3-Traitement et analyse spatial

1-image couleur :

✓Image couleur = 3 plans couleur.

✓Pour la plupart des caméras : Rouge, Vert, Bleu (R,V,B).

✓Image couleur = 3 plans couleur.

✓Pour la plupart des caméras : Rouge, Vert, Bleu (R,V,B).

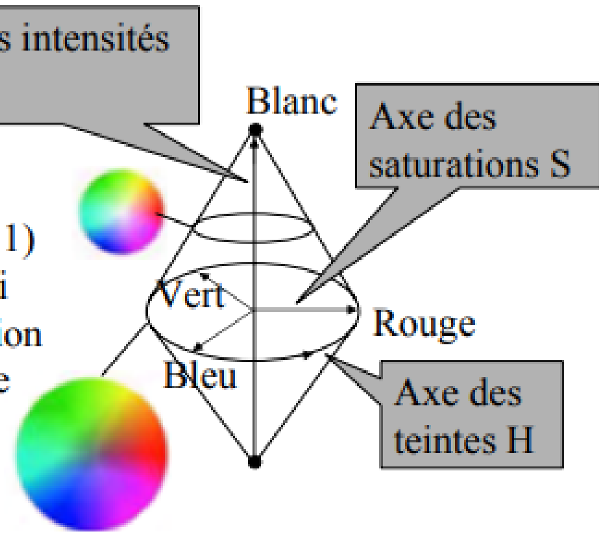
✓Chaque plan est codé comme une image niveaux de gris, avec des valeurs allant de 0 à 255.

✓Lorsque R=V=B, la couleur associé est un niveau de gris.

✓Pour passer d'une image couleur à une image niveau de gris, on réalise :

$$I(y,x) = \frac{R(y,x) + V(y,x) + B(y,x)}{3}$$

✓De nombreux autres systèmes de représentation des couleurs existent parmi lesquels on peut citer le système HSV (Hue, Saturation, Value) :



H : teinte (varie entre 0 et 2π)
S : saturation (varie entre 0 et 1)
une couleur très saturée qui possède une faible proportion de blanc se trouvera loin de l'axe des intensités
V : intensité lumineuse.

2-Traitement spatiale

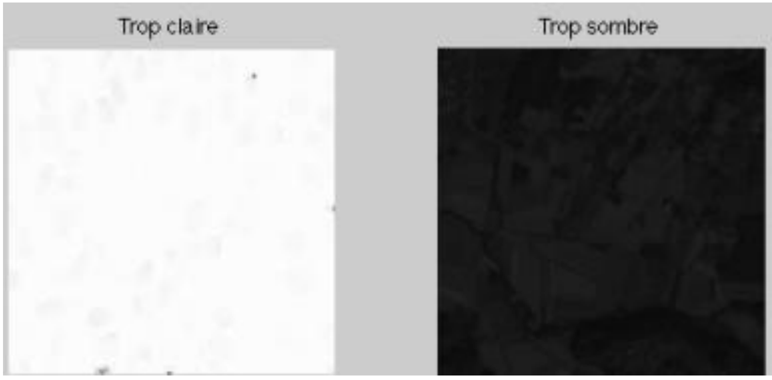
Améliorations d'images

But de l'amélioration

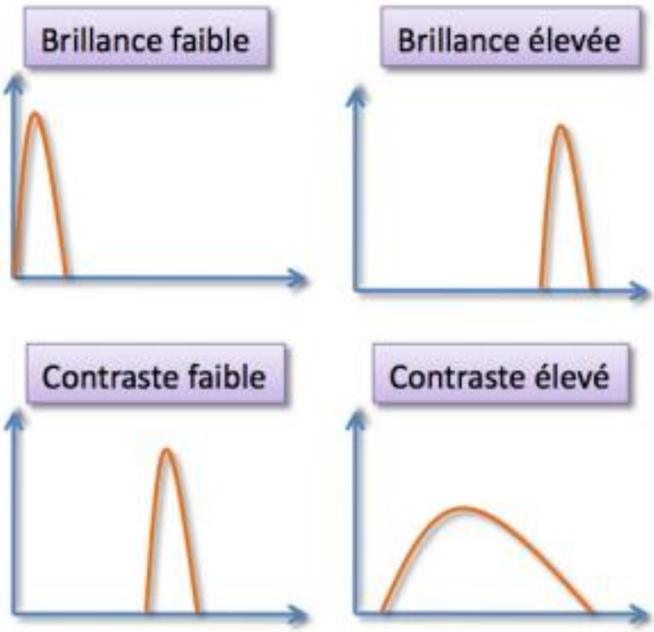
- Rendre les images plus aptes à l'interprétation humaine ou à celle de la machine
- Aucune théorie générale
- Manipulation dans le domaine spatial : accès direct aux valeurs de pixels
- Manipulation dans le domaine fréquentiel : modification de la transformée de Fourier de l'image

Pourquoi améliorer une image ?

- Régions à faire apparaître
- Image trop claire ou trop foncée
- Nécessité de modifier ses niveaux de gris afin de rendre visibles certains détails



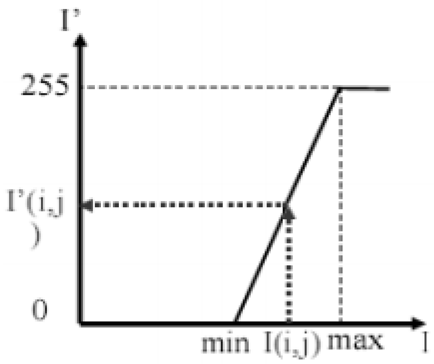
- Modifier la brillance.
- Augmenter le contraste.



Amélioration du Contraste :

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

2.1. Transformation linéaire (Approche Histogramme) :

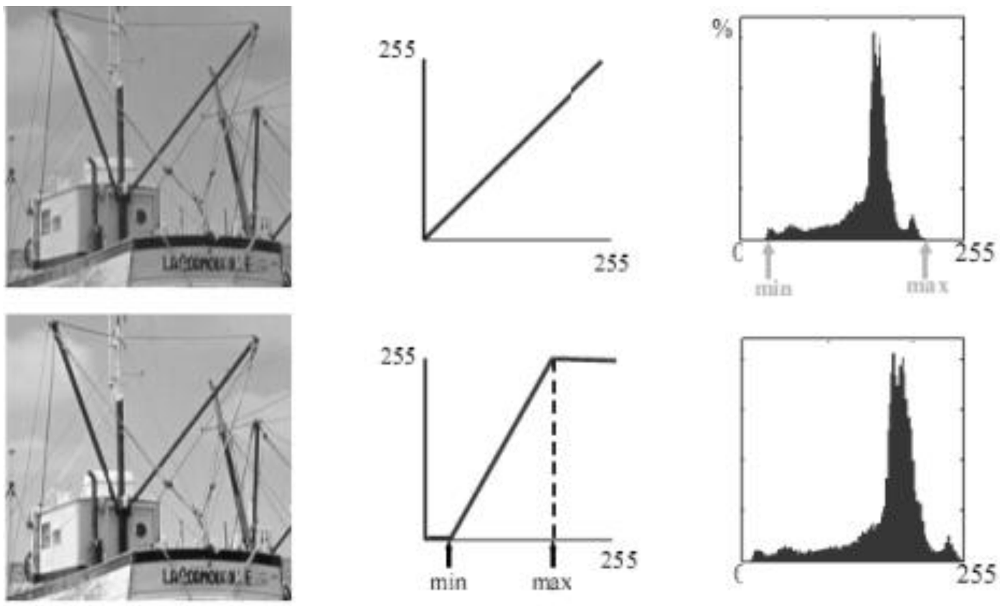


$$\frac{\max - \min}{I(i,j) - \min} = \frac{255 - 0}{I'(i,j) - 0}$$

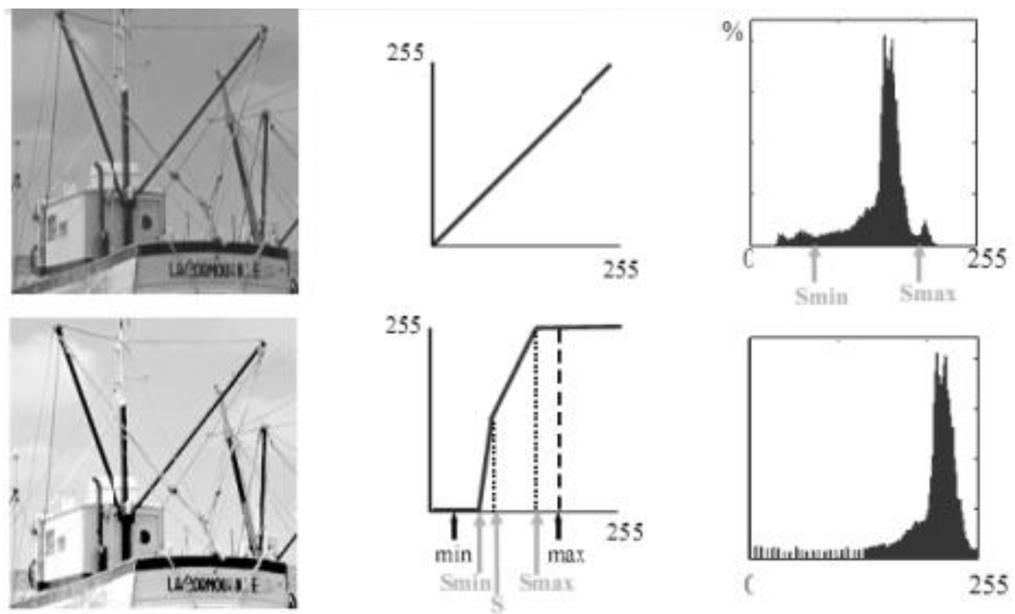
Alors :

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

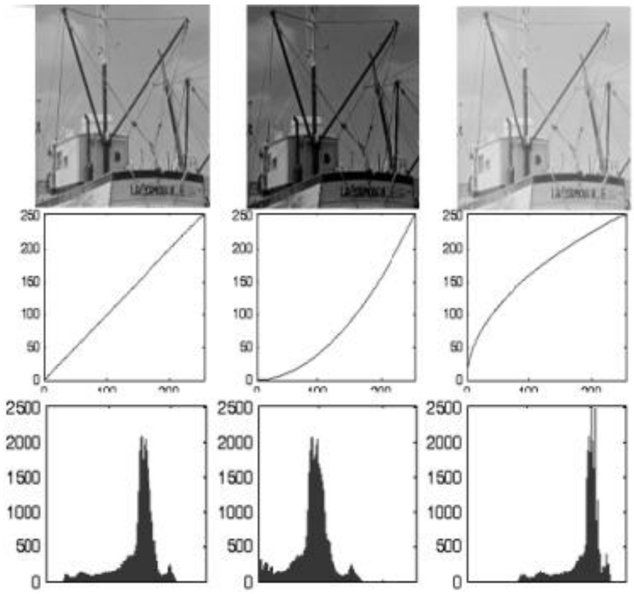
$$I'(i,i) = \frac{255}{\max \neg \min}\left(I(i,j) \neg \min\right) \quad avec\ \frac{(I(i,j) \neg \min)}{\max \neg \min} \in [0,1]$$



2.2. Transformation Linéaire par Morceaux:



2.3. Transformation non linéaire (approche histogramme) :



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

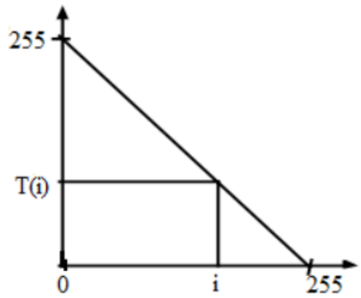
exponentielle gamma :

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

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

2.4. Fonction d'inversion- mise à l'échelle:

Fonction d'inversion Elle consiste à inverser les niveaux i de gris de l'image par la transformation suivante : T(i) = 255 – i



2.5.Fonction de binarisation :

Elle permet de définir deux niveaux d'intensité pour toute l'image en fonction d'un critère ou Seuil. L'image transformée est une image binaire. Tous les niveaux d'intensité qui vérifient le critère seront transformés en niveau Max (2^k-1), sinon en niveau 0 (ou en niveau Min). Seuil alors

I'(p) = 2^k-1 $\leq$ I si I(p) $\in$ p $\forall$ sinon I'(p) = 0

2.6. Fonction de Recadrage dynamique :

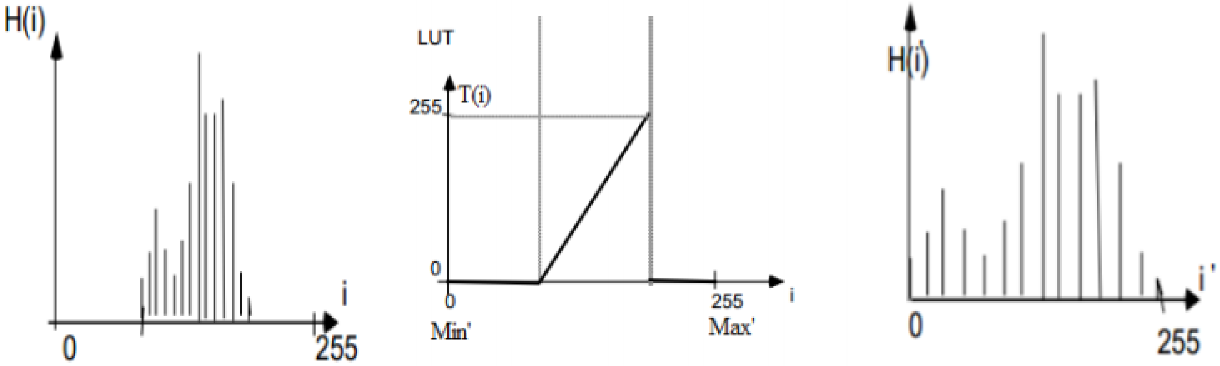
Elle permet de redistribuer les niveaux d'intensité sur le domaine disponible [0-2 k-1]

i $\dashrightarrow$ i' = T(i)

$$\begin{cases} i' = T(\mathrm{i}) = (2^k - 1) * \dfrac{i - min}{max - min} \\ min' = T(min) = 0 \text{ et } max' = T(max) = 2^k\text{-}1 \end{cases}$$

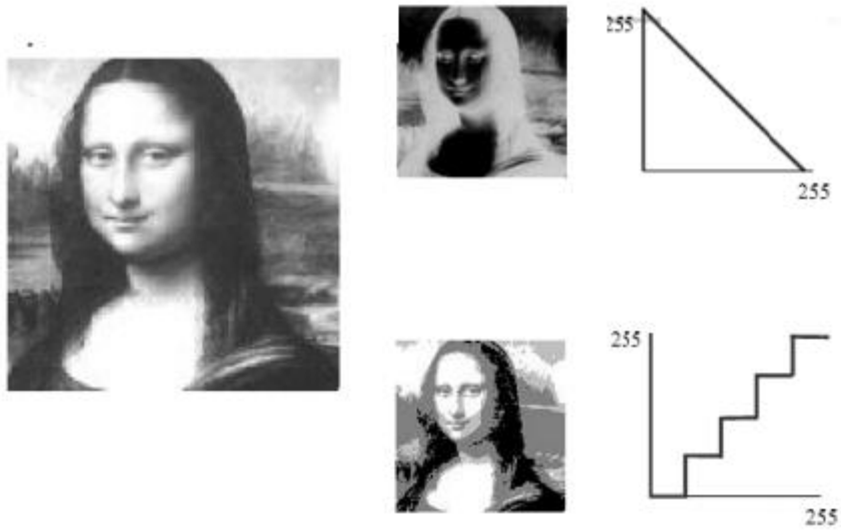
Elle est appliquée pour des images trop claires, ou trop foncées, ou pour des images dont les pixels sont distribués sur une plage étroite L'étalement simple est :

$$i^{'} = T(\mathrm{i}) = \frac{i - min}{max - min}$$



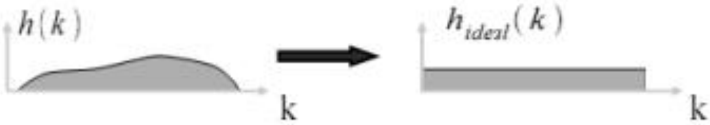
2.7. Fonction de rehaussement logarithmique

Elle permet d'éclaircir globalement l'image. Elle convient pour les images trop sombres.
T(i) = log (i)



2.8. Egalisation de l'histogramme :

- 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 $h_n(i) = \frac{h(i)}{Nbp} \quad i \in [0, 255]$
  (Nbp : nombre de pixels de l'image)
- 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

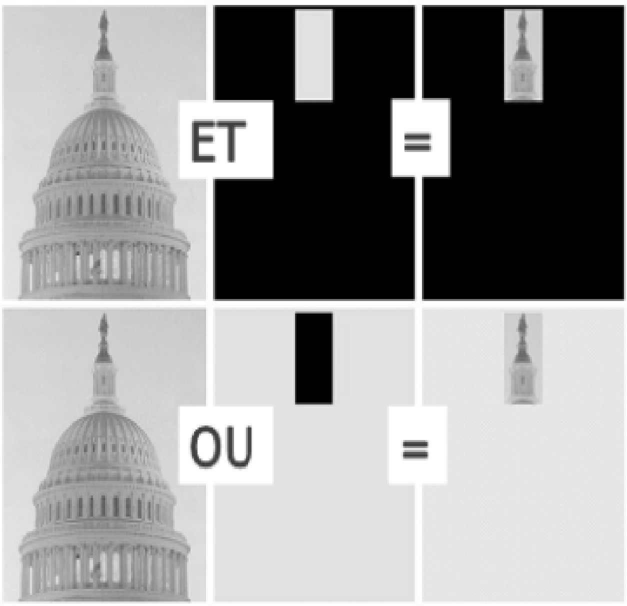
$$f'(x,y) = C(f(x,y)) \times 255$$

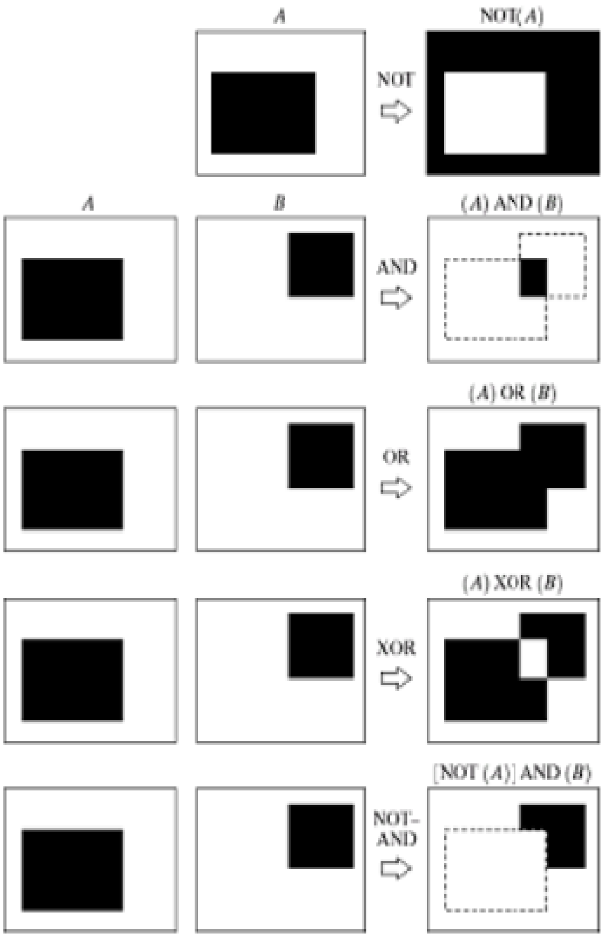
3. Opération sur les images numériques

3.1. Opérations logiques sur les images numériques

Les opérations logiques s'effectuent généralement entre deux images binaires. Elles utilisent les opérateurs logiques du domaine des ensembles tels que l'union, l'union exclusive, l'intersection et le complémentaire. Les opérateurs logiques sont appliqués aux images binaires, c'est à dire des images dont les pixels ne peuvent valoir que la valeur "vrai" ou la valeur "faux". Non logique : "Not" ; Addition logique : "And" ; Ou logique : "Or" ; Ou exclusif logique : "Xor" ; Différence logique : "logical-sub" ; Equivalence logique : "nxo"

A	B	Not A	A and B	A or B	A xor B	A-B	A nxo B
1	1	0	1	1	0	0	1
1	0	0	0	1	1	1	0
0	1	1	0	1	1	0	0
0	0	1	0	0	0	0	1





3.2. Opérations arithmétiques :

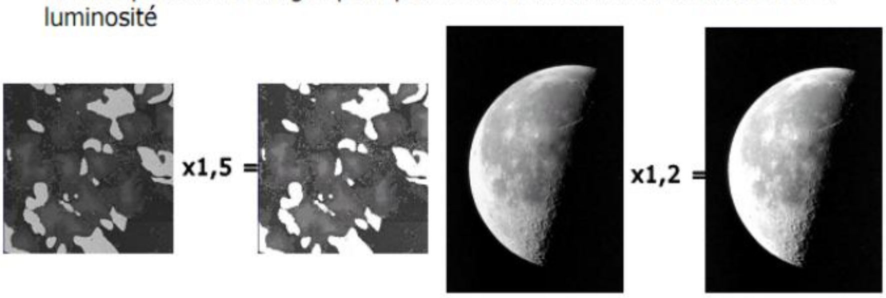
Les opérations arithmétiques des images s'effectuent sur des images de même dimension et sont respectivement l'addition, la soustraction et la division d'images. L'image résultante est l'image dont le niveau d'intensité de chaque pixel est le résultat de l'opération effectuée pixel par pixel des images originales.

A -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



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

Soient deux images I1 et I2 de même dimension,

$$I_3 = I_1 + I_2 \iff \forall\, p(x, y)\ \ I_3(x, y) = \text{Min}(I_1(x, y) + I_2(x, y), 255)$$
$$I_4 = I_1 - I_2 \iff \forall\, p(x, y)\ \ I_4(x, y) = \text{Max}\ (I_1(x, y) - I_2(x, y), 0)$$
$$I_5 = I_1 \,/\, I_2 \iff \forall\, p(x, y)\ \ I_5(x, y) = \text{Int}(I_1(x, y) \,/\, I_2(x, y)\ )$$
$$I_6 = I_1 * \alpha \iff \forall\, p(x, y)\ \ I_6(x, y) = \text{Max}(\ I_1(x, y)\ * \alpha, 255)$$

L'addition des images peut contribuer à :

- Diminuer les informations parasites de l'image

-D'éclaircir l'image en augmentant sa luminance

La soustraction d'images peut contribuer à :

Détecter les défauts de l'image

Détecter les mouvements ou les changements

La division des images peut contribuer à :

Diminuer nettement la luminance de l'image

La multiplication d'image par un facteur peut contribuer à :
Améliorer la luminance ou le contraste de l'image

4. Transformation d'une image

La transformation d'une image I consiste à appliquer une fonction de transformation f à tous les niveaux d'intensité de l'image de telle façon à reproduire une autre image I' équivalente à I.

$$I' = f(I)$$

- 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*

4.1. Transformation ponctuelle

Elle consiste à transformer chaque niveau d'intensité i 'une image en un autre niveau d'intensité i' par une fonction de transformation appliquée à tous les pixels.

$$\forall\, i \in I \quad i' = f(i)$$

La fonction f(.) permet de modifier les caractéristiques de l'image. La fonction f permet d'établir une table de correspondance de niveaux appelée la table de transcodage LUT (Look Up Table).

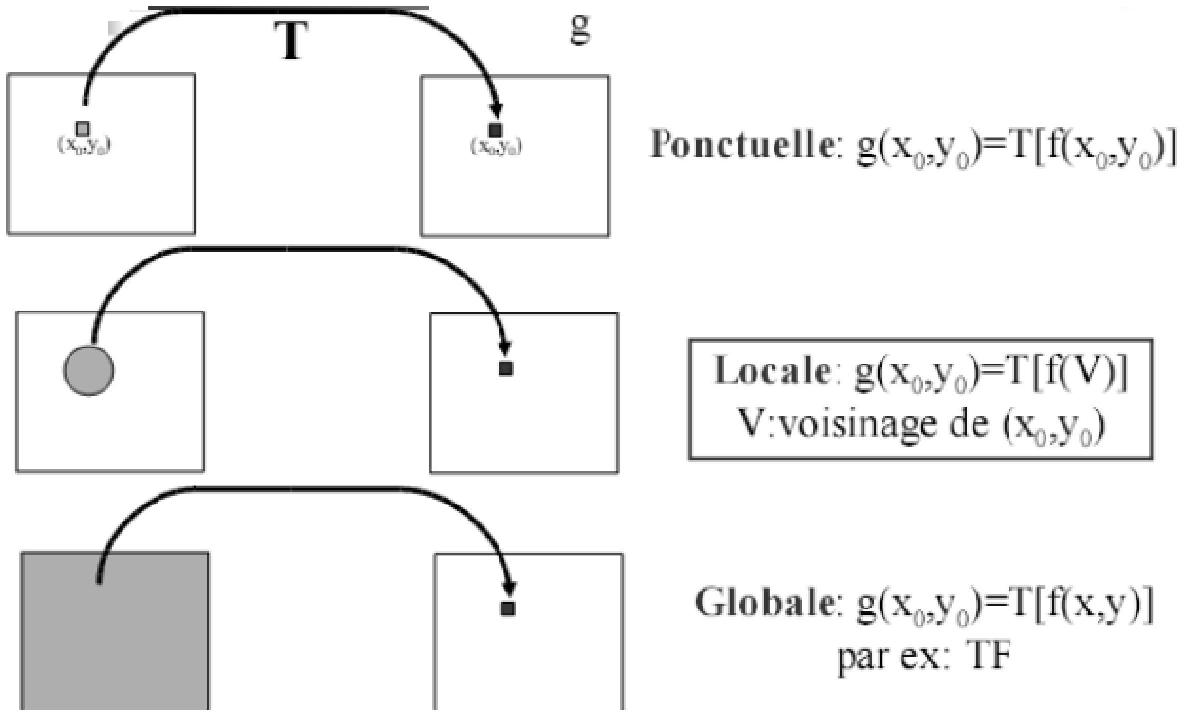
4.2. Transformation locale

Elle consiste à transformer chaque niveau d'intensité i d'un pixel p d'une image en une intensité dépendant de la localisation de p, de son intensité et des intensités de son voisinage par une fonction locale de voisinage.

$$\forall\, p(x, y) \in I \quad i'(p) = f(i(p), x, y)$$

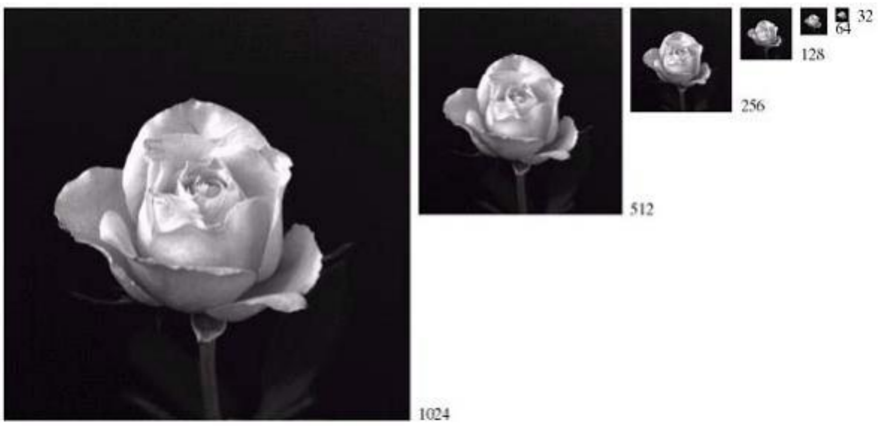
4.3. Transformation globale :

Elle consiste à changer de base ou de domaine de définition de la fonction image. La transformation de Fourier est un exemple du domaine spatial de l'image en un domaine fréquentiel de Fourier.



5. 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

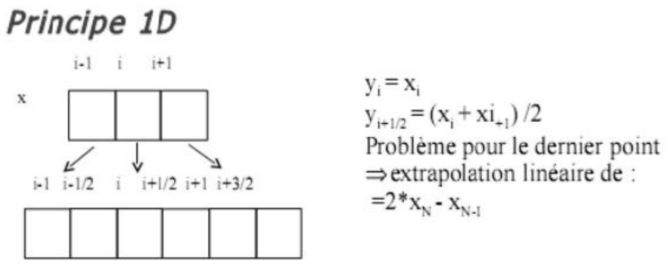
12	14
15	17

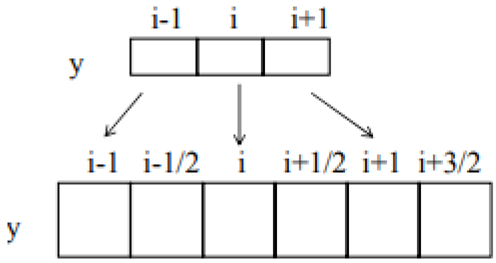
$\rightarrow$

12	12	14	14
12	12	14	14
15	15	17	17
15	15	17	17

B- Interpolation Bilinéaire :

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





On suppose que la loi de variation entre y_i et y_{i+1} est linéaire :

y=a x+b

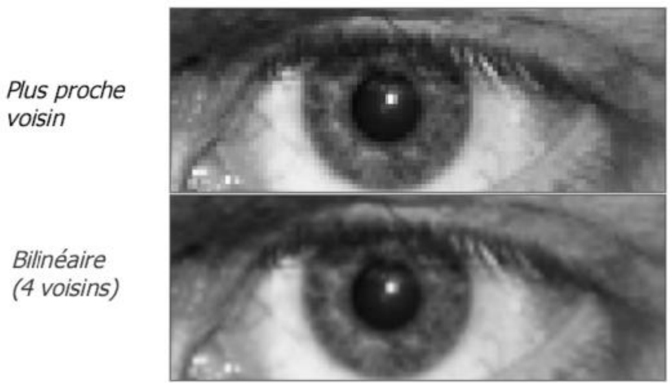
On connaît les couples de points (x_i, y_i) et (x_{i+1}, y_{i+1}). On en déduit facilement la valeur de $y_{i+1/2}$

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

Exemples d'Interpolation :

Exemple de Changement d'Echelle :

L'image originale à petit échelle : 



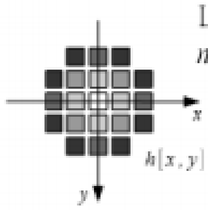
6. convolution :

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

- 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, ...*

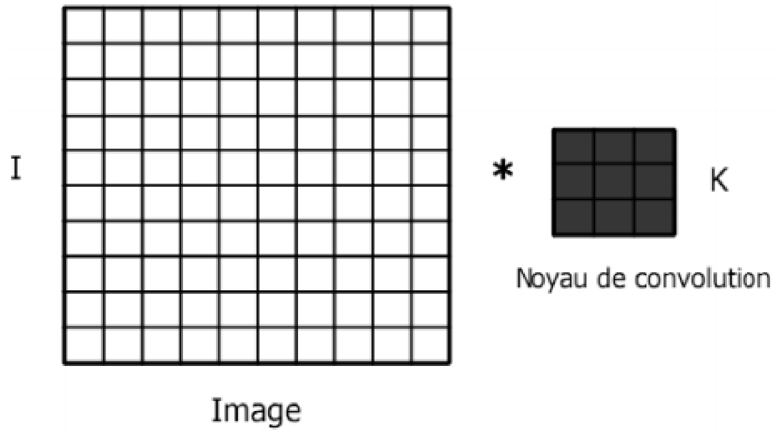
Note : par convention pratique, la taille de l'image résultat est la même que celle de l'image d'origine

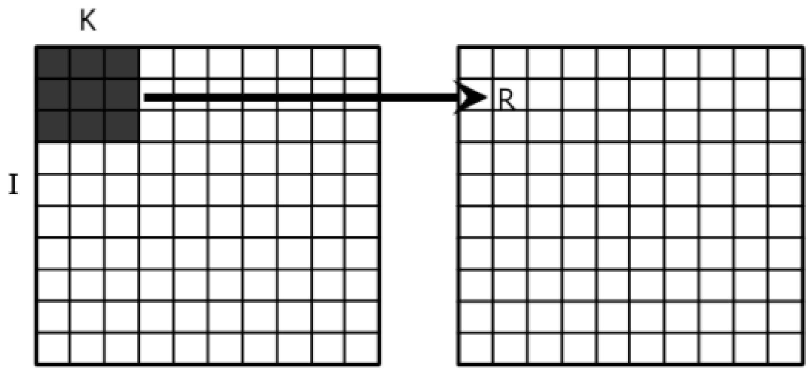
- En pratique, la convolution numérique d'une image se fera par une **sommation de multiplications**
- Un filtre de convolution est une matrice (image) généralement *(mais pas toujours)* de taille impaire et symétrique
  - 3x3, 5x5, 7x7, ...

Convolution d'une image par un filtre 2D :

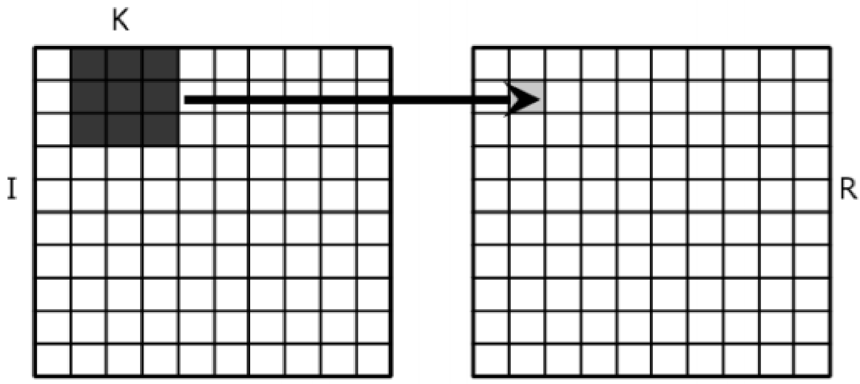
$$I'(i,j) = I(i,j) * filtre(i,j)$$

$$I'(i,j) = \sum_u \sum_v I(i-u, j-v) \cdot filtre(u,v)$$

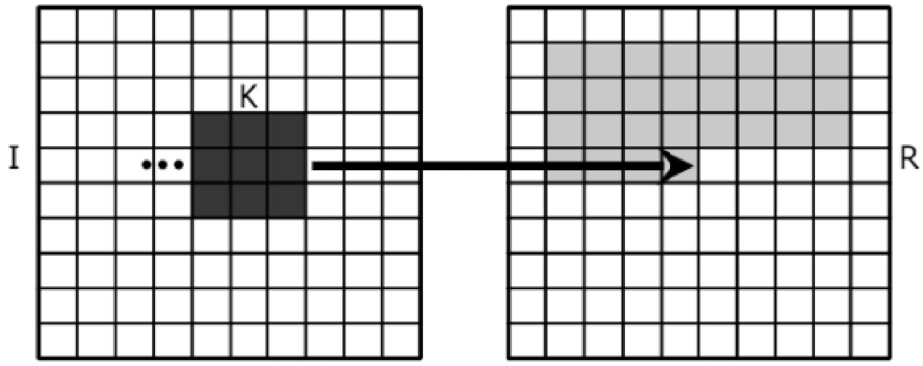




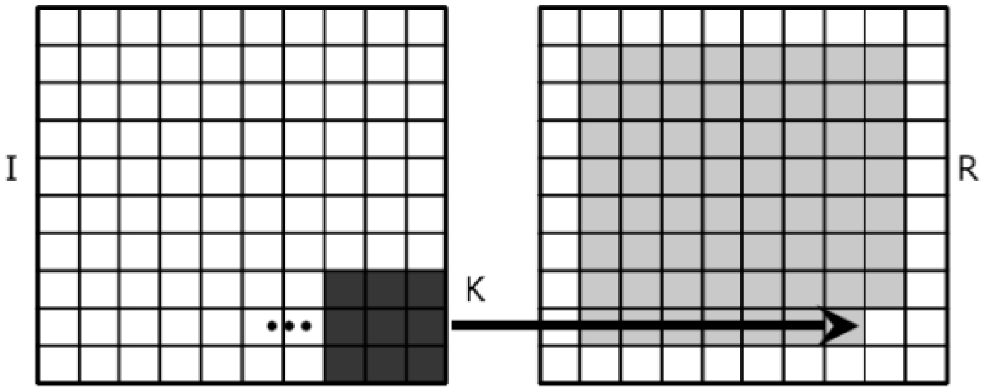
R(1,1) = I(0,0) K(0,0) + I(1,0) K(1,0) + I(2,0) K(2,0)
+ I(0,1) K(0,1) + I(1,1) K(1,1) + I(2,1) K(2,1)
+ I(0,2) K(0,2) + I(1,2) K(1,2) + I(2,2) K(2,2)



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)

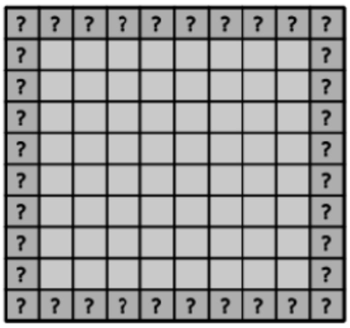


$$R(x,y) = I(x-1,y-1)\,K(0,0) + I(x,y-1)\,K(1,0) + I(x+1,y-1)\,K(2,0)$$
$$+ I(x-1,y)\,K(0,1) + I(x,y)\,K(1,1) + I(x+1,y)\,K(2,1)$$
$$+ I(x-1,y+1)\,K(0,2) + I(x,y+1)\,K(1,2) + I(x+1,y+1)\,K(2,2)$$

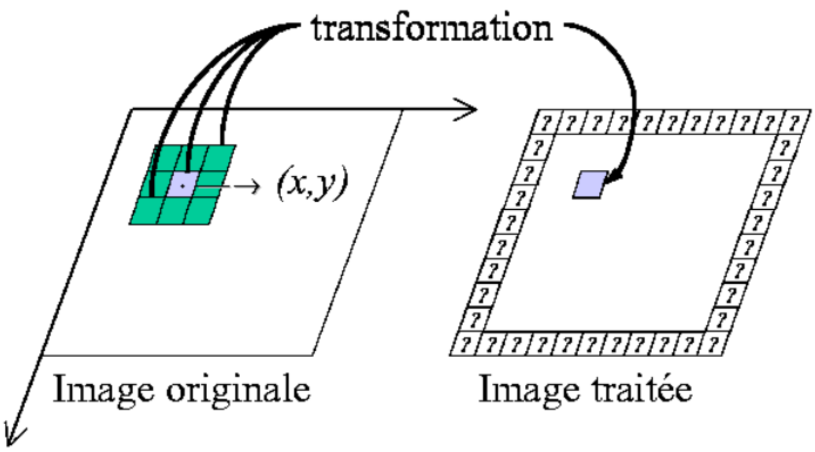


$$R(N-2,M-2) = I(N-3,M-3)\,K(0,0) + I(N-2,M-3)\,K(0,1) + I(N-1,M-3)\,K(0,3)$$
$$+ I(N-3,M-2)\,K(1,0) + I(N-2,M-2)\,K(1,1) + I(N-1,M-2)\,K(1,2)$$
$$+ I(N-3,M-1)\,K(2,0) + I(N-2,M-1)\,K(2,1) + I(N-1,M-1)\,K(2,2)$$

- 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)



7. le filtrage



?: effet de bord => périodicité, miroir,extérieur=0
ou on ne filtre pas les bords…

Le filtrage consiste à appliquer une transformation (appelée filtre) à tout ou partie d'une image numérique en appliquant un opérateur. On distingue généralement les types de filtres suivants :

• **Les filtres passe-bas**, consistant à atténuer les composantes de l'image ayant une fréquence haute (pixels foncés). Ce type de filtrage est généralement utilisé pour atténuer le bruit de l'image, c'est la raison pour laquelle on parle habituellement de lissage. Les filtres moyenneurs sont un type de filtres passe-bas dont le principe est de faire la moyenne des valeurs des pixels avoisinants. Le résultat de ce filtre est une image plus floue.

• **Les filtres passe-haut**, à l'inverse des passe-bas, atténuent les composantes de basse fréquence de l'image et permettent notamment d'accentuer les détails et le contraste, c'est la raison pour laquelle le terme de "filtre d'accentuation" est parfois utilisé.

•**Les filtres passe-bande** permettant d'obtenir la différence entre l'image originale et celle obtenue par application d'un filtre passe-bas.

• **Les filtres directionnels** appliquant une transformation selon une direction donnée.

On appelle filtrage adaptatif les opérations de filtrage possédant une étape préalable de sélection des pixels.

Qu'est-ce qu'un filtre ?

Un filtre est une transformation mathématique (appelée produit de convolution) permettant, pour chaque pixel de la zone à laquelle il s'applique, de modifier sa valeur en fonction des valeurs des pixels avoisinants, affectées de coefficients.

Le filtre est représenté par un tableau (matrice), caractérisé par ses dimensions et ses coefficients, dont le centre correspond au pixel concerné. Les coefficients du tableau déterminent les propriétés du filtre. Ainsi le produit de la matrice image, généralement très grande car représentant l'image initiale (tableau de pixels) par le filtre donne une matrice correspondant à l'image traitée. Le bruit caractérise les parasites ou interférences d'un signal, c'est-à-dire les parties du signal déformées localement. Ainsi le bruit d'une image désigne les pixels de l'image dont l'intensité est très différente de celles des pixels voisins.

<u>Notion de bruit</u>

Le bruit peut provenir de différentes causes :

• Environnement lors de l'acquisition

• Qualité du capteur

• Qualité de l'échantillonnage

<u>Lissage</u>

On appelle "lissage" (parfois débruitage ou filtre anti-bruit) l'opération de filtrage visant à éliminer le bruit d'une image. L'opération de lissage spécifique consistant à atténuer l'effet d'escalier produit par les pixels en bordure d'une forme géométrique est appelée anti-crénelage (en anglais anti-aliasing).

<u>Accentuation</u>

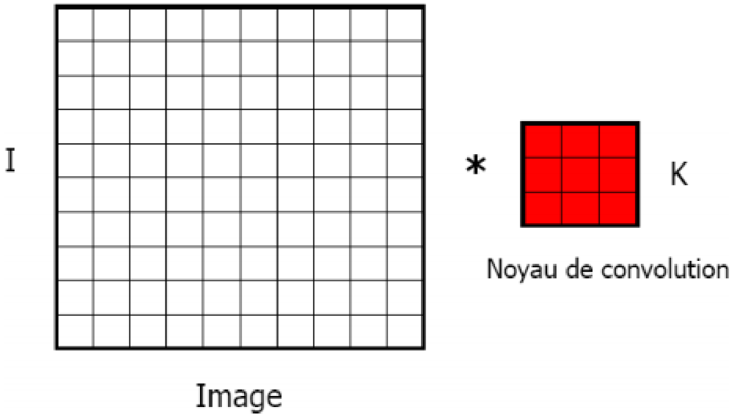
L'accentuation (ou bruitage) est l'inverse du lissage ; il s'agit d'une opération visant à accentuer les différences entre les pixels voisins.

Ainsi l'accentuation peut permettre de mettre en exergue les limites entre les zones homogènes de l'image et est alors appelée extraction de contours (également contourage ou réhaussement de contours).

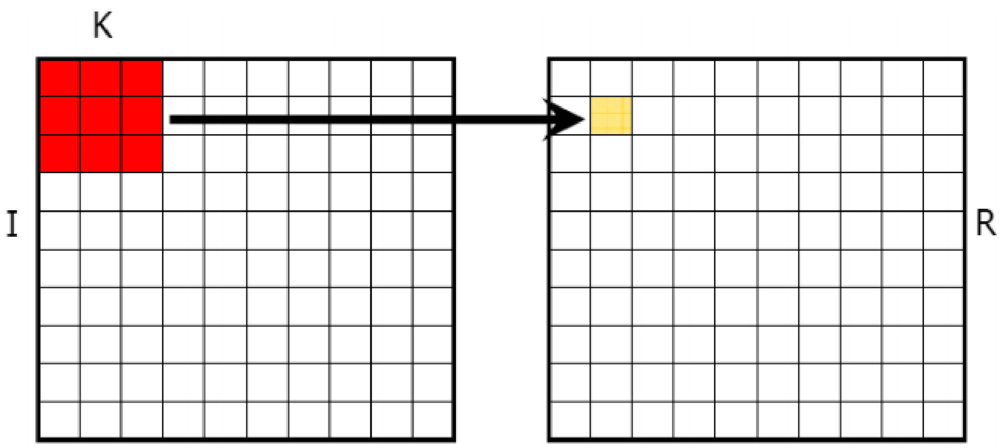
L’opération permet d’obtenir une nouvelle matrice (I_C) :

$$I_C(i,j) = \sum_{k=-\frac{i_F}{2}}^{\frac{i_F}{2}} \left\{ \sum_{l=-\frac{j_F}{2}}^{\frac{j_F}{2}} \left[F(k,l) \times I(i+k, j+l) \right] \right\}$$

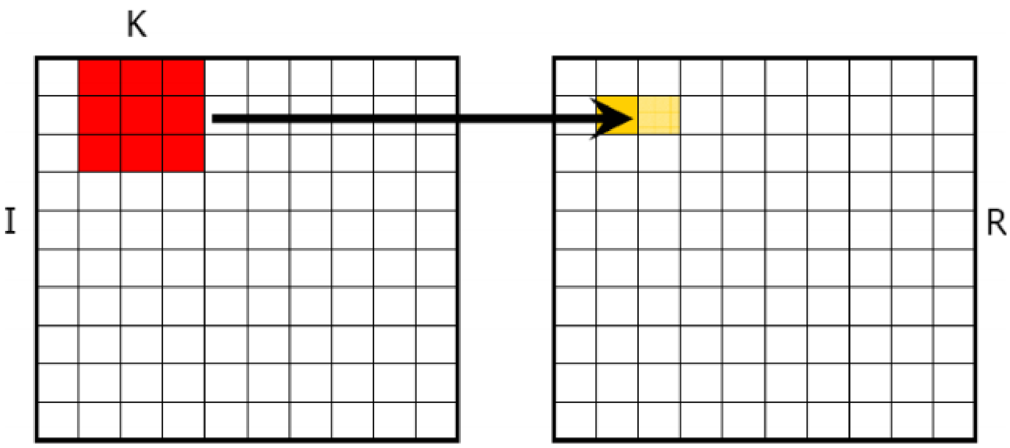
Pour expliquer la convolution, on appelle I l’image initiale, K = F le filtre de convolution, et R = IC l’image finale filtrée. Ainsi :



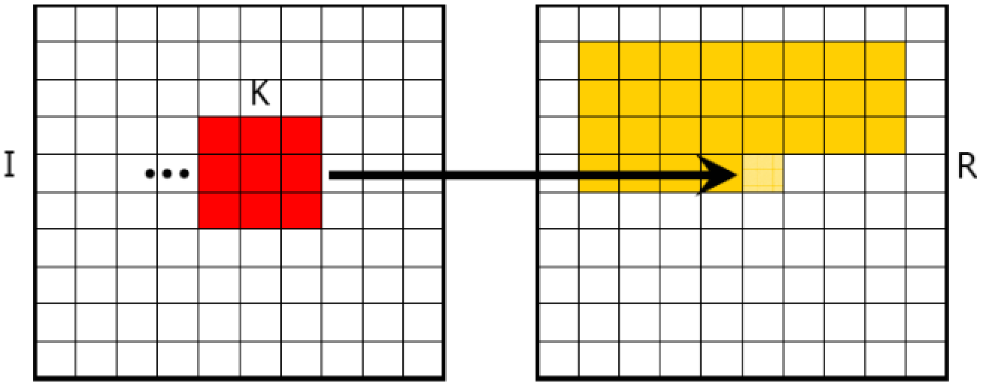
On applique successivement le filtre pour calculer chaque pixel de l’image finale R :



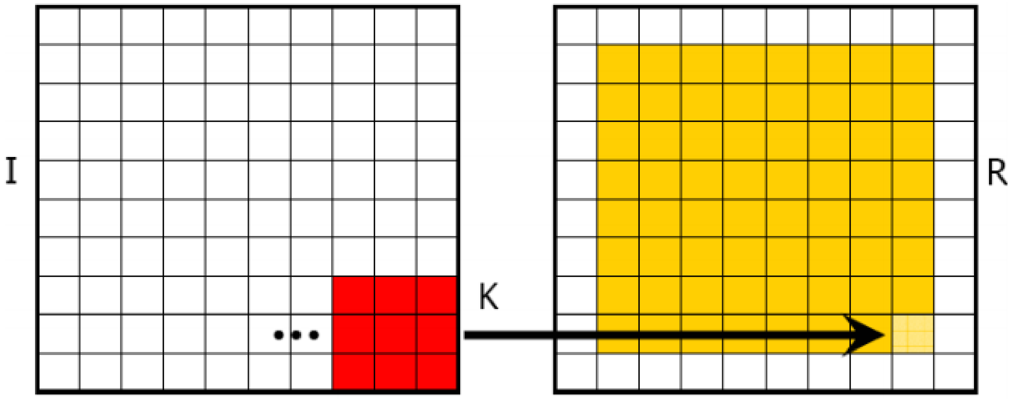
R(1,1) = I(0,0) K(0,0) + I(1,0) K(1,0) + I(2,0) K(2,0)
+ I(0,1) K(0,1) + I(1,1) K(1,1) + I(2,1) K(2,1)
+ I(0,2) K(0,2) + I(1,2) K(1,2) + I(2,2) K(2,2)



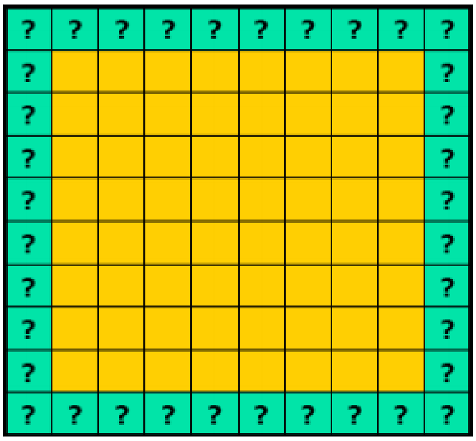
R(2,1) = I(1,0) K(0,0) + I(2,0) K(1,0) + I(3,0) K(2,0)
+ I(1,1) K(0,1) + I(2,1) K(1,1) + I(3,1) K(2,1)
+ I(1,2) K(0,2) + I(2,2) K(1,2) + I(3,2) K(2,2)



$$
\begin{aligned}
R(x,y) = {} & I(x-1,y-1)\ K(0,0) + I(x, y-1)\ K(1,0) + I(x+1, y-1)\ K(2,0) \\
& + I(x-1,y)\ K(0,1) + I(x,y)\ K(1,1) + I(x+1,y)\ K(2,1) \\
& + I(x-1,y+1)\ K(0,2) + I(x,y+1)\ K(1,2) + I(x+1,y+1)\ K(2,2)
\end{aligned}
$$



$$
\begin{aligned}
R(N-2,M-2) = {} & I(N-3,M-3)\ K(0,0) + I(N-2,M-3)\ K(0,1) + I(N-1,M-3)\ K(0,3) \\
& + I(N-3,M-2)\ K(1,0) + I(N-2,M-2)\ K(1,1) + I(N-1,M-2)\ K(1,2) \\
& + I(N-3,M-1)\ K(2,0) + I(N-2,M-1\ )K(2,1) + I(N-1,M-1)\ K(2,2)
\end{aligned}
$$



Lorsque le pixel considéré est proche du bord de l'image, certains points du voisinage sont en dehors de l'image d'origine. Il convient alors de choisir une stratégie pour gérer ces pixels extérieurs.

<u>Les stratégies couramment employées :</u>

• **Mise à zéro :** Si un pixel du voisinage est en dehors de l'image d'origine, sa valeur est considérée comme nulle. C'est à dire : Image[-1][y]=0

• **Continuité :** Si un pixel du voisinage est en dehors de l'image d'origine, sa valeur est celle du pixel le plus proche qui est dans l'image d'origine. C'est à dire : Image[-1][y]= Image[0][y]

• **Miroir** : Si un pixel du voisinage est en dehors de l'image d'origine, sa valeur est celle du pixel symétrique par rapport au bord de l'image. C'est à dire : Image[-1][y]= Image[1][y]

• **Sphérique** : Si un pixel du voisinage est en dehors de l'image d'origine, sa valeur est celle du pixel correspondant si l'image était projetée sur une sphère. C'est à dire : Image[-1][y]= Image[Largeur-1][y]
Si la valeur obtenue n'est dans les limites imposées par le format d'image (entier(s) entre 0…255), alors la valeur est ramenée dans l'intervalle souhaité par normalisation et décalage.

8. Filtrage dans le domaine spatial :

On distingue 2 classes de filtres : les filtres linéaires et les filtres non-linéaires suivant que l'opérateur appliqué est linéaire ou non.

Le filtrage linéaire

Principe des filtres linéaires

Les filtres linéaires sont une classe particulière de filtres, dont les propriétés mathématiques les rendent intéressants à utiliser. Dans le cas d'un filtrage linéaire de l'image, les opérations impliquées ne sont que des additions (ou soustractions) et multiplications (ou divisions) par des valeurs constantes. En pratique, cela consiste à réaliser une moyenne pondérée de la valeur des pixels du voisinage. Par exemple, une opération courante consiste à remplacer la valeur de chaque pixel par la valeur moyenne de ses voisins. La nouvelle valeur du pixel I(u, v) est déterminée par l'équation :

$$\begin{aligned} I'(u,v) \quad \leftarrow 1/9[\quad & I(u-1,v-1) + I(u,v-1) + I(u+1,v-1) \\ & + I(u-1,v) + I(u,v) + I(u+1,y) \\ & + I(u-1,v+1) + I(u,v+1) + I(u+1,v+1)] \end{aligned}$$

On peut classer les filtres linéaires en fonction de leur effet sur l'image de départ. Les filtres moyenneurs réduisent le bruit tout en lissant les structures. Les filtres dérivatifs permettent de mettre en évidence les bords des structures, et sont très utilisés pour détecter les contours. Les filtres dérivatifs du second ordre sont une alternative aux filtres dérivatifs simples, et permettent aussi de mettre en évidence les contours

Exemple de filtrage linéaire :Filtre moyenneur (passe bas)

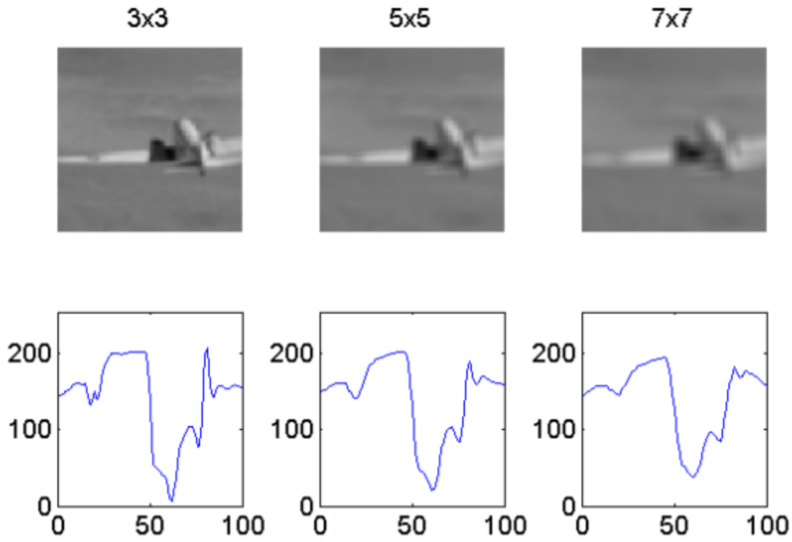
Les filtres moyenneurs, comme leur nom l'indique, calculent la moyenne, éventuellement pondérée, des pixels situés dans le voisinage de chaque pixel. Cette famille de filtres permet de réduire le bruit dans l'image, ce qui rend les zones homogènes plus lisses. Par contre, les contours sont fortement dégradés, et les structures trop fines peuvent devenir moins visibles. Filtres plats Les filtres qui calculent une moyenne directe des pixels sont appelés filtres plats (flat smoothing). Ils utilisent en général des voisinages carrés (3 × 3, 5 × 5...). Pour un filtre plat de taille 3 × 3, la matrice de voisinage correspondante est la suivante :

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

Pour des filtres plats de taille n × n, la matrice est de la forme :

$$\frac{1}{n^2} \times \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix}$$

Le résultat d'un filtre plat avec trois tailles de filtre différentes est donné sur la figure suivante :



Exemple de filtrage linéaire : filtre gaussien (passe-bas) moyennes pondérées

On peut vouloir pondérer le poids associé à la valeur de chaque voisin. En particulier, on peut supposer que l'influence d'un voisin proche est plus importante que celle d'un voisin éloigné. On utilise donc aussi des filtres pondérés selon la distance au pixel central :

$$H = \begin{bmatrix} 0.05 & 0.15 & 0.05 \\ 0.15 & 0.20 & 0.15 \\ 0.05 & 0.15 & 0.05 \end{bmatrix} = \frac{1}{20}\begin{bmatrix} 1 & 3 & 1 \\ 3 & 4 & 3 \\ 1 & 3 & 1 \end{bmatrix}$$

Les pondérations les plus classiques sont d'utiliser un noyau gaussien ou parabolique. Pour un noyau gaussien, les coefficients sont calculés à partir de la distance au pixel central du noyau du filtre :

$$G(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$

Un noyau discret 5×5 approchant un filtrage par un noyau gaussien avec $\sigma = 2$ est le suivant :

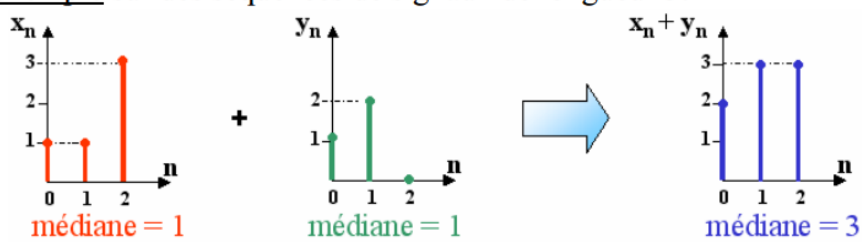
$$H = \frac{1}{101}\begin{bmatrix} 1 & 3 & 4 & 3 & 1 \\ 3 & 5 & 7 & 5 & 3 \\ 4 & 7 & 9 & 7 & 4 \\ 3 & 5 & 7 & 5 & 3 \\ 1 & 3 & 4 & 3 & 1 \end{bmatrix}$$

Exemple de filtrage non-linéaire : le filtrage médian

- Le filtrage médian est une opération non-linéaire :

  médiane $\{ x_m + y_m \} \neq$ médiane $\{ x_m \}$ + médiane $\{ y_m \}$ sauf exception

  - exemple sur des séquences de signaux de longueur 3:



- En traitement d'image, les tailles des fenêtres utilisées pour le filtrage médian sont généralement impaires : ***3 × 3 ; 5 × 5 ; 7 × 7***

  - exemple : fenêtre de 5 pixels en croix 

- Pour les fenêtres de taille paire (2 K valeurs) : après ordonnancement croissant des valeurs, prendre la moyenne des 2 valeurs centrales :

$$\text{valeur de sortie} = \frac{(\text{K}^{\text{ième}} \text{ valeur ordonnée} + (\text{K}+1)^{\text{ième}} \text{ valeur ordonnée})}{2}$$

- Image de référence (taille 3 × 3) :

91	55	90
77	68	95
115	151	210

- Filtrage médian avec une fenêtre de taille 3 × 3 :

on liste les valeurs de l'image de référence sur la fenêtre 3 × 3 :

55 , 68 , 77 , 90 , **91** , 95 , 115 , 151 , 210

valeur médiane = 91

- Filtrage médian sur une fenêtre de taille 3 × 3 en croix:

	55	
77	68	95
	151	

on liste les valeurs de l'image de référence sur la fenêtre :

55 , 68 , **77** , 95 , 151

valeur médiane = 77

- Observation :

Le filtrage médian est plus adapté que le filtrage linéaire pour réduire le bruit impulsionnel

<u>autres filtres spatiales passe-bas</u>

Filtre Binomial

Les coefficients de ce filtre sont obtenus par le binome de Newton. Un filtre 1D Binomial du quatrième ordre donne le vecteur (1/16)(1 4 6 4 1). Le filtre 2D est

$$\frac{1}{256}\begin{array}{|c|c|c|c|c|}\hline 1 & 4 & 6 & 4 & 1 \\ \hline 4 & 16 & 24 & 16 & 4 \\ \hline 6 & 24 & 36 & 24 & 6 \\ \hline 4 & 16 & 24 & 16 & 4 \\ \hline 1 & 4 & 6 & 4 & 1 \\ \hline \end{array}$$

Filtre Pyramidal

$$\frac{1}{81}\begin{array}{|c|c|c|c|c|}\hline 1 & 2 & 3 & 2 & 1 \\ \hline 2 & 4 & 6 & 4 & 2 \\ \hline 3 & 6 & 9 & 6 & 3 \\ \hline 2 & 4 & 6 & 4 & 2 \\ \hline 1 & 2 & 3 & 2 & 1 \\ \hline \end{array}$$

y

Filtre Conique

$$\frac{1}{25}\begin{array}{|c|c|c|c|c|}\hline 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 2 & 2 & 2 & 0 \\ \hline 1 & 2 & 5 & 2 & 1 \\ \hline 0 & 2 & 2 & 2 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 \\ \hline \end{array}$$

Exercice1:

Soit la matrice d'image suivante :

234	212	212
150	150	110
100	110	110

Egaliser la matrice d'image en adoptant la démarche suivante :

1- Calculer l'histogramme H(i)

2- Normaliser l'histogramme Hn(i) = H(i)/Def

3- Calculer la densité de probabilité cumulative Hn(i) = H(i)/Def

4- Transformer les niveaux de gris de l'image

$$I'(x, y) = Hcn(I(x,y)) * 255$$

Exercice2:

Soit M La matrice d'intensités suivante représentant une image à 256 niveaux de gris:

234	212	212
150	150	110
100	110	110

1-Donner la définition de M, sa dynamique, sa luminance et son contraste

2-Déterminer son histogramme

3-Inverser M en M' tel que I'(p) = 255-I(p)

4-Donner la luminance de M' et son contraste.

5-Déterminer son histogramme

6-Transformer M en M'' tel que I''(p) = (I(p)-Min) * (255/ (Max-Min)) recadrage

7-Déterminer son histogramme cumulé