

CHAP II Présentation d'une chaîne de traitement d'image

1. Chaîne de vision par ordinateur. Prétraitement - Amélioration d'images

Rappel : L'essor du traitement des images est lié d'une part à l'avènement des robots de 3ème génération ($\equiv$ capables de percevoir et réagir avec l'environnement) et d'autre part à la puissance de calcul que permet le traitement digital.

I. CHAÎNE DE VISION PAR ORDINATEUR

a) Capteur : caméra vidéo, CCD, ... , plaque photosensible aux rayonnements (Gama, X, ...)

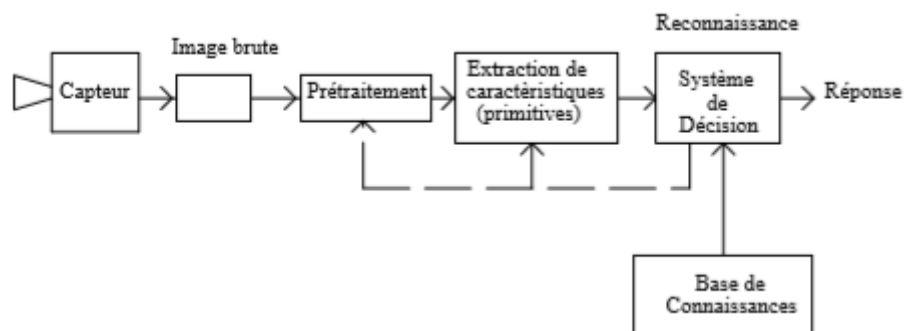
b) Prétraitement : amélioration de l'image (renforcement du contraste, réduction du bruit ...)

c) Extraction de caractéristiques : contours, régions, angles, segments de droites,...

Les caractéristiques ou primitives sont compatibles avec la représentation qui en est donnée dans la Base de Connaissances (mémoire).

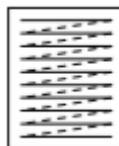
d) Système de Décision : utilise des méthodes de Reconnaissance Des Formes (RDF) communes à d'autres disciplines (reconnaissance de la parole, ...)

Analogie : le cerveau corrige l'image floue et déformée issue de l'oeil (même chose pour les oreilles !) et effectue de nombreux prétraitements (occupant ainsi des ressources du cerveau, et, chez les personnes âgées, cette consommation de ressources peut se faire au détriment de l'attention).



Caméra : - Caméra à tube : Un faisceau électronique ($\equiv$ d'électrons) explore une cible photoconductrice.

Balayage standard : 1/25 ème de seconde pour une image complète. Image de 625 lignes (balayage télévision) (25 images/seconde)



A chaque ligne correspond un signal analogique proportionnel à l'intensité lumineuse (luminance).

Caméra CCD : (Charge Coupled Device) : Assemblage de photodiodes délivrant chacune une intensité proportionnelle à l'intensité lumineuse d'un point de l'image (pixel) →échantillonnage opéré ($\neq$ signal continu fourni par les caméras à tube). (Pas de balayage

CHAP II Présentation d'une chaîne de traitement d'image

télévision comme pour les caméras à tube). Le couple constitué du nombre de pixels en X et en Y est appelé résolution ou définition.

Caméra CCD matricielle : photodiodes disposées en une matrice de n lignes et p colonnes. Résolutions courantes : $NX \times NY = 256 \times 256, 320 \times 200, 512 \times 512, 640 \times 480, 800 \times 600, 1024 \times 768$.

- **Caméra uniligne : (scanner)** : photodiodes disposées en une seule ligne. Résolution supérieure (2048 points par exemple). L'obtention d'une image implique le déplacement du scanner (sur un document par exemple) pour avoir une image bi dimensionnelle par acquisition de plusieurs lignes.

- **Processeur analogique** : gestionnaire de l'entrées-sorties vidéo du système.

Fonctions : - sélection de la caméra active, de l'écran d'affichage - acquisition du signal vidéo : numérisation (digitalisation) et contrôle de mémorisation/affichage :

2) Numérisation = échantillonnage et quantification (codage) : découpage en pixels valeur numérique (sur un nombre de m bits) (échantillonnage spatial) donnée à l'intensité lumineuse par arrondi → niveau de gris

a) Echantillonnage

Contrairement au son pour lequel l'échantillonnage est temporel (période d'échantillonnage de $(1 / 44\,100)$ seconde pour une qualité CD), l'échantillonnage d'une image est spatial :

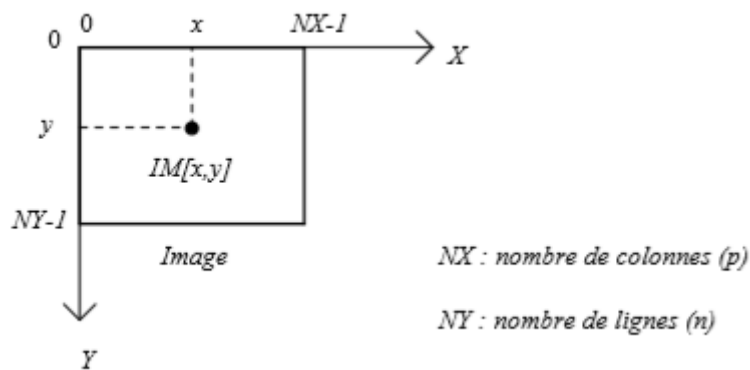
ex.: image aérienne, image satellite : 1 pixel représente 1 mètre au sol par exemple. A vitesse V constante, la distance d est directement liée (proportionnelle) au temps t : $V = d/t \rightarrow d = V.t$. On peut donc considérer indifféremment l'image comme un signal spatial ($\equiv$ dépendant de la distance) ou temporel ($\equiv$ dépendant du temps) et du même coup, l'échantillonnage peut aussi être considéré indifféremment spatial ou temporel.

b) Quantification

Codages classiques sur 16, 32, 64, 256 niveaux de gris correspondant respectivement à un nombre m de bits $m = 4, 5, 6, 8$. Ce nombre de niveaux de gris disponibles (2^m) s'appelle la dynamique de l'image (à partir de 64 niveaux, l'oeil ne fait plus beaucoup de différences).

Mémoire d'image (frame buffer) : Image digitalisée = tableau 2D IM de n lignes et p colonnes : $IM[0: p-1, 0: n-1]$ (n, p entiers > 0) $IM[x, y]$ = niveau de gris du pixel de coordonnées x et y : $0 \leq x \leq p-1$ $0 \leq y \leq n-1$ (x, y entiers) $IM[x, y]$ entier codé sur m bits

CHAP II Présentation d'une chaîne de traitement d'image



L'image peut aussi être mémorisée sous forme 1D : $IM[0 : n.p - 1]$: les lignes sont stockées bout à bout. Un calcul d'adresse permet d'accéder à l'index i du pixel (x, y) : $i = y.p + x$.

En général : Niveau de gris : $0 \rightarrow \text{noir}$ (2^m-1) $\rightarrow$ blanc (255 pour $m = 8$ bits)

Les mémoires d'images réservent en général : 8 bits (mot standard) pour la dynamique ($\rightarrow$ 256 niveaux) ou 24 bits (1 octet par couleur : R (Rouge), V (Vert) ou B (Bleu) ($2^{24} \approx 16.7$ millions de couleurs).

Résolutions courantes : $n \times p = 256 \times 256$ ou $n \times p = 512 \times 512$. (Bon compromis entre la définition de l'image et le temps de traitement nécessaire). Elles possèdent en général plusieurs plans mémoire pour traiter plusieurs images simultanément. Certaines possèdent des opérateurs intégrés (zoom, histogramme, opérateurs logiques ...).

- Le Processeur numérique d'images

Processeur(s) câblé(s) ou programmable(s) contenant les algorithmes de traitement. (Temps Réel $\rightarrow$ temps de traitement $<$ temps de présence image de $1/25^{\text{ème}}$ de seconde avant l'arrivée de la suivante). (Le traitement comprend ici l'algorithme mais aussi le chargement et le stockage ou affichage de l'image).

- L'Ordinateur hôte : Il contrôle : - le processeur analogique - les échanges mémoire.

Il effectue les traitements s'il n'y a pas de contrainte temps réel.

- Mémoire de masse

Stockage des fichiers images.

Exemple : une image noir et blanc (N/B) 256×256 sur 256 niveaux occupe déjà 64 KOctets!

3. Représentation des images

Types d'Images

De nombreux types de format d'image existent, incluant éventuellement une compression des données : formats de fichier d'extension .BMP de Windows, .TIF, PCX, .GIF, .JPG(JPEG), .MPG (MPEG) (images animées), ...

Exemple :

CHAP II Présentation d'une chaîne de traitement d'image

Une image en niveaux de gris ($\equiv$ monochrome) de taille $NX \times NY$ de dynamique $DYN=256$ niveaux peut être codée par un plan mémoire (Bitmap) de $NY \times NX$ octets consécutifs en stockant ligne après ligne l'image. On peut aussi coder cette image par une table dont chaque case représente non plus directement la luminance d'un pixel mais un index référant à une palette de niveaux de gris située en tête ou en fin de fichier. Ces 2 types de codage d'image revêtent le nom d'images mosaïque, ou raster ou encore bitmap. L'objet graphique élémentaire est le pixel (picture element).

On trouve aussi le format d'images vectoriel pour les images constituées de segments de droites (polygones) (images de synthèse). Après l'en-tête de fichier, la liste des sommets et leur correspondance est mémorisée. L'objet graphique élémentaire est le segment. Le format d'images vectoriel est l'équivalent du format midi pour la synthèse de la musique, et le format bitmap d'images d'acquisition, l'équivalent du format wav pour l'acquisition de musique.

Pour les images couleur ($\equiv$ multispectrales), il en est de même :

Index sur une palette de couleurs ou codage Bitmap non plus du niveau de gris mais de 3 luminances pour chaque pixel : niveau de Rouge, niveau de Vert et niveau de Bleu (codage RVB) ou des 3 informations équivalentes : Teinte, Saturation, Luminosité (codage ILS), ...

En traitement d'images, l'objectif ultime est la Reconnaissance Des Formes : pour cela, on réalise des prétraitements d'amélioration et ensuite on s'attache à ne retenir que les informations nécessaires à la compréhension de l'image pour se donner une représentation minimale de l'image, apte à permettre la reconnaissance (extraction des contours ...).

Autant dire que l'image brute en niveaux de gris et à fortiori en couleurs est trop riche en informations (il y en a trop !). On peut quasiment toujours comprendre une image sans avoir tout le flot d'informations présentes dans une image en niveaux de gris ou en couleurs (la compréhension est souvent une réduction d'informations).

On veillera à ne retenir que les informations pertinentes de l'image et à les insérer dans une structure adaptée à sa reconnaissance.

Voilà pourquoi on ne travaillera pas avec des images couleur mais des images en niveaux de gris. (Il faudrait 3 fois plus de traitement sans pour autant aider de façon notable la reconnaissance).

La conversion d'une image couleur RVB en niveaux de gris N/B se fait par la relation issue de la colorimétrie :

Niveau de gris = $30\% \cdot R + 59\% \cdot V + 11\% \cdot B$.

Les images utilisées en Travaux Pratiques sont donc en niveaux de gris et codées avec le format bmp. Sauf indication contraire, tous les exemples donnés dans ce qui suit feront référence à une image $IM[0:255,0:255]$ codée sur 256 niveaux de gris ($0 \rightarrow 255$), dont le nombre de colonnes est noté NX et le nombre de lignes NY .

Caractéristiques principales des images raster

Une image rectangulaire bitmap est caractérisée par :

- le nombre de colonnes NX de l'image

CHAP II Présentation d'une chaîne de traitement d'image

- le nombre de lignes NY
- la dynamique (nombre de bits par pixel) : le nombre de niveaux de quantification varie de 2 (noir ou blanc des dessins « au trait » des scanners), à 256 par composante de couleur (256 niveaux de Rouge, 256 niveaux de Vert, 256 niveaux de Bleu) (ou 256 niveaux de gris pour les images monochromatiques → 8 bits par pixel).
- la résolution spatiale (nombre de pixels par unité de surface, indiquant ainsi la distance que représente 1 pixel). La résolution spatiale d'une image bitmap est l'équivalent du taux d'échantillonnage temporel (fréquence d'échantillonnage) d'un fichier audio non synthétique (format type wav/mp3 ...).

Sur un écran d'ordinateur, la résolution spatiale est d'environ 75 dpi ou ppp (dot per inch, point par pouce), meilleure que la télévision (environ 25 dpi pour un écran télévision de 50 cm). La résolution spatiale passe à 300 dpi pour une imprimante (et à une dizaine de milliers de dpi pour une diapositive). Le satellite SPOT fournit des images correspondant à une distance au sol de l'ordre de la dizaine de km. Les meilleurs satellites américains peuvent descendre à une résolution spatiale de l'ordre du cm.

Une résolution de 300 dpi correspond à un pixel de 85 µm de côté (1 pouce = 2.54 cm).

Etapes de l'analyse d'images

La plupart des applications de traitement d'image passent par les étapes suivantes : • Acquisition : Echantillonnage, Quantification.

• **Analyse globale de l'image et transformations ponctuelles** : Histogramme ; Statistiques : moyenne, écart-type, etc... Transcodage (palette de couleur [LUT]) et classification

• **Opérations entre images : Indices, ratio** ; Différence ; Opérations logiques, masques ; Seuillage • Amélioration, filtrage et segmentation : Anamorphose (≡ recadrage de dynamique) ; Opérations de convolution : Lissage, Rehaussement, Détection de contours ; Squelettisation ; Vectorisation

• **Interprétation et sémantique** : Cartographie thématique : classification automatique et supervisée ; Cartographie vecteur ; Représentation des graphes et de la topologie.

4. PRETRAITEMENT - AMELIORATION D'IMAGES

1. Amélioration et restauration d'images

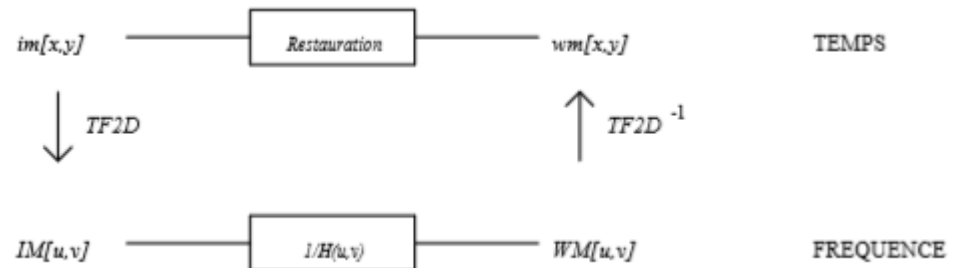
Bruits présents dans l'image :

- **sur éclairage ou sous- éclairage** → saturation ou mauvais rapport S/B de la caméra.
- **distorsions optiques** : aberrations, astigmatisme, problèmes de prise de vue : mise au point.
- bruit électronique lié au capteur (bruit thermique des composants).
- défauts du système de numérisation.
- **Restauration (image restoration)**

CHAP II Présentation d'une chaîne de traitement d'image

Restitution de l'image aussi proche que possible qu'avant dégradation. → détermination du modèle mathématique du processus de la dégradation (Fonction de Transfert 2D : FT 2D: $H(u, v)$), et application du modèle inverse ($1/H(u, v)$) sur l'image. Détermination de 2 manières : à priori et à postériori :

- **à priori** : modèle $H(u, v)$ construit à partir d'images tests (identification) :



Inconvénient : $H(u,v)$ peut être nul dans un domaine et le filtre inverse $1/H(u,v)$ est indéterminé ou instable. On préfère alors déterminer la Réponse Impulsionnelle (RI) du filtre qui minimise l'erreur (critère de moindres carrés) entre l'image restaurée et l'image non dégradée (filtrage optimal). De plus, les paramètres du filtre évoluent dynamiquement pour adapter la restauration (filtrage adaptatif): c'est la méthode à postériori: - à postériori (filtre adaptatif) : restauration à l'aide de mesures réalisées sur l'image à restaurer, selon un processus adaptatif. Ces méthodes sont lourdes et encore peu utilisées à cause de leur coût élevé en temps calcul, et de la nécessité de la connaissance du processus de dégradation. Exemple : Restauration d'une image dégradée à cause du « bougé » du photographe lors de la prise de vue. La dégradation se traduit par un flou et une traînée sur l'image, la rendant « illisible ». La modélisation (direction, vitesse) du bougé autorise la restauration de l'image en appliquant la transformation inverse :



- **Amélioration d'images (image enhancement)** : Prétraitements visant à atténuer les bruits dans l'image sans nécessiter la connaissance du modèle de la dégradation.

Différentes approches :

- à partir de l'histogramme de l'image (méthode statistique).
- par lissage :
- linéaire (filtrage linéaire)
- non linéaire

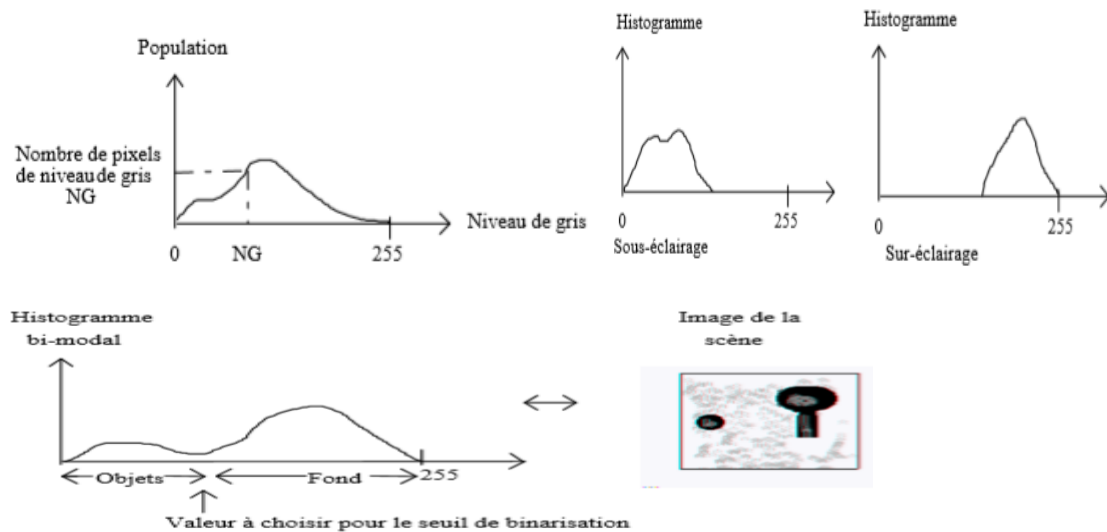
2. Histogramme d'une image (traitement statistique de l'image)

But : L'histogramme des niveaux de gris d'une image donne la fréquence d'apparition ($\equiv$ population) de chaque niveau de gris dans l'image :

Inconvénient du traitement statistique : Perte de l'information de localisation.

CHAP II Présentation d'une chaîne de traitement d'image

Avantage du traitement statistique : Traitement global.



Algorithme du calcul de l'Histogramme d'une image de taille $NX \times NY$ et de dynamique D :

```

Début
  Mise à zéro des D éléments du vecteur Histo ;
  Pour tout point [x,y] de l'image IM Faire
    Histo [ IM [x,y] ] := Histo [ IM [x,y] ] + 1 ;   ou   n := IM [x,y];
    Histo_n := Histo_n + 1 ;
  Fp
Fin
  
```

L'itération : "Pour tout point $[x,y]$ de l'image IM" consiste en un Balayage de l'image par exemple horizontal (ligne par ligne d'image), de haut en bas et de gauche à droite :

```

  Pour y = 0 à NY-1 Faire
    Pour x = 0 à NX-1 Faire
      Fp
    Fp
  Fp
  
```

5. Transformations d'histogrammes

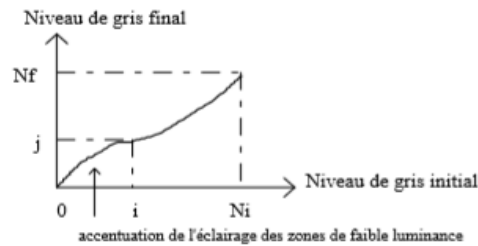
5.1. Table de conversion : LUT : Look Up Table

LUT : fonction qui transforme un niveau de gris i en un niveau de gris j , sans modification spatiale de l'image.

But : Elle permet de gagner du temps calcul. (Les calculs sont effectués une fois pour toutes et mémorisés dans la LUT - on applique alors le transcodage (simple lecture d'un vecteur : le vecteur LUT)).

CHAP II Présentation d'une chaîne de traitement d'image

Exemple :



Algorithme utilisant une LUT :

```

Début
  Calculer initialement et une fois pour toutes la LUT : Constitution de la LUT : for i = 0 to 255 LUT[i] = 255-i
  Pour tout point [x,y] de IM Faire
    IM [x,y] := LUT [ IM [x,y] ] ; /* l'image de départ IM est modifiée */
    ou
    WM [x,y] := LUT [ IM [x,y] ] ; /* l'image de départ IM n'est pas modifiée */
  Fp
Fin
  
```

Exemple : LUT d'inversion vidéo :

i	0	1	2	...	254	255
LUT [i]	255	254	253	...	1	0

Inversion vidéo sans LUT : (inversion vidéo algorithmique)

```

Début
  Pour tout point [x,y] de IM Faire
    IM [x,y] := 255 - IM [x,y] ; /* l'image de départ IM est modifiée */
    ou
    WM [x,y] := 255 - IM [x,y] ; /* l'image de départ IM n'est pas modifiée */
  Fp
Fin
  
```

5.3. Binarisation d'image (≡Seuillage d'image)

But : Réduction importante de la quantité d'information (l'image résultat est codée sur 2 niveaux → dynamique réduite à 2 : image bi-niveaux).

L'image reste compréhensible dans la plupart des cas.



Algorithme de Binarisation à 1 seuil :

```

Début
  Pour tout point [x,y] de IM Faire
    si IM [x,y] > seuil alors WM [x,y] := 255 sinon WM [x,y] := 0
  Fp
Fin
  
```

Le seuil peut être choisi de façon statistique par utilisation de l'histogramme (ex. : caractère bimodal de l'histogramme,...)

CHAP II Présentation d'une chaîne de traitement d'image

Résultat du seuillage (1 seuil) :

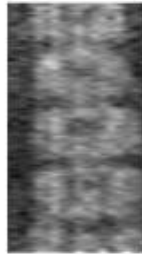


Image initiale (vertebre.bmp)



Image seuillée (seuil = 110)

6.4. Egalisation d'Histogramme (Rehaussement d'image)

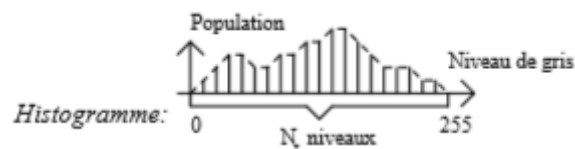
Problème : l'image naturelle quantifiée de manière linéaire présente une majorité de pixels de luminance $<$ luminance moyenne : les détails dans les régions sombres sont difficilement perceptibles.

But : augmenter la dynamique pour les niveaux de gris faibles sans en perdre sensiblement pour les niveaux élevés. (L'amélioration du rapport Signal/Bruit transforme la quantification linéaire en une quantification exponentielle $\rightarrow$ compression logarithmique).

$\rightarrow$ **Égalisation :** modification de l'histogramme (et aussi de l'image) pour qu'il soit uniforme (population constante pour tous les niveaux de gris). (On peut généraliser à une modification telle que la forme de l'histogramme soit exponentielle, logarithmique...)

L'idée est de modifier la répartition des niveaux pour obtenir un histogramme plat étendu à l'ensemble des valeurs possibles. Dans cette opération la dynamique originale $[\min, \max]$ est étalée à $[0, 255]$. On cherche à affecter le même nombre de pixels à chaque niveau de gris ; c'est pourquoi on appelle cette opération équipopulation. L'effet obtenu permet de mieux séparer les valeurs les plus représentées dans l'image et de rapprocher les valeurs marginales. Cela a pour effet d'améliorer le contraste de l'image. Cette transformation peut aussi être appliquée pour réduire le nombre de niveaux pour la visualisation ou pour l'impression (passer de 256 à 16 niveaux) ; elle s'apparente alors à une classification puisqu'il s'agit de représenter plusieurs valeurs initiales par la même valeur finale.

Soit une image de taille $p \times p$ ayant N_I niveaux (ex: image 256×256 sur 256 niveaux) :



CHAP II Présentation d'une chaîne de traitement d'image

7. Filtrage

Pourquoi filtrer une image ?

- Pour réduire le bruit dans l'image (sujet de ce cours)
- Pour détecter les contours d'une image.
- Convolution entre une image f et un filtre h , appelé aussi masque de convolution
- Opération de voisinage qui effectue une combinaison linéaire (ou non) de pixels de l'image f , produisant une nouvelle image f'
- h est un opérateur sur f défini en chaque pixel (i, j) et sur son voisinage

Phénomène parasite aléatoire (suivant une distribution de probabilité connue ou non) dont les origines sont divers (capteur, acquisition, lumière, ...)

Dans le cas du filtrage linéaire, on considère que le bruit est additif

Pour le cas du bruit additif, si fb est l'image alors on peut l'écrire de la forme :

$$fb(i,j) = f(i,j) + b(i,j).$$

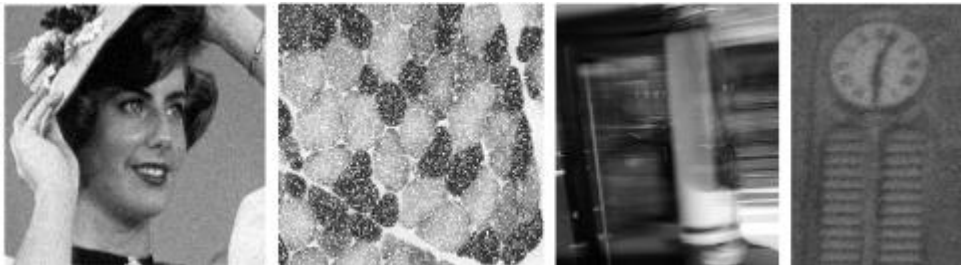
Exemples de bruits additifs : bruits gaussiens et impulsionnels

• **Bruit gaussien** : $I(x,y) = I(x,y) + N(0,\sigma)$

• **Bruit impulsionnel** ("poivre et sel") d'ordre n : ajouter n pixels blancs et n pixels noirs aléatoirement dans l'image caractérisé par un % de pixels remplacés.

Autres types de bruits : flou (convolutif), grain (multiplication)

Exemples d'images bruitées



➤ Hypothèse fondamentale pour la réduction du bruit :

- Signal utile et bruit ont des composantes fréquentielles différentes
- Signal utile $\leftrightarrow$ basses fréquences
- Bruit $\leftrightarrow$ hautes fréquences

Il existe deux grands types de filtres :

a) Filtrage linéaire

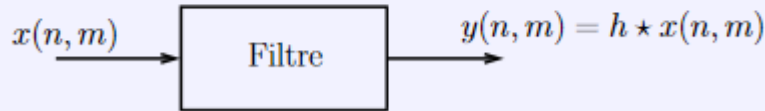
b) Filtrage non linéaire

Filtrage spatial

Filtrage linéaire spatial : Convoluer l'image avec la réponse impulsionnelle du filtre.

CHAP II Présentation d'une chaîne de traitement d'image

$$T[x(n, m)] = \sum_{k=-\infty}^{+\infty} \sum_{l=-\infty}^{+\infty} x(k, l)h(n-k, m-l) = x \star h(n, m)$$



Dans le domaine spatial Comment s'interprète le calcul de la convolution ?

- ✓ Comment fait-on le calcul de la convolution pour des images (signaux discrets 2D) ?

Produit de convolution 2D discret

Le produit de convolution d'un signal 2D $f(i, j)$ (une image) avec un filtre $h(i, j)$ est donné par :

$$f'(i, j) = (f \star h)(i, j) = \sum_{n=-\infty}^{+\infty} \sum_{m=-\infty}^{+\infty} f(i-n, j-m)h(n, m)$$

On distingue 2 situations :

- 1) Filtres de Réponse Impulsionnelle $h(n, m)$ Finie (RIF)
- 2) Filtres de Réponse Impulsionnelle $h(n, m)$ Infinie (RII).

Filtrage par convolution

Filtres de Réponse Impulsionnelle Finie (RIF) v.s. Infinie (RII)

- 1) Filtres de Réponse Impulsionnelle $h(n, m)$ Infinie (RII) : $h(n, m)$ ne s'annule pas
- 2) Filtres de Réponse Impulsionnelle $h(n, m)$ Finie (RIF) : $h(n, m)$ est nul en dehors d'un intervalle en n et m .

•On parle de masque de convolution

•Par exemple, si $h(n, m)$ est un masque carré de taille d impaire, centré en $(0,0)$,

$h(n, m) \neq 0$ pour $|n| \leq \frac{d-1}{2}$ et $|m| \leq \frac{d-1}{2}$:

$$f'(i, j) = (f \star h)(i, j) = \sum_{n=-\frac{d-1}{2}}^{\frac{d-1}{2}} \sum_{m=-\frac{d-1}{2}}^{\frac{d-1}{2}} f(i-n, j-m)h(n, m)$$

Filtrage par convolution Filtres de Réponse Impulsionnelle Finie (RIF)

Exemple : si $h(n, m)$ est un masque carré de taille $d=3$ impaire :

$$f'(i, j) = (f \star h)(i, j) = \sum_{n=-1}^1 \sum_{m=-1}^1 f(i-n, j-m)h(n, m)$$

$$\begin{aligned} f'(i, j) &= h(1, 1)f(i-1, j-1) + h(1, 0)f(i-1, j) + h(1, -1)f(i-1, j+1) \\ &+ h(0, 1)f(i, j-1) + h(0, 0)f(i, j) + h(0, -1)f(i, j+1) \\ &+ h(-1, 1)f(i+1, j-1) + h(-1, 0)f(i+1, j) + h(-1, -1)f(i+1, j+1) \end{aligned}$$

Si on pose $g(n, m) = h(-n, -m)$: somme pondérée image avec g

$$\begin{aligned} f'(i, j) &= g(-1, -1)f(i-1, j-1) + g(-1, 0)f(i-1, j) + g(-1, 1)f(i-1, j+1) \\ &+ g(0, -1)f(i, j-1) + g(0, 0)f(i, j) + g(0, 1)f(i, j+1) \\ &+ g(1, -1)f(i+1, j-1) + g(1, 0)f(i+1, j) + g(1, 1)f(i+1, j+1) \end{aligned}$$

CHAP II Présentation d'une chaîne de traitement d'image

Principe de calcul de la convolution pour un filtre RIF Soit un filtre RIF de réponse impulsionnelle $h(n, m)$ Au pixel $p=f(i, j)$:

1) Faire une symétrie centrale du noyau par rapport à son centre :

$$h(n, m) \Rightarrow h(-n, -m) = g(n, m)$$

2) Centrer le filtre sur p en le superposant à l'image

3) Effectuer la somme pondérée entre les pixels de l'image et les coefficients du filtre $g(n, m)$

4) Le pixel p dans l'image but (filtrée) aura comme valeur cette somme pondérée

Exemple de convolution

Cas avec un filtre de taille $d=3$:

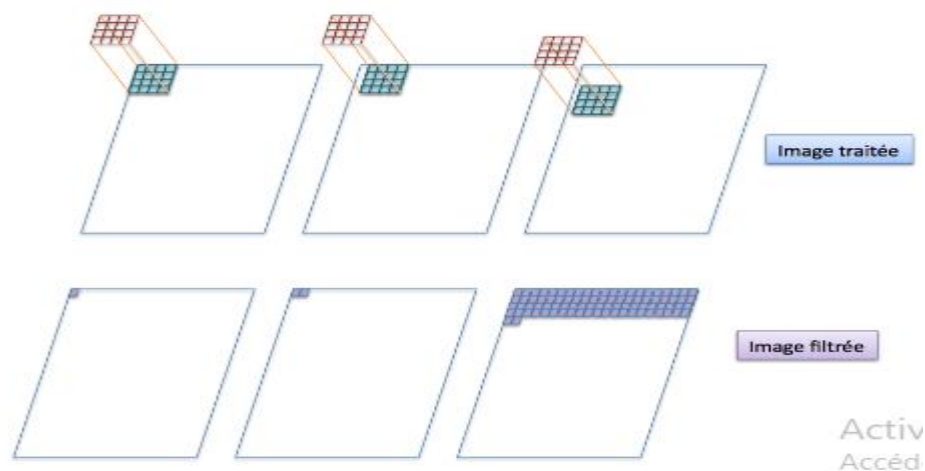
$$h = \begin{pmatrix} w_9 & w_8 & w_7 \\ w_6 & w_5 & w_4 \\ w_3 & w_2 & w_1 \end{pmatrix} \quad g = \begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix}$$

La convolution au pixel (i, j) de f par le noyaux h est donnée par :

$$\begin{aligned} f'(i, j) = & w_1 f(i-1, j-1) + w_2 f(i-1, j) + w_3 f(i-1, j+1) \\ & + w_4 f(i, j-1) + w_5 f(i, j) + w_6 f(i, j+1) \\ & + w_7 f(i+1, j-1) + w_8 f(i+1, j) + w_9 f(i+1, j+1) \end{aligned}$$

Les filtres que nous étudions dans ce chapitre (débruiteurs) doivent conserver la moyenne de l'image ; les coefficients du filtre doivent donc vérifier $\sum_i w_i = 1$.

Principe de la fenêtre glissante



Convolution 2D

Types de convolution Comment faire quand le masque recouvre des zones en dehors de l'image? Convolution linéaire : on considère que l'image est entourée de noir, donc de valeurs nulles : zero - padding Convolution circulante : on considère que l'image est entourée d'elle-même (i.e. support infini de l'image).

Convolution d'une image avec répétition du bord : on considère que l'image est entourée des mêmes valeurs que sur son bord.

CHAP II Présentation d'une chaîne de traitement d'image

Convolution miroir

Filtrage spatial linéaire : exemple de filtres Lissage par moyennage

Propriété : la valeur d'un pixel est relativement similaire à celle de ses voisins.

Dans le cas où l'image contient un bruit et que la propriété précédente est préservée, un moyennage local peut atténuer ce bruit → Cette opération est appelée lissage (smoothing)
Pour effectuer un moyennage dans un bloc voisinage de taille $d \times d$, on obtient la sortie f' .

$$f'(i,j) = \frac{1}{d^2} \sum_{n=-\frac{d-1}{2}}^{\frac{d-1}{2}} \sum_{m=-\frac{d-1}{2}}^{\frac{d-1}{2}} f(i+n, j+m)$$

Lissage par moyennage Exemple : Le filtre de taille $d=3$:

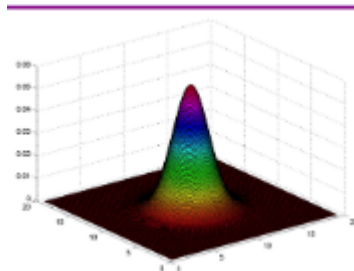
D'une manière générale, si on a un filtre de taille d , tous les coefficients du filtre ont comme valeur $w_i = 1/d^2$. Plus d est grand, plus le lissage sera important, et plus l'image filtrée perd les détails de l'image originale.

Filtrage spatial linéaire : exemple de filtres Lissage gaussien

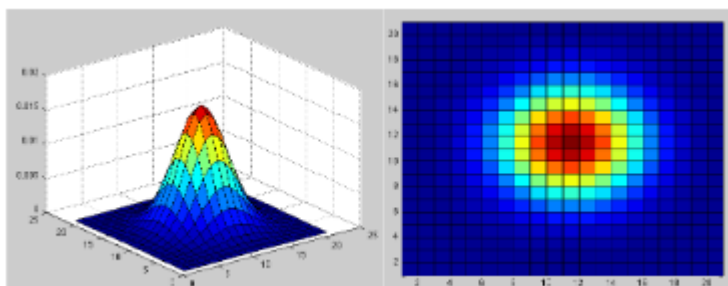
Définition : noyau gaussien centré et d'écart-type σ :

$$g_{\sigma}(i,j) = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}}$$

Lissage par moyennage pondéré de l'image en fonction de la distance du pixel voisin



Lissage gaussien du continu au discret le noyau gaussien est défini par un ensemble de coefficients qui sont des échantillons de la gaussienne 2D



Lissage gaussien Calcul des coefficients du filtre :

CHAP II Présentation d'une chaîne de traitement d'image

La largeur du filtre est donnée par son écart-type σ :

- Largeur du filtre de part et d'autre du point central : $\text{Ent}^+(3\sigma)$ ($\text{Ent}^+(\cdot)$ est l'entier supérieur)

- Largeur totale du filtre : $2\text{Ent}^+(3\sigma) + 1$

Si σ est plus petit qu'un pixel le lissage n'a presque pas d'effet

Plus σ est grand, plus on réduit le bruit, mais plus l'image filtrée est floue

Si σ est choisi trop grand, tous les détails de l'image sont perdus $\leftrightarrow$ On doit trouver un compromis entre la quantité de bruit à enlever et la qualité de l'image en sortie.

Lissage gaussien

Exemple : pour $\sigma=0.625$

Largeur du filtre de part et d'autre du point central : $\text{Ent}^+(3\sigma)=2$.

Largeur totale du filtre : $2\text{Ent}^+(3\sigma) + 1 = 5$ On obtient le filtre suivant :

$$h = 0.4 \times 10^{-2} \times \begin{pmatrix} 0.03 & 0.16 & 5.98 & 0.16 & 0.03 \\ 0.16 & 7.7 & 27.8 & 7.7 & 0.16 \\ 5.98 & 27.8 & 100 & 27.8 & 5.98 \\ 0.16 & 7.7 & 27.8 & 7.7 & 0.16 \\ 0.03 & 0.16 & 5.98 & 0.16 & 0.03 \end{pmatrix}$$

Lissage gaussien : des petites aux grandes échelles

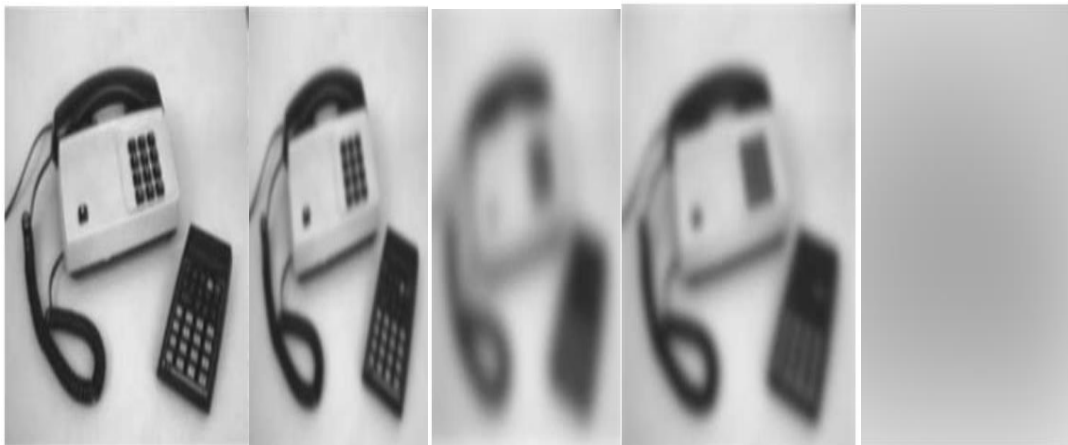


Image originale

$\sigma = 2$

$\sigma = 4$

$\sigma = 8$

$\sigma = 64$

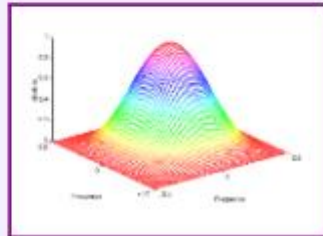
Filtrage linéaire Filtrage non linéaire Autres filtres Filtre binomial

Coefficients obtenus par le binôme de Newton

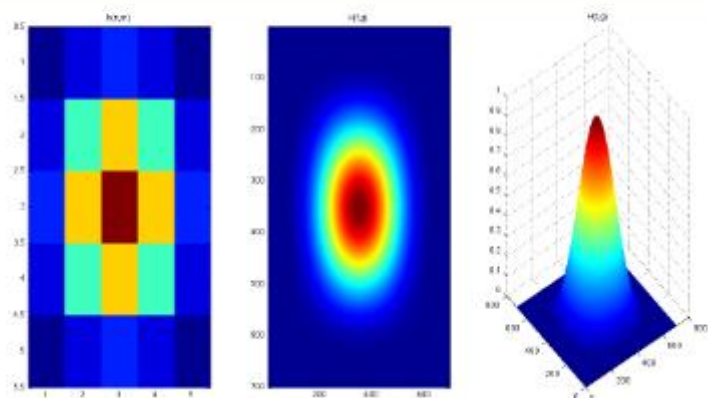
CHAP II

Présentation d'une chaîne de traitement d'image

$$h = \frac{1}{256} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$



$$h = \frac{1}{256} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

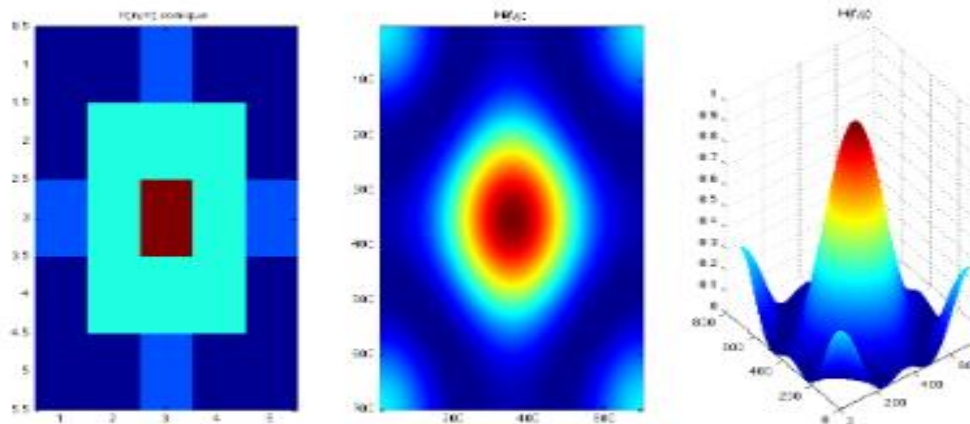


Filtre passe-bas, non idéal (voir partie filtrage fréquentiel)

Filtre pyramidal :

CHAP II Présentation d'une chaîne de traitement d'image

$$h_c = \frac{1}{25} \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 2 & 2 & 0 \\ 1 & 2 & 5 & 2 & 1 \\ 0 & 2 & 2 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$



Filtre passe-bas, non idéal (voir partie filtrage fréquentiel)