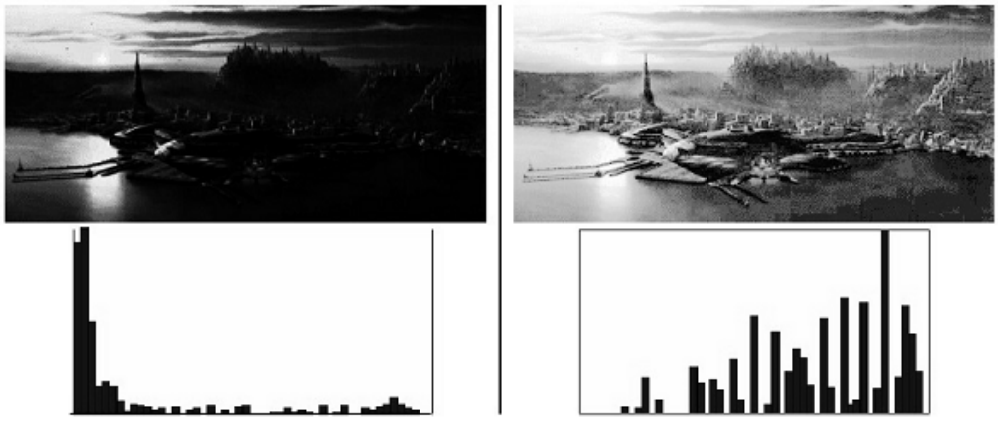
Image originale *Image plus contrastée*

- Pour améliorer le contraste, on cherche à aplanir l'histogramme



- Etape 1 : Calcul de l'histogramme $h(i) \quad i \in [\ 0,\ 255]$
- Etape 2 : Normalisation de l'histogramme (Nbp : nombre de pixels de l'image) $h_n(i) = \frac{h(i)}{Nbp} \quad i \in [\ 0,\ 255]$
- Etape 3 : Densité de probabilité normalisé $C(i) = \sum_{j=0}^{i} h_n(j) \quad i \in [\ 0,\ 255]$
- Etape 4 : Transformation des niveaux de gris de l'image

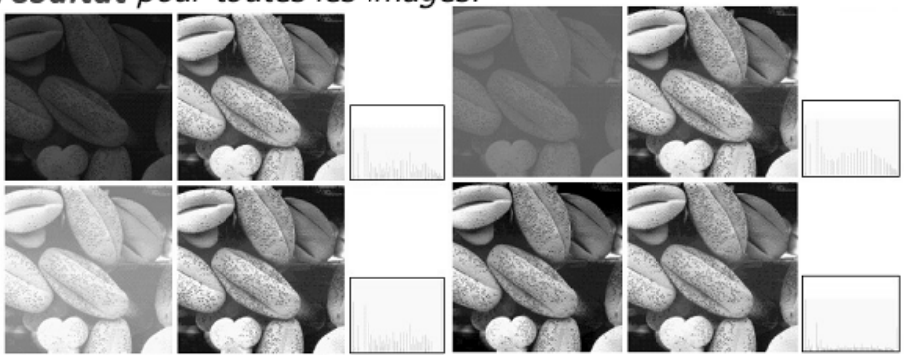
$$f'(x,y) = C(f(x,y)) \times 255$$



L'égalisation d'histogramme peut améliorer une image là où la correction de dynamique de l'histogramme est inefficace.

Remarque Importante :

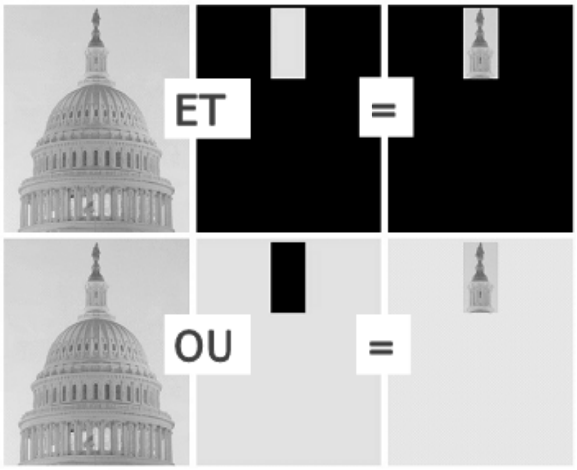
*Si on prend la **même image** avec des **contrastes différents**, l'égalisation d'histogramme donne le **même résultat** pour toutes les images.*

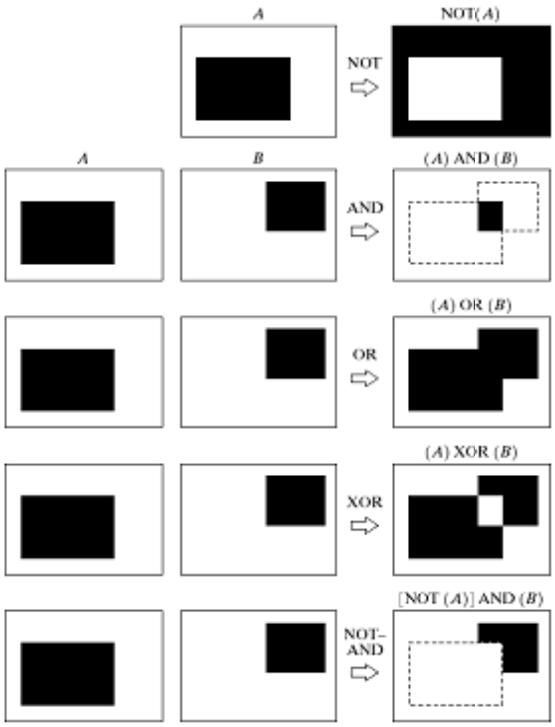


2- Opérations sur les Images Numérique :

A- Opérations Logiques sur l'Images Numérique :

Les opérations logiques fonctionnent aussi sur les images.





B- Addition d'Images Numérique :

- Si f et g sont deux images, on peut définir l'addition R pixel à pixel de ces deux images par :

  $$R(x,y) = Min(f(x,y)+g(x,y) ; 255)$$

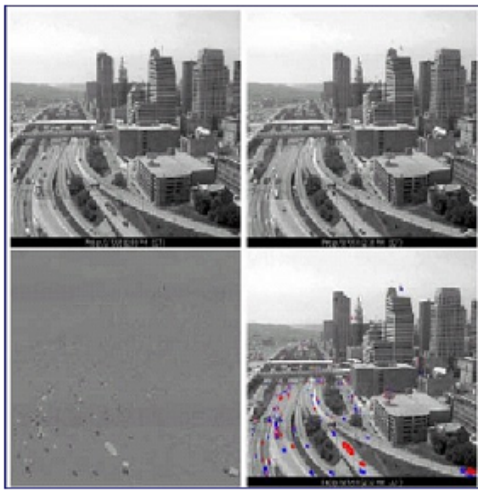
- L'addition d'images peut permettre
  - De diminuer le bruit d'une vue dans une série d'images
  - D'augmenter la luminance en additionnant une image avec elle-même

C- Soustraction d'Images Numériques :

- On peut définir la soustraction S pixel à pixel de deux images f et g par :

  $$S(x,y) = Max(f(x,y)-g(x,y) ; 0)$$

- La soustraction d'images peut permettre
  - Détection de défauts
  - Détection de mouvements



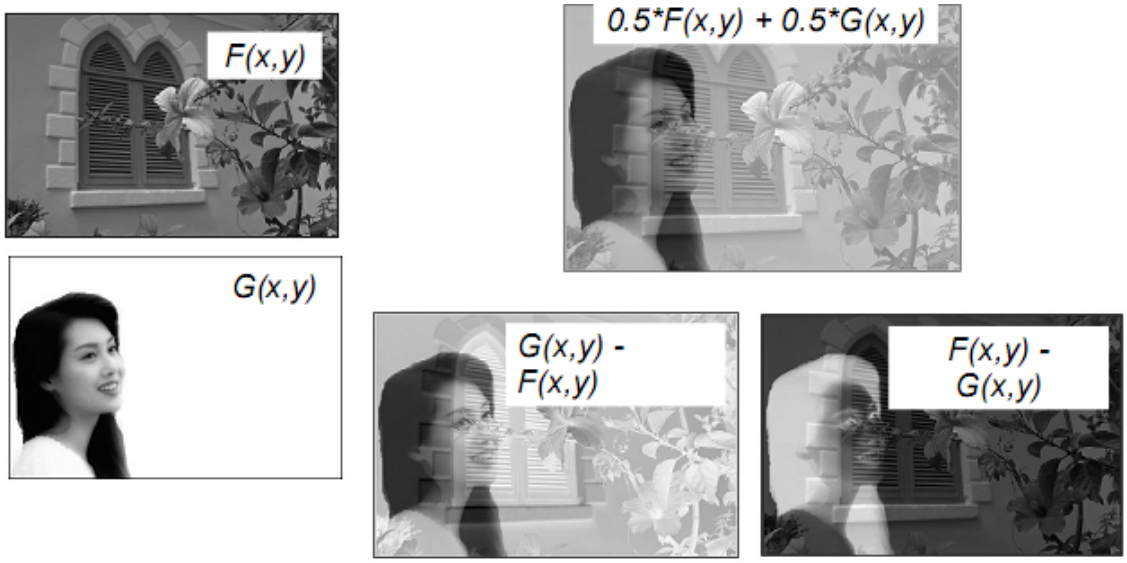
D- Multiplication d'Images Numériques :

- La multiplication S d'une image f par un ratio (facteur) peut se définir par :

$$S(x,y) = Max(f(x,y)*ratio ; 255)$$

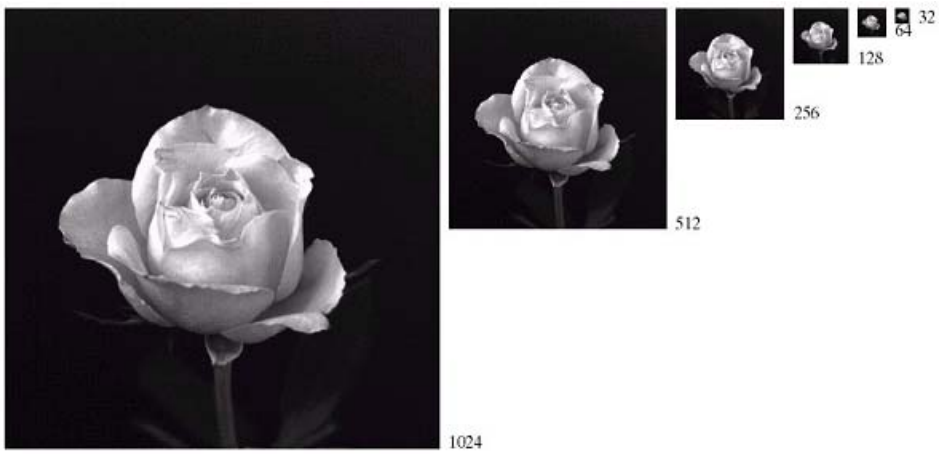
- La multiplication d'images peut permettre d'améliorer le contraste ou la luminosité

Quelques Exemples :



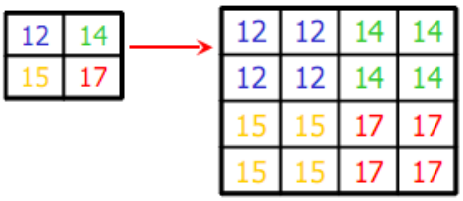
3- Interpolation d'Images Numériques :

Changement d'échelle :



A- Interpolation par plus proche voisin :

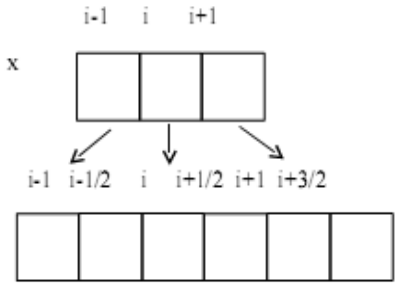
- Interpolation du plus proche voisin par copie des pixels
  - Copie de chaque colonne et de chaque rang



B- Interpolation Bilinéaire :

- Zoom !
- Interpolation bilinéaire
- Fonction bilinéaire de 4 pixels voisins (en 2D)

Principe 1D



$y_i = x_i$

$y_{i+1/2} = (x_i + xi_{+1}) /2$

Problème pour le dernier point

$\Rightarrow$ extrapolation linéaire de :

$=2*x_N - x_{N-1}$

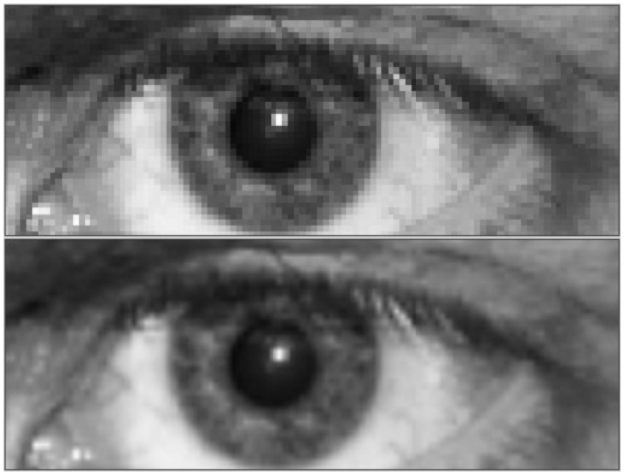
Exemples d'Interpolation :

Exemple de Changement d'Echelle :

L'image originale à petit échelle :

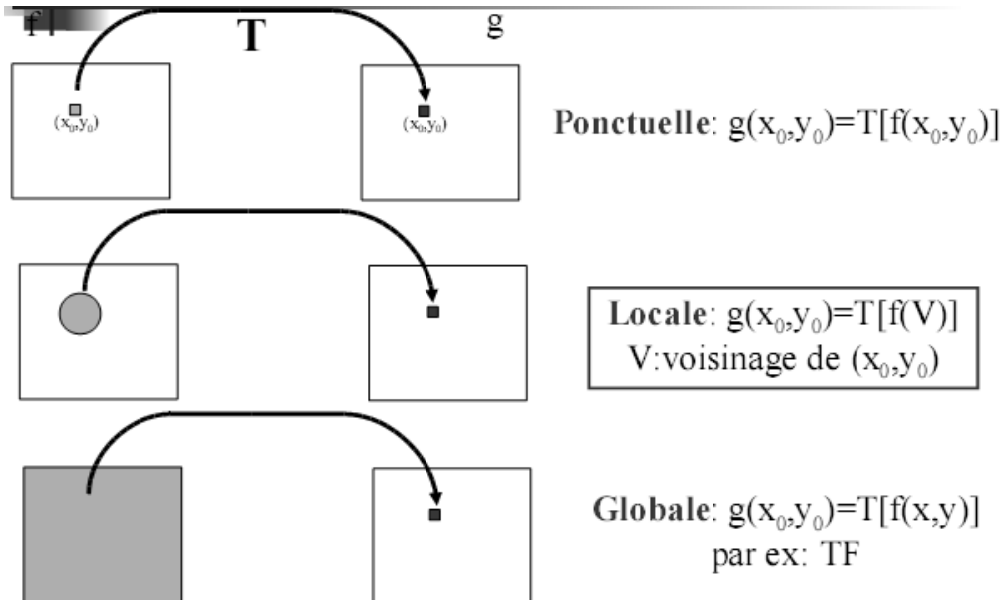
Plus proche voisin

Bilinéaire (4 voisins)



4- Opération de Convolution Spatiale :

- Pour l'instant, nous avons vu surtout des **transformations ponctuelles** des pixels d'une image
  - *Lire la valeur d'un pixel → la remplacer par une autre*
- Il existe aussi des **transformations locales**
  - *Lire la valeur de quelques pixels voisins → calculer une nouvelle valeur pour un pixel*
- …et des **transformations globales**
  - *Lire la valeur de tous les pixels de l'image → calculer une nouvelle valeur pour un seul pixel*



- La convolution discrète est un outil permettant l'utilisation de **filtres linéaires** ou de **filtres de déplacements invariants**

- L 'équation générale de la convolution, notée g(x), de la fonction d'origine f(x) avec une fonction h(x) est :

$$g(x) = f(x) * h(x) = \sum_{\forall k} h(x-k) f(k)$$

- f(x) est la fonction d'origine et g(x) la fonction convoluée (résultat de la convolution)
  - Dans notre cas, une image est vue comme une fonction mathématique

- h(x) est appelé *masque de convolution*, *noyau de convolution*, *filtre*, *fenêtre*, *kernel*, …

Exemples :

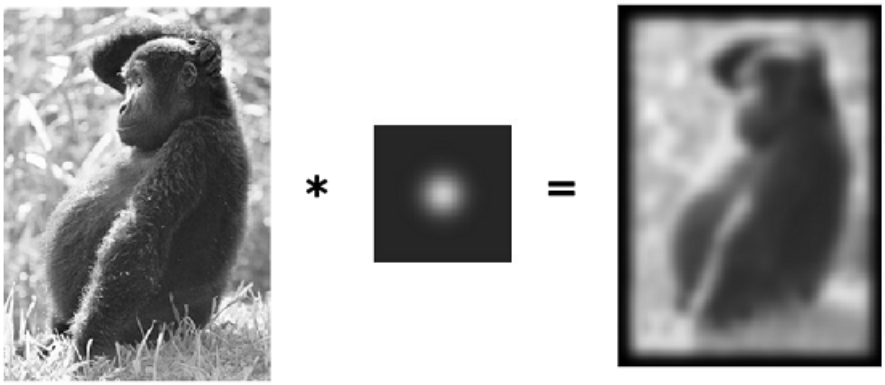


Image d'origine | Filtre de convolution (masque) | Image convoluée (résultat)

Note : par convention pratique, la taille de l'image résultat est la même que celle de l'image d'origine

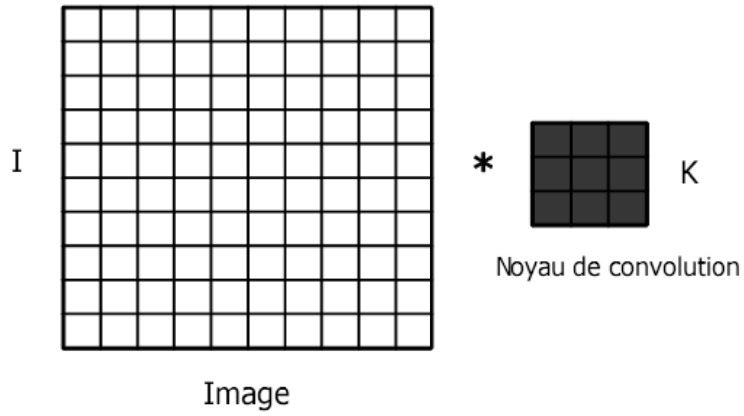
- En pratique, la convolution numérique d'une image se fera par une **sommation de multiplications**

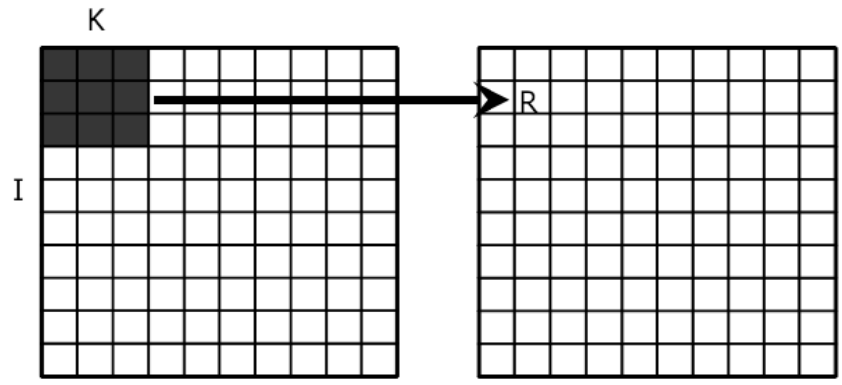
- Un filtre de convolution est une matrice (image) généralement *(mais pas toujours)* de taille impaire et symétrique
  - 3x3, 5x5, 7x7, …

Convolution d'une image par un filtre 2D :

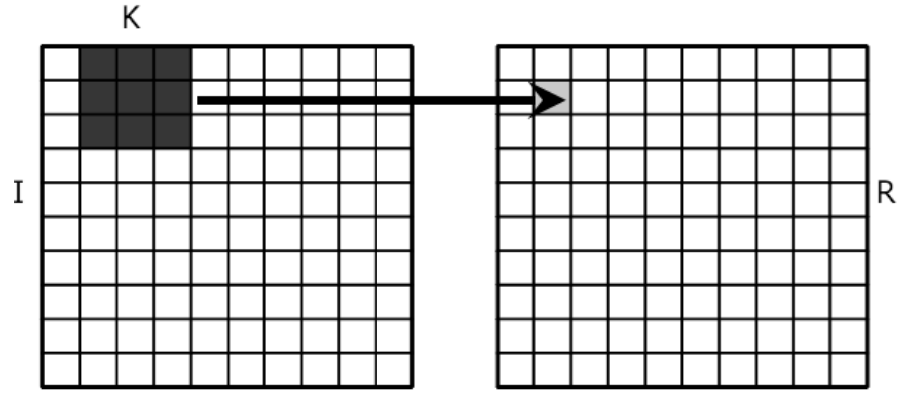
$$I'(i,j) = I(i,j) * filtre(i,j)$$

$$I'(i,j) = \sum_{u} \sum_{v} I(i-u, j-v) \cdot filtre(u,v)$$

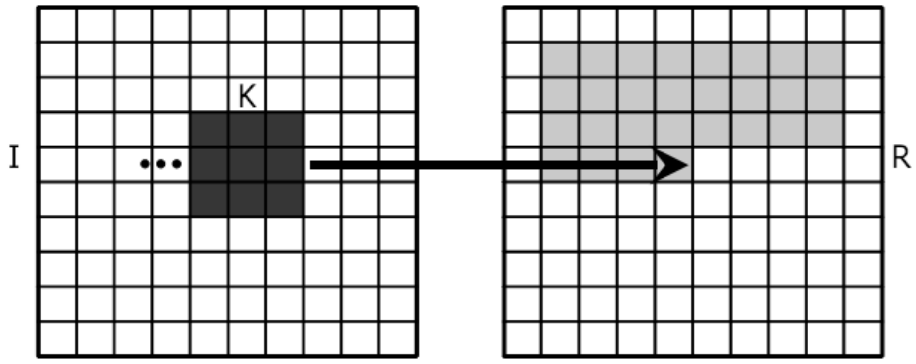




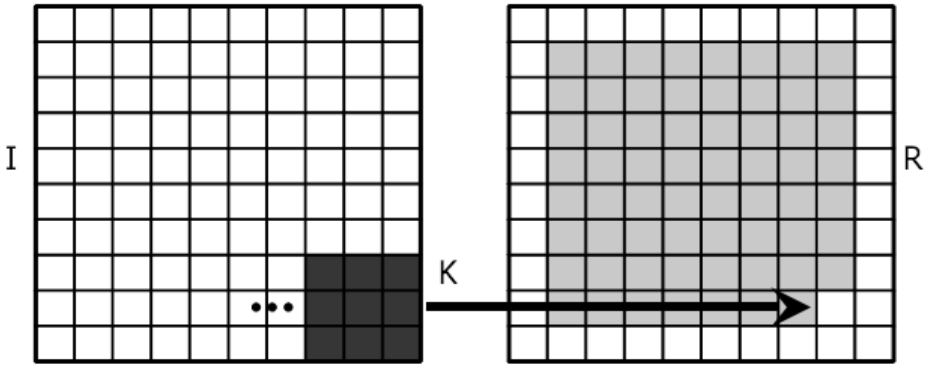
$$R(1,1) = I(0,0)\,K(0,0) + I(1,0)\,K(1,0) + I(2,0)\,K(2,0) + I(0,1)\,K(0,1) + I(1,1)\,K(1,1) + I(2,1)\,K(2,1) + I(0,2)\,K(0,2) + I(1,2)\,K(1,2) + I(2,2)\,K(2,2)$$



$$R(2,1) = I(1,0)\,K(0,0) + I(2,0)\,K(1,0) + I(3,0)\,K(2,0) + I(1,1)\,K(0,1) + I(2,1)\,K(1,1) + I(3,1)\,K(2,1) + I(1,2)\,K(0,2) + I(2,2)\,K(1,2) + I(3,2)\,K(2,2)$$

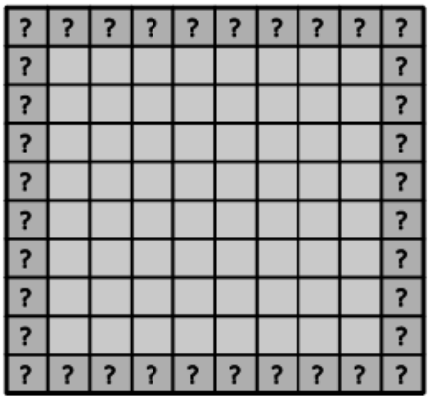


$$R(x,y) = I(x-1,y-1)\,K(0,0) + I(x,y-1)\,K(1,0) + I(x+1,y-1)\,K(2,0) + I(x-1,y)\,K(0,1) + I(x,y)\,K(1,1) + I(x+1,y)\,K(2,1) + I(x-1,y+1)\,K(0,2) + I(x,y+1)\,K(1,2) + I(x+1,y+1)\,K(2,2)$$



R(N-2,M-2) = I(N-3,M-3) K(0,0) + I(N-2,M-3) K(0,1) + I(N-1,M-3) K(0,3)
+ I(N-3,M-2) K(1,0) + I(N-2,M-2) K(1,1) + I(N-1,M-2) K(1,2)
+ I(N-3,M-1) K(2,0) + I(N-2,M-1)K(2,1) + I(N-1,M-1) K(2,2)

- Problème : Que faire avec les bords de l'image ?
  - Mettre à zéro (0)
  - Convolution partielle
    - Sur une portion du noyau
  - Miroir de l'image
    - f(-x,y) = f(x,y)
  - … (pas de solution miracle)



Le Masque de Convolution :

- Le masque de convolution représente un filtre linéaire permettant de modifier l'image
- On divisera le résultat de la convolution par la somme des coefficients du masque
  - Pour éviter de modifier la luminance globale de l'image, la somme des coefficients doit être égale à 1

En réalité il existe deux types de filtrage Spatiale;

- **Filtres passe-bas**
  - Atténue le bruit et les détails (basses fréquences)
    ➔ lissage

- **Filtres passe-haut**
  - Accentue les détails et les contours (hautes fréquences)
    ➔ accentuation

A- Le Filtre Moyenneur :

- Le filtre moyenneur
  - Permet de lisser l'image *(smoothing)*
  - Remplace chaque pixel par la valeur moyenne de ses voisins
  - Réduit le bruit
  - Réduit les détails non-important
  - Brouille ou rend floue l'image *(blur edges)*
- Filtre dont tous les coefficients sont égaux
- Exemple de filtres moyenneurs :

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

ou 1/9

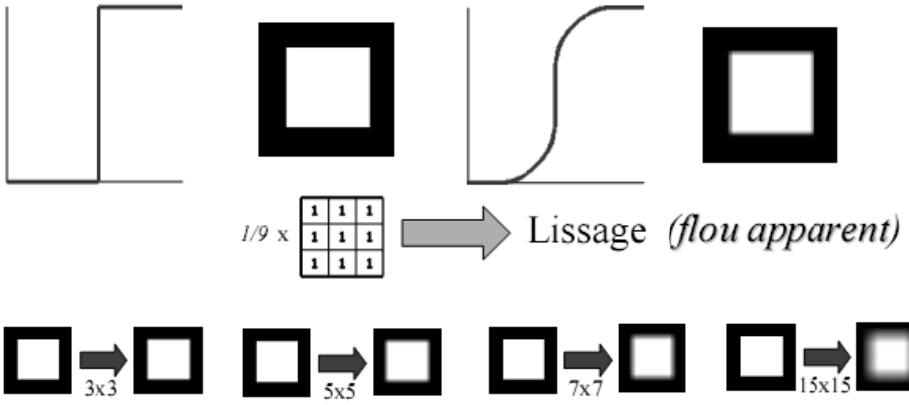
1	1	1
1	1	1
1	1	1

3x3

ools : Utilities>Filter>Mean

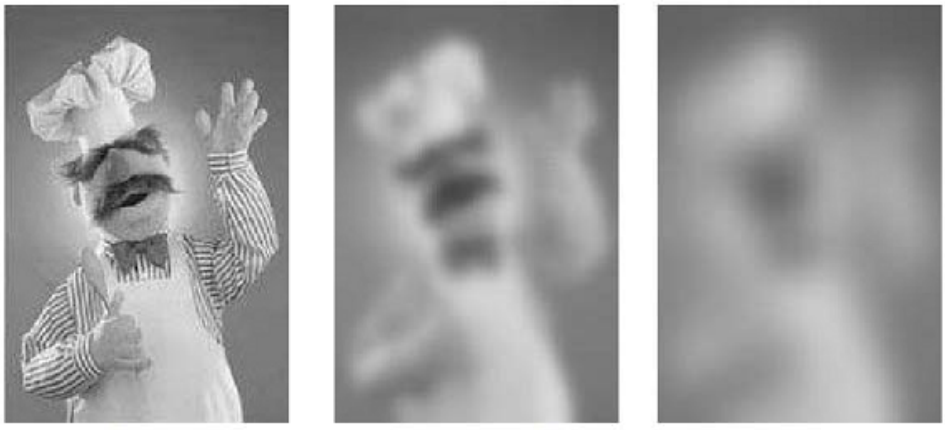
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

5x5



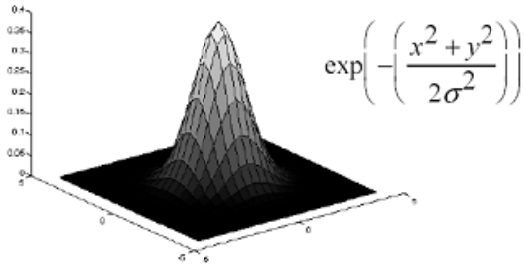
Plus le filtre grossit , plus le lissage devient important et plus le ***flou s'accentue*** *!*

Exemple :



Original | Moyenne 5x5 | Moyenne 11x11

B- Le Filtre Gaussien :



$$\exp\left(-\left(\frac{x^2+y^2}{2\sigma^2}\right)\right)$$

Fonction gaussienne en 3D

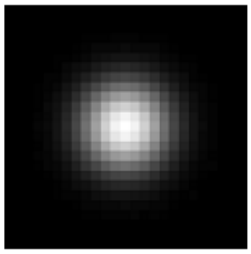


Image d'une gaussienne

Le filtre gaussien donnera un meilleure lissage et une meilleure réduction du bruit que le filtre moyenne

$$\frac{1}{98}\times\begin{bmatrix}1 & 2 & 3 & 2 & 1\\ 2 & 6 & 8 & 6 & 2\\ 3 & 8 & 10 & 8 & 3\\ 2 & 6 & 8 & 6 & 2\\ 1 & 2 & 3 & 2 & 1\end{bmatrix}$$

Exemple :

Original | Gauss 5x5 | Gauss 11x11

C- Le Filtre Médian (Non-Linéaire) :

- Pour nettoyer le bruit dans une image, il existe mieux que le filtre moyenneur ou le filtre gaussien
- Il s'agit du filtre médian
- C'est un filtre non-linéaire, qui ne peut pas s'implémenter comme un produit de convolution
- On remplace la valeur d'un pixel par la valeur médiane dans son voisinage NxN

30	10	20
10	250	20
25	10	30

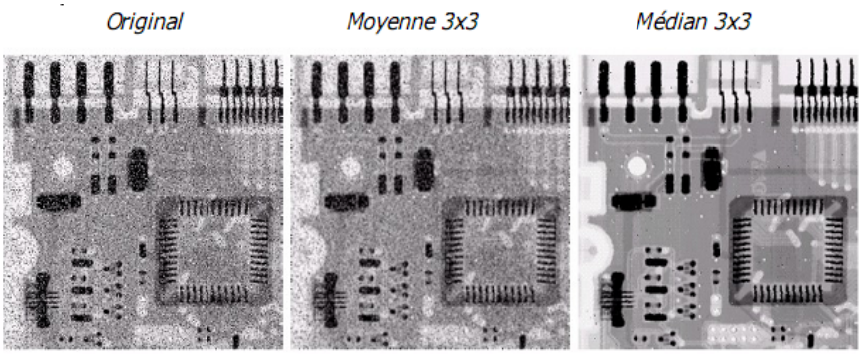
Moyenne = 45

bruit

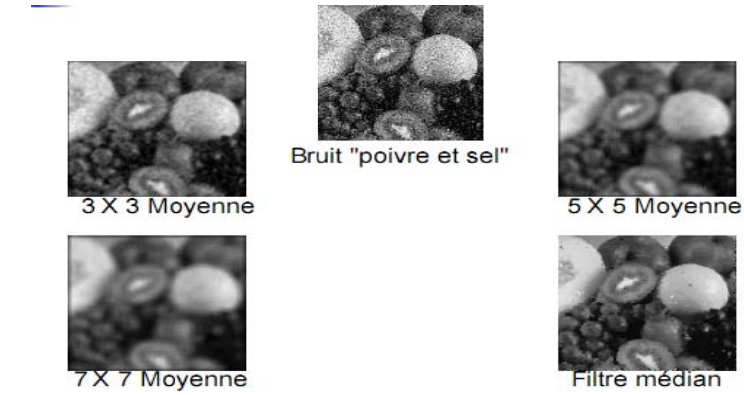
10, 10, 10, 20, **20**, 25, 30, 30, 250

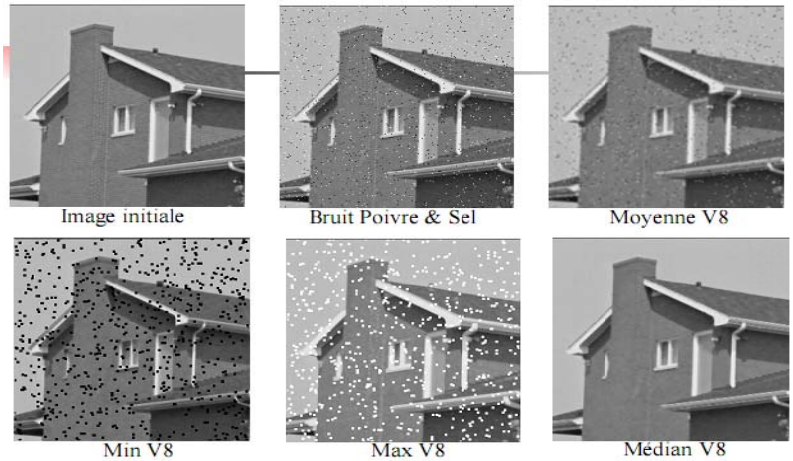
médiane

Exemple :

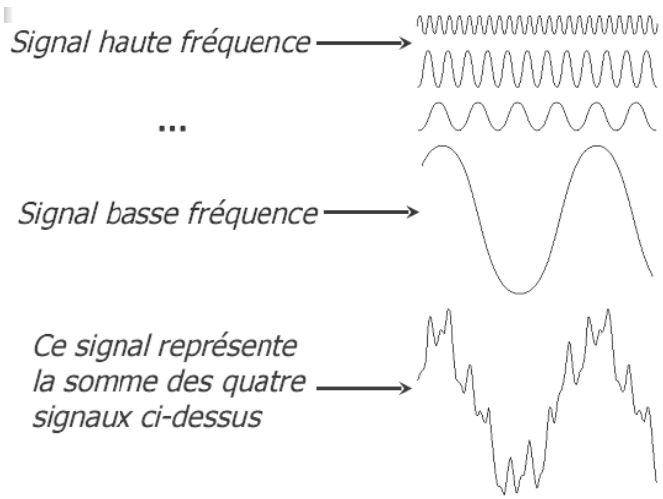


Exemples de Nettoyage du Bruit de type Additif :

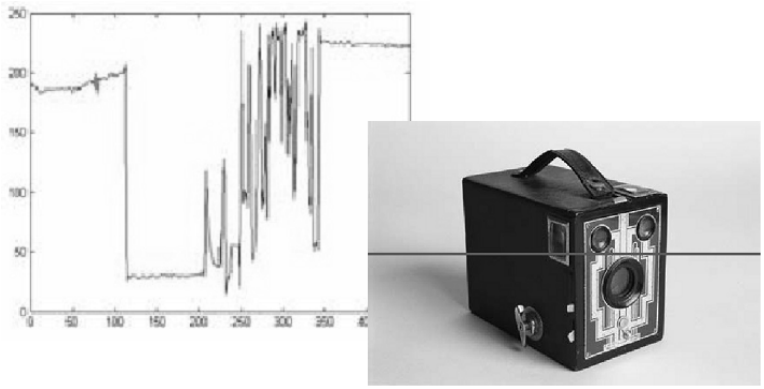




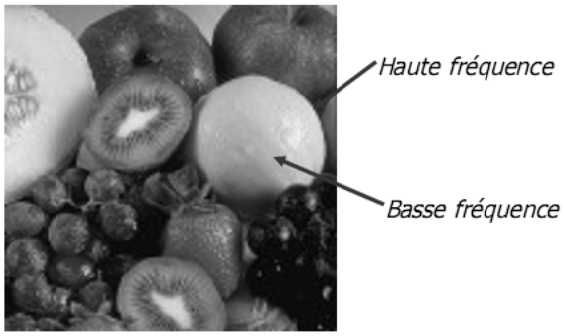
II. Traitement et Analyse Fréquentiel :



Les Fréquences dans L'image Numérique :

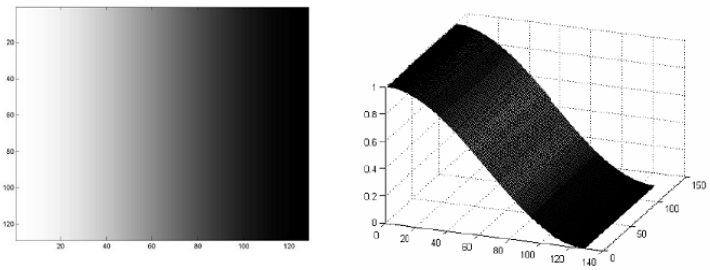


- Qu'est-ce qu'une fréquence dans une image ?
  - Fréquence = changement d'intensité
  - Basses fréquences : régions homogènes, flou
  - Hautes fréquences : contours, changement brusque d'intensité, bruit



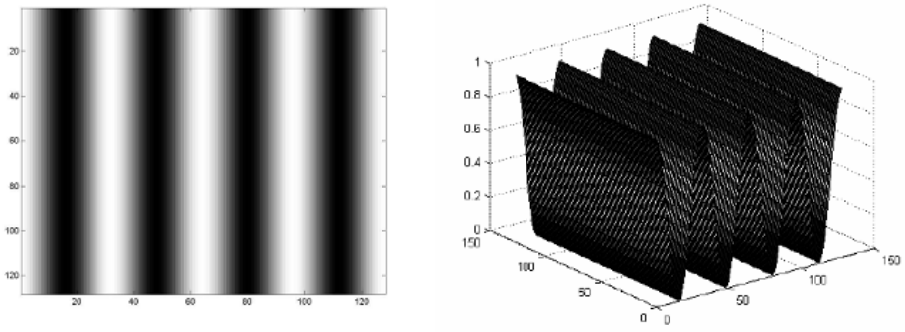
La plus grande partie de l'énergie d'une image se situe dans les basses fréquences.

Les Basses Fréquences :



Les basses fréquences correspondent à des changements d'intensité lents

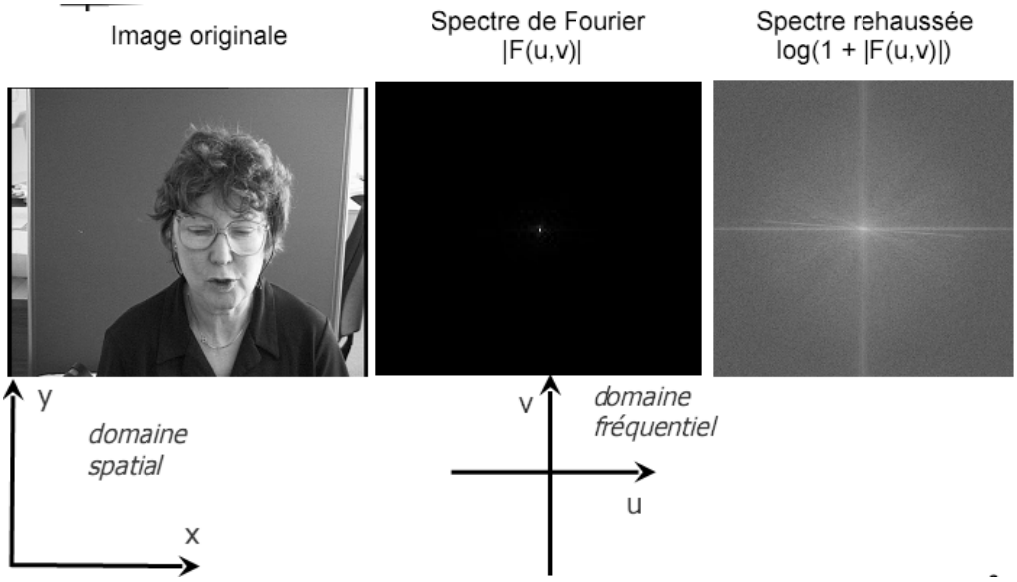
Les Hautes Fréquences :



Les hautes fréquences correspondent à des changements d'intensité rapides

A- Analyse Spectrale d'une Image :

- Une image est avant tout un signal (visuel)
  - *Tout comme le son est un signal audio*
- On peut analyser les fréquences de ce signal
- Pour cela, on crée un nouvel « histogramme » qui va représenter les fréquences de l'image
  - L'outil de base pour cela est la **Transformée de Fourier**
- On parle de **domaine fréquentiel**, par opposition au **domaine spatial** (de l'image)



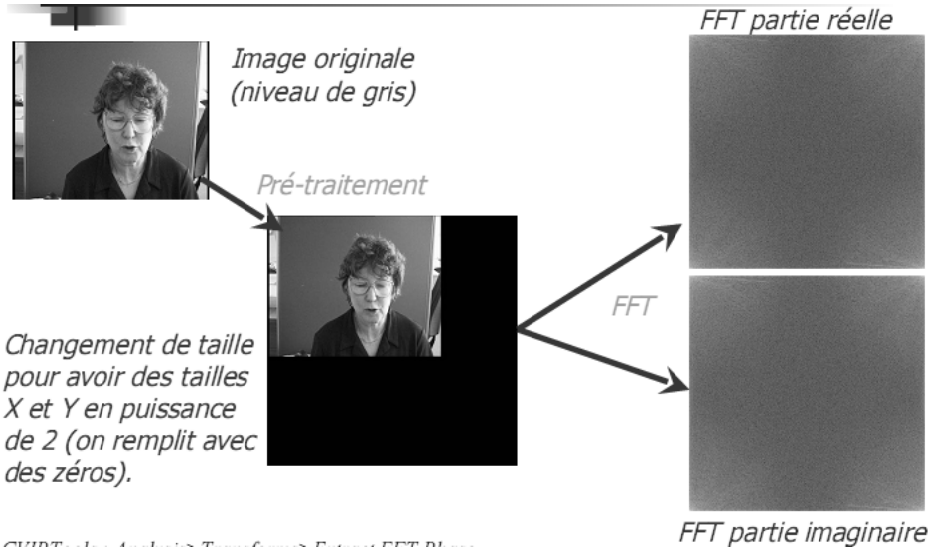
- **Hautes fréquences** : loin du centre de la TF
- **Basses fréquences** : proche du centre de la TF
- **Composante continue** : centre de l'image
  - *fréquence zéro = moyenne de l'image*
- La Transformée de Fourier d'une fonction réelle donne une fonction complexe
  - **Entrée** : une image
  - **Sortie** : image(partie réelle) + image(partie imaginaire)
- Rappel sur la définition d'un nombre complexe :

$$z = x + iy \quad \text{où} \quad i = \sqrt{-1}$$

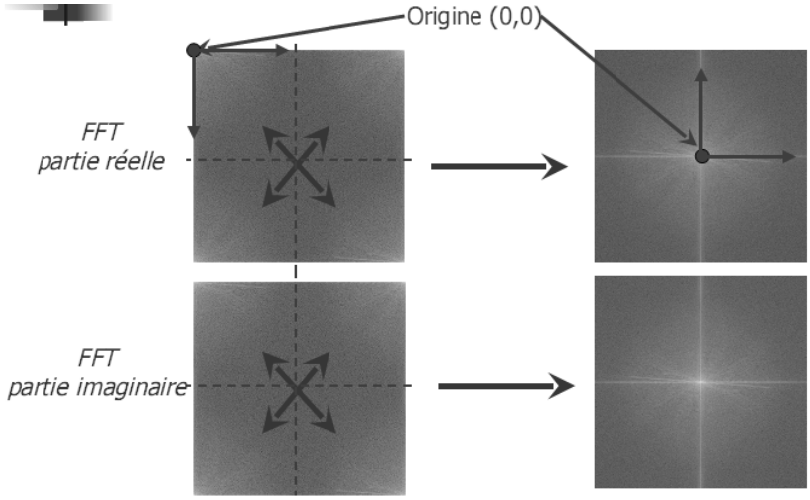
- On utilise beaucoup la norme de la Transformée :

$$\text{Norme} = |F(u,v)| = \sqrt{\text{Réel}^2 + \text{Imag}^2}$$

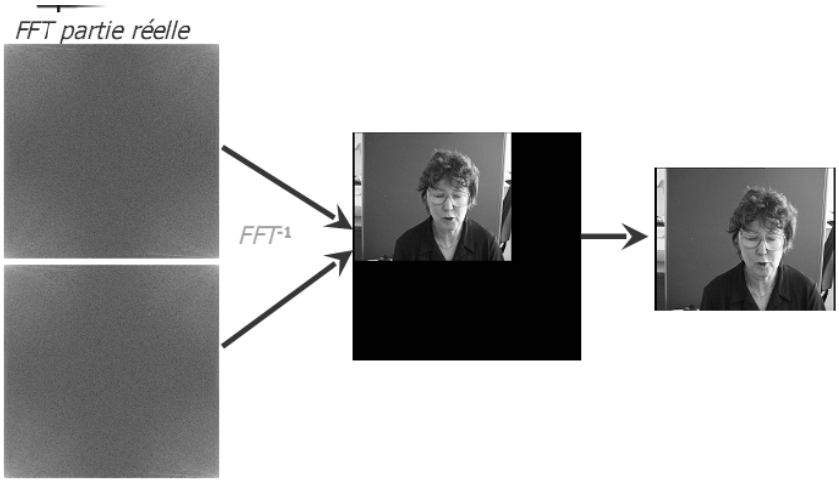
- L'algorithme le plus connu et le plus utilisé est la FFT
  - FFT = Fast Fourier Transform



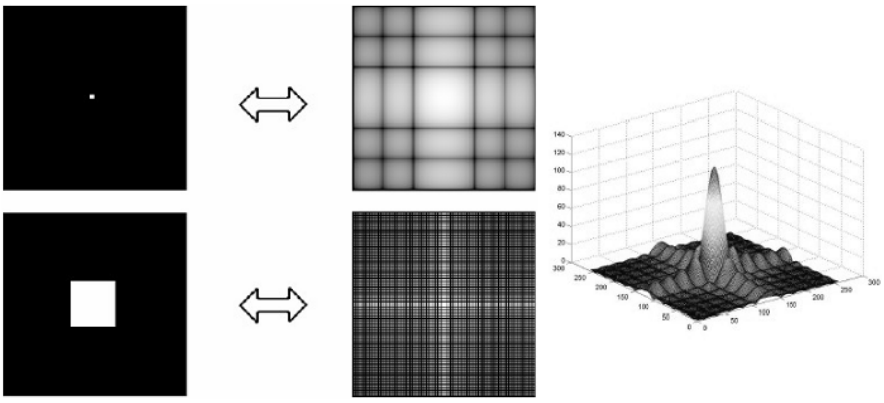
Inversion des Quadrants de la FFT :



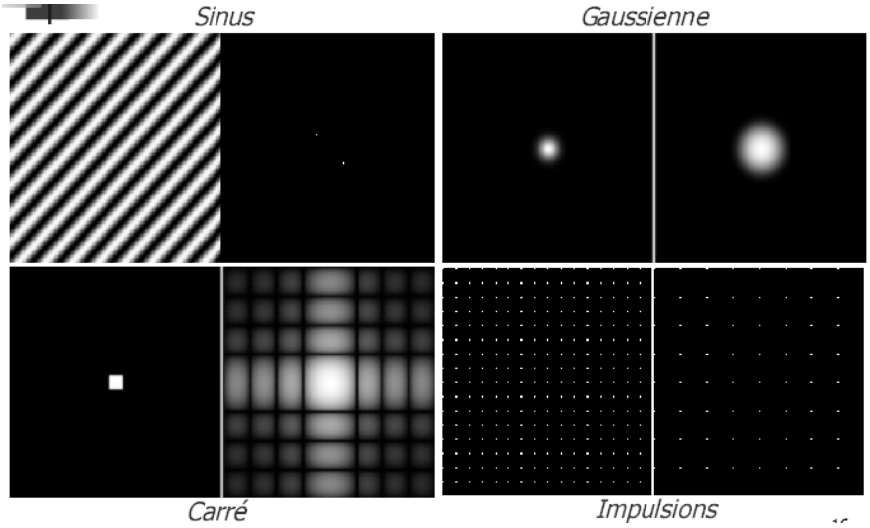
La Transformée Inverse :



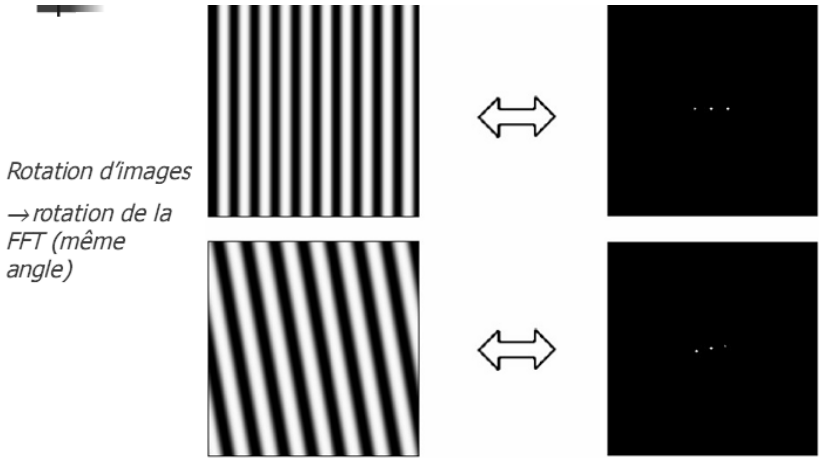
Exemple de FFT :



Images originales *FFT* *Vue 3D de la FFT*



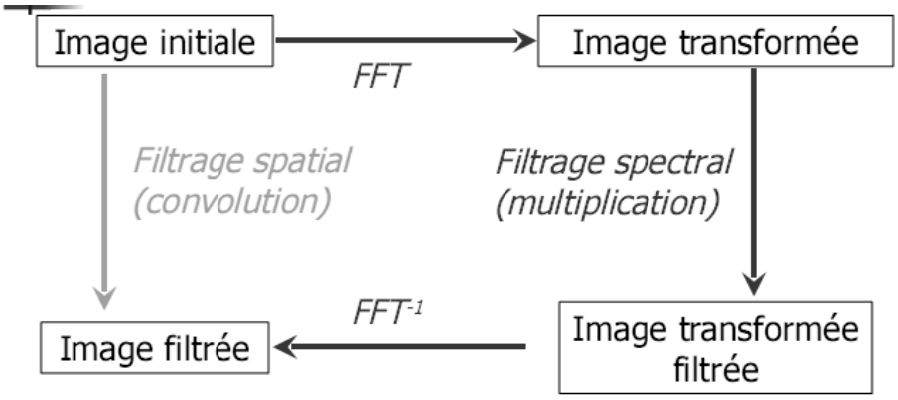
Rotation d'Image :



Rotation d'images

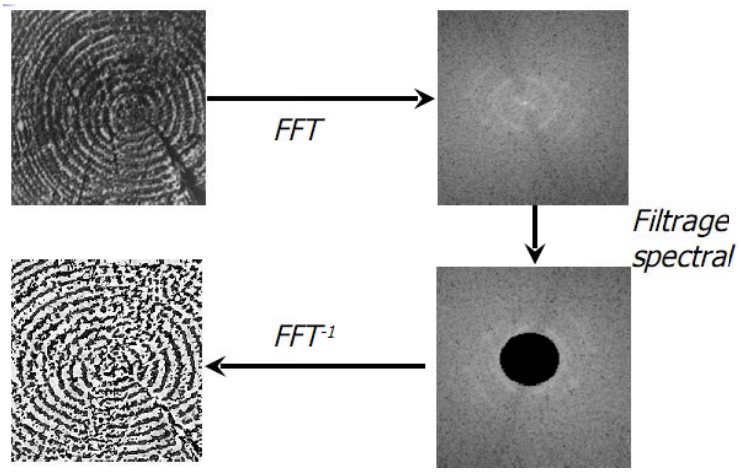
→ rotation de la FFT (même angle)

B- Filtrage dans le Domaine Spectrale :

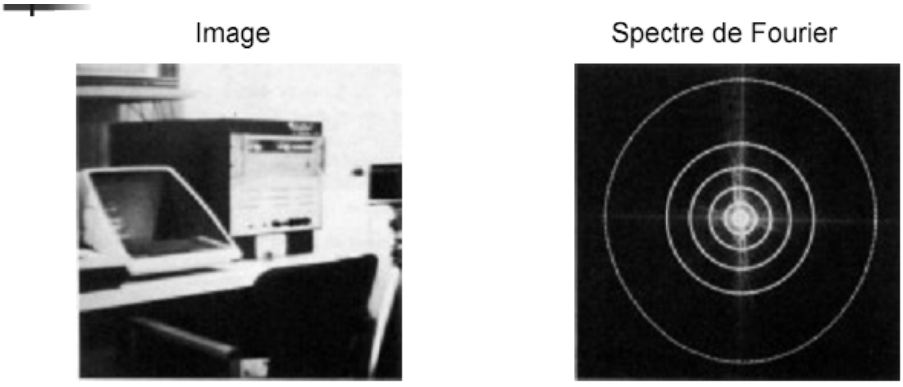


Dans le domaine spatial, le filtrage se fait par convolution. Dans le domaine spectral (ou fréquentiel), il se fait par multiplication (ou masquage de l'image).

Dans le cas des filtres dans le domaine fréquentiel non-multiplicatif, on ne peut pas obtenir le même résultat par convolution dans le domaine spatial.



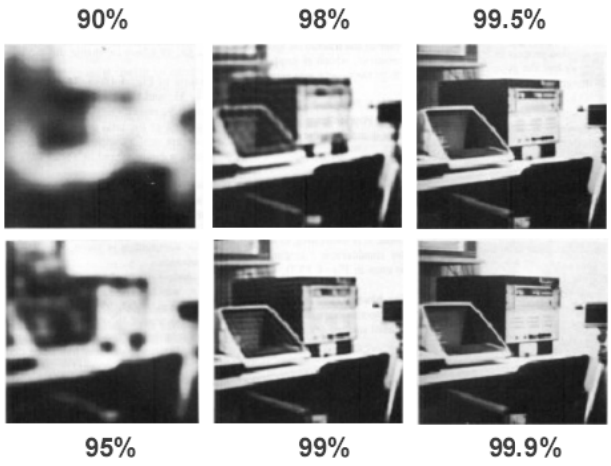
La Bande Fréquentielle :

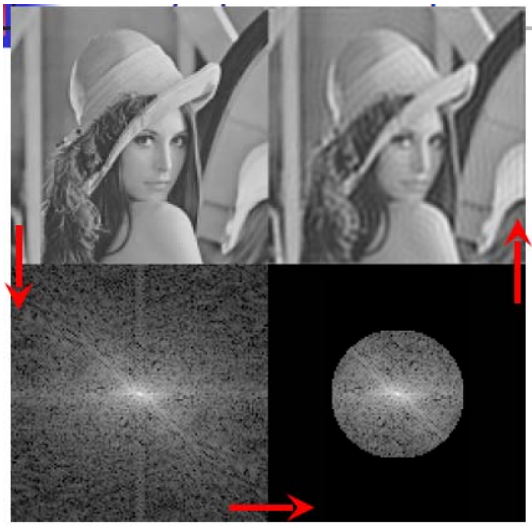


Pourcentage de l'information de l'image inclus dans les cercles (plus petit vers le plus grand) :

90%, 95%, 98%, 99%, 99.5%, 99.9%

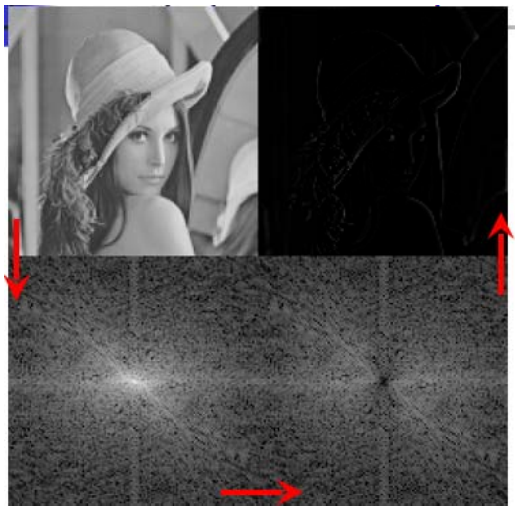
Filtrage Passe-Bas :



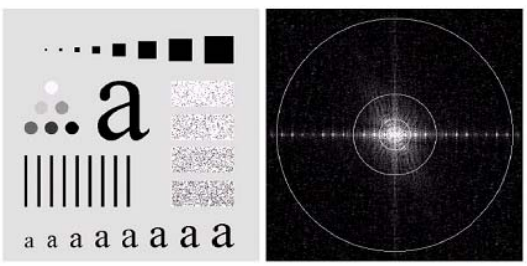


On efface les hautes fréquences de la FFT en mettant les pixels loin du centre à zéro

Filtrage Passe-Haut :



On efface les basses fréquences de la FFT en mettant les pixels au centre à zéro



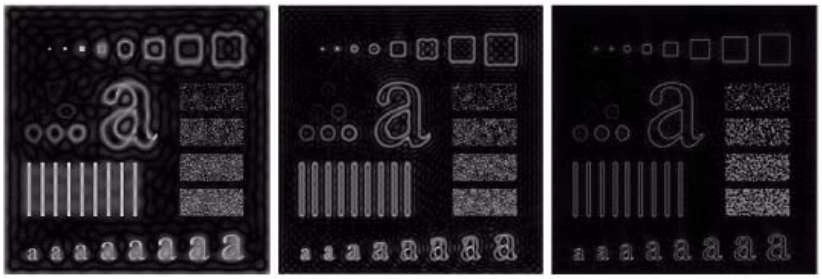


Image originale

Image filtrée passe-haut

Quelques Applications :

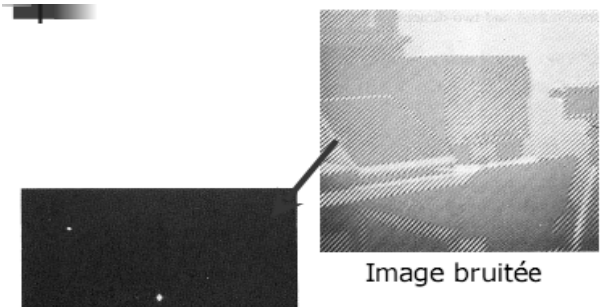


Image bruitée

Spectre de Fourier

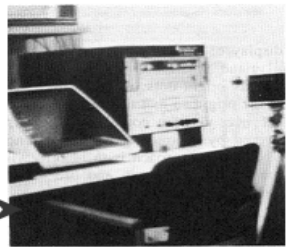


Image filtrée

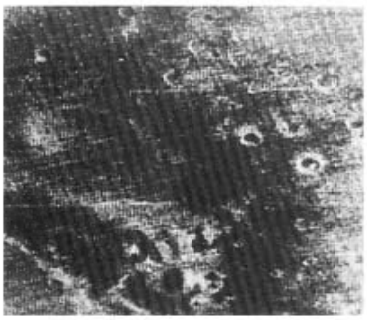
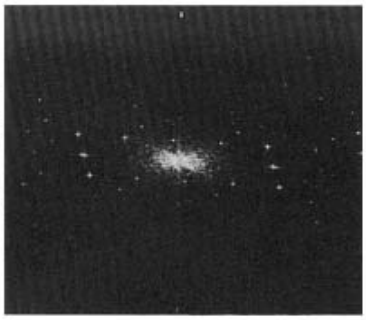
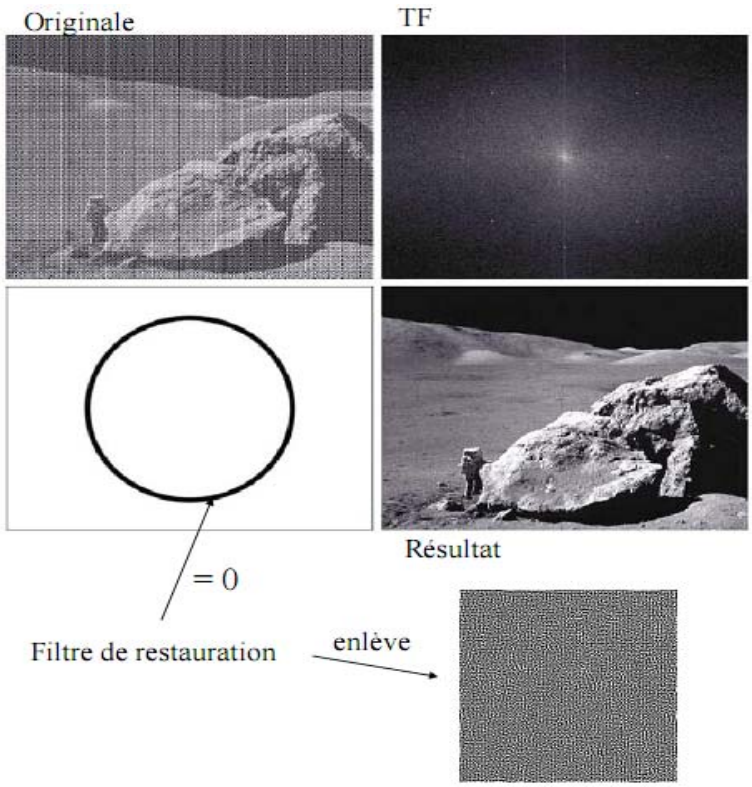


Image bruitée

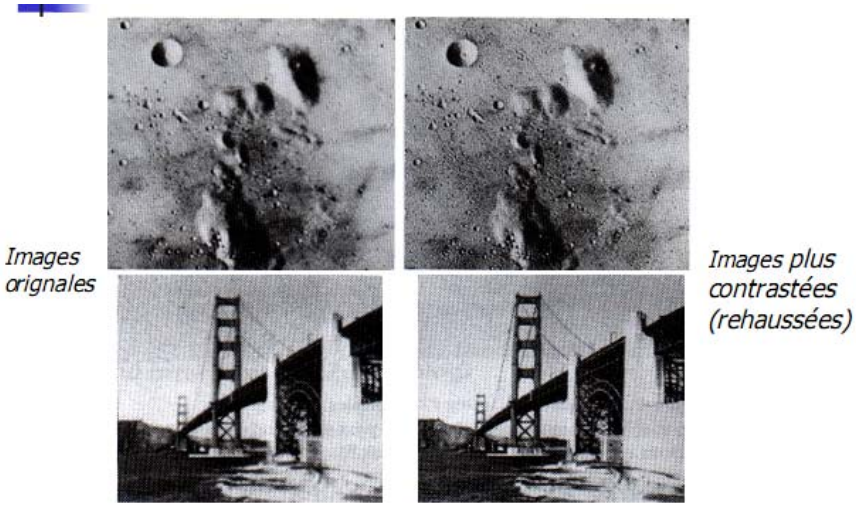


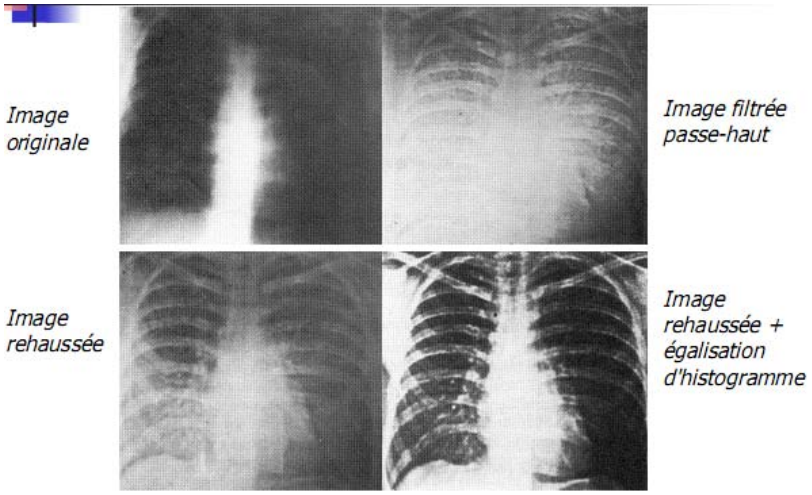
Spectre de Fourier

Image filtrée

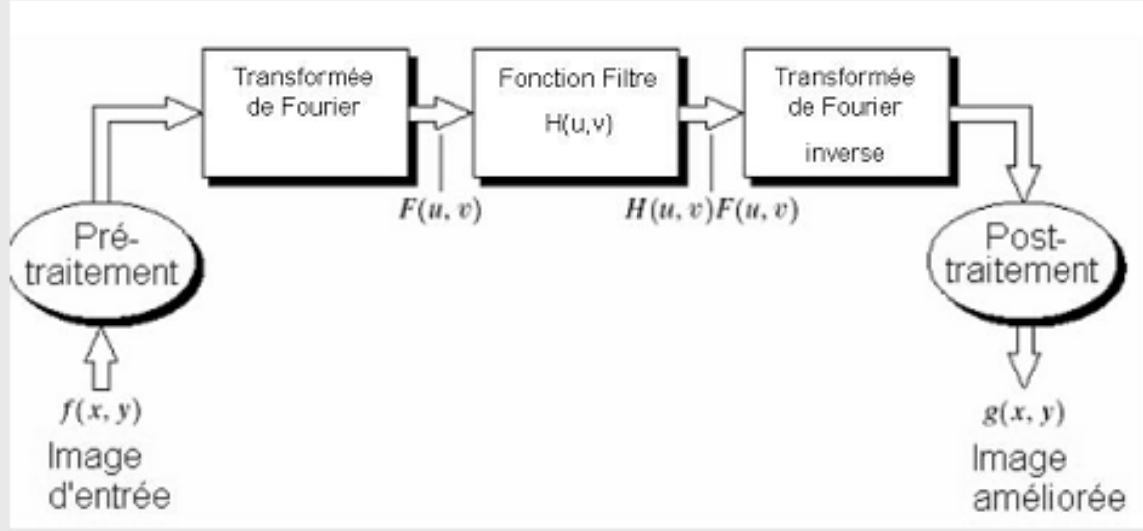


Rehaussement de Contraste :



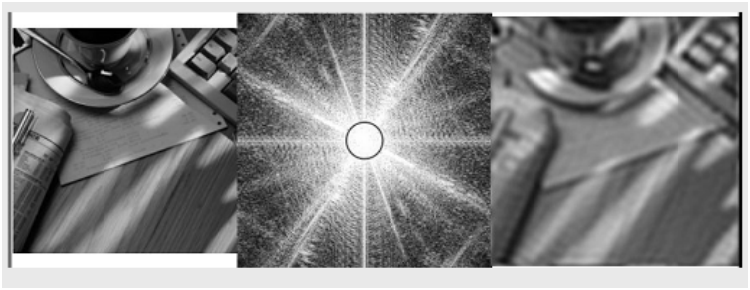


III. Amélioration des Images Numériques dans le Domaine Fréquentiel :



Suivant le type de filtre utilisé, on aboutit à différents résultats :
– coupe bande : élimination ponctuelle
– passe-bas
– passe-haut
On s'intéresse ici aux deuxième type.

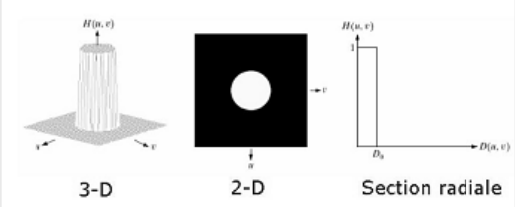
A- Filtre Passe-Bas Idéal :



– Coupe toutes les hautes fréquences après une distance D_0 du centre

$$H(u,v) = \begin{cases} 1 & \text{si} D(u,v) \leq D_0 \\ 0 & \text{sinon} \end{cases}$$

avec $D(u,v) = \sqrt{\left[\left(u - \frac{M}{2}\right)^2 + \left(v - \frac{N}{2}\right)^2\right]}$

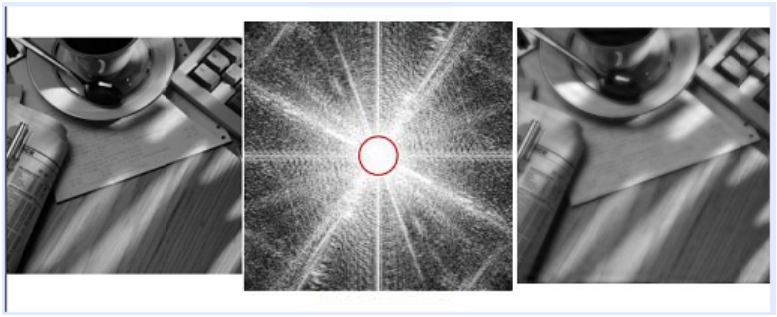


B- Filtre Passe-Bas Idéal 1 et ½ :

– Coupe 1/2 hautes fréquences après une distance D_0 du centre

$$H(u,v) = \begin{cases} 1 & \text{si} D(u,v) \leq D_0 \\ \frac{1}{2} & \text{sinon} \end{cases}$$

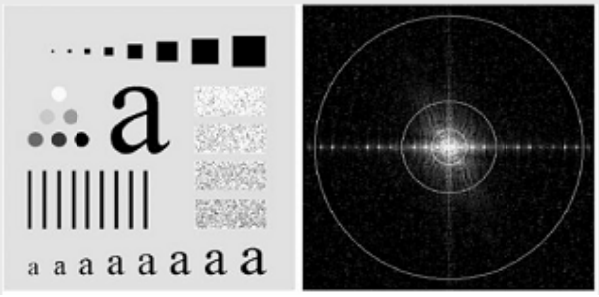
avec $D(u,v) = \sqrt{\left[\left(u - \frac{M}{2}\right)^2 + \left(v - \frac{N}{2}\right)^2\right]}$

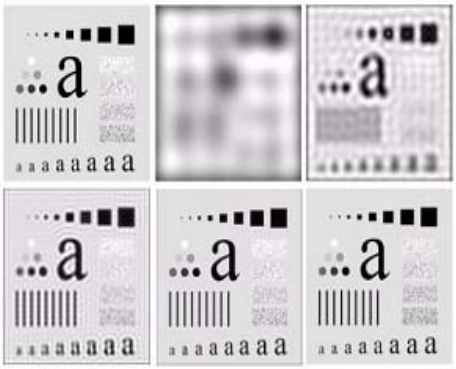


Effet de la fréquence de coupure D_0

Evalué en fonction de l'énergie comprise dans le cercle de rayon D_0 $P(u,v) = F(u,v)^2$, $P_T = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} P(u,v)$

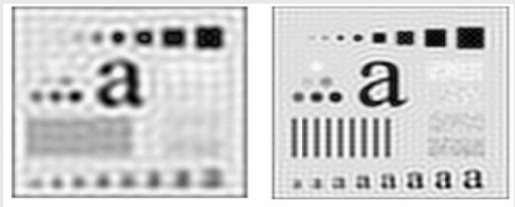
$$\% \text{ puissance} = 100 \sum_u \sum_v \frac{P(u,v)}{P_T} \quad (u,v) \in \text{ cercle}$$

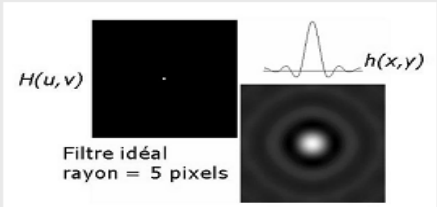




Rayon : 5, 15, 30, 80 et 230 pixels
Energie enlevée : 8, 5.4, 3.6, 2, et 0.5%

Phénomène de réverbération





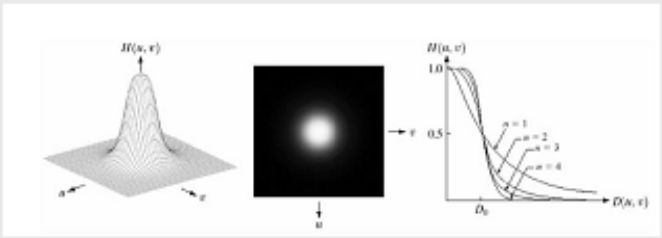
C- Filtre Butterworth Passe-Bas :

Coupe graduellement les hautes fréquences selon D_0 et n

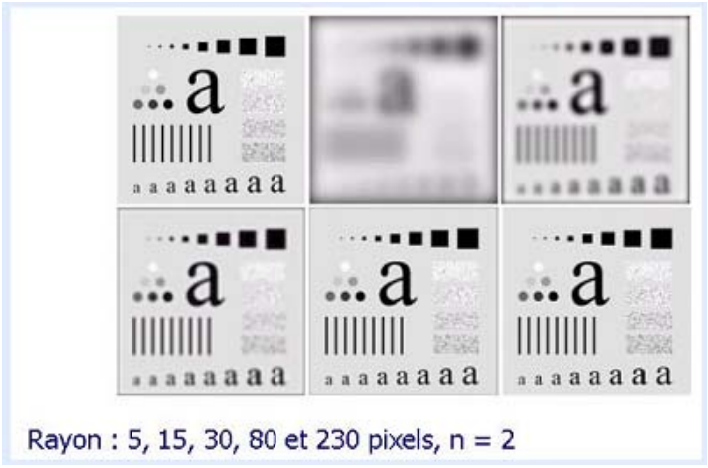
$$H(u,v) = \frac{1}{1 + \left[\frac{D(u,v)}{D_0}\right]^{2n}}$$

avec

$$D(u,v) = \sqrt{\left[\left(u - \frac{M}{2}\right)^2 + \left(v - \frac{N}{2}\right)^2\right]}$$

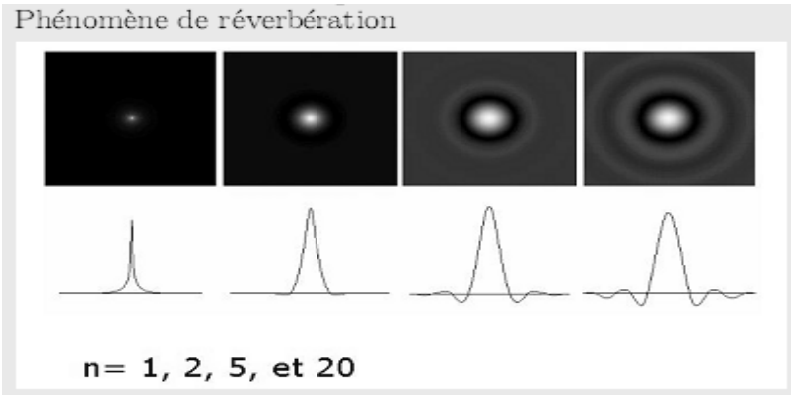


D_0 est choisie pour $H(u,v) = 0.5$



Rayon : 5, 15, 30, 80 et 230 pixels, n = 2

Phénomène de réverbération



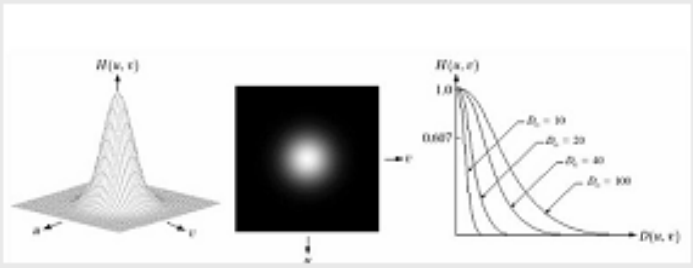
n= 1, 2, 5, et 20

D- Filtre Gaussien Passe-Bas :

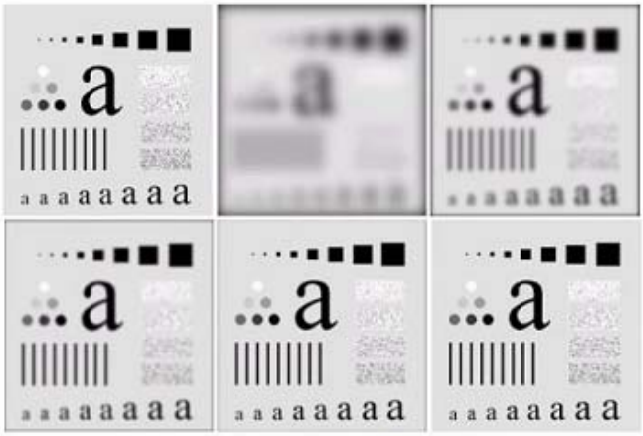
Coupe graduellement les hautes fréquences selon σ

$$H(u,v) = e^{-\frac{D^2(u,v)}{2\sigma^2}}$$

$\sigma = D_0$



$D(u,v) = D_0$ quand $H(u,v) = 0.607$



Rayon : 5, 15, 30, 80 et 230 pixels, n = 2

- Pas de phénomène de réverbération
- Moins agressif que le filtre idéal ou le filtre Butterworth
- Moins de contrôle sur la sélection précise de D_0

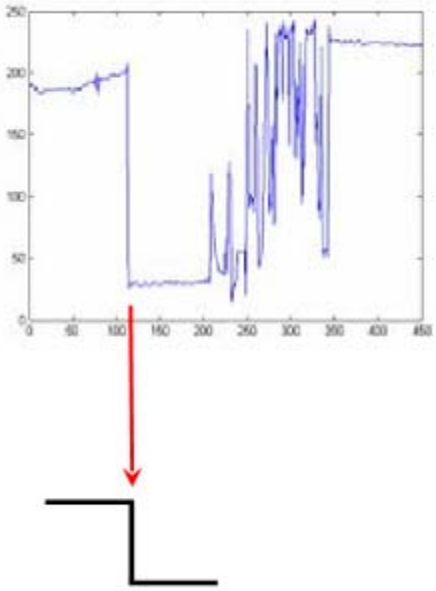
Chapitre 3 : Détection de Contour & Segmentation des Images Numériques

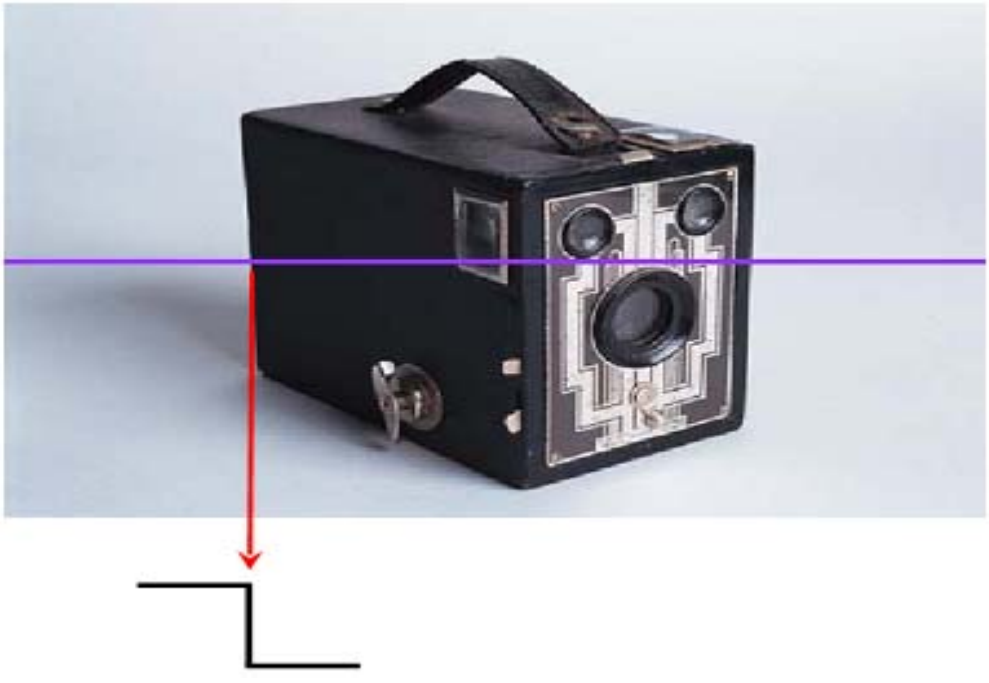
1- La Détection de Contour :

1.1 Introduction à la Détection de Contour :

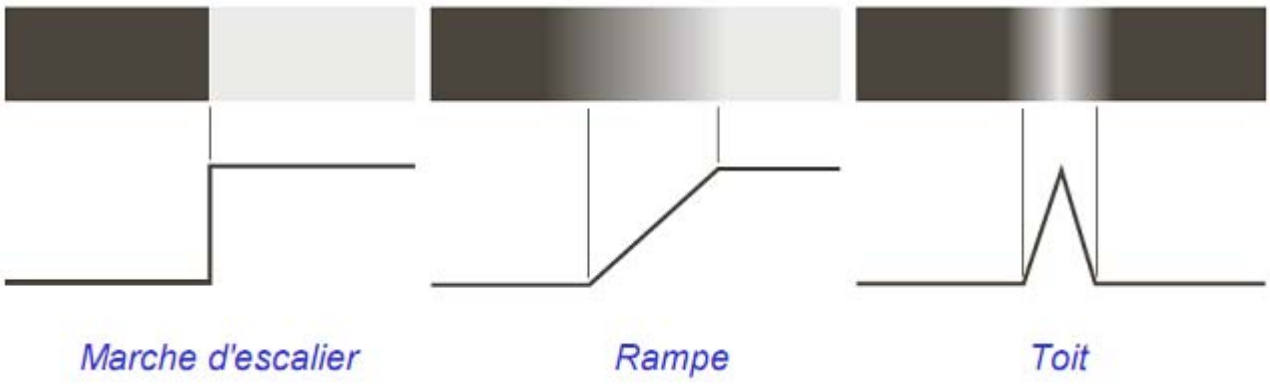
Un contour est une variation brusque d'intensité

- Par définition, un contour est la frontière qui sépare deux objets dans une image.
  - *Une discontinuité de l'image*
- Dans notre cas, nous détecterons toutes les lignes marquant des changements d'intensité
  - *Pas seulement les contours !*
  - *Abus de langage sur la notion de contours !*

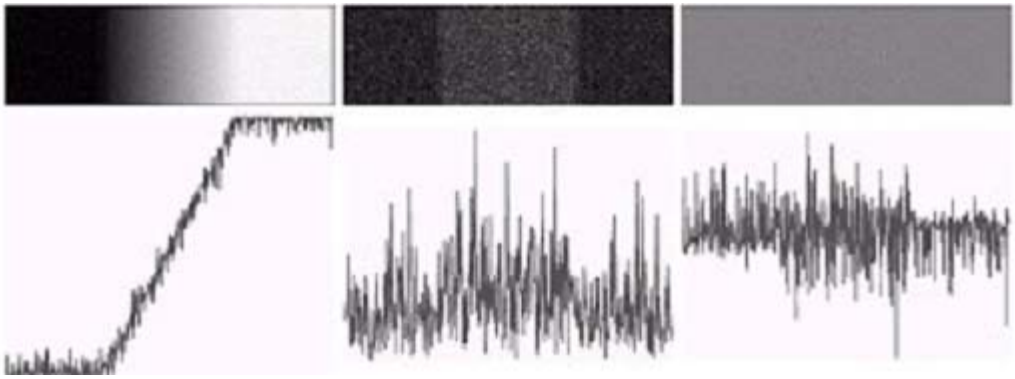




1.2 Différents Types de Contours :

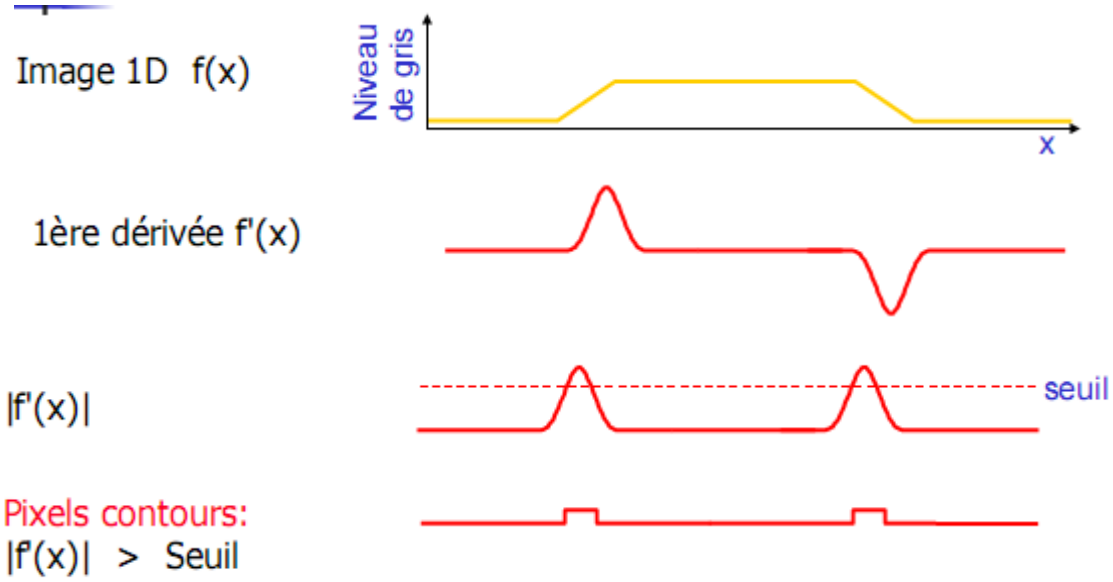


Contour Type



Contour plus le Bruit

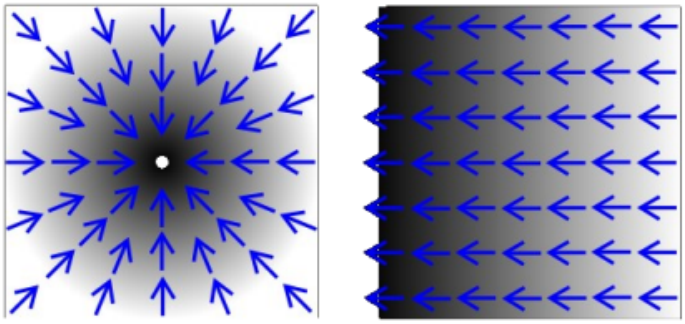
1.3 La Drivée d'une Image (Contour) :



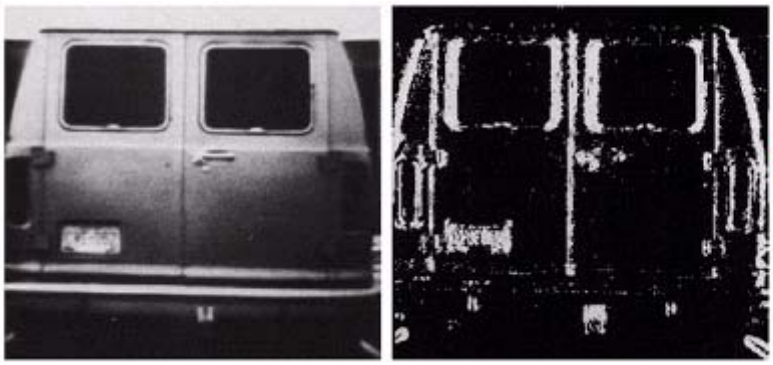
1.3.1 Le Gradient d'Une Image :

- En 2 dimensions, nous parlons de gradient de l'image
  - *dérivée en X + dérivée en Y*

→ vecteur avec une norme et une direction



Les lignes bleues représentent le gradient de couleur du plus clair vers le plus foncé

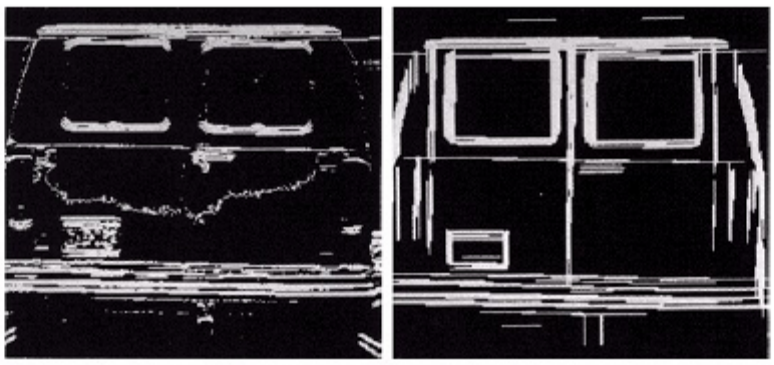


$\frac{\partial f}{\partial x}$

Contours verticaux

$\frac{\partial f}{\partial y}$

Contours horizontaux



$\frac{\partial f}{\partial y} + \frac{\partial f}{\partial x}$

Norme

1.3.2 La Dérivée Discréte :

On utilise la première dérivée de l'image pour les contours : $\frac{\Delta I}{\Delta x} = \frac{I(x+\Delta x) - I(x)}{\Delta x}$

Approximation simple de la dérivée discrète :

-1	1

-1
1

Ou encore :

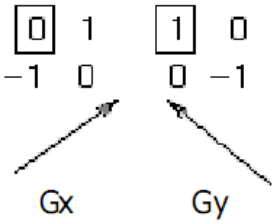
-1	0	1

-1
0
1

1.3.3 Les Diférents Filtres de Contours :

A- Filtre de Robert

- Roberts (1965) fournit une première approximation de la première dérivée d'une image discrète
- Le calcul se fait avec 2 masques de convolution pour les 2 directions de la dérivée



B- Autres Filtres :

Plusieurs autres filtres existent pour la détection des contours dans l'image

On fait lissage de l'image + dérivée de l'image (sauf Roberts)

Il existe beaucoup d'autres filtres pour détecter les contours

Roberts:

1	0
0	-1

0	1
-1	0

Prewitt:

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

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

Sobel:

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

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

1.3.4 Lissage et Dérivée de l'Image :

Filtre de Prewitt : Moyenneur + Dérivée

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} * \begin{pmatrix} -1 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} * \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$$

Filtre de Sobel : Gaussienne + Dérivée

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

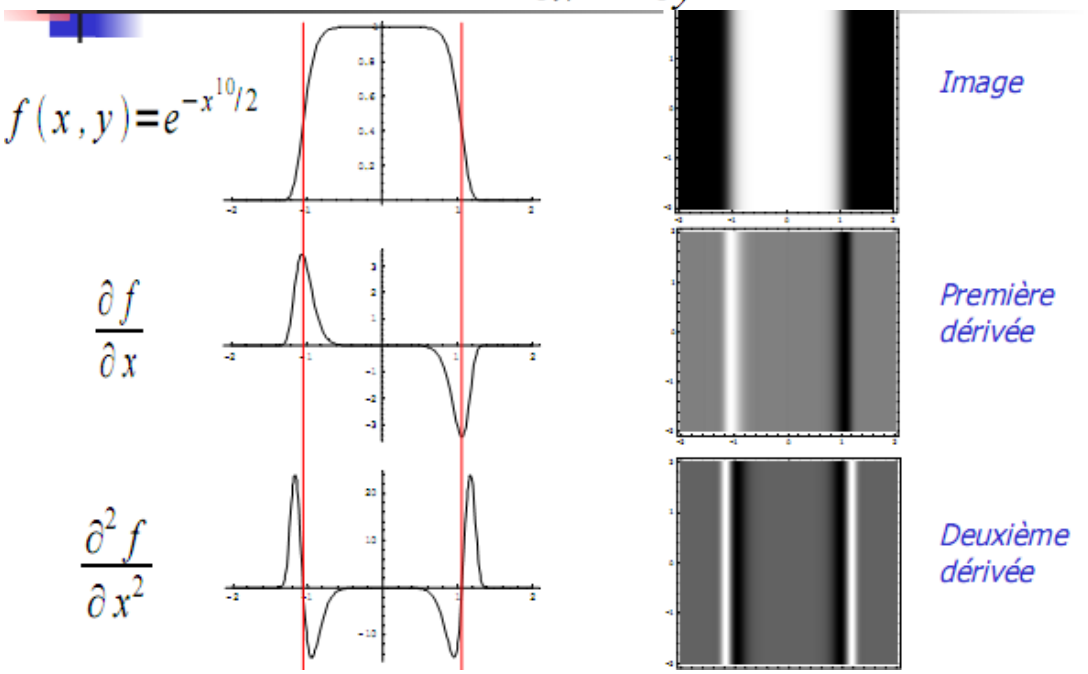
⇒ Détection des contours moins sensible au bruit

1.4 Deuxiéme Dérivée : (Laplacien) :

- Une autre approche pour trouver les contours de l'image est d'utiliser la seconde dérivée de l'image

- Pour cela, on utilise le Laplacien comme opérateur

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



Le Contour Rampe :

- Plusieurs approximations discrètes du Laplacien existent

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{ou} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- Une seule matrice de convolution
- Symétrique en rotation

Exemple de Comparaison entre Gradient et Laplacien

Remarques Importantes:

- Aucun opérateur n'est parfait pour détecter les contours
- En pratique, on obtient des contours incomplets
  - *il y a des pixels superflus*
  - *il y a des manques*
  - *il y a des erreurs de position et d'orientation des pixels contours*

- Chacun semble avoir sa préférence pour une méthode ou une autre
- Un opérateur de détection de contour n'est qu'une première étape dans la chaîne de segmentation

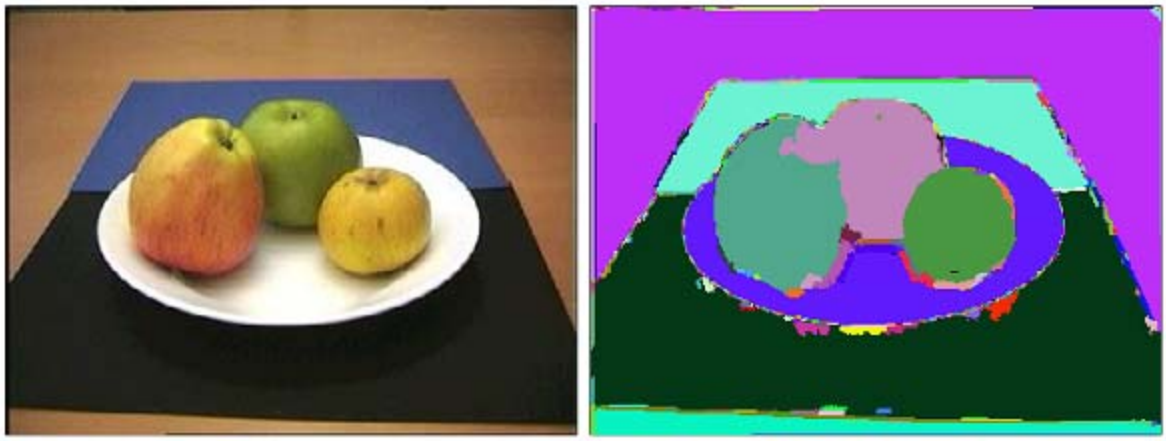
Il existe d'autres filtre poure la détection de Contour qui sont plus efficaces dans des domaines d'application bien définis, tel que, le Filtre de Hough vis-à-vis l'image médicale.

2- Introduction à la Segmentation :

- La segmentation vise à diviser l'image en morceaux
  - *Ces morceaux correspondent aux objets dans l'image*
- La segmentation est liée à la reconnaissance
  - *Quels objets voit-on dans l'image ?*

- Extraire (séparer) les entités d'une image
  - *Pour y appliquer un traitement spécifique*
  - *Pour interpréter le contenu de l'image*
- Dans la pratique : construire une image de masques
- Chaque masque est une composante connexe

Exemple d'une image ségmentée :



- La segmentation est normalement basée sur:
  - *les discontinuités : contours*
    - les changements abruptes, frontières entre régions…
  - *les zones homogènes : régions*
    - Mêmes couleurs, textures, intensités, …
- La segmentation est le découpage d'une image en différentes régions et/ou contours
- Un contour fermé est équivalent à une région

2.1 Approche par Seuil :

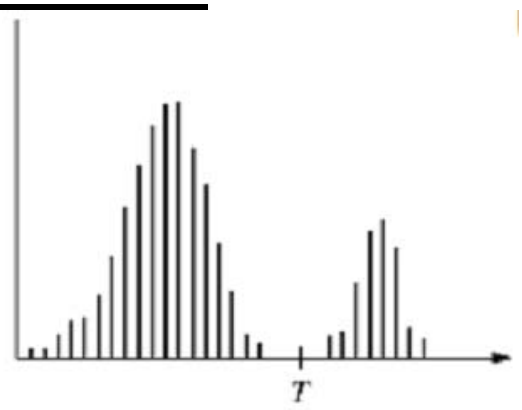
- Le seuillage est une méthode simple et très populaire pour le traitement des images numériques
- Ce n'est pas une méthode de segmentation en régions
  - ***Approche pixel*** *(pas région ni contour)*
  - *Mais on l'utilise souvent en segmentation (avec post-traitements)*
- Le seuillage peut être
  - Global : un seuil pour toute l'image
  - Local : un seuil pour une portion de l'image
  - Adaptatif : un seuil s'ajustant selon les parties de l'image

- Seuillage de base (2 classes) :
  - *Si valeur(pixel) ≥ seuil alors valeur(pixel) = 1*
  - *Si valeur(pixel) < seuil alors valeur(pixel) = 0*

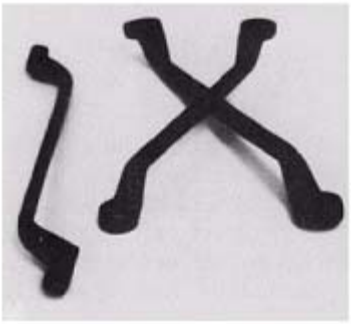
- Le résultat du seuillage est une image binaire
  - 0 ou 1 *(qu'on transforme parfois en 0:255 pour l'affichage)*

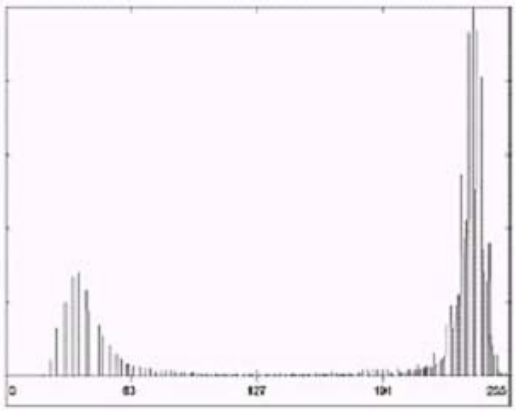
- Problème : choix du seuil !

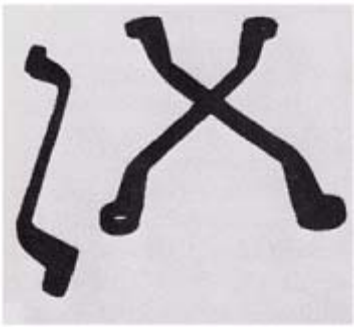
$$g(x,y)=\begin{cases}1 \text{ si } f(x,y)\geq T\\ 0 \text{ si } f(x,y)<T\end{cases}$$



Exemple :







Quelques Avantages & Inconvénients :

- Universel, temps réel, simplicité
- Fonctionne bien sur des histogrammes multi-modaux

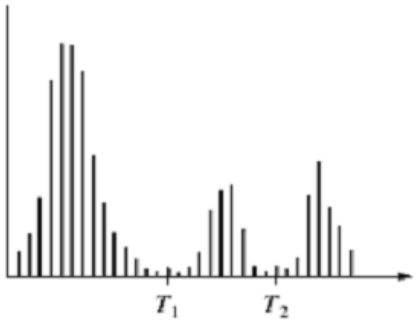
- Connaître le nombre de classes
- Apparition de faux éléments (aucune prise en compte de la composante spatiale)
- Nombre de modes souvent nombre de classes attendu

Remarque Importante

Dans le cas, d'une image conteant une forte variation de régions ou ségments, on peut utiliser l'approche Muti-Seuils.

Exemple :

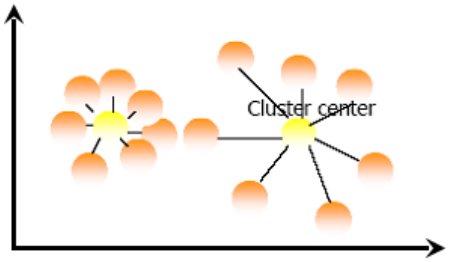
$$g(x,y)=\begin{cases}2 \; si \; f(x,y)\geq T_2\\ 1 \; si \; f(x,y)\geq T_1\\ 0 \; si \; f(x,y)<T_1\end{cases}$$



- Comment trouver le bon seuil (T) ?
  - Une valeur obtenue par tests
  - La valeur moyenne des tons de gris
  - La valeur médiane entre le ton maximum et le ton minimum
  - Une valeur qui balance les deux sections de l'histogramme

2.2 Approche par K-moyennes

- On veut diviser les points en *k* groupes *(clusters)*
  - k est donné à l'avance (un paramètre de l'algorithme)
  - On définit le centre d'un groupe comme la moyenne des éléments (pixels) du groupe



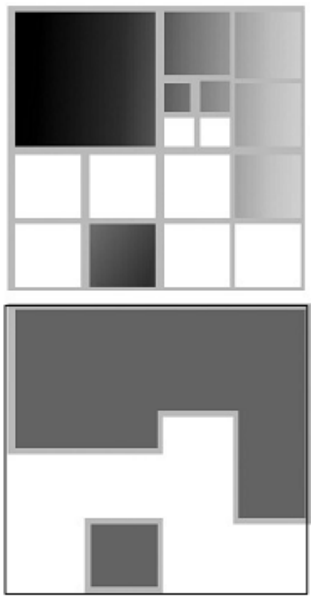
L'algorithme est composé de 4 étapes :

1. Partitionnement des données en k sous-ensembles (non vides)
2. Calcul des centres des groupes de la partition courante
3. Les données sont affectées au groupe dont le centre leur est le plus proche
4. Retour à l'étape 2

Arrêt lorsque les groupes sont ~ constants

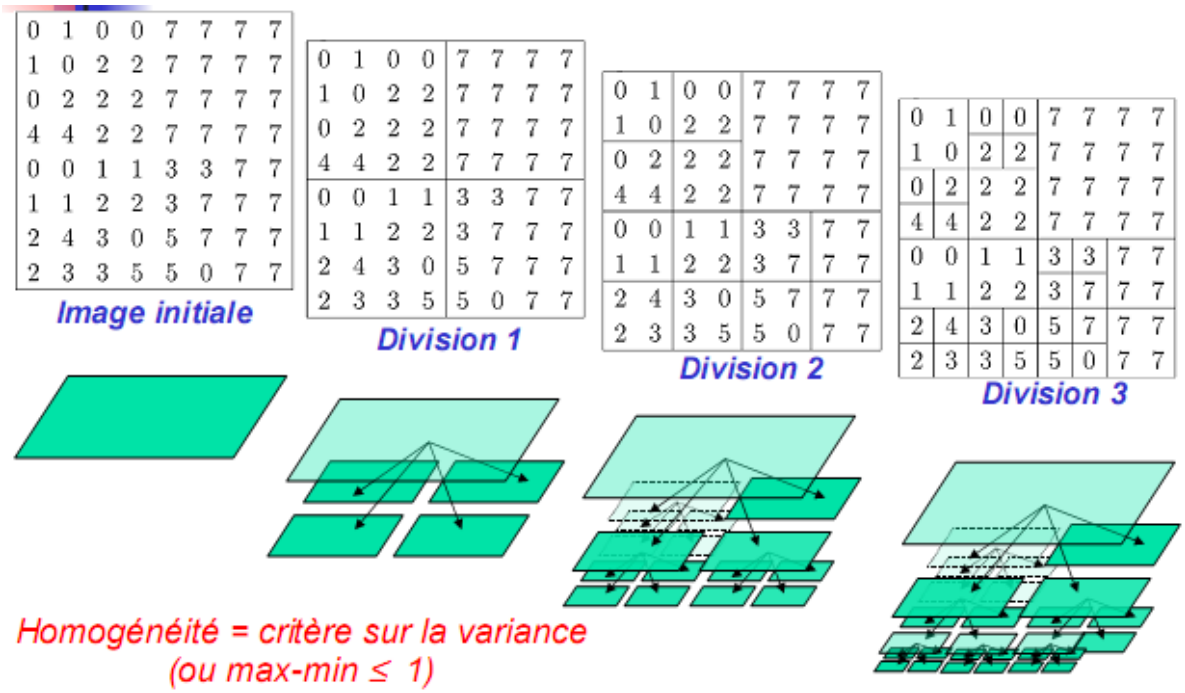
2.3 Approche Division-Fusion : Split and Merge :

- ***Etape de division*** *(split)*
  - Diviser récursivement tout bloc non-homogène selon un prédicat défini
    - *variance, max-min, …*
  - La division d'un bloc donne 4 sous-blocs
  - Les attributs de chaque sous-bloc sont recalculés
- ***Etape de fusion*** *(merge)*
  - Regrouper les blocs adjacents représentant des régions homogènes selon un prédicat défini

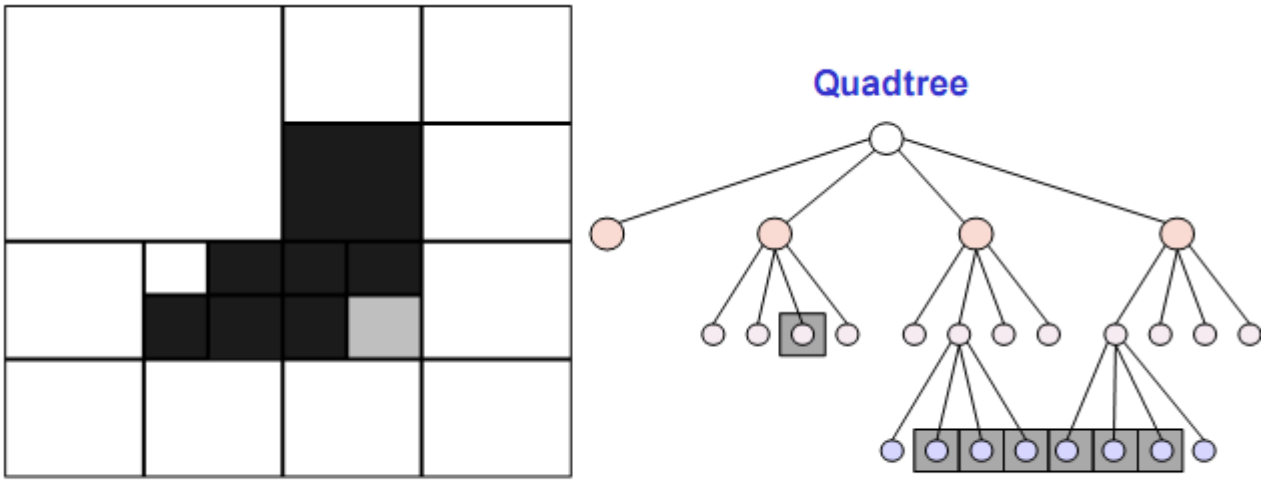


Exemple Concrét :

Etape de Division :



Etape de Fusion :



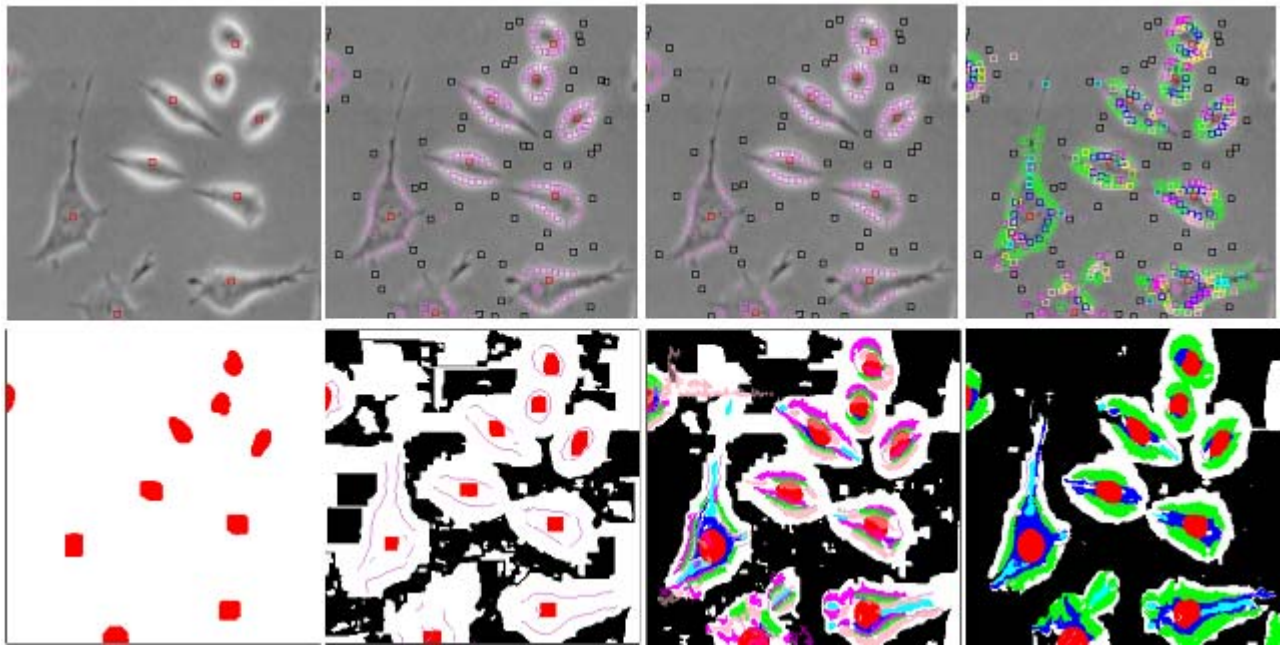
Connecter les régions adjacentes homogènes

2.4 Approche de Croissance en Régions :

- On débute avec un pixel, et on « ajouter » les pixels voisins qui répondent à un critère d'appartenance :
  - *Variance faible*
  - *Niveau de gris répondant un seuil*
  - *…*
- Les pixels initiaux sont appelés « germes », « amorces » ou semences«
  - *Choix des pixels initiaux automatiques ou semi-automatiques*
- La région « grandit » à partir de son germe
  - *Besoin d'une critère (ou prédicat) pour choisir les pixels à ajouter*
- On part d'un germe *(seed)* et on l'étend en ajoutant les pixels voisins qui satisfont le critère d'homogénéité
- Le germe peut être choisi soit par un humain, soit de manière automatique en évitant les zones de fort contraste *(gradient important)*

Exemple avec un seul germe :

Exemple avec plusieurs germes :



2.5 Remarques Globales :

- La segmentation d'une image cause encore aujourd'hui beaucoup de problèmes
  - *Aucune méthode ne fonctionne pour toutes les images*
  - *Pas de garantie, pas de recette miracle !*
- Le pré-traitement des images, la sélection de capteurs et sources d'énergie appropriées, et la prise contrôlée des images rendent cette étape plus facile et plus efficace
- La segmentation aide beaucoup pour la reconnaissance, mais elle n'est pas obligatoire dans tous les cas
  - *Voir cours de Vision par Ordinateur, semestre III*
- Evaluer le résultat d'une segmentation n'est pas facile
  - *Il dépend de l'application*
  - *Il dépend de ce qu'on veut*
  - *Il est subjectif et varie d'une personne à l'autre*
- Un des principaux problème est de définir le but de la segmentation
  - ***Qu'est-ce qu'on recherche exactement dans l'image ?***
    - *Eléments globaux de l'image ou détails fin de la composition ?*
    - *Présence d'un humain ou détails du visage ?*
- Il faut toujours se poser la question de ce que l'on veut faire avec la segmentation
  - *Cela permet de définir le degré de précision nécessaire*

Chapitre 4 : Codage et Compression des Images Numériques

1. Codage des images :

Il y a deux modes de codage d'une image numérique :

- Bitmap
- Vectoriel

Pour stocker ces deux types d'images sur disque, il existe de très nombreux formats de fichiers. Ces formats diffèrent par les quantifications supportées (binaire, niveaux de gris, palette, 24 bits, alpha-channel, etc.), par leur algorithmes de compression, par la volonté d'avoir un format propriétaire, ou au contraire ouvert, etc.

1.1 Images bitmap

Une image bitmap est un ensemble de points (=pixels), caractérisée par sa discrétisation et sa quantification.

Formats de fichiers d'images bitmap

- Formats limités à 256 couleurs : GIF, PCX, PGM, ...
- Formats acceptant différentes quantifications : BMP, TIFF, TGA, PNG, ...
- Formats limités à 16 millions de couleurs : JPEG, JPEG 2000, ...

Remarque : Certains de ces formats permettent une compression des données.

Avantages des images bitmap

1. Les images bitmaps peuvent facilement être créées et stockées dans un tableau de pixels représentant l'image.
2. Lecture/écriture d'un pixel aisée de part la représentation de l'image comme une grille.
3. Les images bitmaps peuvent facilement être affichées sur un écran ou être imprimées.

Inconvénients des images bitmap

1. Les fichiers peuvent être très gros (→ nécessité de compression)
2. Problème de changement d'échelle (apparition d'effets de marches d'escalier ou de flou avec interpolation).
3. Les dimensions de l'image doivent être prévues pour la résolution de l'interface de sortie (écran, imprimante).

1.2 Images vectorielles

Ne stocke pas le résultat du dessin sous la forme de pixels (=bitmap), mais la façon de dessiner par un ensemble d'objets géométriques (lignes, cercles, polygones, courbes de Bézier, texte, ...) définis par différents attributs (coordonnées, couleur, épaisseur de trait, remplissage, ...)
Les domaines d'utilisation de ces images sont très variés : dessin industriel, PAO, Systèmes d'Information Géographiques (SIG), Internet,…

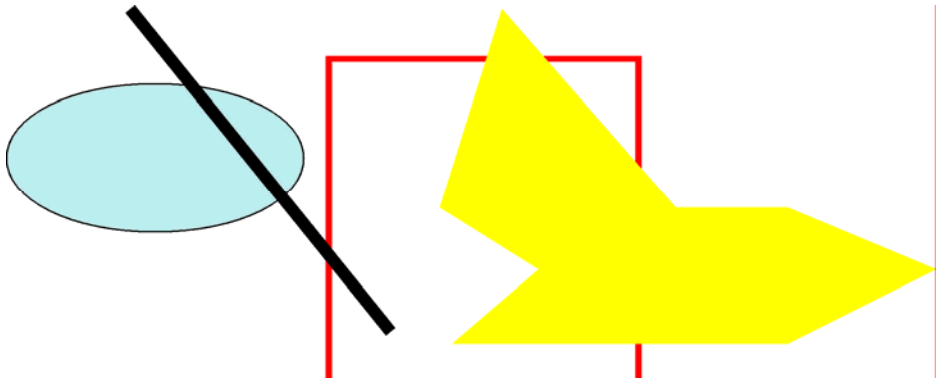


Image vectorielle

Avantages des images vectorielles

1. Adaptées au stockage d'images composées de formes géométriques.
2. Peuvent aisément être redimensionnées.
3. Prennent moins de place qu'une image bitmap.

Inconvénients des images vectorielles

1. Peuvent difficilement stocker des images complexes comme des photographies.
2. L'affichage d'une image vectorielle peut prendre plus de temps que l'affichage d'une image bitmap de complexité égale.

Formats d'images vectorielles

- AI (Adobe Illustrator)
- CDR (Corel Draw)
- EPS (Encapsulated PostScript)
- PDF (Portable Document Format)
- PS (PostScript)
- PSD (Adobe Photoshop)
- SVG (Scalable Vector Graphics)
- SWF (Flash)
- WMF (Windows MetaFile) (cliparts Windows)
- EMF (Enhanced MetaFile) (cliparts Windows)

2. Compression

Compresser une image = réduire la taille de son fichier en remplaçant la série de ses données par une autre plus réduite. Processus inverse : décompression (par exemple pour afficher l'image). Concerne essentiellement les images bitmap.
Pourquoi compresser une image (ou tout autre type de données : texte, son, vidéo, etc.) ?

- Pour réduire sa place sur disque.
- Pour accélérer sa transmission sur un réseau.

Ex: Une image en 16 millions de couleurs de 1600 x 1200 pixels . taille = 1600 x 1200 x 3 octets = 5,49 Mo

Taux de compression :

$$\text{taux de compression} = \frac{\text{taille originale}}{\text{taille compressée}}$$

Exemple :

Soit une image originale de 100 Ko, compressée à 20 Ko, le taux de compression est de 5:1. Le taux de compression est variable, il dépend de la nature de l'image à compresser et de l'algorithme de compression utilisé.

En réalité, il existe Deux types de compression :

- Réversible, sans perte (« lossless »)
- Non réversible, avec pertes (« lossy »)

2.1 Compression sans perte

- Pas de modification de l'image.
- Ne change que la façon dont elle est codée sur disque.

➔ Affichage (=décompression) de l'image : identique à l'original.

Algorithmes de compression sans perte :

- Méthodes à base de redondances (RLE) : utilisées par les formats d'images PCX, TGA, BMP, ...
- Méthodes statistiques (Huffman, ...) : utilisées par les formats d'images JPEG (en partie), ...
- Méthodes à base de dictionnaires (LZW, ...) : utilisées par les formats d'images GIF, TIFF, ...

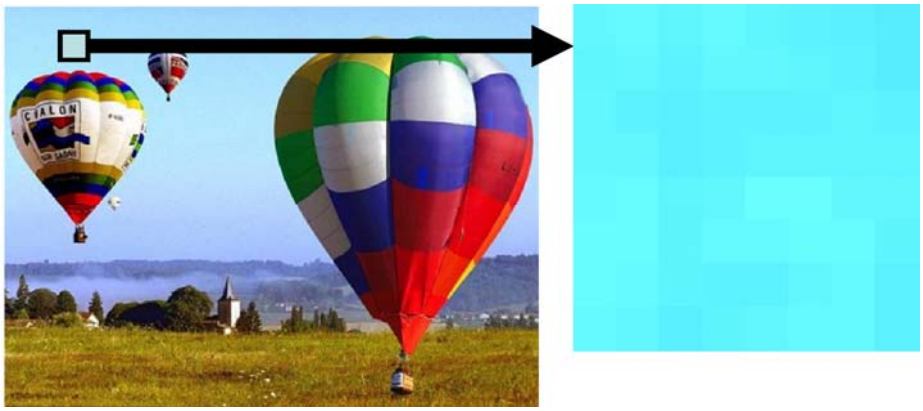
2.1.1 Codage RLE (Run Length Encoded)

Code la répétition de pixels identiques en donnant le nombre de répétitions et le pixel à répéter.

Exemple :

ABCCCDDDDEF --> AB3C4DEF

- Surtout efficace pour des images <_ 256 couleurs.
- Inefficace pour des images en 16 millions de couleurs (trop de nuances différentes pour qu'il y ait de longues séquences de pixels identiques).



2.1.2 Codage de Huffman

Certaines couleurs de pixels reviennent plus souvent que d'autres.

Idée : réduire le nombre de bits utilisés pour le codage des pixels les plus fréquents et d'augmenter ce nombre pour les pixels plus rares. Surtout efficace pour des images ≤ 256 couleurs. Inefficace pour des images en 16 millions de couleurs.

Algorithme de compression (ex pour un texte) :

rien de nouveau sous le soleil

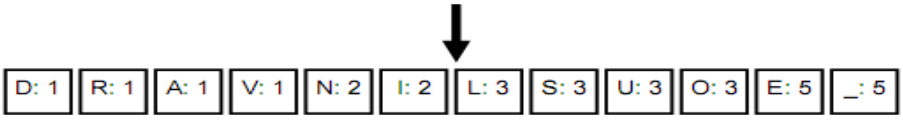
Cette chaîne comporte 30 lettres, soit 30 octets ou 240 bits.

1. On cherche la fréquence des caractères.

R	I	E	N	_	D	O	U	V	A	S	L
1	2	5	2	5	1	3	3	1	1	3	3

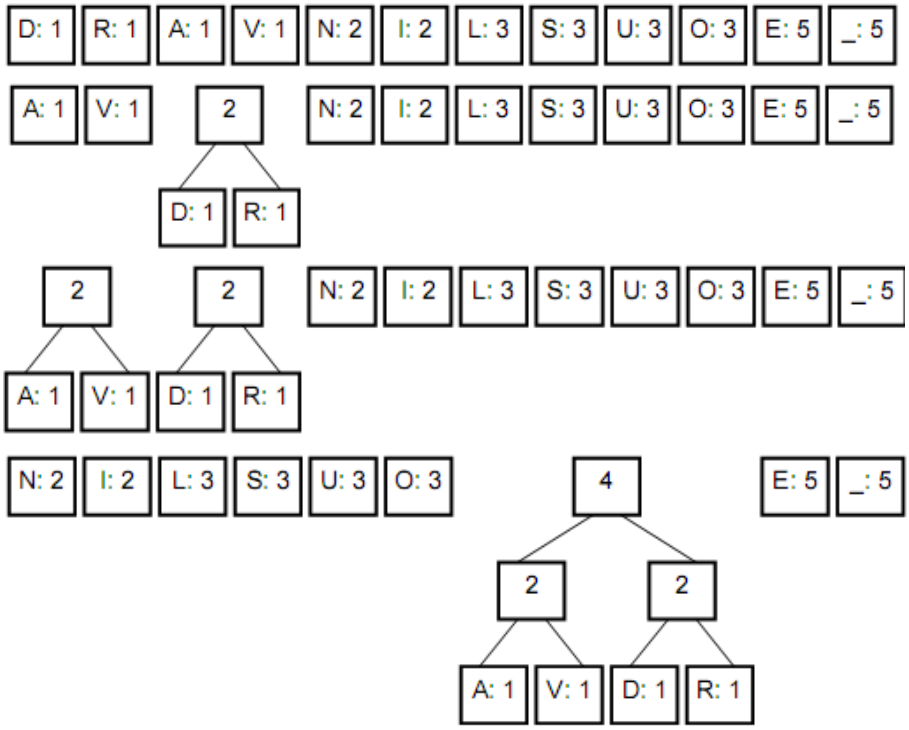
2. On trie les caractères par ordre croissant de fréquence.

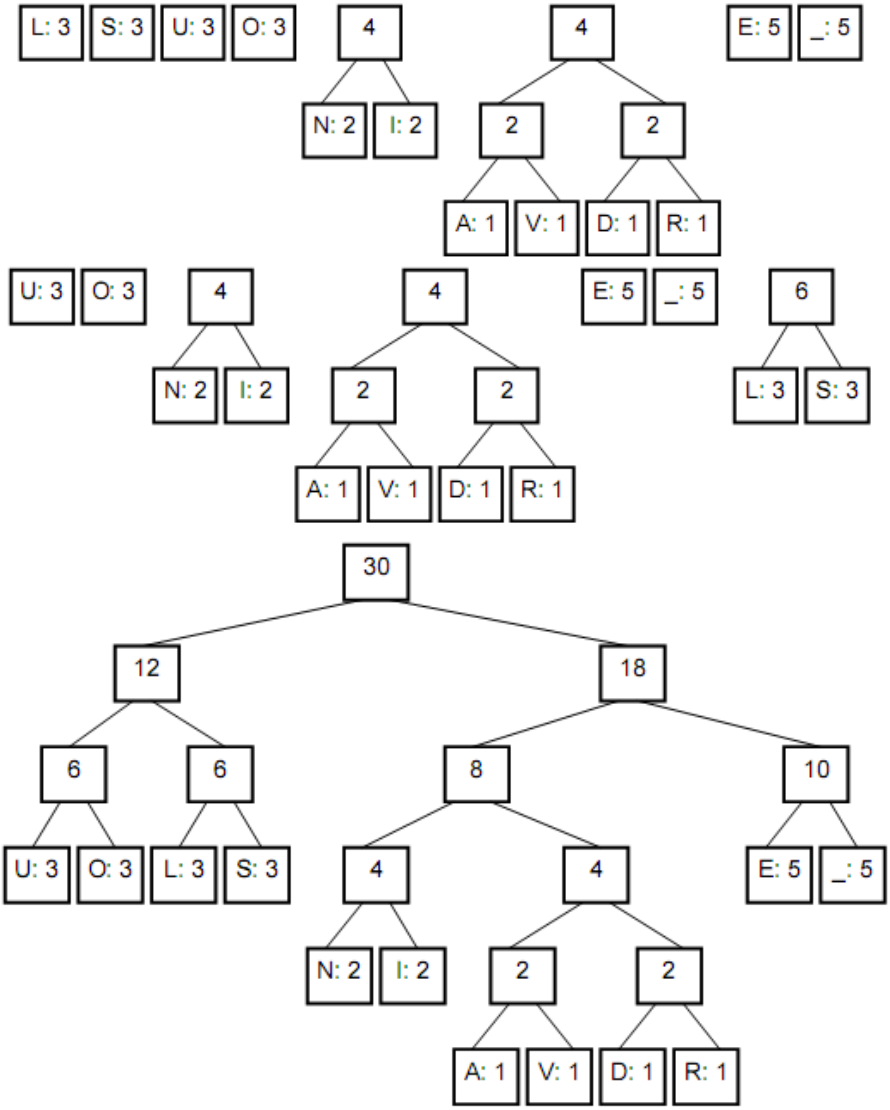
D	R	A	V	N	I	L	S	U	O	E	_
1	1	1	1	2	2	3	3	3	3	5	5



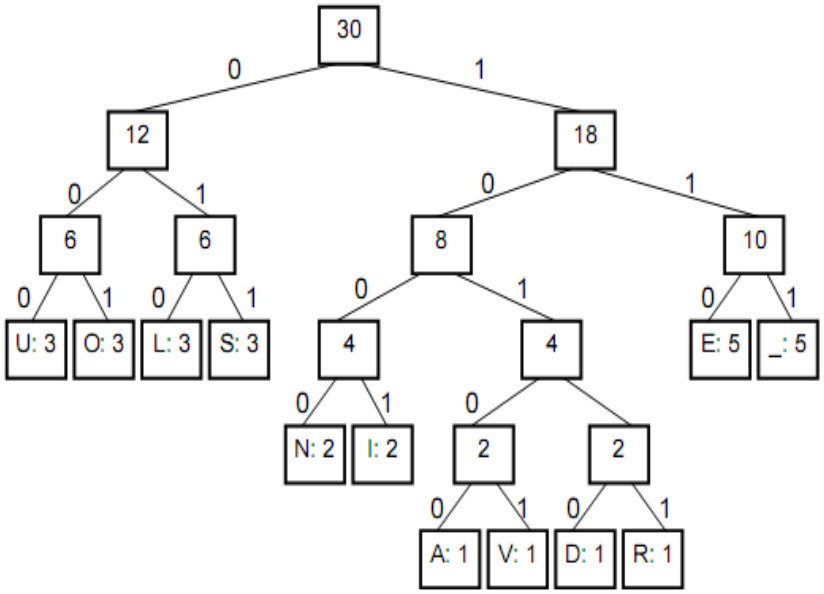
3. On construit un arbre pour donner le code binaire de chaque caractère.

Construction de l'arbre : on relie deux à deux les caractères de fréquence les plus basses et on affecte à ce nœud la somme des fréquences des caractères. Puis on répète ceci jusqu'à ce que l'arbre relie toutes les lettres.





L'arbre étant construit, on met un 1 sur la branche à droite du nœud et un 0 sur celle de gauche.



lettre	code
_	111
E	110
O	001
U	000
S	011
L	010
I	1001
N	1000
V	10101
A	10100
R	10111
D	10110

Un nouveau code est donné à chaque lettre. Il est construit par le chemin d'accès à la feuille correspondante, depuis la racine, en notant 0 la branche de gauche et 1 la branche de droite. Et si on recode la phrase de départ, on obtient :

R I E N _ D E _
10111 1001 110 1000 111 10110 110 111
N O U V E A U _
1000 001 000 10101 110 10100 000 111
S O U S _ L E _ S O L E I L
011 001 000 011 111 010 110 111 011 001 010 110 1001 010

Ce qui fait 102 bits au lieu de 240 au départ, soit un taux de compression de 2,35 : 1.
Le fichier compressé se compose d'une suite de codes sans séparateur, bien que les codes comportent un nombre variable de bits, car aucun code ne peut être le préfixe d'un autre.

Algorithme de décompression :

On transmet la bibliothèque de codage, puis on associe les caractères à leur code.

2.1.3 Codage LZW (Lempel-Ziv-Welch)

- Trouver des séquences de pixels qui se répètent.
- Construire un dictionnaire associant un code unique à chaque séquence.
- Remplacer la séquence de pixels par le code.

➔ Algorithme utilisé par les formats GIF, TIFF, ...

Remarque :

- Intéressant lorsqu'il y a peu de couleurs.
- Algorithme non spécifique aux images.

2.1.4 Bilan des méthodes de compression sans perte

- Méthodes surtout utilisables sur des images comportant peu de couleurs (9 256 couleurs).
- Taux de compression dépendant du contenu des images mais souvent peu important.

➔ D'autres méthodes de compression ont été mises au point, spécifiques aux images comportant beaucoup de couleurs (16 millions), permettant des taux de compression très élevés en autorisant des pertes de qualité.

2.2 Compression avec pertes

Principe : suppression des informations les moins indispensables pour l'œil humain.
Réduction du nombre de données

➔ Taux de compression plus élevé que compression sans perte.

➔ Plus le taux de compression est élevé, plus le niveau de pertes est important et plus la qualité d'image est dégradée.

Original

JPEG

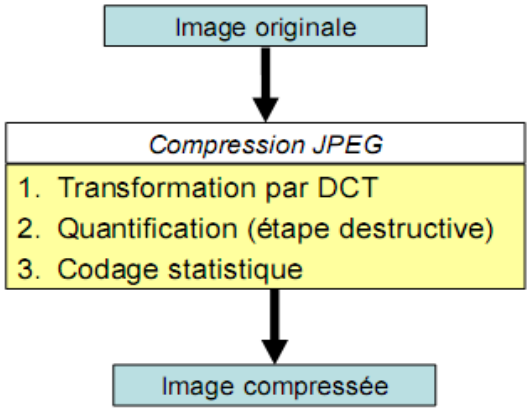
Image 512x512 pixels, 16 millions de couleurs :

- Brut : 769 Ko
- LZW : 631 Ko (ratio = 1.2 : 1)
- PEG : 52 Ko (qualité 75 %) (ratio = 15 : 1)

2.2.1 Compression JPEG

JPEG = standard mis au point par le Joint Photographic Experts Group (1992). Méthode de compression d'images la plus utilisée.

Principe :



1) Transformation par DCT (Discret Cosinus Transform)

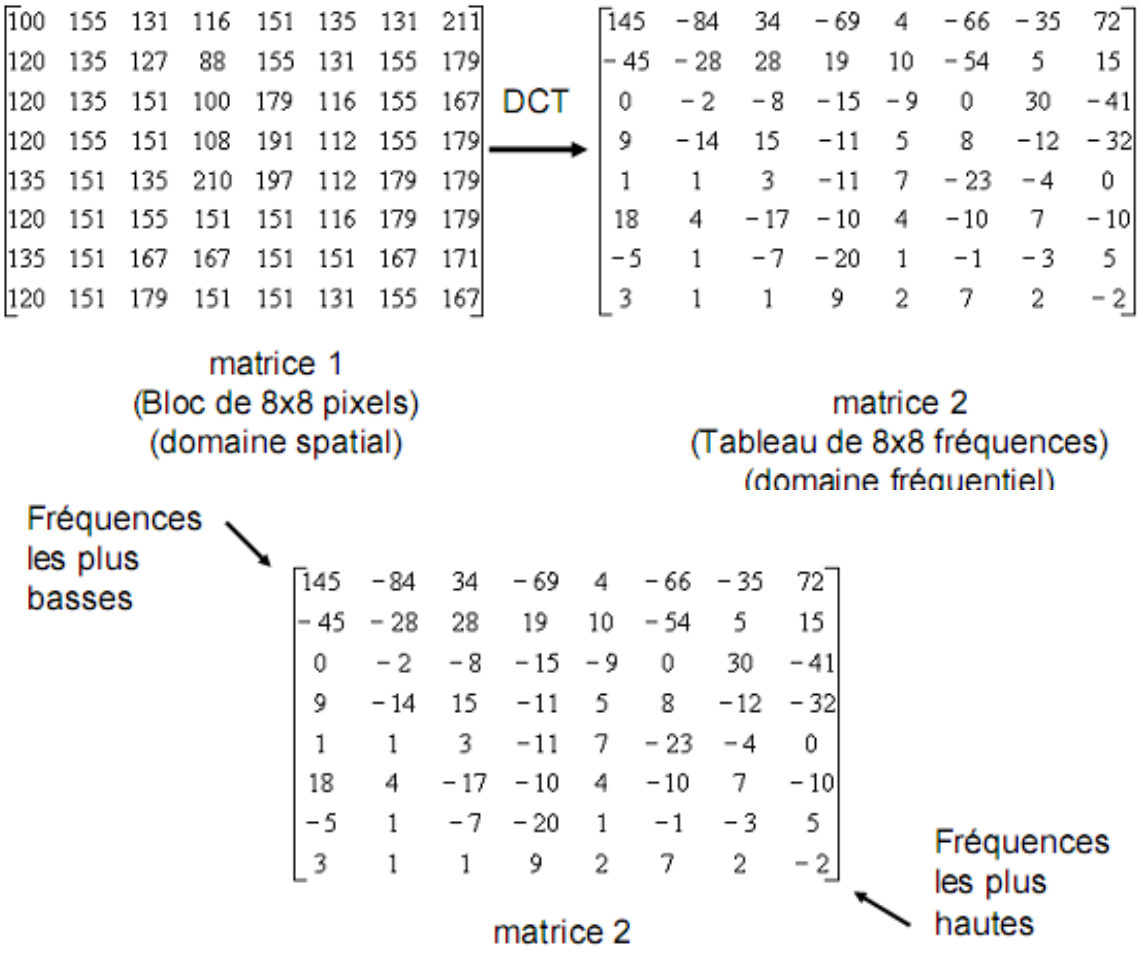
➔ Analyse de l'image par blocs de 8x8 pixels (tableaux de 64 valeurs).

➔ Application d'une DCT (Transformée de Cosinus Discrète) au tableau de 64 valeurs ~ tableau de 64 fréquences (opération complètement réversible)

La formule de la DCT :

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos\frac{\pi(2m+1)p}{2M} \cos\frac{\pi(2n+1)q}{2N}, \quad \begin{matrix} 0 \le p \le M-1 \\ 0 \le q \le N-1 \end{matrix}$$

$$\alpha_p = \begin{cases} 1/\sqrt{M}, & p = 0 \\ \sqrt{2/M}, & 1 \le p \le M-1 \end{cases} \qquad \alpha_q = \begin{cases} 1/\sqrt{N}, & q = 0 \\ \sqrt{2/N}, & 1 \le q \le N-1 \end{cases}$$



- Basses fréquences : plage de couleurs uniforme.
- Hautes fréquences : variations brusques de couleur d'un pixel à l'autre.

On observe que les basses fréquences ont des coefficients plus élevés (donc plus significatifs) que les hautes fréquences.

Idée principale de la compression JPEG :

Retirer des détails pour diminuer la taille du fichier image en supprimant les hautes fréquences.
Le taux de compression va dépendre du seuil de l'écrêtage : plus on conserve des composantes à amplitude faible, meilleure sera la qualité (car plus il y aura de détails fins), mais plus faible sera le taux de compression.

2) Quantification

Diviser chaque valeur de la matrice de fréquences (matrice 2) par une matrice de quantification (matrice 3)

➔ On obtient une matrice quantifiée (matrice 4) dans laquelle beaucoup de coefficients sont nuls.

$$\begin{bmatrix} 145 & -84 & 34 & -69 & 4 & -66 & -35 & 72 \\ -45 & -28 & 28 & 19 & 10 & -54 & 5 & 15 \\ 0 & -2 & -8 & -15 & -9 & 0 & 30 & -41 \\ 9 & -14 & 15 & -11 & 5 & 8 & -12 & -32 \\ 1 & 1 & 3 & -11 & 7 & -23 & -4 & 0 \\ 18 & 4 & -17 & -10 & 4 & -10 & 7 & -10 \\ -5 & 1 & -7 & -20 & 1 & -1 & -3 & 5 \\ 3 & 1 & 1 & 9 & 2 & 7 & 2 & -2 \end{bmatrix}$$ matrice 2

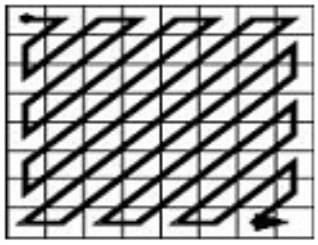
$$/ \begin{bmatrix} 6 & 11 & 16 & 21 & 26 & 31 & 36 & 41 \\ 11 & 16 & 21 & 26 & 31 & 36 & 41 & 46 \\ 16 & 21 & 26 & 31 & 36 & 41 & 46 & 51 \\ 21 & 26 & 31 & 36 & 41 & 46 & 51 & 56 \\ 26 & 31 & 36 & 41 & 46 & 51 & 56 & 61 \\ 31 & 36 & 41 & 46 & 51 & 56 & 61 & 66 \\ 36 & 41 & 46 & 51 & 56 & 61 & 66 & 71 \\ 41 & 46 & 51 & 56 & 61 & 66 & 71 & 76 \end{bmatrix}$$ matrice 3

$$= \begin{bmatrix} 24 & -7 & 2 & -3 & 0 & -2 & 0 & 1 \\ -4 & -1 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$ matrice 4

La matrice de quantification est définie à partir de critères psycho-visuels et du taux de compression souhaité.

3) Codage statistique

$$\begin{bmatrix} 24 & -7 & 2 & -3 & 0 & -2 & 0 & 1 \\ -4 & -1 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$ matrice 4



Séquence zigzag

1. La matrice quantifiée contient beaucoup de 0.
2. Les coefficients de la matrice sont parcourus selon une séquence en zigzag --> longues suites de 0 consécutifs

➔ compression RLE.

3. Les coefficients non nuls sont compressés avec la méthode statistique de Huffman.

➔ Taux de compression très élevés.

22.2 Algorithme de décompression JPEG

Procédé inverse de celui de la compression. Pour chaque bloc 8x8 de pixels :

- Décompression par la méthode de Huffman.
- Multiplication de la matrice obtenue par la matrice de quantification.
- Application de la DCT inverse pour retrouver une image plus ou moins dégradée par rapport à l'image initiale.

La formule inverse de la DCT :

$$A_{mn} = \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} \alpha_p \alpha_q B_{pq} \cos\frac{\pi(2m+1)p}{2M} \cos\frac{\pi(2n+1)q}{2N}, \quad \begin{matrix} 0 \le m \le M-1 \\ 0 \le n \le N-1 \end{matrix}$$

$$x_p = \begin{cases} 1/\sqrt{M}, & p = 0 \\ \sqrt{2/M}, & 1 \le p \le M-1 \end{cases} \qquad \alpha_q = \begin{cases} 1/\sqrt{N}, & q = 0 \\ \sqrt{2/N}, & 1 \le q \le N-1 \end{cases}$$

Performances de la compression JPEG :

Original (769 Ko) | JPEG 99% (200 Ko) Ratio = 4:1 | JPEG 1% (6 Ko) Ratio = 128:1

Artefacts de la compression JPEG

- Compression avec pertes --> baisse de qualité de l'image.
- Apparition de blocs (car traités séparément), d'autant plus visibles que le taux de compression augmente.

Bibliographie

[01] T Guyer. « Cours de traitement d'images ». Université de Chambéry.

[02] R.C. Gonzalez et Woods. « Digital Image Processing - 2d edition ». Addison Wesley 2002.

[03] JC Burie. « Cours de traitement d'images ». Université de La Rochelle.

[04] R.C. Gonzalez, Richard E. Woods (Auteur), Steven L. Eddins. « Digital Image Processing Using MATLAB® ». 16 décembre 2003.

[05] I.T. Young, J.J. Gerbrands et L.J. Van Vliet. « Fundamentals of Image Processing ». Université de Delft.

[06] Alain Boucher. « Cours de traitement d'images ». Institut de la francophonie pour l'informatique.

[07] J.P. Cocquerez et S. Philipp. « Analyse d'images : filtrage et segmentation ». Masson 1995.

[08] Caroline ROUGIER. « Cours de traitement d'images ». Iniversité de Montréal

[09] Weidong Kou. « Digital Image Compression: Algorithms and Standards ». The Springer International Series in Engineering and Computer Science 1995.

[10] Elise Arnaud – Edmond Boyer. « Cours de traitement d'images ». Université Joseph Fourier

[11] Cocquerez. « Analyse d'images : filtrage et segmentation ». Edition Broché.

[12] V Eglin. « Cours de traitement d'images ». INSA de Lyon.