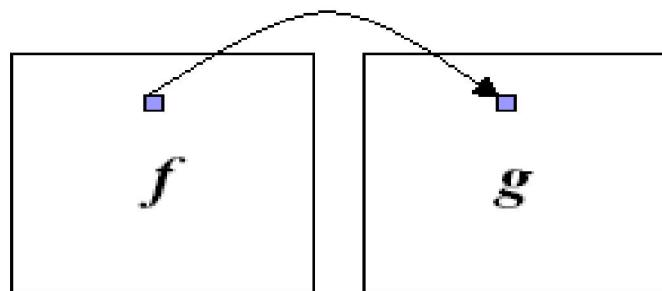
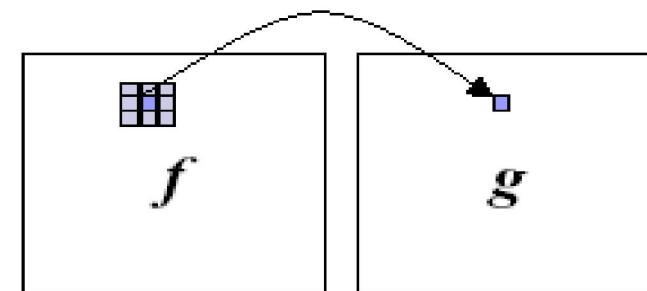


7. Traitements de base d'une image

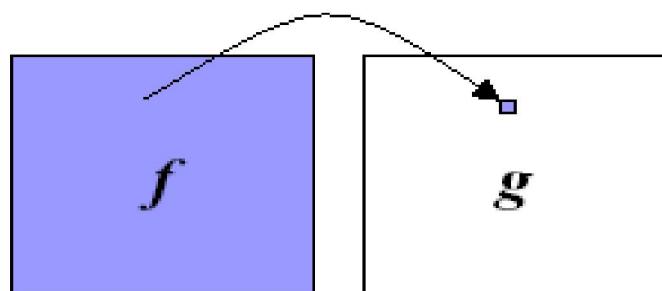
Types d'opérations



Ponctuelle



Locale



Globale

Ponctuelle	$g(x_0, y_0) = T[f(x_0, y_0)]$
Locale	$g(x_0, y_0) = T[V(f(x_0, y_0))]$
Globale	$g(x_0, y_0) = T[f(x, y)]$

- T est le même pour tous les points de l'image
- $V(a)$ = voisinage de a

7. Traitements de base d'une image

Opérations ponctuelles

- **Définitions**
 - Moyenne et contraste
 - Histogramme
- **Opérations ponctuelles linéaires**
 - Décalage additif
 - Mise à l'échelle multiplicative
 - Inversion
 - Amélioration du contraste
- **Opération arithmétiques entre les images**
 - Opérations de base
 - Réduction du bruit par moyennage

7. Traitements de base d'une image

Opérations ponctuelles

- Moyenne
 - Image en tons de gris

$$\mu = \frac{1}{NM} \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} f(x, y)$$

- Image couleur
 - Une moyenne par canal

7. Traitements de base d'une image

Opérations ponctuelles

- Moyenne (Luminosité)

Intensité moyenne dans l'image



7. Traitements de base d'une image

Opérations ponctuelles

- Contraste
 - Peut être défini de plusieurs façons
 - Écart type des variables de niveaux d'intensité

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

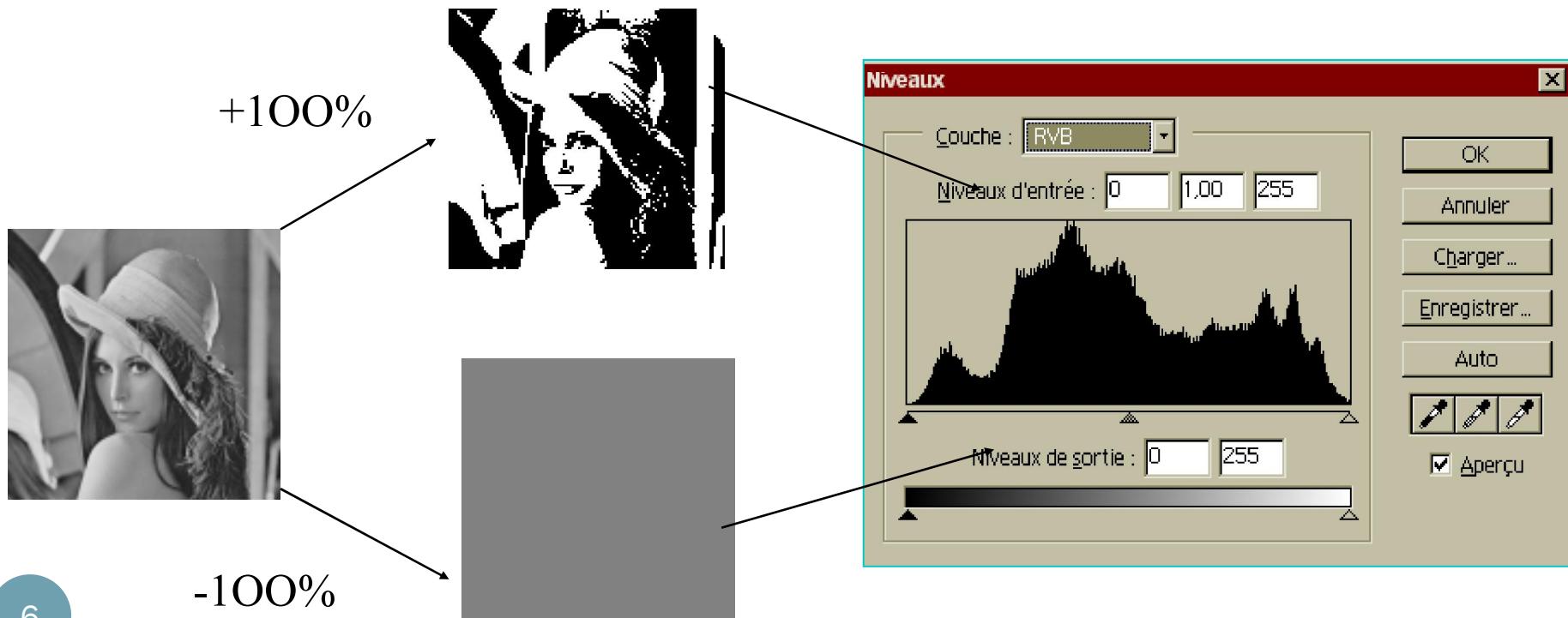
- Variation entre valeurs maximale et minimale de niveaux d'intensité (moins utile , plus rapide)

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

7. Traitements de base d'une image

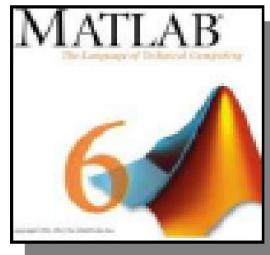
Opérations ponctuelles

- Contraste
 - Plage de valeurs utilisée : $C = \max - \min$



7. Traitements de base d'une image

Opérations ponctuelles



Code Matlab

■ Moyenne et contraste

```
img = imread('gray1.bmp');  
moyenne = mean2(img)  
contraste = std2(img)
```

7. Traitements de base d'une image

Opérations ponctuelles

Amélioration des images basée sur les pixels

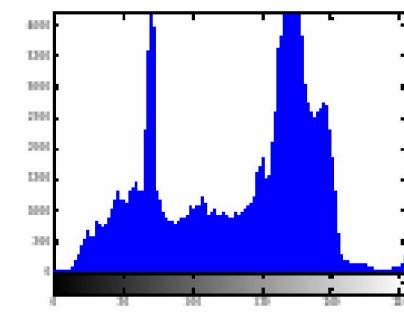
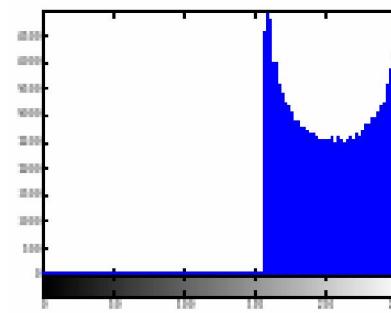
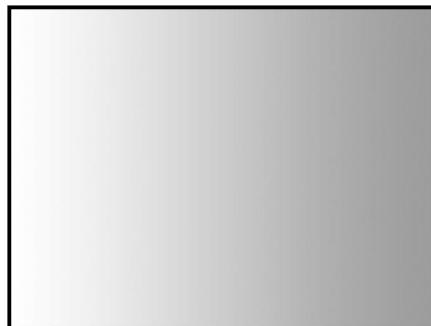
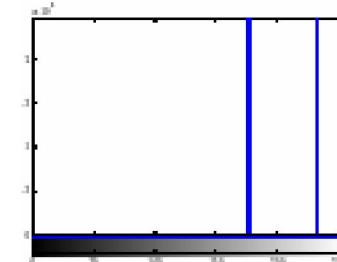
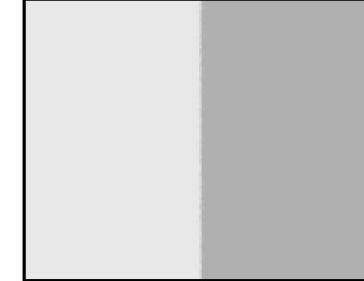
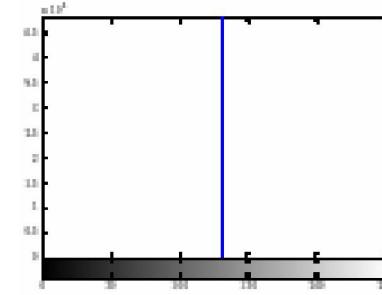
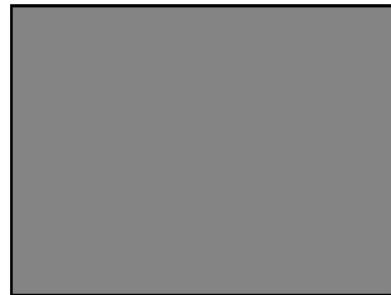
- Histogramme des niveaux de gris
 - Transformation des niveaux de gris
- Amélioration du contraste

7. Traitements de base d'une image

Opérations ponctuelles

Histogramme des niveaux de gris (distribution des niveaux de gris)

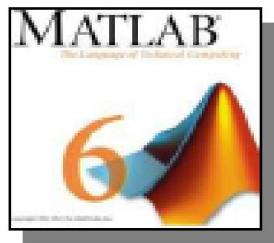
- Histogrammes pour différents types d'images



7. Traitements de base d'une image

Opérations ponctuelles

Histogramme des niveaux de gris (distribution des niveaux de gris)



Code Matlab

■ Histogramme

```
img = imread('gray1.bmp');
imshow(img)
histo = imhist(img,256); %calcul
figure, imhist(img,256); %affichage
```

7. Traitements de base d'une image

Opérations ponctuelles

Caractérisation par Histogramme des niveaux de gris

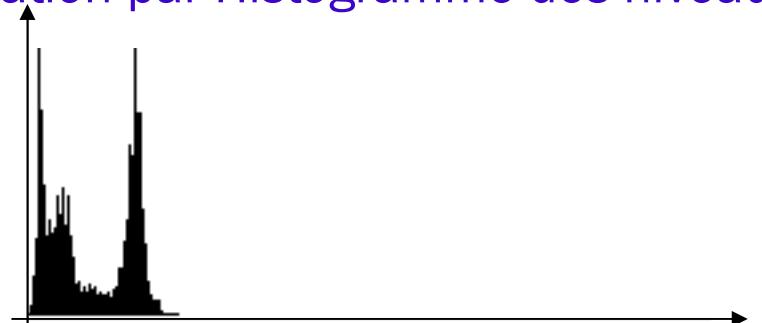


Image sous-exposée

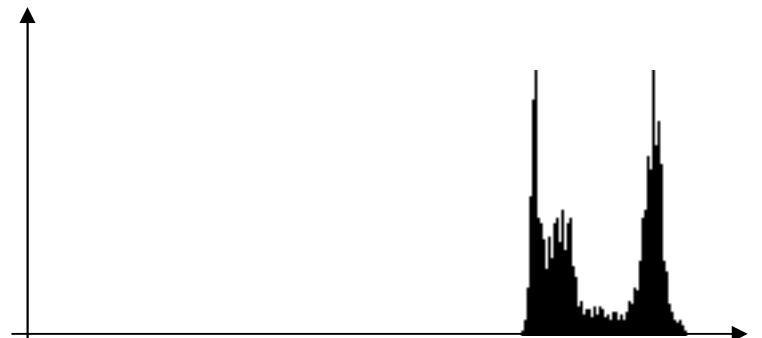


Image sur-exposée

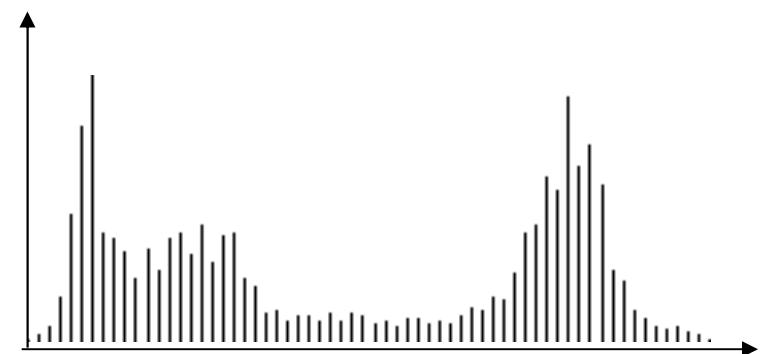
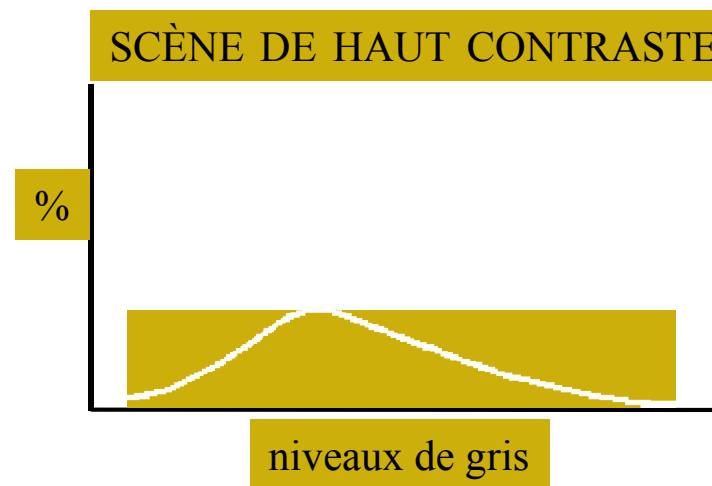
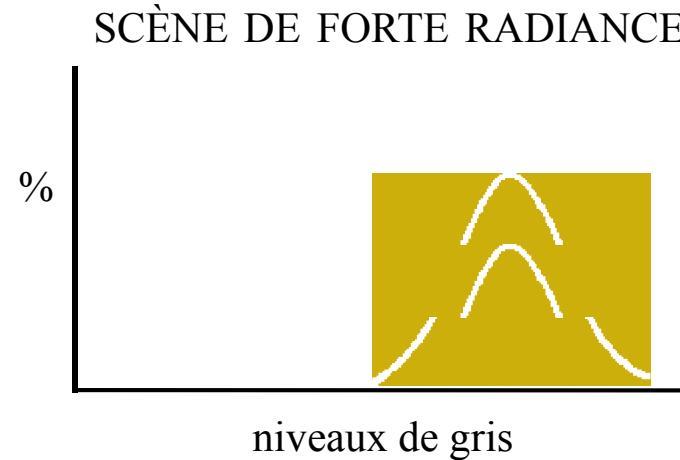
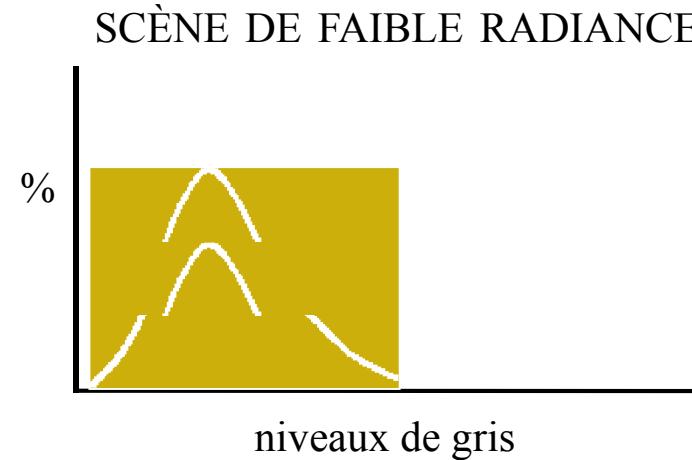


Image mal quantifiée

7. Traitements de base d'une image

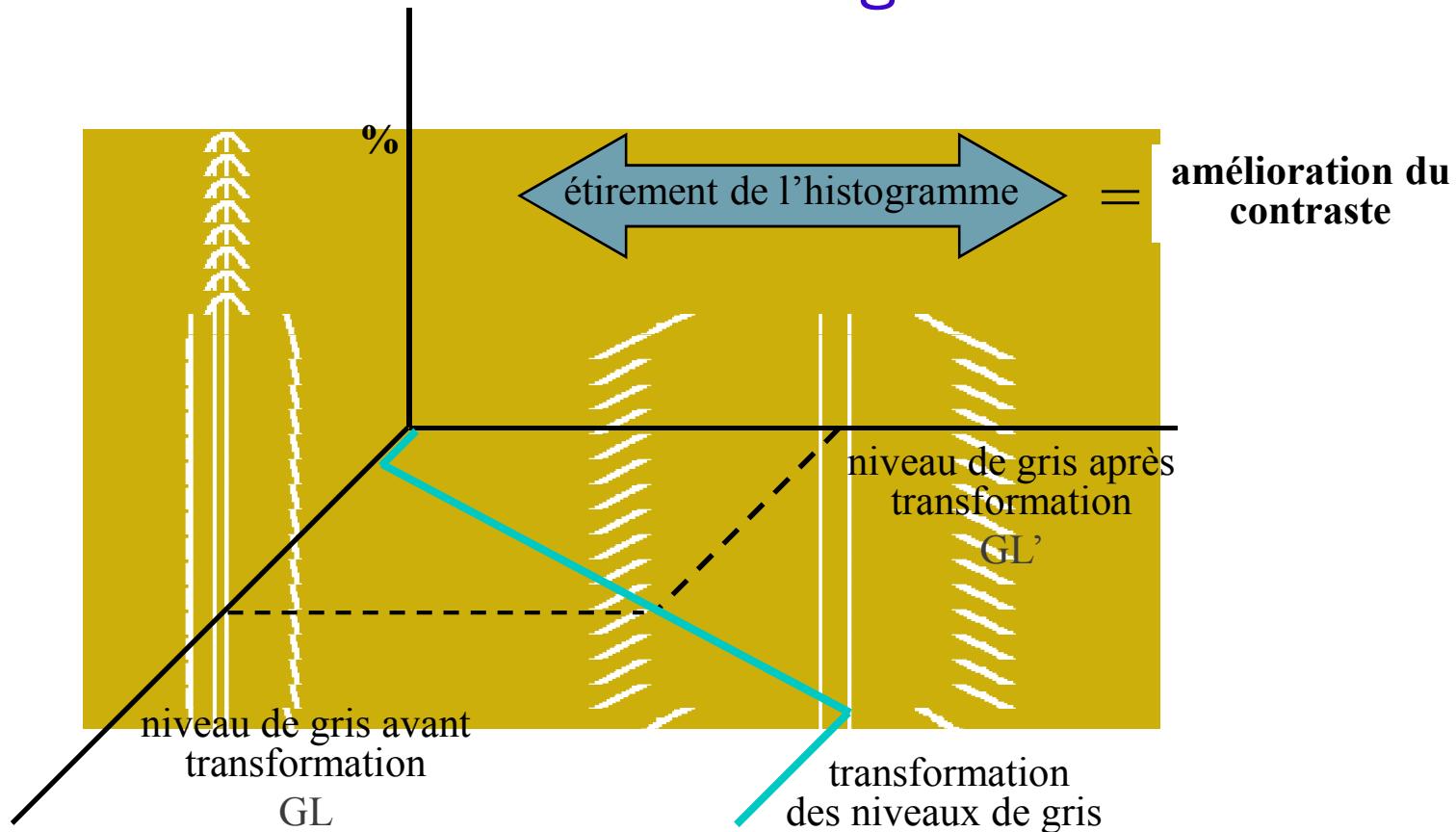
Opérations ponctuelles



7. Traitements de base d'une image

Opérations ponctuelles

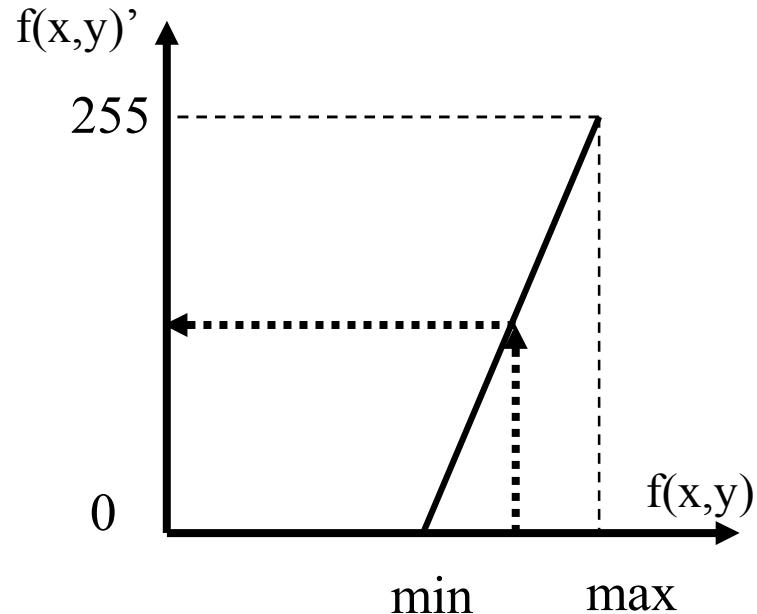
Transformation des niveaux de gris



7. Traitements de base d'une image

Opérations ponctuelles

Transformation des niveaux de gris



$$f(x,y)' = m \cdot f(x,y) + b$$

$$m = \frac{PIXMAX}{\max - \min} = \frac{255}{\max - \min}$$

$$b = -\left(\frac{PIXMAX \cdot \min}{\max - \min} \right)$$

$$f(x,y)' = \frac{PIXMAX}{\max - \min} f(x,y) - \left(\frac{PIXMAX \cdot \min}{\max - \min} \right)$$

$$f(x,y)' = \frac{PIXMAX}{\max - \min} (f(x,y) - \min)$$

$$\frac{PIXMAX}{\max - \min} = \frac{f(x,y)'}{f(x,y) - \min}$$

$$f(x,y)' = \frac{PIXMAX}{\max - \min} (f(x,y) - \min)$$

7. Traitements de base d'une image

Opérations ponctuelles

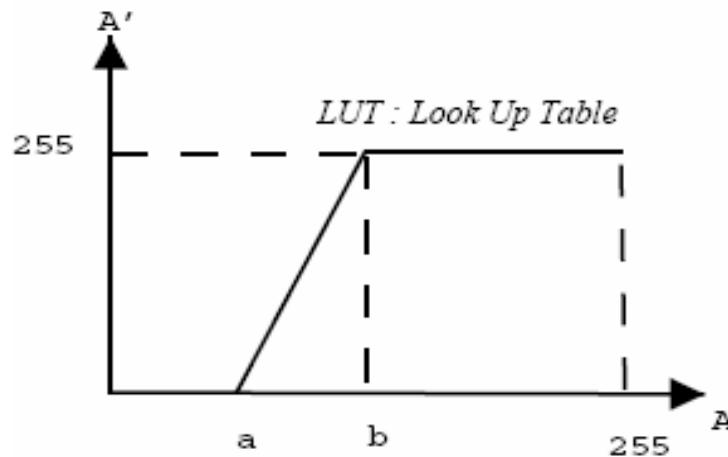
Transformation des niveaux de gris

- Expression de la dynamique (recadrage)

$$A[i, j] \xrightarrow{t} A'[i, j]$$

$$t : [a, b] \xrightarrow{t} [0, 255]$$

$$n \xrightarrow{t} n' = 255 * \frac{n - a}{b - a}$$



Recadrage d'histogramme

7. Traitements de base d'une image

Opérations ponctuelles

Amélioration du contraste

- Transformation linéaire
- Transformation linéaire avec saturation
- Transformation linéaire par bout avec saturation
- Égalisation de l'histogramme

7. Traitements de base d'une image

Opérations ponctuelles

Amélioration du contraste

- Transformation linéaire

$$f'(x, y) = \frac{PIXMAX}{\max - \min} (f(x, y) - \min)$$

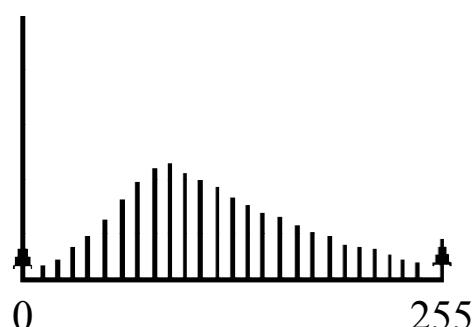
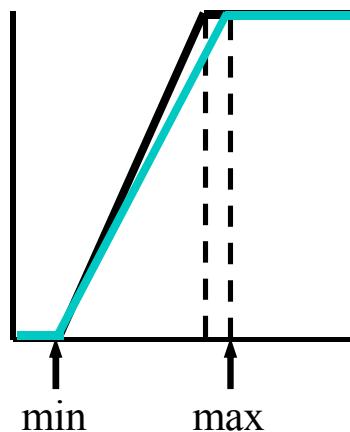
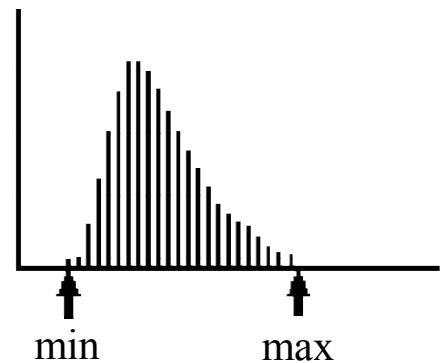
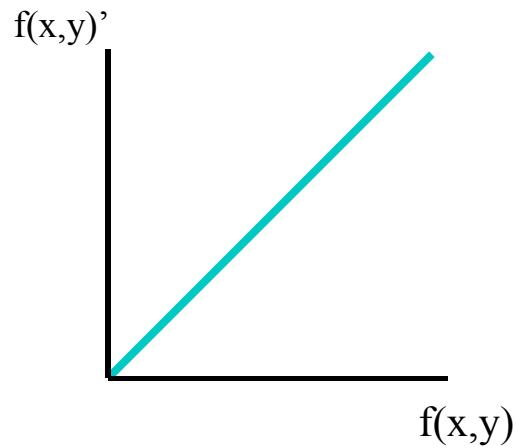
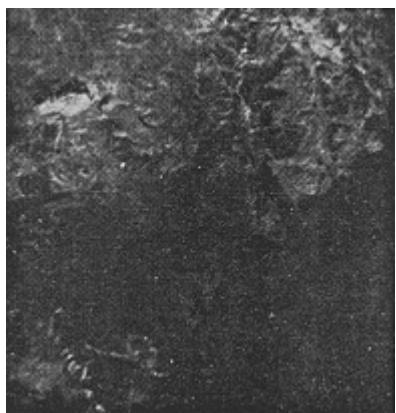
$$f'(x, y) = \frac{(f(x, y) - \min)}{\max - \min} PIXMAX$$

$$\frac{(f(x, y) - \min)}{\max - \min} \in [0, 1]$$

7. Traitements de base d'une image

Opérations ponctuelles

Transformation linéaire



7. Traitements de base d'une image

Opérations ponctuelles

Amélioration du contraste

◆ Transformation linéaire avec saturation

$$f'(x, y) = \frac{PIXMAX}{S_{\max} - S_{\min}} (f(x, y) - S_{\min})$$

$$f'(x, y) = \frac{(f(x, y) - S_{\min})}{S_{\max} - S_{\min}} PIXMAX$$

$$\frac{(f(x, y) - S_{\min})}{S_{\max} - S_{\min}} \in [0, 1]$$

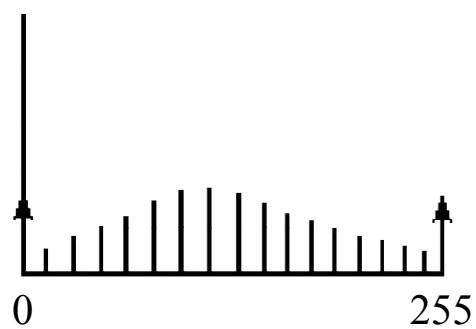
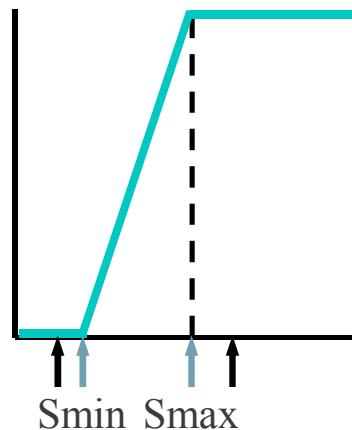
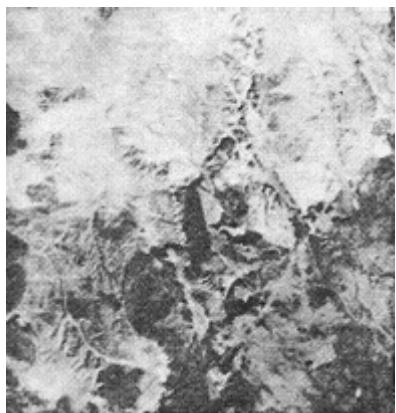
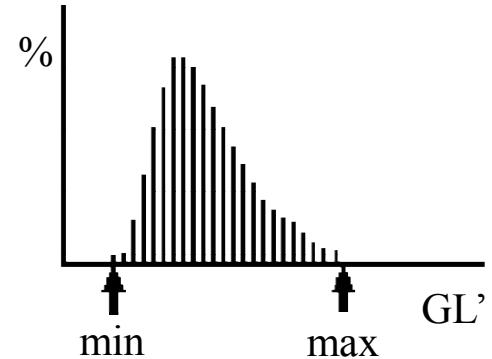
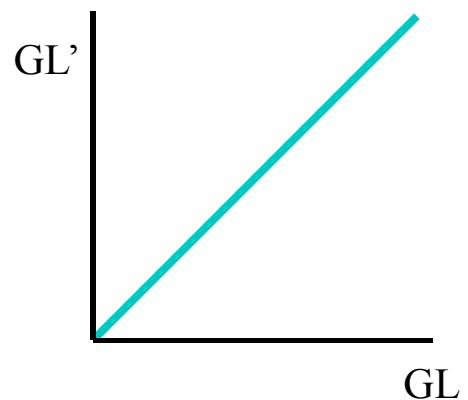
$$\min(f(x, y)) \leq S_{\min} < S_{\max} \leq \max(f(x, y))$$

7. Traitements de base d'une image

Opérations ponctuelles

Amélioration du contraste

Transformation linéaire avec saturation

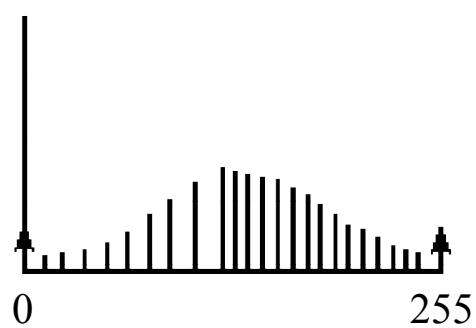
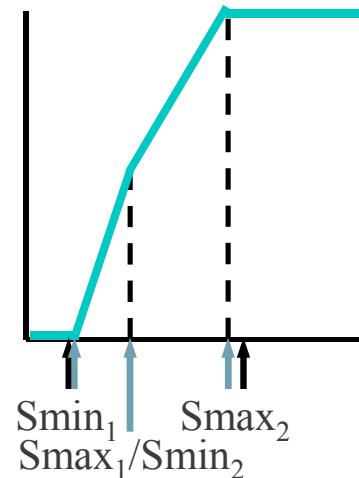
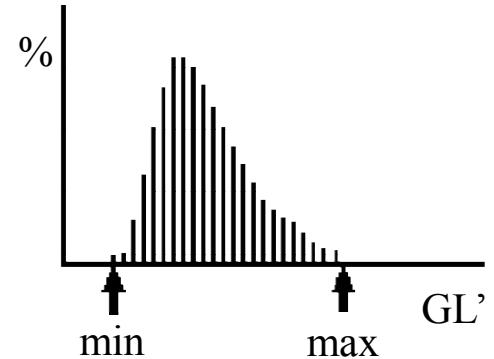
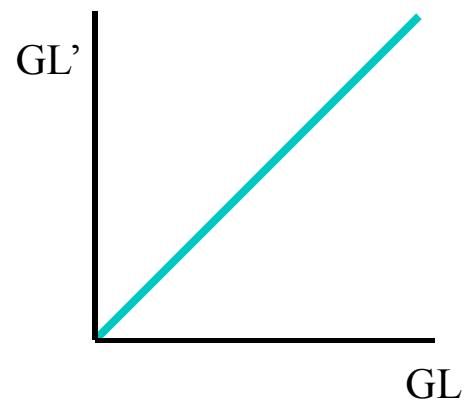


7. Traitements de base d'une image

Opérations ponctuelles

Amélioration du contraste

Transformation linéaire par bout avec saturation



7. Traitements de base d'une image

Opérations ponctuelles

Amélioration du contraste

◆ Transformation par égalisation d'histogramme

$$h[i], i \in [0, 255]$$

Histogramme non-normalisé

$$h_n[i] = \frac{h[i]}{DIMX \cdot DIMY}$$

Histogramme normalisé

$$C[i] = \sum_{j=0}^i h_n[j]$$

Densité de probabilité cumulative

$$f'(x, y) = C[f(x, y)] \cdot PIXMAX$$

$$C[f(x, y)] \in [0, 1]$$

7. Traitements de base d'une image

Opérations ponctuelles

Amélioration du contraste

Égalisation de l'histogramme

- Le but est de rendre l'histogramme le plus plat possible
 - Distribution uniforme des niveaux de gris
 - Maximisation de l'entropie de l'image
 - Image donnant une information maximale

7. Traitements de base d'une image

Opérations ponctuelles

Amélioration du contraste

Égalisation de l'histogramme

$$f(x, y)' = (2^D - 1) \frac{HC(f(x, y))}{wh}$$

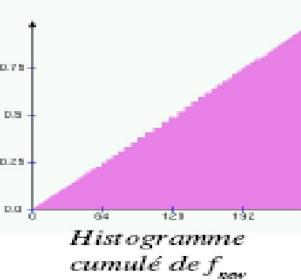
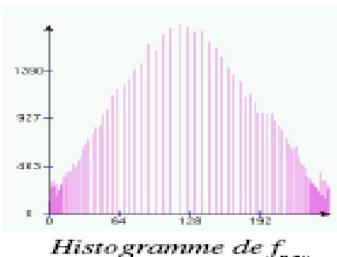
D: Dynamique

(w,h): Dimension de l'image

HC(.): Histogramme cumulé



Original $f[x, y]$



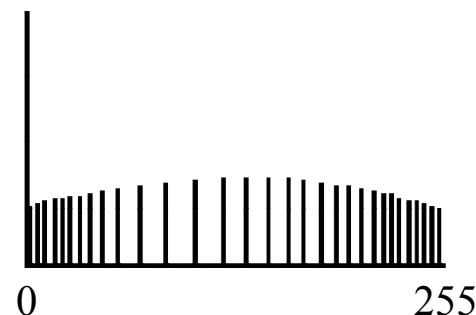
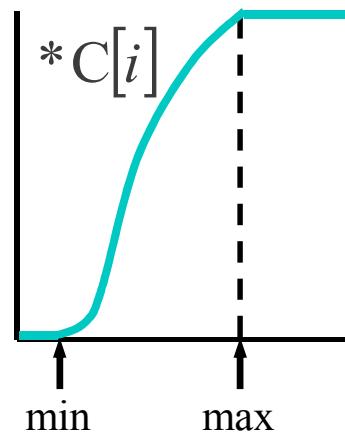
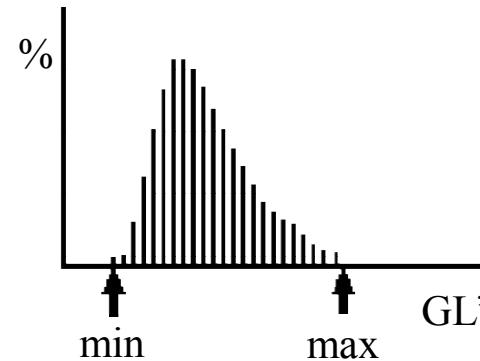
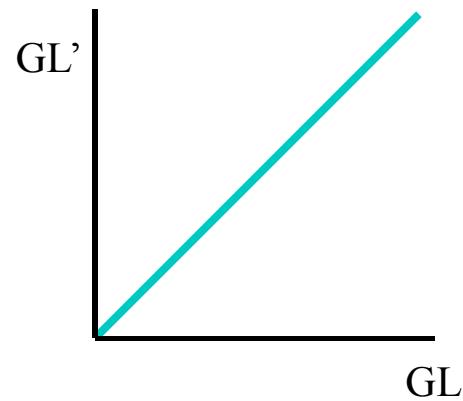
Après égalisation $f_{new}[x, y]$

7. Traitements de base d'une image

Opérations ponctuelles

Amélioration du contraste

Égalisation de l'histogramme [densité de probabilité cumulative]



7. Traitements de base d'une image

Opérations ponctuelles

Opérations ponctuelles linéaires

Forme générale

- $g(x,y) = Pf(x,y) + L$
 - Il faut éviter $g(x,y) < 0$ et $g(x,y) > K-1$
 - Où $K-1 =$ intensité maximale (255, souvent)

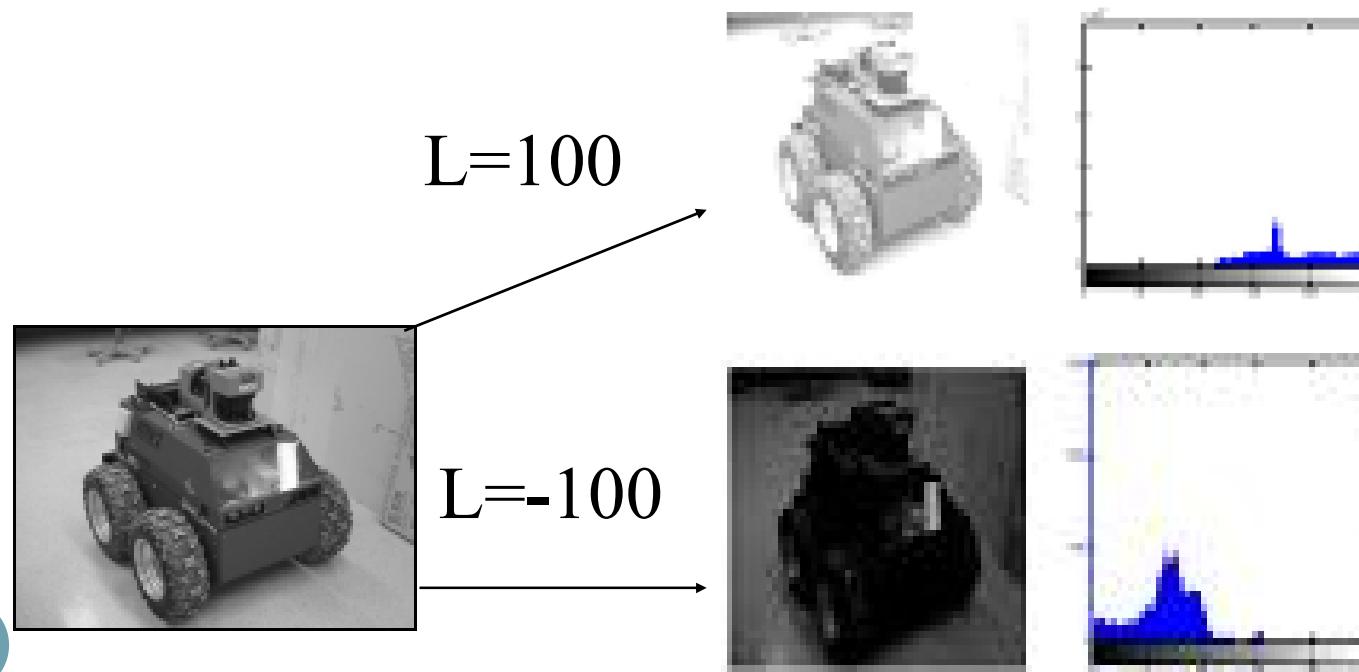
7. Traitements de base d'une image

Opérations ponctuelles

Opérations ponctuelles linéaires

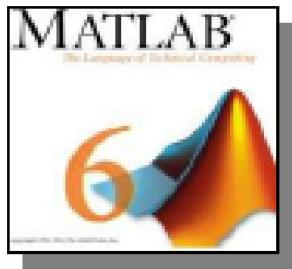
Décalage additif

- Contrôle de la luminosité
 - Ici $P=1$ et $-(K-1) \leq L \leq K-1$
 - $g(x,y)=f(x,y)+L$



7. Traitements de base d'une image

Opérations ponctuelles



Code Matlab

■ Décalage additif

```
img = imread('atrvgrey.bmp');
L = 100;
img = uint8( double(img) + L );
imshow(img)
figure, imhist(img,256);
```

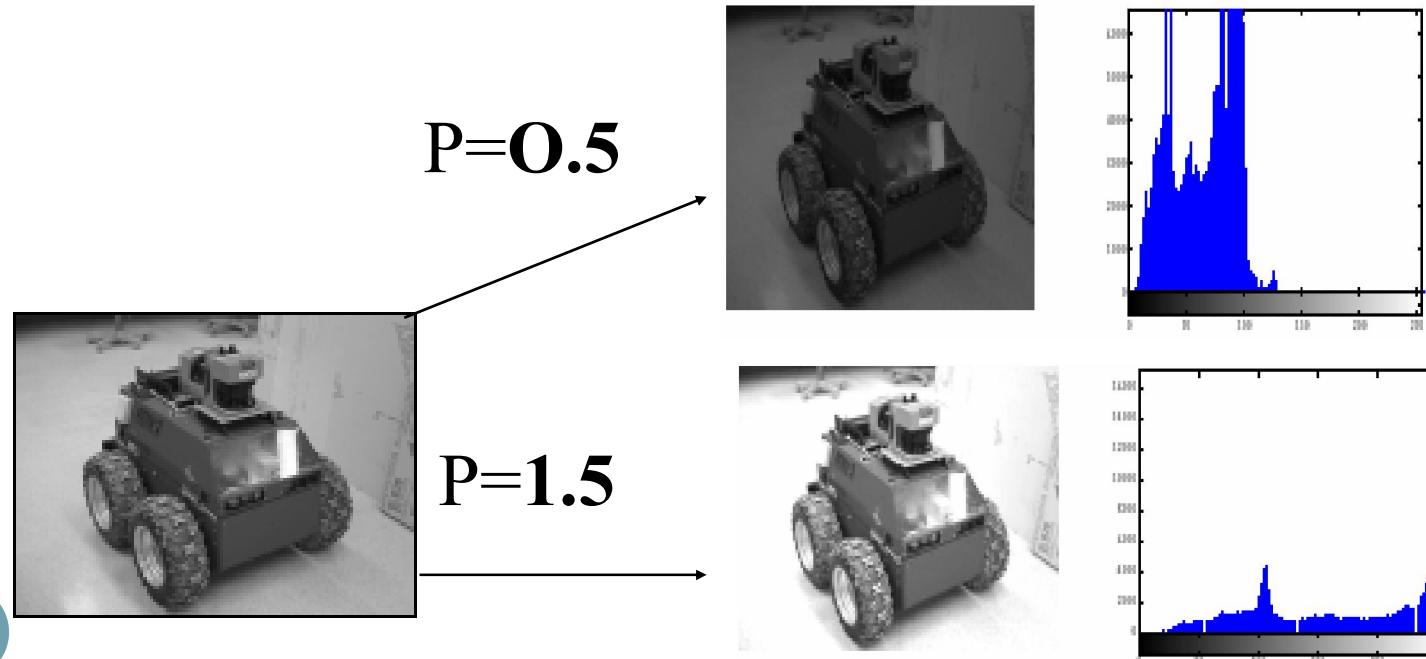
7. Traitements de base d'une image

Opérations ponctuelles

Opérations ponctuelles linéaires

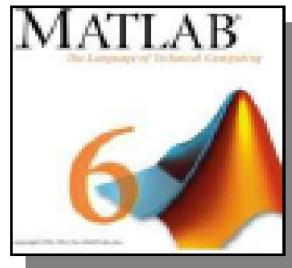
Mise à l'échelle multiplicative

- Ici $L=0$ et $P>0$
 - $g(x,y)=Pf(x,y)$
 - Modifie la luminosité et le contraste



7. Traitements de base d'une image

Opérations ponctuelles



Code Matlab

■ Mise à l'échelle multiplicative

```
img = imread('atrvgray.bmp');
P = 0.5;
img = uint8( P*double(img) );
imshow(img)
figure, imhist(img,256);
```

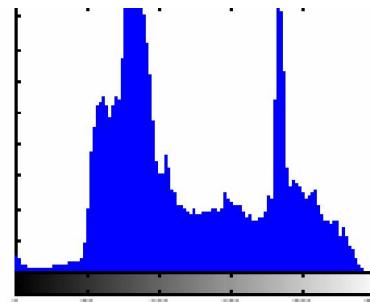
7. Traitements de base d'une image

Opérations ponctuelles

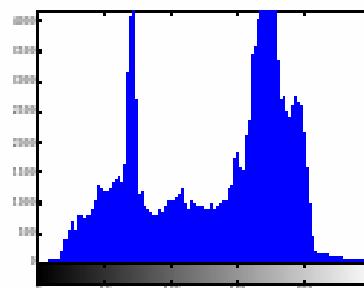
Opérations ponctuelles linéaires

Inversion

- Ici $P=-1$ et $L=255$
 - $g(x,y) = -f(x,y) + 255$

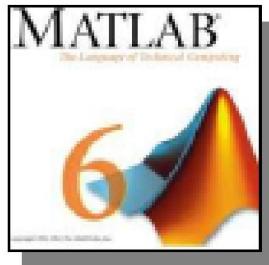


Inversion



7. Traitements de base d'une image

Opérations ponctuelles



Code Matlab

■ Inversion

```
img = imread('atrvgrey.bmp');
img = uint8( -double(img) + 255 );
imshow(img)
figure, imhist(img,256);
```

7. Traitements de base d'une image

Opérations ponctuelles

Opérations ponctuelles linéaires

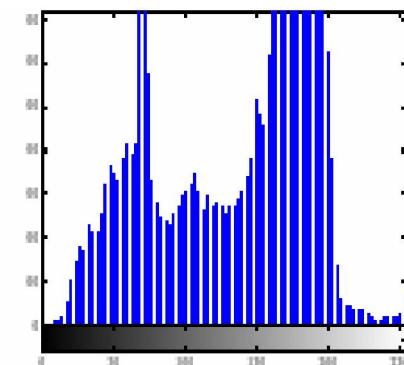
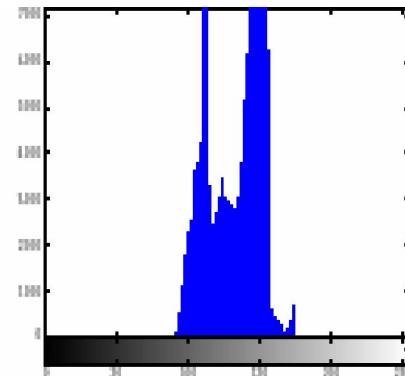
Amélioration du contraste

- On étire l'histogramme pour qu'il utilise toute la plage disponible
 - On pose $A = \min(f(x,y))$ et $B = \max(f(x,y))$
 - On veut que les nouvelles limites soient A' et B'
 - On cherche $PA + L = A'$ et $PB + L = B'$
 - 2 éq et 2 inconnues : P et L



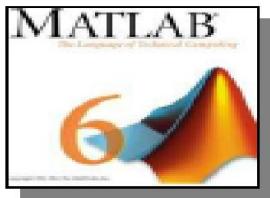
$$P = (B' - A') / (B - A)$$

$$L = (A'B - AB') / (B - A)$$



7. Traitements de base d'une image

Opérations ponctuelles



Code Matlab

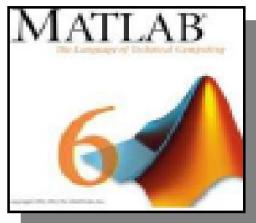
■ Amélioration du contraste

```
img = imread('atriv_lowc.bmp');
A = double(min(min(img)));
B = double(max(max(img)));
% formules simplifiées pour [0..255]
P = 255/(B-A);
L = -P*A;
img = uint8( P*double(img) + L );
imshow(img)
figure, imhist(img,256);
```

8. Opérations arithmétiques et logiques entre les images

- Somme d'images
 - Ajout de bruit, fusion d'images,...
- Soustraction d'images
 - Élimination de l'arrière-plan, dérivation par rapport au temps,...
- Moyenne point à point d'images
 - Réduction du bruit
- Produit point à point
 - Convolution (lissage, dérivation, ...)
- Seuillage
 - Donne une **image binaire** à partir d'une image en niveaux de gris

8. Opérations arithmétiques et logiques entre les images



Code Matlab

■ Opérations arithmétiques et logiques

```
img = imread('atriv_lowc.bmp');
img2 = imread('atriv_highc.bmp');
seuil = 128;

somme      = uint8( double(img) + double(img2) );
difference = uint8( double(img) - double(img2) );
produit    = uint8( double(img) .* double(img2) );
moyenne    = uint8( (double(img) + double(img2)) ./2 );
img( img < seuil ) = 0;
img( img >= seuil ) = 255;
```

8. Opérations arithmétiques et logiques entre les images

Les sources de bruit

- Bruits liés aux conditions de prise de vues
 - Ce sont des événements vérifiant les conditions d'acquisition du signal
 - Le bougé
 - Problèmes liés à l'éclairage de la scène observée
- Bruits liés au capteur
 - Capteur mal réglé
 - Capteur de mauvaise qualité

8. Opérations arithmétiques et logiques entre les images

Les sources de bruit

- Bruits liés à l'échantillonnage
 - Ces bruits reflètent essentiellement des problèmes de quantification (CCD)
 - Précision d'environ 1/512
 - Problèmes dans le cas d'applications de grande précision
- Bruits liés à la nature de la scène
 - Nuage sur les images satellitaires
 - Poussières dans les scènes industrielles
 - Brouillard pour les scènes routières

8. Opérations arithmétiques et logiques entre les images

- Réduction du bruit par moyennage
 - Soit n images d'une même scène, chaque image peut s'exprimer $g_i(x, y) = f(x, y) + b_i(x, y)$
 - $f(x, y)$ est l'image non bruitée
 - $b_i(x, y) \sim N(0, \sigma^2)$ est un bruit blanc non corrélé

$$E\{b_i(x, y)b_j(x, y)\} = 0 \text{ pour } i \neq j$$

- L'image moyenne

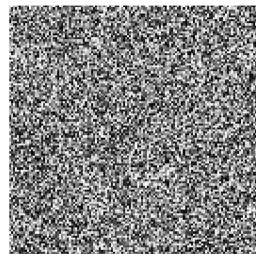
$$\bar{g}(x, y) = \frac{1}{n} \sum_{i=1}^n g_i(x, y)$$

8. Opérations arithmétiques et logiques entre les images

- Différents types de bruit
 - Bruit additif ou multiplicatif
 - $P'(i) = b(i) + P(i)$
 - $P'(i) = b(i) * P(i)$

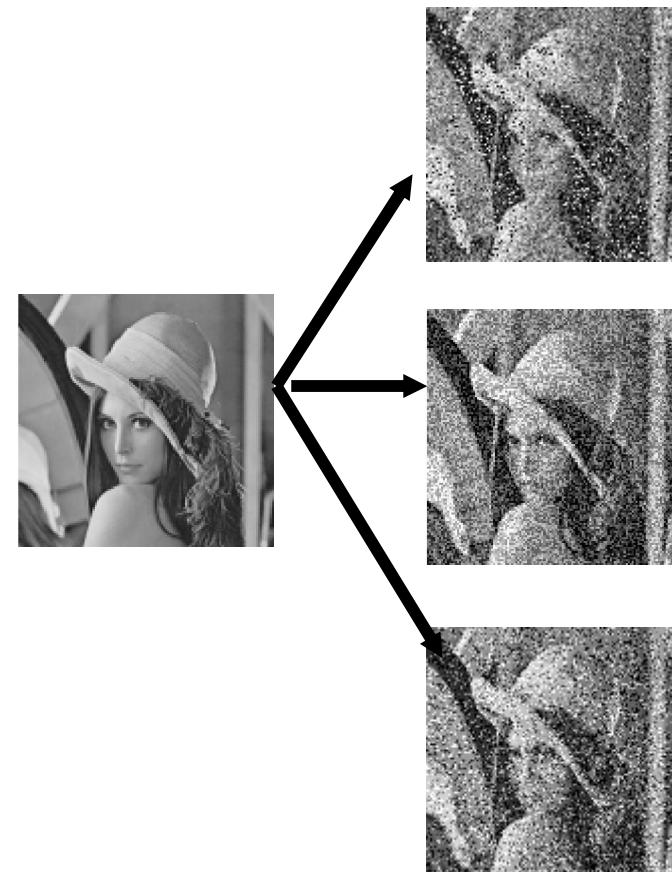


et



8. Opérations arithmétiques et logiques entre les images

- Différents types de bruit
 - Comment est générée $b(i)$?
 - Loi gaussienne
 - Loi uniforme
 - Aléatoire
- Principaux types de bruit
 - Gaussien
 - Poivre et sel



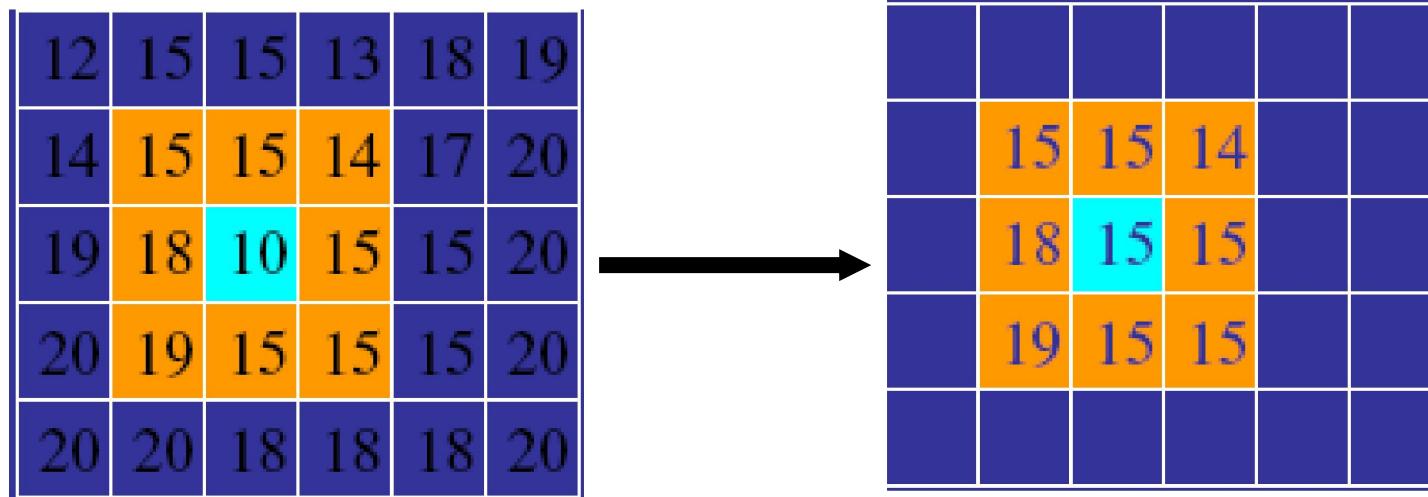
8. Opérations arithmétiques et logiques entre les images

- Comment réduire le bruit ?
 - Démarche générale :
 - Caractériser le bruit présent dans l'image
 - Choisir un algorithme approprié
 - Pour chaque pixel de l'image
 - Étudier la valeur du pixel
 - Étudier les valeurs des voisins
 - Modifier la valeur du pixel

8. Opérations arithmétiques et logiques entre les images

- Filtre Moyenneur
 - On affecte à chaque pixel la valeur moyenne obtenue dans le voisinage : avec j les voisins de i

$$P'(i) = \frac{1}{n} \sum_{j=1}^n (P(j))$$



8. Opérations arithmétiques et logiques entre les images

- Élimination du bruit par moyennage

- L'image moyenne:

$$\bar{g}(x, y) = \frac{1}{n} \sum_{i=1}^n (f(x, y) + b_i(x, y))$$

$$g(x, y) = f(x, y) + \frac{1}{n} \sum_{i=1}^n (b_i(x, y))$$

- Signal inchangé et un nouveau bruit
 - Moyenne du nouveau bruit nulle, comme avant

$$\begin{aligned} E\left\{\frac{1}{n} \sum_{i=1}^n (b_i(x, y))\right\} &= \frac{1}{n} \sum_{i=1}^n E(b_i(x, y)) \\ &= 0 \end{aligned}$$

8. Opérations arithmétiques et logiques entre les images

- Calcul de la variance du bruit/

$$\text{Var}(X) = E\{(X - E\{X\})^2\}$$

$$\begin{aligned}\sigma^2 &= E\left\{\left(\frac{1}{n} \sum_{i=1}^n b_i^2(x, y)\right)^2\right\} \\ &= \frac{1}{n^2} \sum_{i=1}^n \sum_{j \neq i} E\{b_i(x, y)b_j(x, y)\} \\ &= \frac{1}{n^2} \sum_{i=1}^n \left(E\{b_i(x, y)b_i(x, y)\} + \sum_{j \neq i}^n E\{b_i(x, y)b_j(x, y)\} \right) \\ &= \frac{1}{n^2} \sum_{i=1}^n E\{b_i^2(x, y)\} \quad \text{Car bruit non corrélé} \\ &= \frac{\sigma^2}{n}\end{aligned}$$

9. Traitements de base d'une image

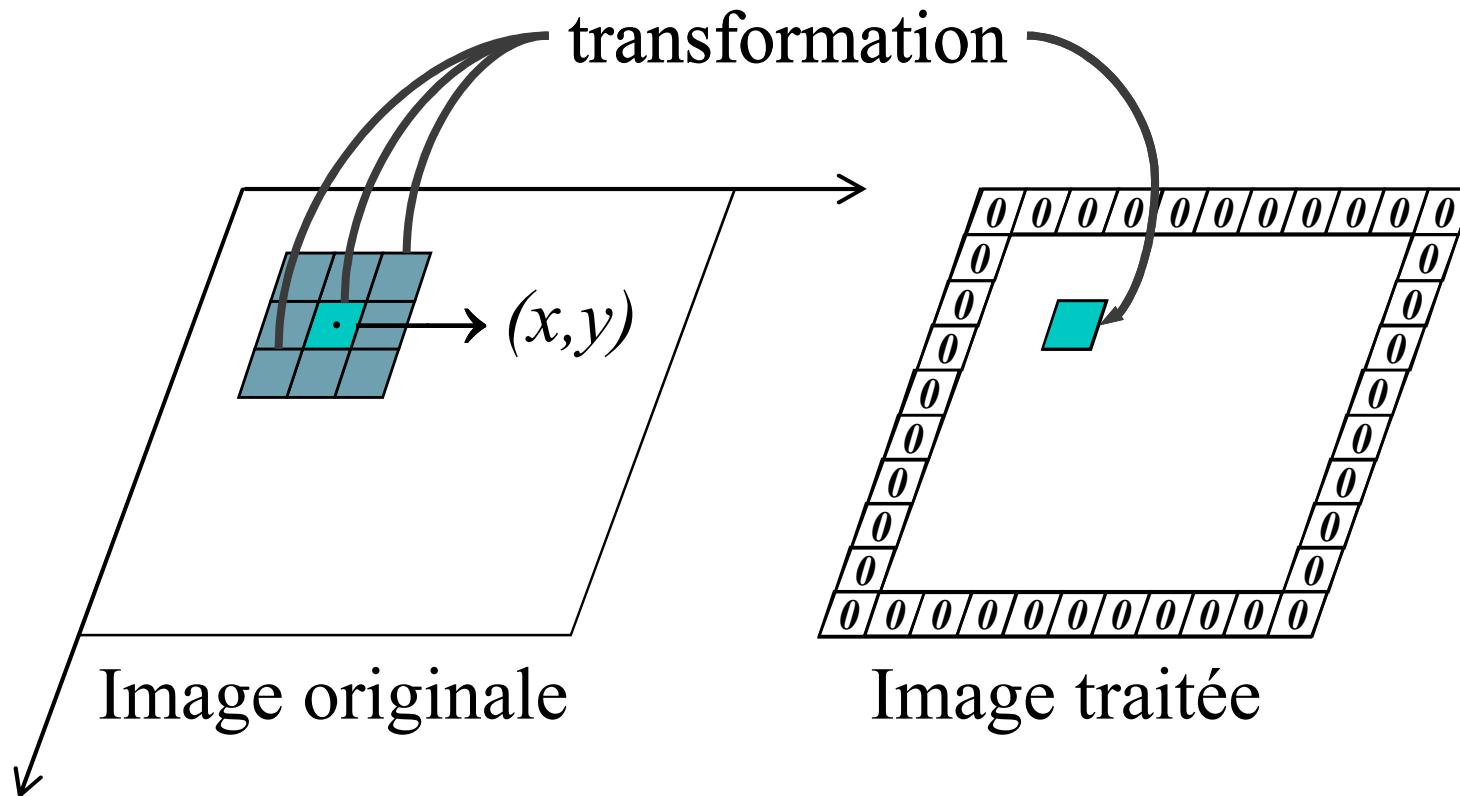
Opérations locales

- Filtres passe-bas
 - Filtre de moyenne
 - Filtre Gaussien
 - Triangle de Pascal
 - Autres filtres
- Moyenne versus médiane
- Filtre médian
- Filtre adaptatif
- Filtres passe-haut
 - Opérateurs mathématique
 - Masque de détection
 - Gradient
 - Laplacien
 - Kirsch
 - Marr-Hildreth
 - Marr-Hildreth versus sobel

9. Traitements de base d'une image

Opérations locales

Transformation basée sur le voisinage d'un point (x,y) :



9. Traitements de base d'une image

Opérations locales

Convolution discrète

- La convolution discrète est un outil permettant la construction de **filtres linéaires** ou de **filtres de déplacements invariants**
- L 'équation de convolution, notée $g(x)$, de la séquence $f(x)$ avec une fonction $h(x)$ est :

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

- $h(x)$ est appelée *masque de convolution, noyau de convolution, filtre, fenêtre, kernel, ...*
- En pratique, la convolution numérique d'une image se fera simplement par une **sommation de multiplications**.

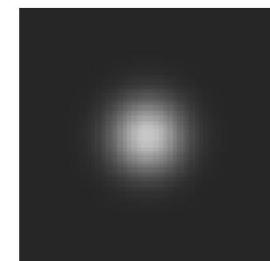
9. Traitements de base d'une image

Opérations locales

Exemple de convolution 2D



=



*

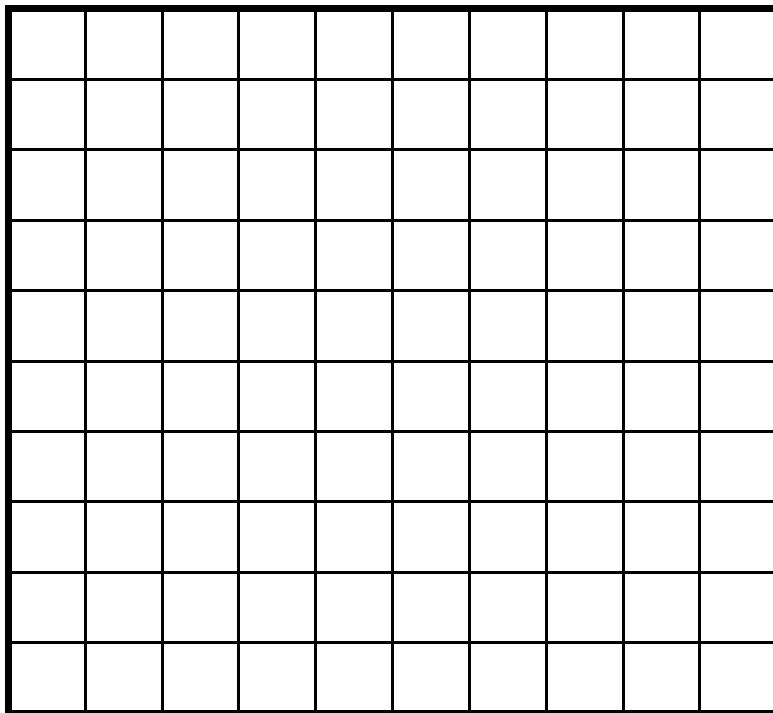


9. Traitements de base d'une image

Opérations locales

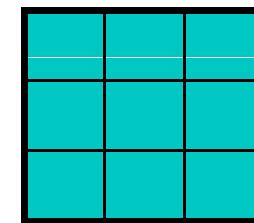
Exemple de convolution 2D

I



Image

*

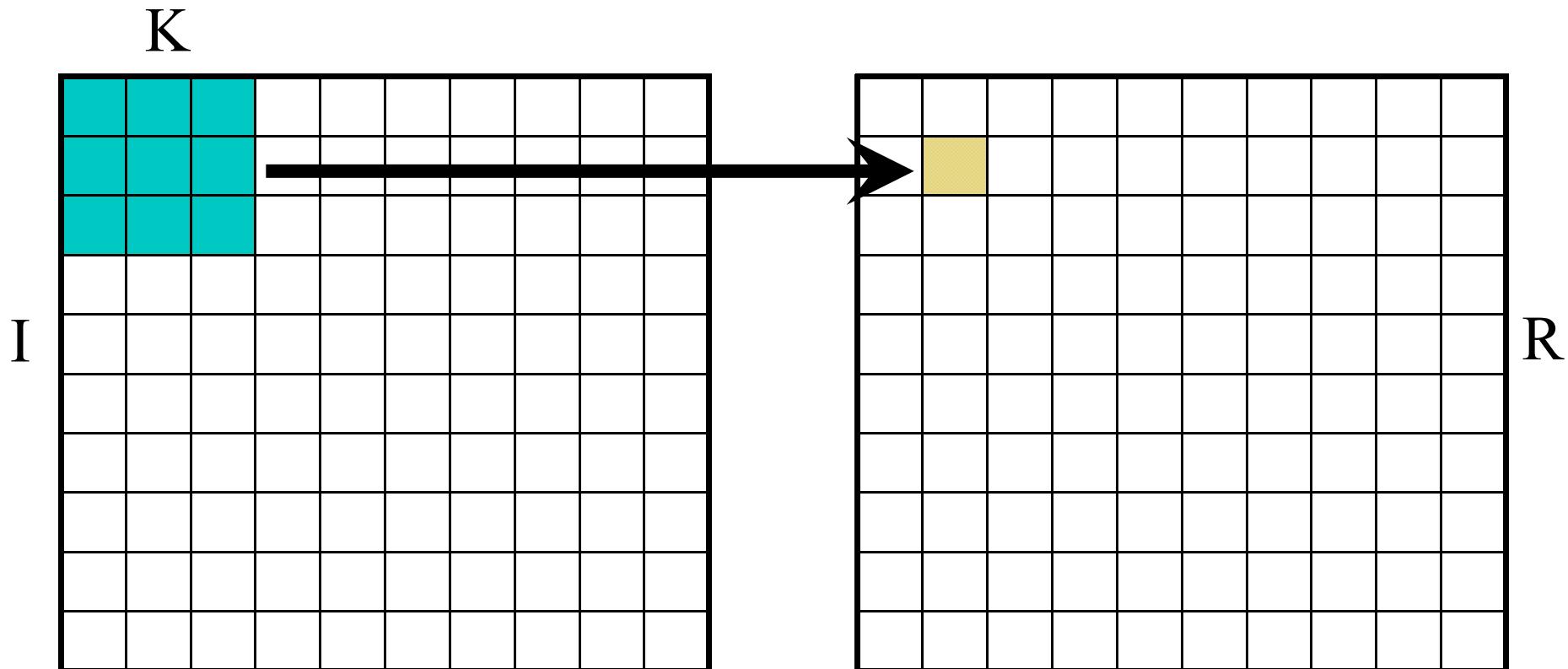


Noyau de convolution

9. Traitements de base d'une image

Opérations locales

Convolution numérique $R = I * K$

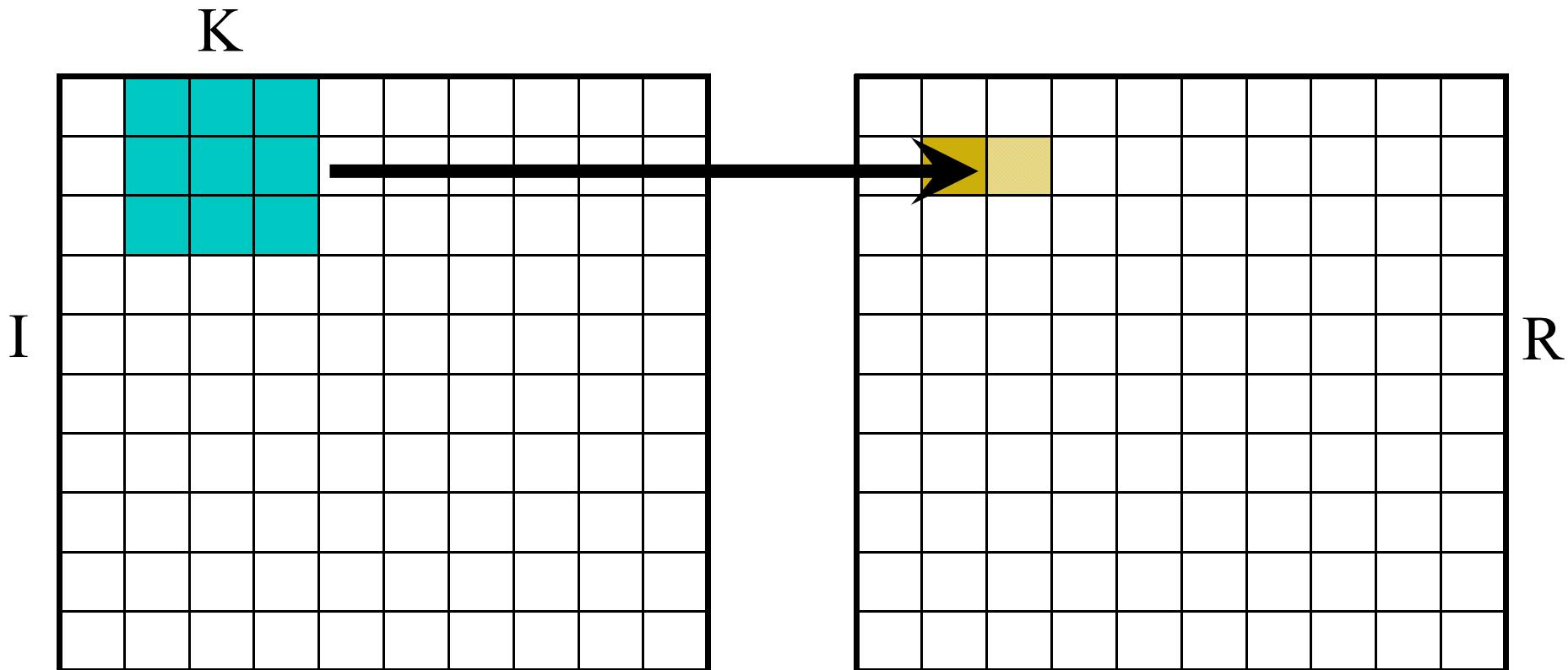


$$\begin{aligned} R(1,1) &= I(0,0) K(0,0) + I(1,0) K(1,0) + I(2,0) K(2,0) \\ &\quad + I(0,1) K(0,1) + I(1,1) K(1,1) + I(2,1) K(2,1) \\ &\quad + I(0,2) K(0,2) + I(1,2) K(1,2) + I(2,2) K(2,2) \end{aligned}$$

9. Traitements de base d'une image

Opérations locales

Convolution numérique $R = I * K$

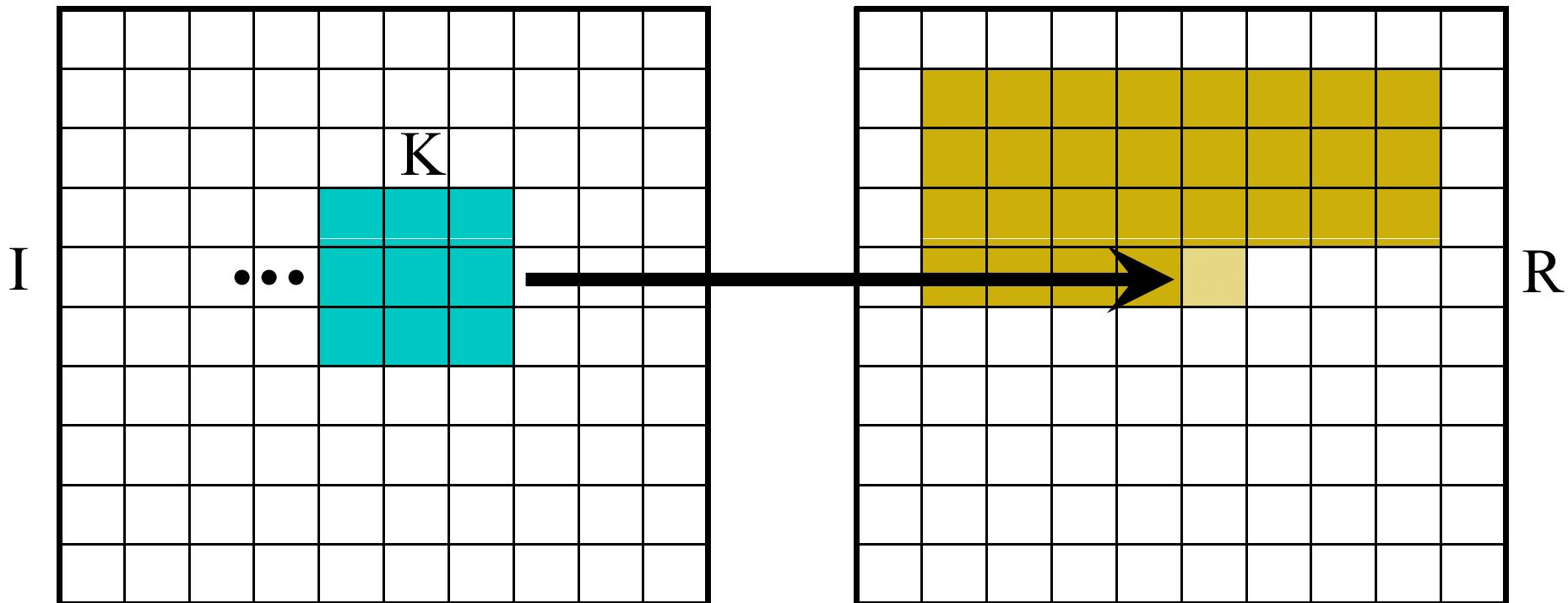


$$\begin{aligned} R(2,1) = & I(1,0) K(0,0) + I(2,0) K(1,0) + I(3,0) K(2,0) \\ & + I(1,1) K(0,1) + I(2,1) K(1,1) + I(3,1) K(2,1) \\ & + I(1,2) K(0,2) + I(2,2) K(1,2) + I(3,2) K(2,2) \end{aligned}$$

9. Traitements de base d'une image

Opérations locales

Convolution numérique $R = I * K$

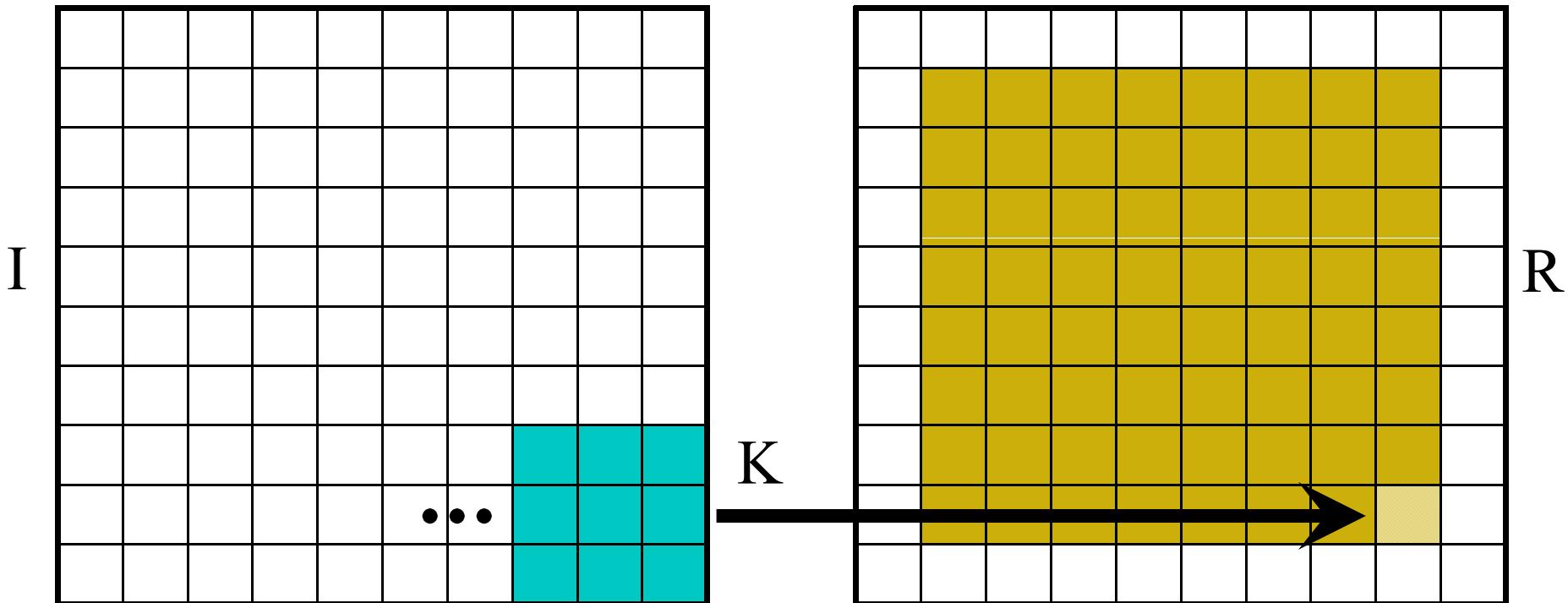


$$\begin{aligned} R(x,y) = & I(x-1,y-1) K(0,0) + I(x, y-1) K(1,0) + I(x+1, y-1) K(2,0) \\ & + I(x-1,y) K(0,1) + I(x,y) K(1,1) + I(x+1,y) K(2,1) \\ & + I(x-1,y+1) K(0,2) + I(x,y+1) K(1,2) + I(x+1,y+1) K(2,2) \end{aligned}$$

9. Traitements de base d'une image

Opérations locales

Convolution numérique $R = I * K$



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

9. Traitements de base d'une image

Opérations locales

Convolution numérique

- Problème : Que faire avec les bords de l'image ?
 - Mettre à zéro (0)
 - Convolution partielle
 - Sur une portion du noyau
 - Miroir de l'image
 - $f(x-1,y) = f(x+1,y)$
 - ... (pas de solution miracle)

9. Traitements de base d'une image

Opérations locales

Masque de convolution

- Le masque de convolution est le plus souvent
 - Carré
 - De taille 3x3 ou 5x5 (ou plus, mais impair)
- Ce masque représente un filtre linéaire permettant de modifier l'image.
- La plupart du temps, on divisera le résultat de la convolution par la somme des coefficients du masque.
 - Pour éviter de modifier l'entropie de l'image, la somme des coefficients doit être égale à 1.
 - Dans le cas du Laplacien (plus loin), la somme sera égale à zéro.

9. Traitements de base d'une image

Opérations locales

Transformation par convolution

$$g(x, y) = \sum_{m=-\frac{DF}{2}}^{\frac{DF}{2}} \sum_{n=-\frac{DF}{2}}^{\frac{DF}{2}} f(x+m, y+n) \cdot PSF\left(\frac{DF}{2} + m, \frac{DF}{2} + n\right)$$



OU

Point Spread Function

$1/9 X$

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

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

$\left. \begin{array}{l} DF \text{ ou} \\ Dimension \\ Filtre : 3 \end{array} \right\}$

résultat divisé par la somme
des poids $[w_x]$ du filtre (P-B)
ou son nombre de pixels (P-H)

9. Traitements de base d'une image

Opérations locales

Typologie des Filtres

- moyenneur : $H = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ séparable : $\frac{1}{3} [1 \ 1 \ 1] * \frac{1}{3} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$
- binomial gaussien : $H = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$ séparable : $\frac{1}{4} [1 \ 2 \ 1] * \frac{1}{4} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$
- laplacien : $H = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$
- Sobel horizontal : $H = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$ séparable : $[-1 \ 0 \ 1] * \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$

Pour accélérer les traitements, il est possible de décomposer les filtres en sous-filtres équivalents qu'on passe un après l'autre.

9. Traitements de base d'une image

Opérations locales

Typologie des Filtres

- gradient oblique : $H = \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$

- rehausseur de contraste : $H = \begin{bmatrix} 1 & -3 & 1 \\ -3 & 9 & -3 \\ 1 & -3 & 1 \end{bmatrix}$ séparable : $\begin{bmatrix} -1 & 3 & -1 \end{bmatrix} * \begin{bmatrix} -1 \\ 3 \\ -1 \end{bmatrix}$

9. Traitements de base d'une image

Opérations locales

Typologie des Filtres : suite

- La mise en cascade de filtres (associativité de la convolution) permet de générer des filtres de grande taille à moindre coût de calculs

$$H_{3 \times 3} * H_{3 \times 3} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} * \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} = H_{5 \times 5}$$

9. Traitements de base d'une image

Opérations locales

Filtres passe-bas

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

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

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

9. Traitements de base d'une image

Opérations locales

Filtres passe-bas

- **Filtre de moyenne**

- Plus le filtre est grand, plus on **coupe les hautes fréquences** (les changements rapides)

1/9*

1	1	1
1	1	1
1	1	1

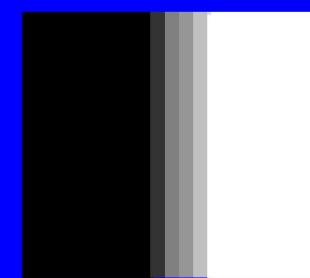
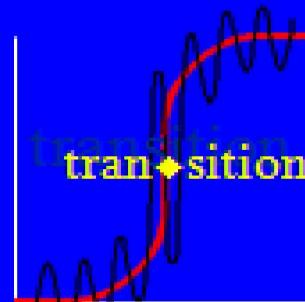
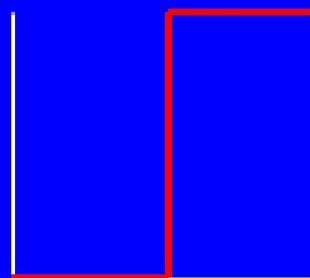
1/25*

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

9. Traitements de base d'une image

Opérations locales

Filtre de moyenne (passe-bas)



→ Lissage (flou apparent)

$I/25 \times$

I	I	I	I	I
I	I	I	I	I
I	I	I	I	I
I	I	I	I	I
I	I	I	I	I

N.B. plus le filtre grossit,
plus le lissage devient important
et plus le flou s'accentue !

9. Traitements de base d'une image

Opérations locales

Filtres passe-bas

- Filtre de [moyenne](#)



Image originale



Image filtrée

9. Traitements de base d'une image

Opérations locales

Filtres passe-bas

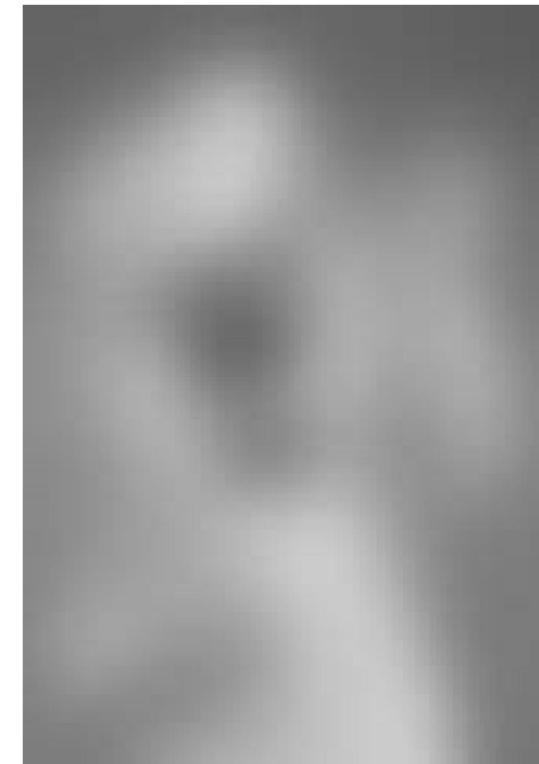
- Exemples de filtres moyennes



Original



Moyenne 5x5



Moyenne 11x11

9. Traitements de base d'une image

Opérations locales

Filtres passe-bas

- Filtre **Gaussien**

(1/16)

1	2	1
2	4	2
1	2	1



Ce filtre est construit à partir du binôme de Newton (ou du triangle de Pascal)... La suite du binôme converge vers une gaussienne

$$(1/4) \begin{matrix} 1 & 1 \\ 1 & 1 \end{matrix}$$

$$* (1/4) \begin{matrix} 1 & 1 \\ 1 & 1 \end{matrix}$$

= (1/16)

1	2	1
2	4	2
1	2	1

$$(1/4) \begin{matrix} 1 \\ 2 \\ 1 \end{matrix}$$

$$* (1/4) \begin{matrix} 1 & 2 & 1 \end{matrix}$$

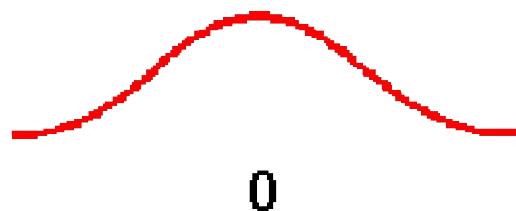
1	2	1
2	4	2
1	2	1

9. Traitements de base d'une image

Opérations locales

Filtre gaussien (passe-bas)

$$gauss(x, y) = PSF(x, y) = \frac{1}{2\pi\sigma^2} \cdot e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$



fonction gaussienne 2-D

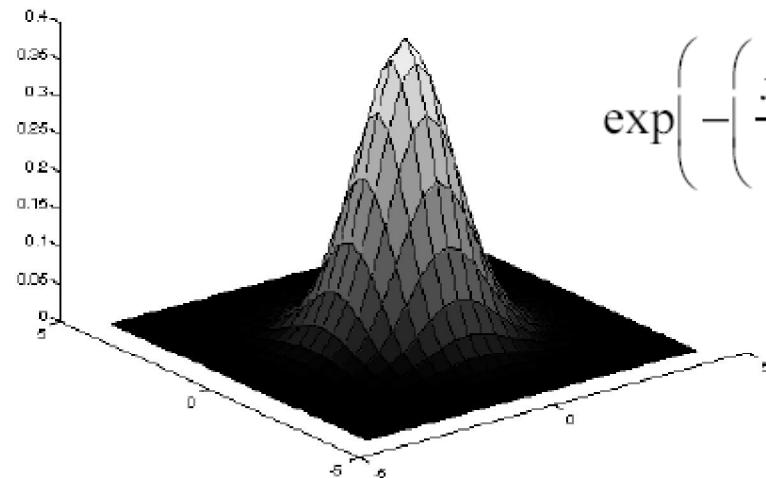
w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

$$\left. \begin{array}{l} Dim X = Dim Y = 8\sigma + 1 \\ \text{où } \sigma \in]0,3[\end{array} \right\}$$

9. Traitements de base d'une image

Opérations locales

Filtre gaussien (passe-bas)



Fonction gaussienne en 3D

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

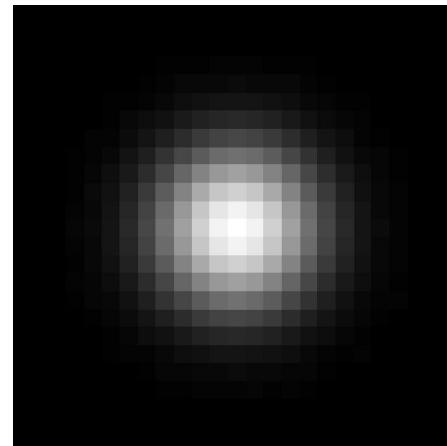


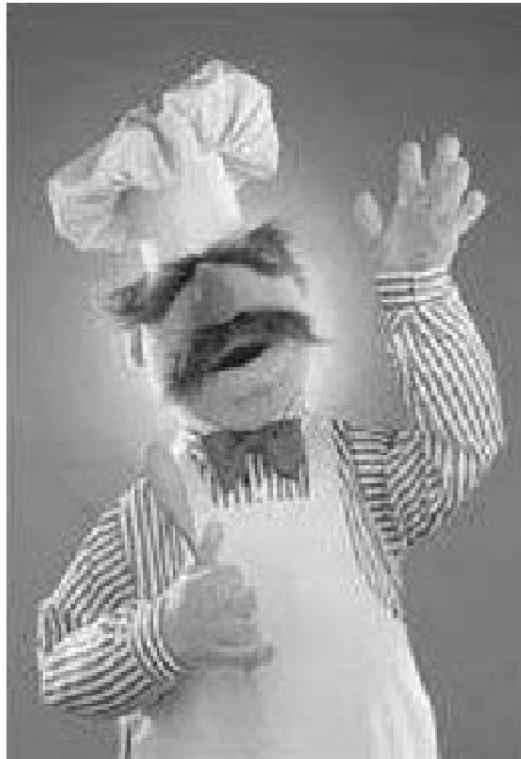
Image d'une gaussienne

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

9. Traitements de base d'une image

Opérations locales

Exemples de filtres gaussiens



Original



Gauss 5x5



Gauss 11x11

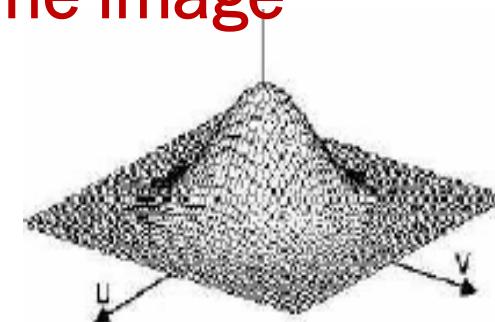
9. Traitements de base d'une image

Opérations locales

Filtres passe-bas

- Triangle de Pascal

- Chaque élément est la somme des éléments supérieur et supérieur gauche
- On choisit une ligne impaire
- On convolute un filtre horizontal avec un filtre vertical
 - On obtient un filtre carré
- On normalise par la somme de tous les éléments du filtre



1						
1	1					
1	2	1				
1						
1	3	3	1			
1	4	6	4	1		
1	5	10	10	5	1	
1	6	15	20	15	6	1

9. Traitements de base d'une image

Opérations locales

Filtres passe-bas

- Autres filtres passe-bas

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

1/256

Filtre Gaussien ou binomial

1	2	3	2	1
2	4	6	4	2
3	6	9	8	3
2	4	6	4	2
1	2	3	2	1

1/81

Filtre pyramidal

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

1/25

Filtre Conique



Obtenu par la convolution de deux filtres de moyenne 3X3

9. Traitements de base d'une image

Opérations locales

Moyenne versus médiane

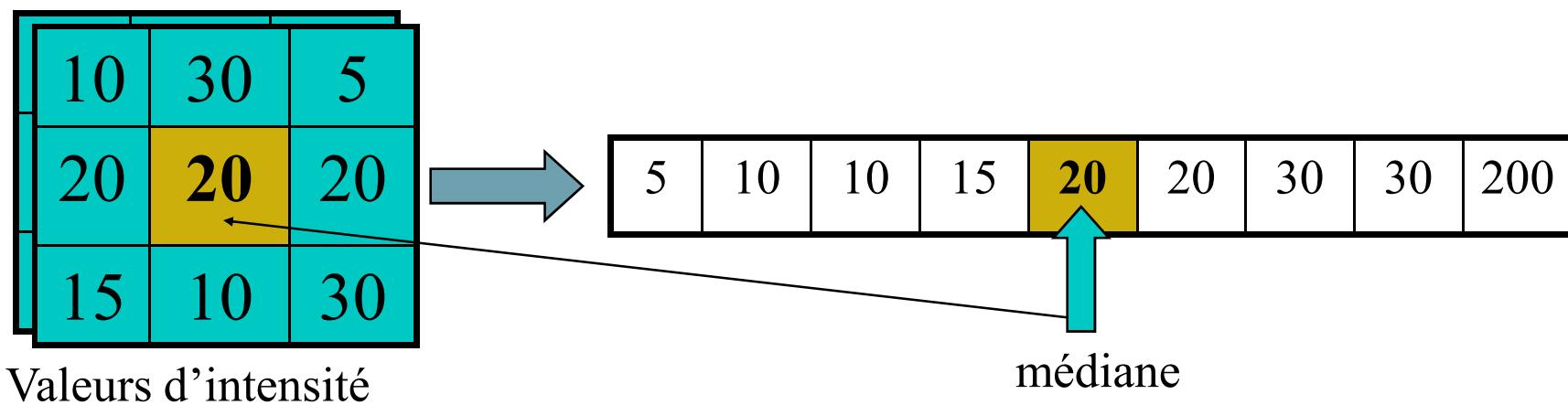
- Moyenne d'une suite de x_i : $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$
 - Facile à calculer
 - Complexité $O(n)$
 - Permet de trouver le "centre" d'une distribution x_i
 - Sensible aux valeurs extrêmes

9. Traitements de base d'une image

Opérations locales

Filtre Médian

- Filtre non linéaire (ce n'est pas une convolution)
 - Coûteux en temps de calcul
 - Complexité $O(n \log(n))$
 - On remplace le pixel central par la médiane du groupe
 - Sert à éliminer les points isolés sans filtrer par un filtre passe-bas
 - Plus robuste aux valeurs extrêmes



9. Traitements de base d'une image

Opérations locales

Filtre Médian

- On affecte à chaque pixel la valeur médiane obtenue dans le voisinage
- On évite ainsi les valeurs extrêmes

12	90	15	13	18	90
14	15	15	90	17	20
0	18	90	15	15	0
20	0	15	15	90	20
20	20	90	18	18	20



	15	15	90		
	18	15	15		
	0	15	15		

Moy = 30

9. Traitements de base d'une image

Opérations locales

Filtre Médian

- Exemple bruit poivre et sel



Image bruitée



Filtre gaussien $r=3$ pix



Filtre médian $r=3$ pix

9. Traitements de base d'une image

Opérations locales

Filtre Médian

- Exemple bruit quelconque

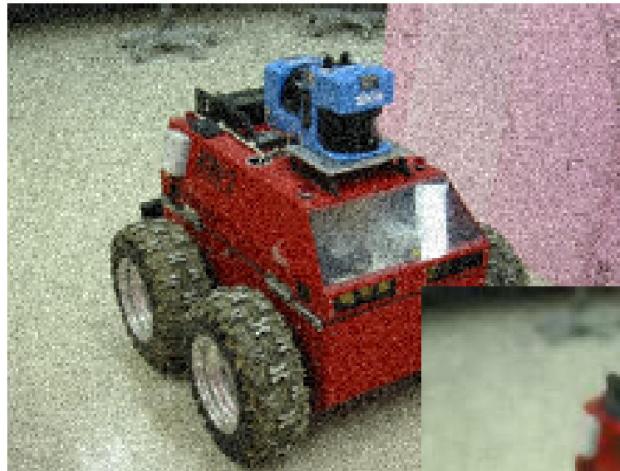


Image bruitée



Filtre gaussien $r=3$ pix



Filtre médian $r=3$ pix

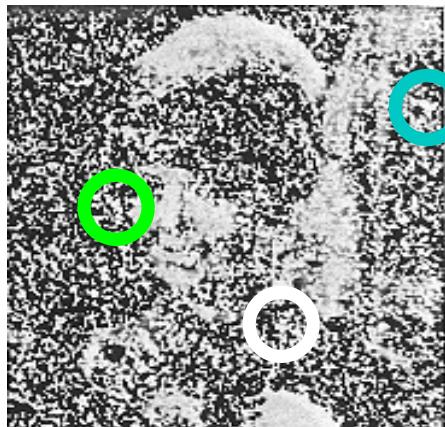
9. Traitements de base d'une image

Opérations locales

Filtre médian



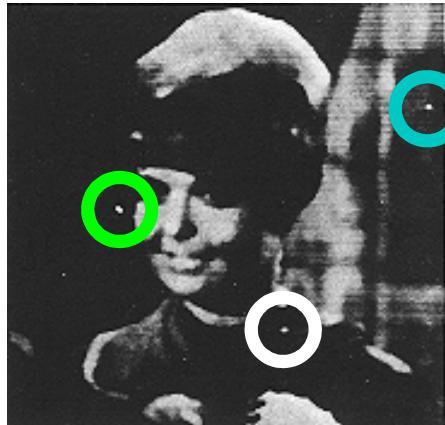
(a)



(b)

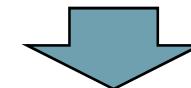


(c)



(d)

- (a) image originale
- (b) image bruitée
- (c) filtre de moyenne 5×5
- (d) filtre médian 5×5



si le bruit ajouté à l'image est supérieur à la dimension du filtre, celui-ci est inefficace !

9. Traitements de base d'une image

Opérations locales

Choix du filtre

- Bruit gaussien : [filtre moyenneur](#)



- Bruit poivre et sel : [filtre médian](#)



9. Traitements de base d'une image

Opérations locales

Filtres adaptatifs

- Simple

$$g(x, y) = \begin{cases} PasseBas[f(x, y)] & \text{si } |PasseBas[f(x, y)] - f(x, y)| < \text{seuil} \\ f(x, y) & \text{sinon} \end{cases}$$



Image bruitée



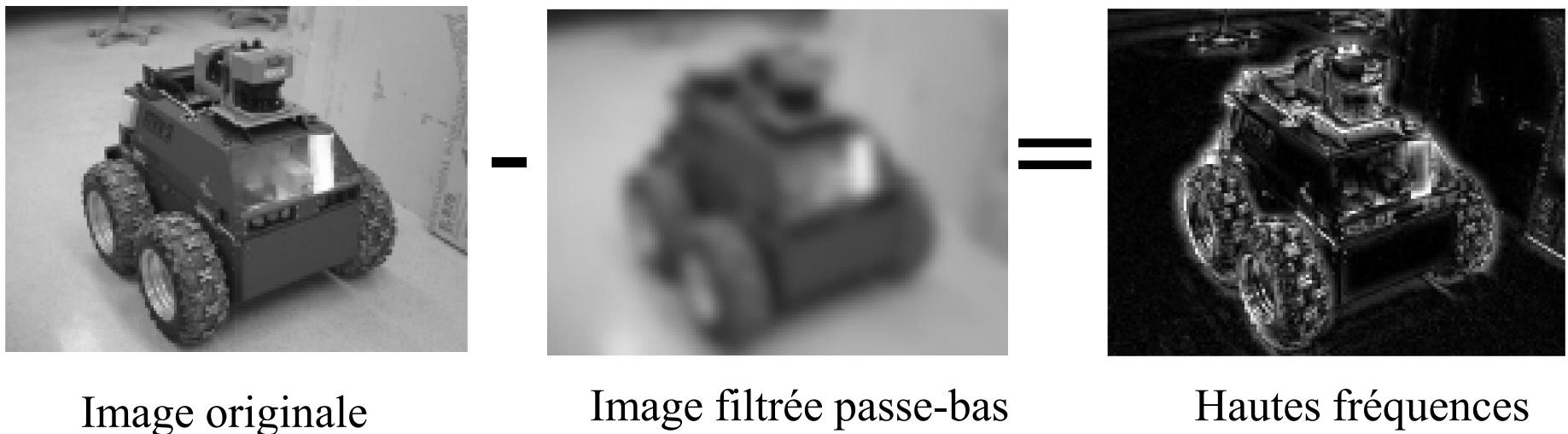
Filtre Gaussien r=9
Filtre adaptatif r=9

9. Traitements de base d'une image

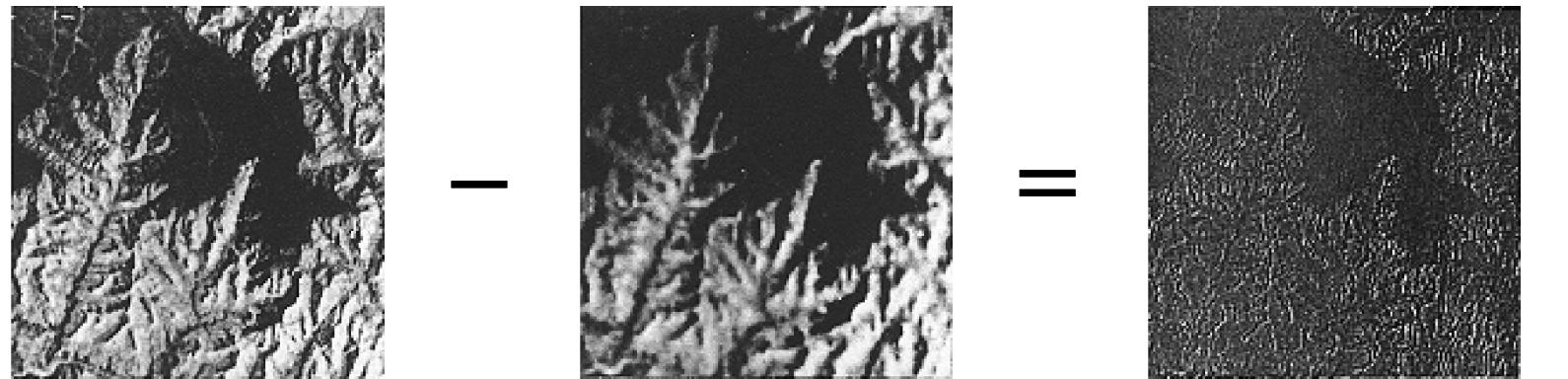
Opérations locales

Filtres passe-haut: Détection de contours

- Opérations mathématiques

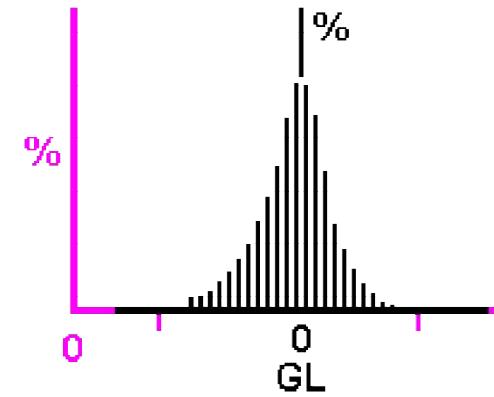
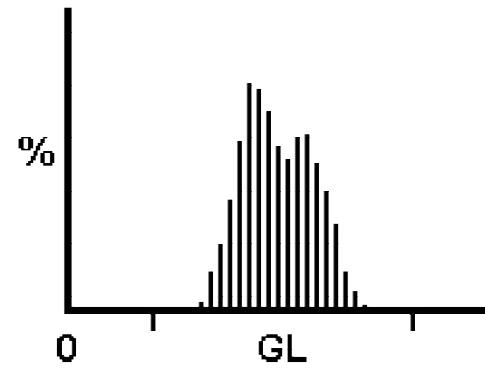
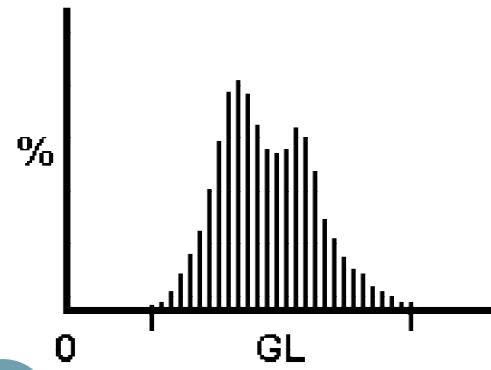


Caractéristiques des images obtenues par filtrage spatial



Lissage
(filtre passe-bas)

Rehaussement
(filtre passe-haut)



9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours

- **Au niveau des pixels**

- Ex: filtre 3X3 (à partir du filtre de moyenne)

$$\delta(x, y) - h(x, y) = \frac{1}{9} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \frac{1}{9} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

- Ex: filtre 5X5

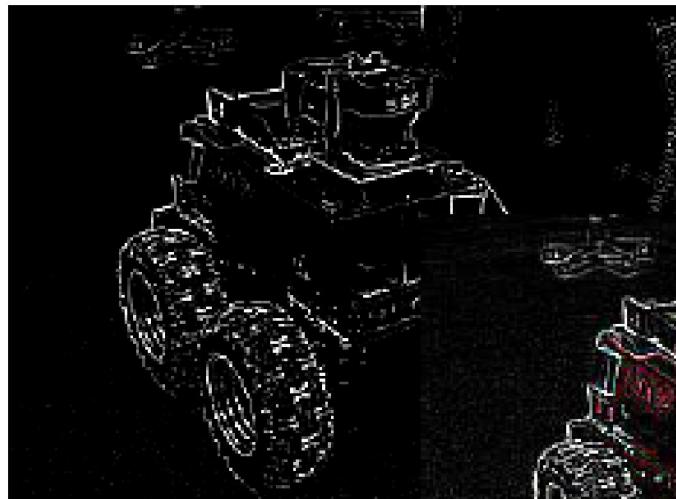
$$\frac{1}{25} \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

9. Traitements de base d'une image

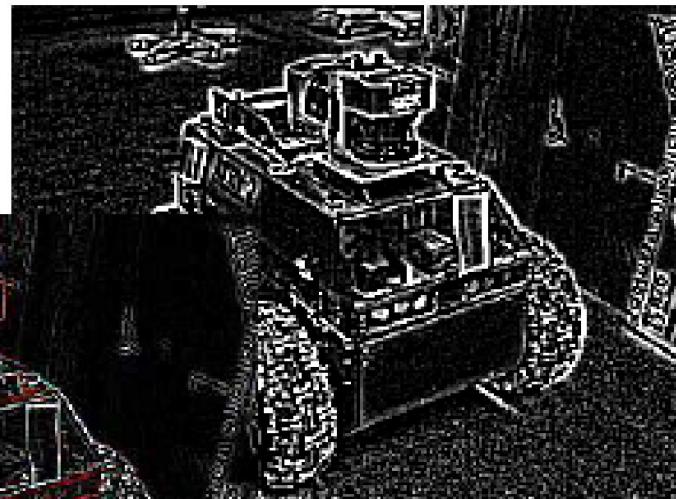
Opérations locales

Filtres passe-haut: Détection de contours

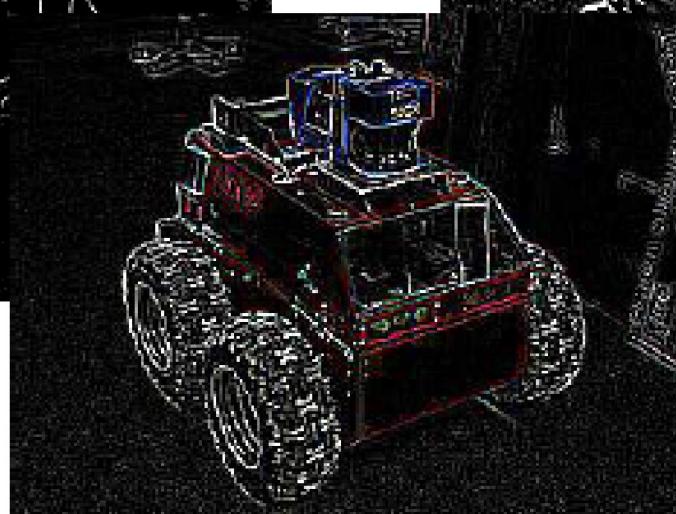
■ Résultats



Passe-haut 3x3



Passe-haut 5x5



Passe-haut 3x3

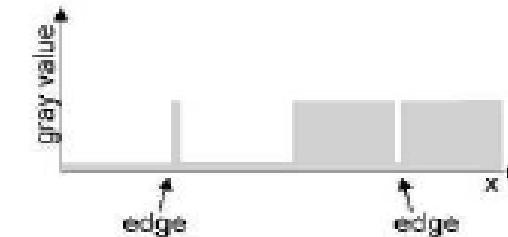
9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours

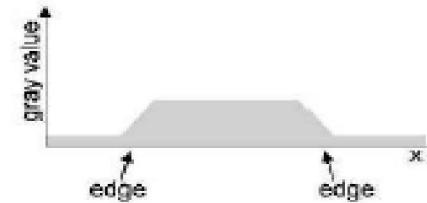
- **Masque de détection**

- Contour d'une ligne



$$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} \quad \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

— / | \ —

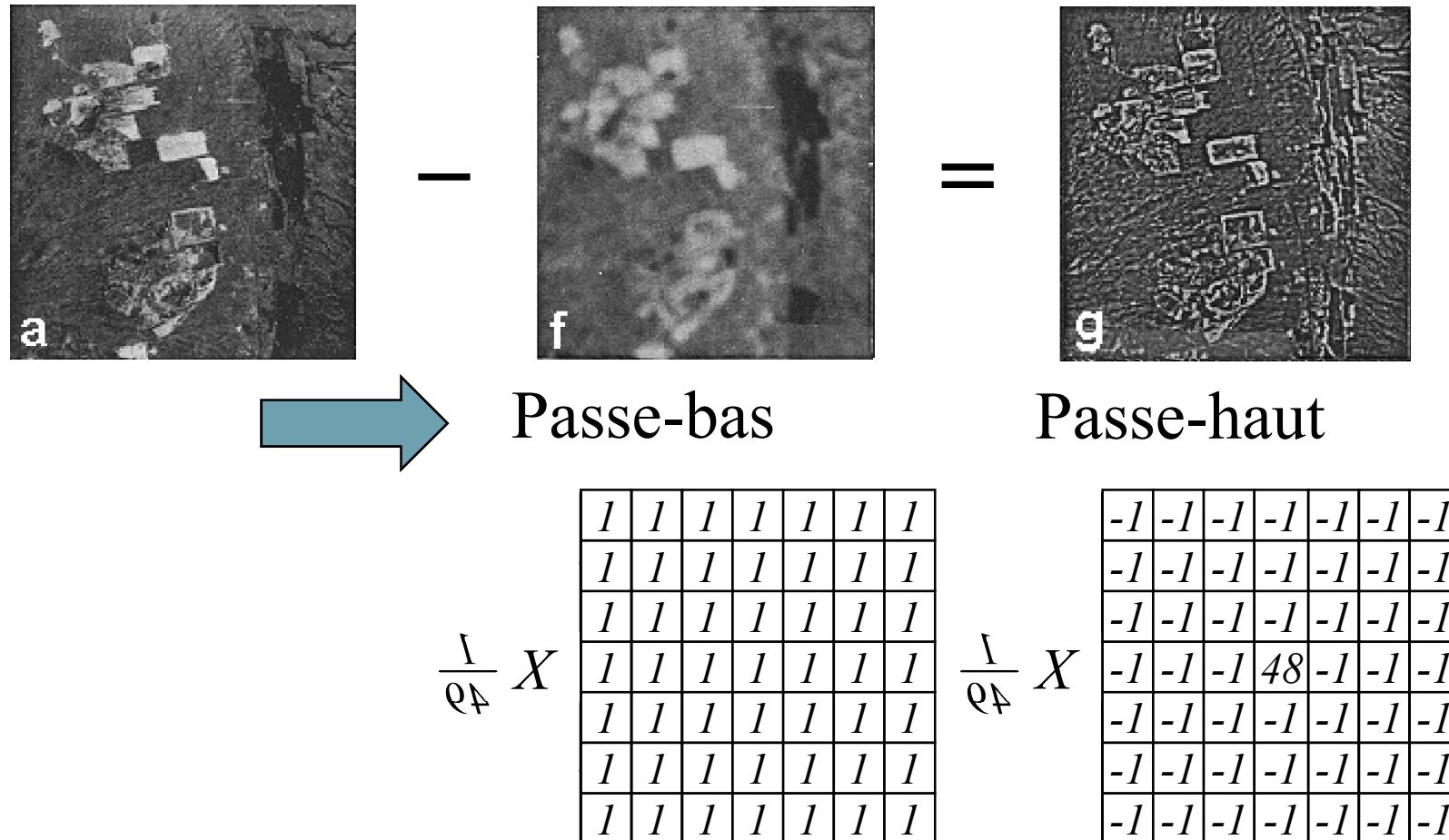


- Contour d'un objet

$$\begin{bmatrix} 1 & 1 \\ -1 & 1 \\ -1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$



Filtre passe-haut: Détection de contours



9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours

- **Détection de contours par gradient**

- Contour = gradient élevé
- Technique en 2 étapes :
 - Estimer le gradient (convolution)
 - Localiser les valeurs élevées (seuillage)

9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours

- **Détection de contours par gradient:** Convolution

- Multiplication locale par un masque
- $I'(x,y) = I(x,y) * M$

$$\rightarrow P'(x,y) = \sum_{i=0}^N P_i(x,y)M(i)$$

12	15	15	13	18	19
14	15	15	14	17	20
19	18	10	15	15	20
20	19	15	15	15	20
20	20	18	18	18	20

0	1	0
1	1	1
0	1	0

9. Traitements de base d'une image

Opérations locales

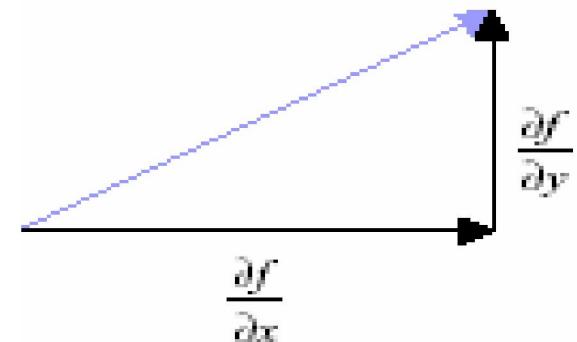
Filtres passe-haut: Détection de contours

- **Estimation du Gradient**

- Soit un pixel $f(x,y)$, le gradient y sera

$$\nabla f(x, y) = \begin{pmatrix} G_x \\ G_y \end{pmatrix} = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix}$$

$$G_x = f * M_x$$
$$G_y = f * M_y$$



- La norme du gradient
- La direction du gradient

$$|\nabla f(x, y)| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} = \left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right|$$
$$\theta = \arctan\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours

- Approximation du gradient (en x –arêtes verticales)

Différences avant et arrière

$$\frac{\partial f}{\partial x} = \lim_{\Delta_x \rightarrow 0} \frac{f(x + \Delta_x, y) - f(x, y)}{\Delta_x} = \lim_{\Delta_x \rightarrow 0} \frac{f(x, y) - f(x - \Delta_x, y)}{\Delta_x}$$

$$\Delta_x = 1 \longrightarrow \text{Masque de convolution } \rightarrow \begin{array}{|c|c|} \hline 1 & -1 \\ \hline \end{array} \text{ ou } \begin{array}{|c|c|} \hline -1 & 1 \\ \hline \end{array}$$

$$\frac{\partial f}{\partial x} = \lim_{\Delta_x \rightarrow 0} \frac{f(x + \Delta_x, y) - f(x - \Delta_x, y)}{2\Delta_x}$$

Différence centrée

$$\Delta_x = 1 \longrightarrow \text{Masque de convolution } \rightarrow \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline \end{array}$$

- Approximation du gradient (en y –arêtes horizontales)

Masque de convolution

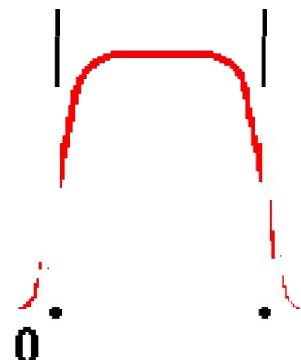
$$\rightarrow \begin{array}{|c|c|} \hline 1 & \\ \hline -1 & \\ \hline \end{array} \text{ ou } \begin{array}{|c|c|} \hline & -1 \\ \hline & 1 \\ \hline \end{array} \text{ ou } \begin{array}{|c|c|} \hline -1 & \\ \hline 0 & \\ \hline 1 & \\ \hline \end{array} \text{ ou } \begin{array}{|c|c|} \hline 1 & \\ \hline 0 & \\ \hline -1 & \\ \hline \end{array}$$

9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours

Filtres différentiels

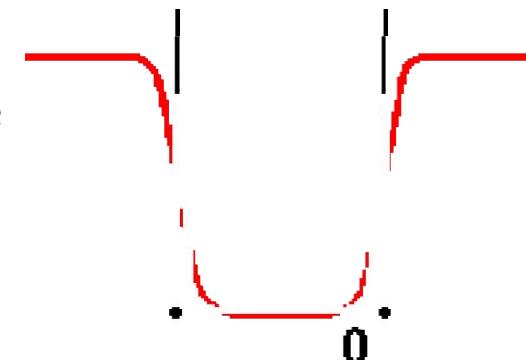
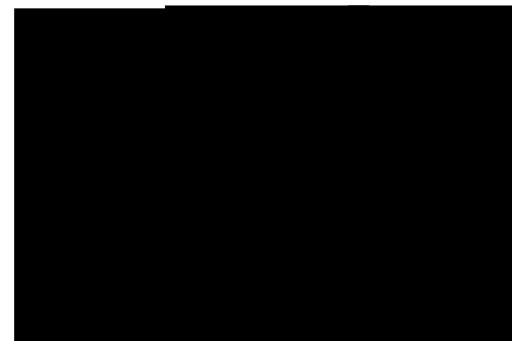


(a)

Image



profil d'une
ligne
horizontale
(dérivée
seconde)



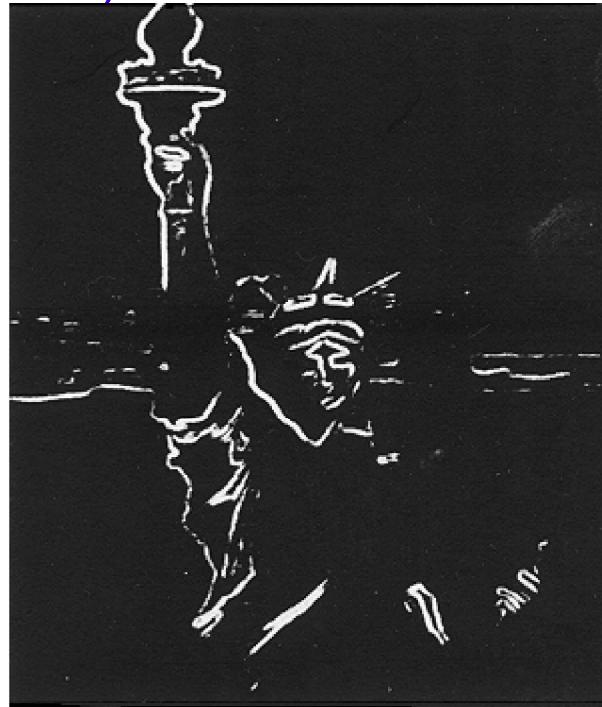
(b)

9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours

Dérivée première (gradient)



$$|G_x| + |G_y|$$

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

$$|G_x|$$

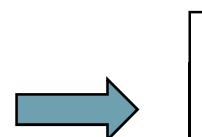
9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours

- **Sobel**

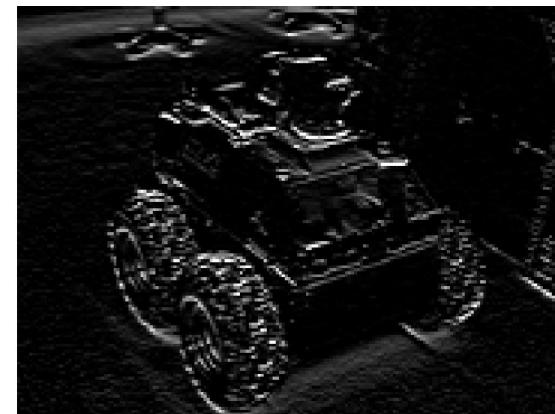
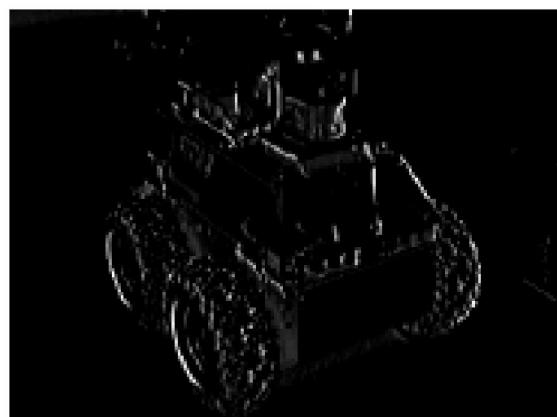
- Le passe-haut intensifie le bruit; on fait donc un passe-bas pour le diminuer.
- Composition d'un filtre gaussien et d'un gradient



$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

et

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$



9. Traitements de base d'une image

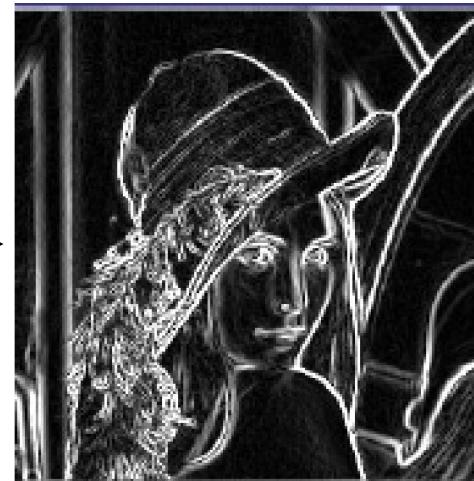
Opérations locales

Filtres passe-haut: Détection de contours

- **Sobel (=Gaussien+Gradient)**

Deux masques de taille 3x3 :

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \text{ et } \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$



9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours



Filtres de Prewitt et Sobel

(a)

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

(d) Sobel

operators

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

$$\frac{\partial f}{\partial y} = \begin{bmatrix} (z_7 + 2z_8 + z_9) \\ -(z_1 + 2z_2 + z_3) \end{bmatrix}$$

où z_x : valeur du niveau de gris

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

$$\frac{\partial f}{\partial x} = \begin{bmatrix} (z_3 + 2z_6 + z_9) \\ -(z_1 + 2z_4 + z_7) \end{bmatrix}$$

$$\nabla f \approx |G_x| + |G_y| \approx \left| \frac{\partial f}{\partial x} \right| + \left| \frac{\partial f}{\partial y} \right|$$

9. Traitements de base d'une image

Opérations locales

Filtre de Prewitt

Filtre Moyenneur + Gradient

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

et

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

1
1
1

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

et

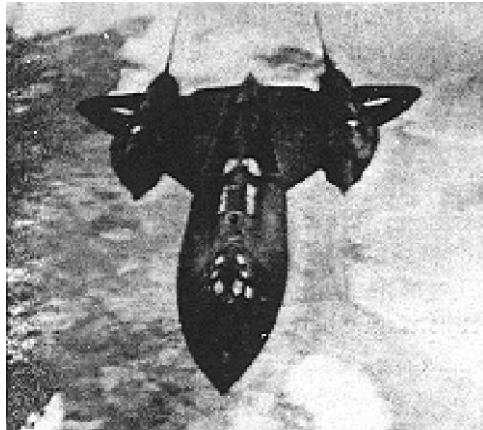
-1
0
1

1	1	1
---	---	---

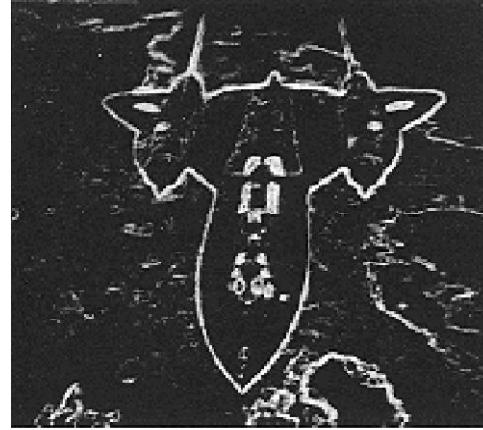
9. Traitements de base d'une image

Opérations locales

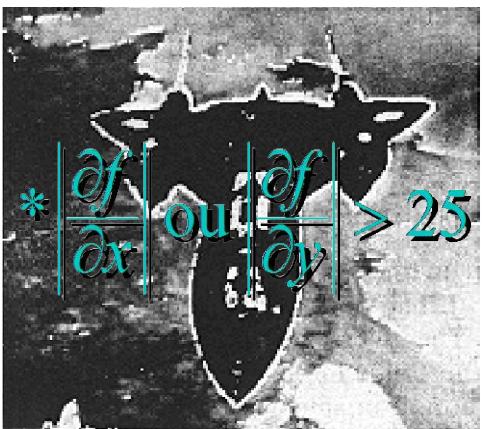
Filtres passe-haut: Détection de contours
Amélioration des arêtes et des contours



(a)



(b)



(c)



(d)

- (a) image originale
- (b) image obtenue à partir des valeurs de magnitude du gradient ['masques' de Prewitt]
- (c) image originale dont les pixels ayant 1 gradient $> 25^*$ ont été mis à 255
- (d) idem à c [sauf pour les pixels dont les 2 gradients $\leq 25^*$ qui ont été mis à 0 - image binaire]

9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours

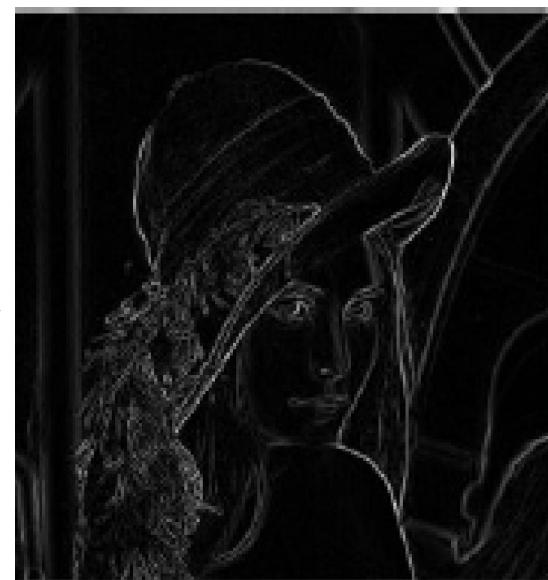
- **Roberts**

Deux masques de taille 2x2 :

$$\begin{array}{|c|c|} \hline -1 & 0 \\ \hline 0 & 1 \\ \hline \end{array}$$

et

$$\begin{array}{|c|c|} \hline 0 & -1 \\ \hline 1 & 0 \\ \hline \end{array}$$



9. Traitements de base d'une image

Opérations locales

Exemple du Gradient de Roberts

$$G(x, y) = |A(x, y) - A(x+1, y+1)| + |A(x, y+1) - A(x+1, y)|$$

Image d'un carré de 8x8 pixels

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

Image « Contours »

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

9. Traitements de base d'une image

Opérations locales

Généralisation des opérateurs de contours

- Opérateur de Roberts : 2 Masques de Convolution

$$Hr1 = \begin{vmatrix} 0 & -1 \\ 1 & 0 \end{vmatrix}$$

$$Hr2 = \begin{vmatrix} -1 & 0 \\ 0 & 1 \end{vmatrix}$$

$$g(x, y) = |f(x, y) - f(x+1, y+1)| + |f(x, y+1) - f(x+1, y)| = |Hr1 * f(x, y)| + |Hr2 * f(x, y)|$$

- Opérateur de Sobel et Prewitt : 2 Masques de Convolution

$$Hs1 = \begin{vmatrix} 1 & c & 1 \\ 0 & 0 & 0 \\ -1 & -c & -1 \end{vmatrix}$$

$$Hs2 = \begin{vmatrix} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{vmatrix}$$

Prewitt, $c = 1$
Sobel $c = 2$

$$g(x, y) = |Hs1 * f(x, y)| + |Hs2 * f(x, y)|$$

9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours

- **Laplacien**

- Dérivée seconde
 - Différence centrée

$$\nabla^2 = \left(\frac{\partial}{\partial x^2} + \frac{\partial}{\partial y^2} \right)$$

$$\frac{\partial^2 f(x, y)}{\partial x^2} = f'(x+1, y) - f'(x, y)$$

$$= [f(x+1, y) - f(x, y)] - [f(x, y) - f(x-1, y)]$$

$$= f(x+1, y) - 2f(x, y) + f(x-1, y)$$

- Noyau de convolution

$$\longrightarrow [1 \quad -2 \quad 1]$$

9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours

- **Laplacien** (suite)

- Hautes fréquences dans toutes les directions

$$\nabla^2 = \left(\frac{\partial}{\partial x^2} + \frac{\partial}{\partial y^2} \right) \longrightarrow [1 \quad -2 \quad 1] + \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Autres formes

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

Ou

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

9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours

- **Laplacien** (suite)

- Dérivée seconde du signal image
- Localisation des contours par le passage par 0
- Différents masques :

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



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

9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours

- **Kirsch**

- 8 masques, dans chaque direction (45°)

-3	-3	-3
5	0	-3
5	5	-3



9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours

- **Marr-Hildreth**

- Le Laplacien augmente encore plus le bruit
- On filtre l'image avec une gaussienne
- On prend le Laplacien de l'image filtrée
- On peut donc filtrer directement avec le laplacien d'une gaussienne!

$$g = (f * N) * L = f * (N * L)$$

9. Traitements de base d'une image

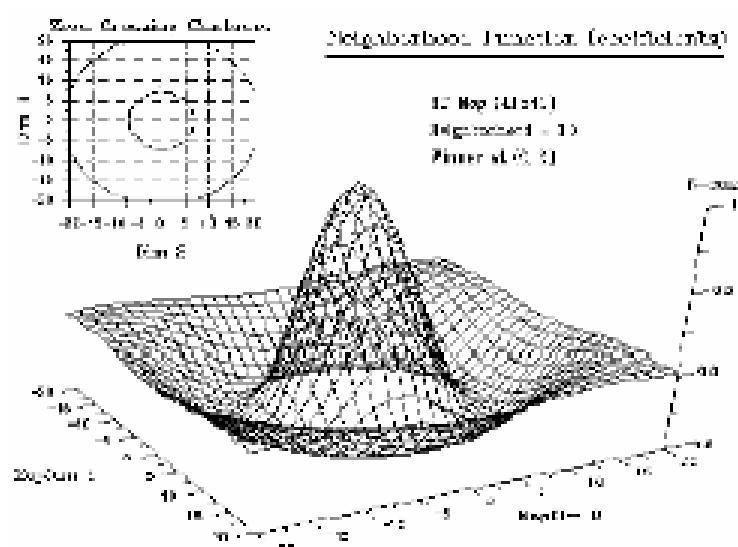
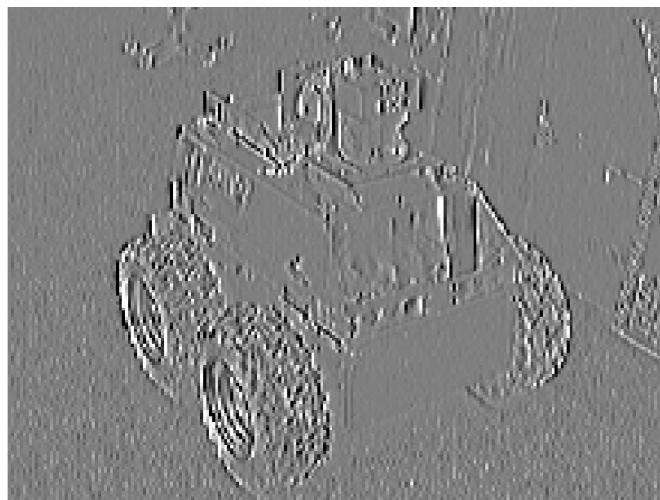
Opérations locales

Filtres passe-haut: Détection de contours

- Marr-Hildreth

0	-1	-2	-1	0
-1	0	2	0	-1
-2	2	8	2	-2
-1	0	2	0	-1
0	-1	-2	-1	0

-1	-3	-4	-3	-1
-3	0	6	0	-3
-4	6	20	6	-4
-3	0	6	0	-3
-1	-3	-4	-3	-1



9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours

- Marr-Hildreth versus Sobel
 - Plus rapide
 - Une seule convolution pour toutes les directions
 - Pas de somme de valeurs absolues ou de racine
 - Moins robuste (plus sensible au bruit)
 - Non directionnel
 - Donne des contours plus « **fermés** »

9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours

- Filtres passe-haut
 - Sobel
 - Marr-Hildreth
 - Canny
 - ...

9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours

- On utilise un filtre passe-haut et on compare la force du contour avec un seuil



Image originale



Sobel



Laplacien

9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours

- **Sobel**

- On applique le filtre horizontal, puis vertical
- On somme les valeurs absolues, ou racine carrée des carrées des valeurs
- Seuillage: si la valeur est assez élevée, la dérivée est forte et il y'a un contour

9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours

- **Marr-Hildreth**

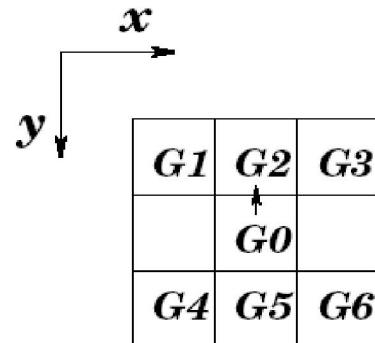
- On applique le filtre de Marr-Hildreth
- On détecte les **PPZ** (**P**assages **P**ar **Z**éro)
 - Seuillage particulier, sur un Kernel (souvent 3X3)
 - S'il y'a changement de signe entre la valeur centrale et un des huit voisins, alors il y'a PPZ
 - Changer de signe n'est plus suffisant, il faut passer de $-\varepsilon$ à $+\varepsilon$

9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours

Décision Contour



- G_0 contour si $|G_0| >$ seuil
- G_0 contour si $\begin{cases} G_2 < G_0 \\ G_5 < G_0 \end{cases}$ ou G_0 contour si $\begin{cases} G_2 < G_0 > G_5 \\ G_1 < G_0 > G_6 \\ G_3 < G_0 > G_4 \end{cases}$
...
...

9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours

Canny

- **Critères**
 - Taux d'erreur bas
 - Points bien localisés
 - Réponse unique pour un contour

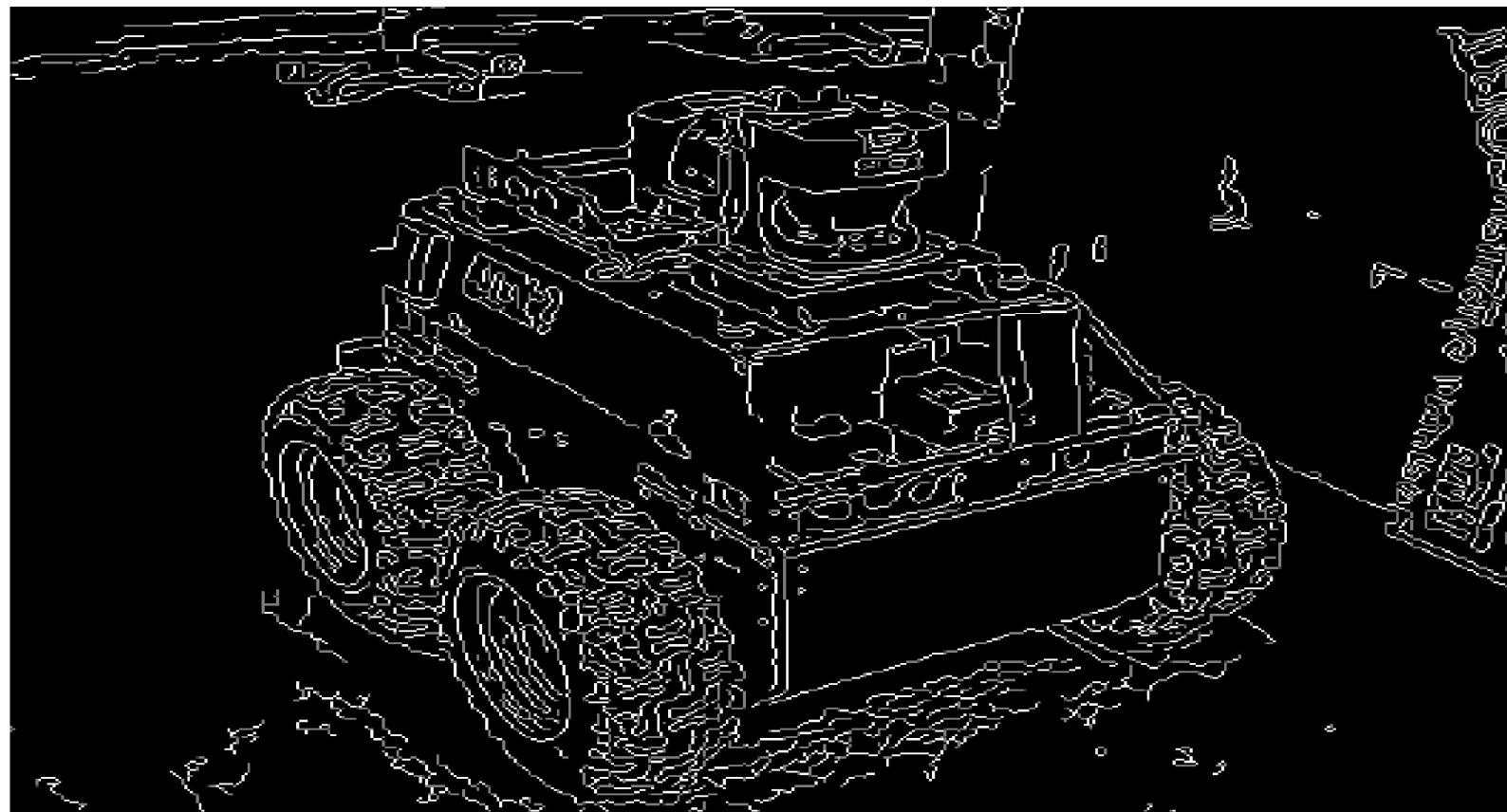


9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours

Canny



9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours

Seuillage

- Différents types de seuillage

- si $ndg > sb$ alors $ndg = 255$
- si $ndg < sh$ alors $ndg = 0$

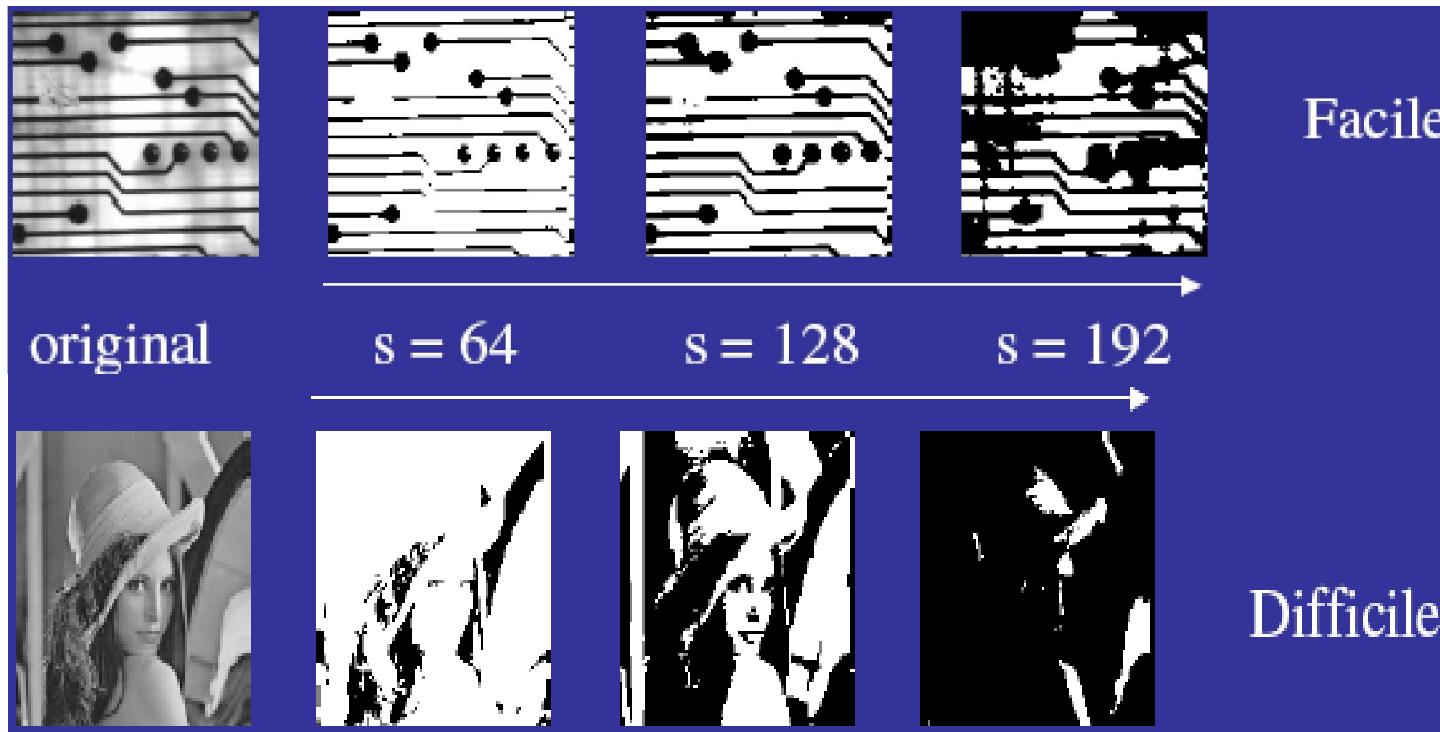


- si $ndg < s$ alors $ndg = 0$ sinon $ndg = 255$

Binarisation

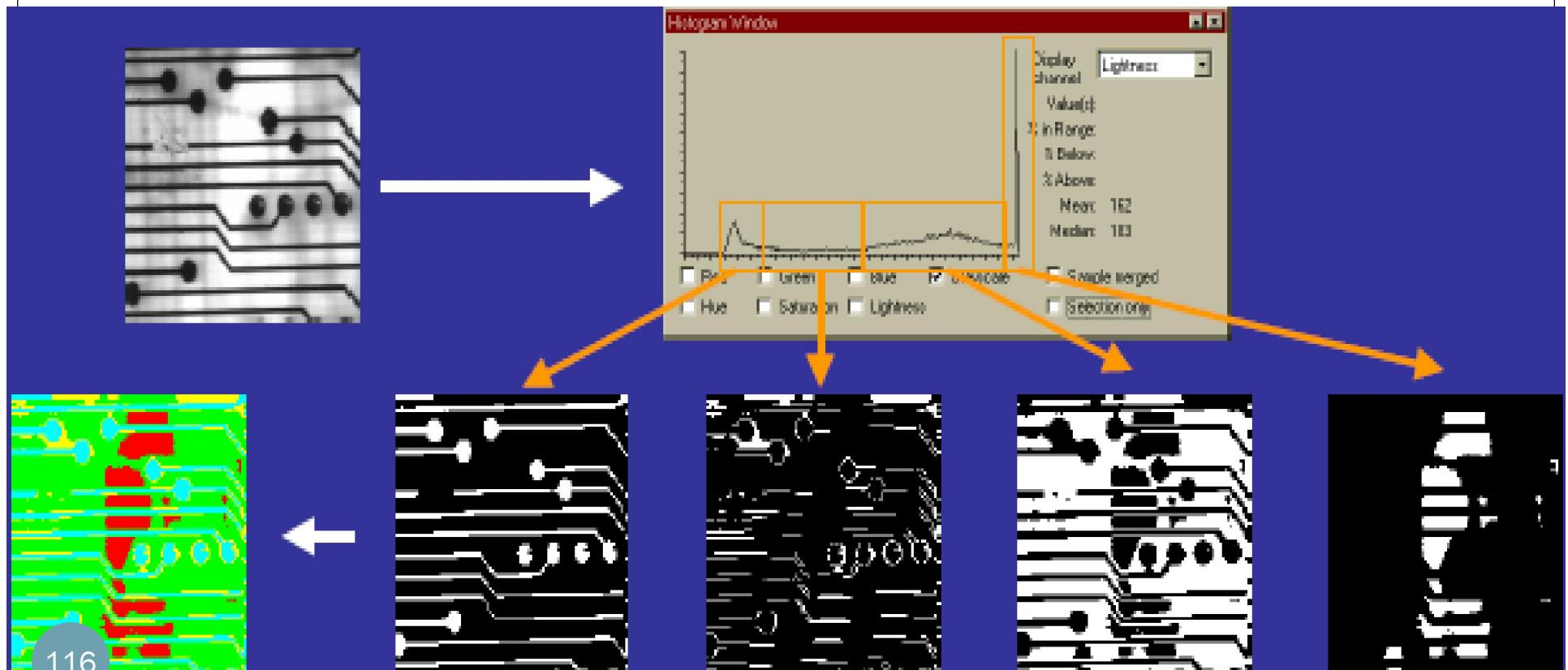
Binarisation

- Réduction du nombre de couleurs à 2 (N & B)



Binarisation

- Comment trouver le bon seuil ?
- Localisation des modes de l'histogramme



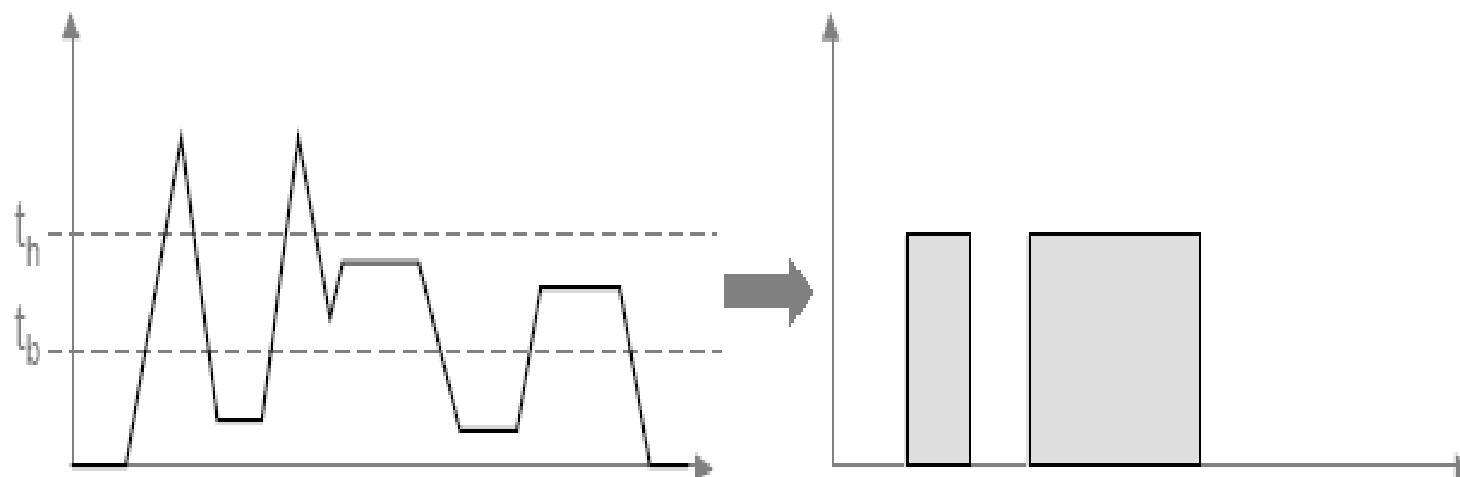
9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours
Canny

- **Seuillage par Hystérésis**

- On utilise 2 seuils T_1 et T_2 ($T_1 < T_2$)
 - Tous les pixels plus petits que T_1 sont classés « **non contours** »
 - Tous les pixels entre T_1 et T_2 sont classés « **peut-être** »
 - Tous les pixels plus grande que T_2 sont classés « **contours** »
 - Tous les pixels classés « **peut-être** » adjacents à des pixels contours sont classés « **contours** »
- On commence à suivre un contour avec un pixel $> T_2$ et on termine par un pixel $< T_1$



9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours

- **Images couleur**

- Utilisation de R,G,B puis combinaison
- Autres méthodes plus robustes



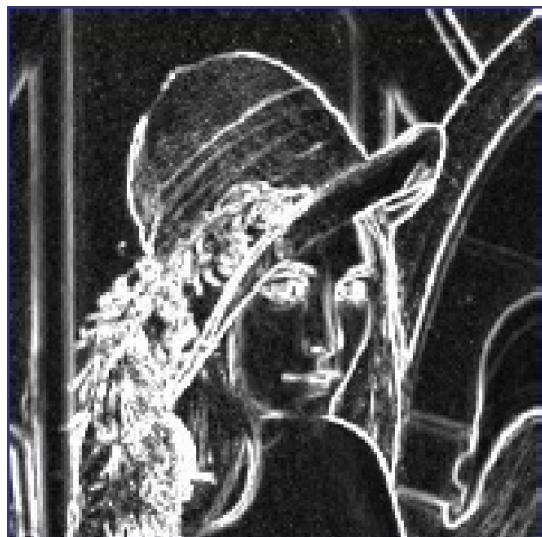
9. Traitements de base d'une image

Opérations locales

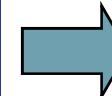
Filtres passe-haut: Détection de contours

Fermeture de contours

- **Problème des contours ouverts**



Seuillage



fermeture

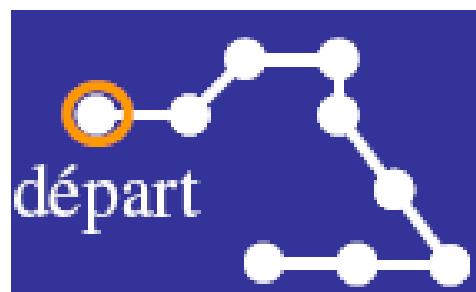
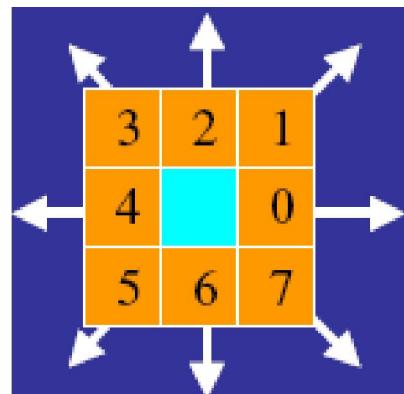
9. Traitements de base d'une image

Opérations locales

Filtres passe-haut: Détection de contours

Fermeture de contours:

- **Codage de Freeman**
 - Représentation discrète d'un contour



$X_0, Y_0 \quad \textbf{01067744}$