Introduction au traitement d'images

Transformation d'images

Nicholas Journet

7 février 2011

Plan

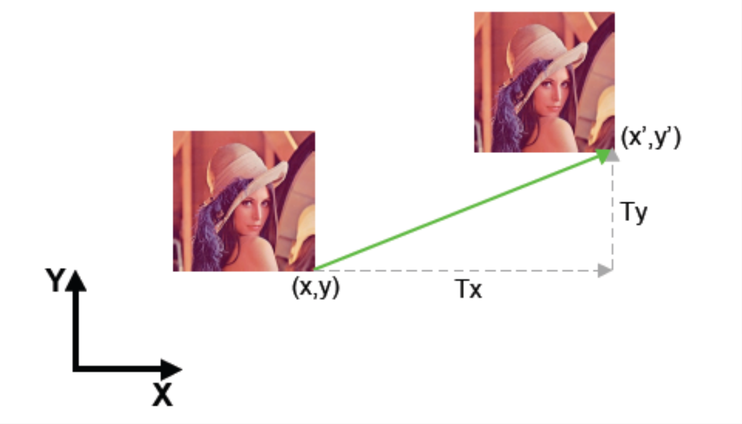
- Transformations 2D (translation, rotation, homothétie)
- Morpho-math
- Convolution

Bibliographie

- Cours de traitement d'images Elise Arnaud - Edmond Boyer Université Joseph Fourier
- Cours de traitement d'images Alain Boucher
- Cours de traitement d'images T Guyer Université de Chambéry
- Cours de traitement d'images Caroline ROUGIER université de Montréal
- Analyse d'images : filtrage et segmentation (Edition Broché) - Cocquerez
- Cours de traitement d'images V Eglin INSA de Lyon
- Cours de traitement d'images JC Burie Université de La Rochelle

Translation

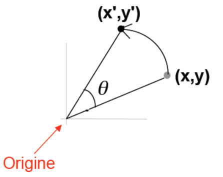
$x' = x + T_x$ et $y' = y + T_y$
Avec T_x et T_y sont les déplacements en x et en y de la translation



Rotation autour de l'origine

$$x' = x\cos(\theta) - y\sin(\theta)$$
$$y' = y\cos(\theta) + x\sin(\theta)$$

avec θ le sens de l'angle de rotation dans le sens positif.



Démonstration :
Par définition $x = r\cos(a)$ et $x = r\sin(a)$
Après rotation d'angle θ :
$x' = r\cos(a+\theta)$ et $y' = r\sin(a+\theta)$

$$x' = r\cos(a)\cos(\theta) - r\sin(\theta)\sin(a)$$
$$y' = r\cos(a)\sin(\theta) + r\cos(\theta)\sin(a)$$

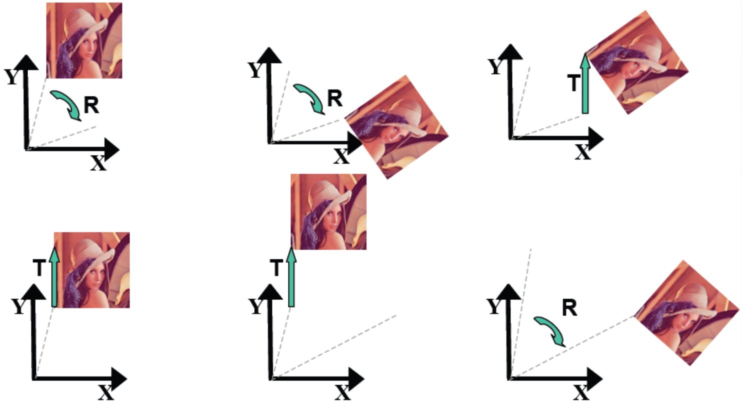
et comme : $x = r\cos(a)$ et $x = r\sin(a)$

$$x' = x\cos(\theta) - y\sin(\theta)$$
$$y' = y\cos(\theta) + x\sin(\theta)$$

Rotation - remarque

Les transformations ne sont pas commutatives

rotation o translation $\neq$ translation o rotation



Rotation - remarque

Par contre, on peut inverser 2 rotations et 2 translations :

- Rotation1 o Rotation2 = Rotation2 o Rotation1
- Translation1 o Translation2 = Translation2 o Translation1

Homothétie

Changement d'échelle par rapport à l'origine.

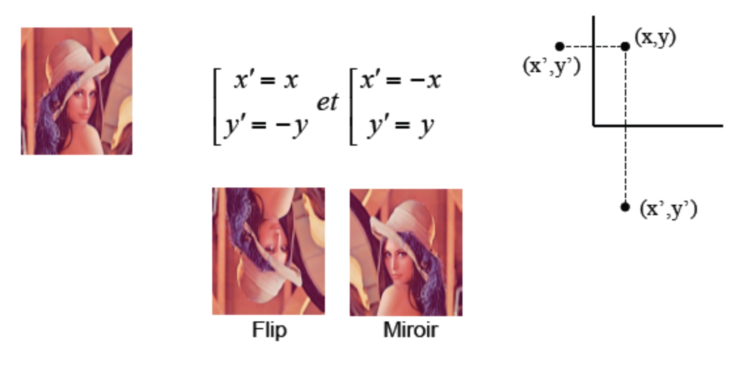
$$x' = S_x.x$$

$$y' = S_y.y$$

Avec S_x et S_y sont les facteurs d'agrandissement ou de réduction. cf. Cours précédent.

Réflexions

Réflexion par rapport aux axes (flip, miroir)



Représentation matricielle

Rotation : $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x' = x\cos\theta - y\sin\theta \\ y' = y\cos\theta + x\sin\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$

Mise à l'échelle : $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \rightarrow x' = S_x.x$ et $y' = S_y.y$

Translation : $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix} \rightarrow$ Représentation matricielle

impossible. $\rightarrow$ On utilise les coordonnées homogènes

Représentation Matricielle

Coordonnées homogènes (on ajoute une dimension) : $(x, y) \rightarrow (x, y, 1)$ Exemple de la translation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$x' = x + T_x$$

Représentation Matricielle

Changement d'échelle

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Rotation :

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Composition de transformations

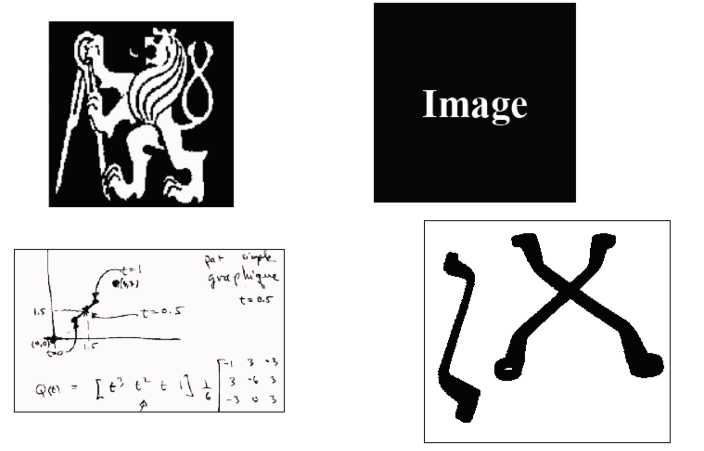
$$T_1(T_{x_1}, T_{y_1}).T_2(T_{x_2}, T_{y_2}) = T_3(T_{x_1} + T_{x_2}, T_{y_1} + T_{y_2})$$

$$S_1(S_{x_1}, S_{y_1}).S_2(S_{x_2}, S_{y_2}) = S_3(S_{x_1}.S_{x_2}, S_{y_1}.S_{y_2})$$

$$R_1(\theta_1).R_2(\theta_2) = R_3(\theta_1 + \theta_2)$$

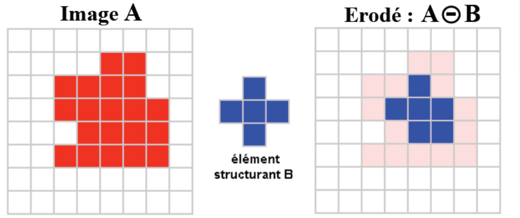
$$M_{st} = S(S_x, S_y).T(T_x, T_y) = \begin{bmatrix} S_x & 0 & S_x.T_x \\ 0 & S_y & S_y.T_y \\ 0 & 0 & 1 \end{bmatrix}$$

Rappel images binaires

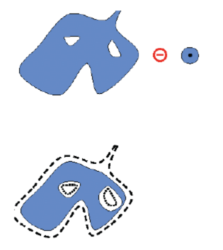


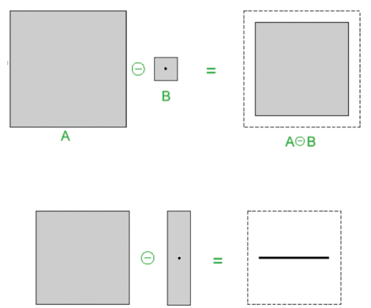
Erosion

Soit B un élément structurant et B_x l'élément positionne sur le pixel x.
Algorithme :
On positionne l'origine de B en chaque pixel x de l'objet A.
Si tous les pixels de B font partie de l'objet A, alors l'origine de B appartient à l'érodé



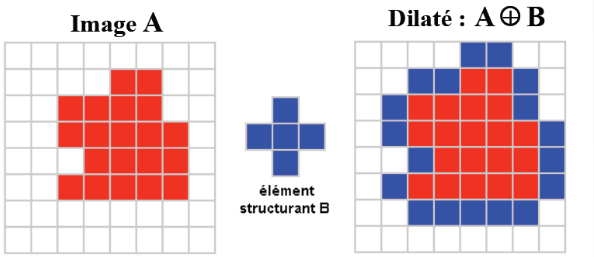
Erosion exemple



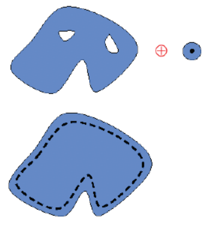


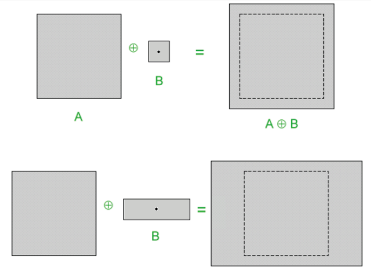
Dilatation

Soit B un élément structurant et B_x l'élément positionné sur le pixel x.
Algorithme :
On positionne l'origine de B en chaque pixel x de l'objet A. Si l'intersection de B et de A est non vide, alors l'origine de B appartient à l'image dilatée



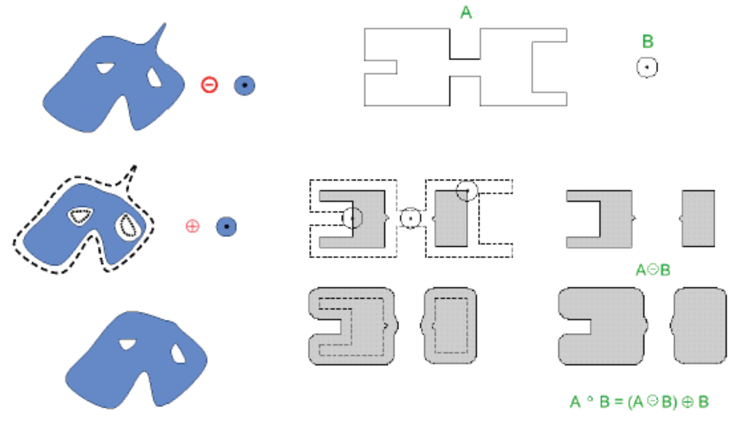
Dilatation exemple





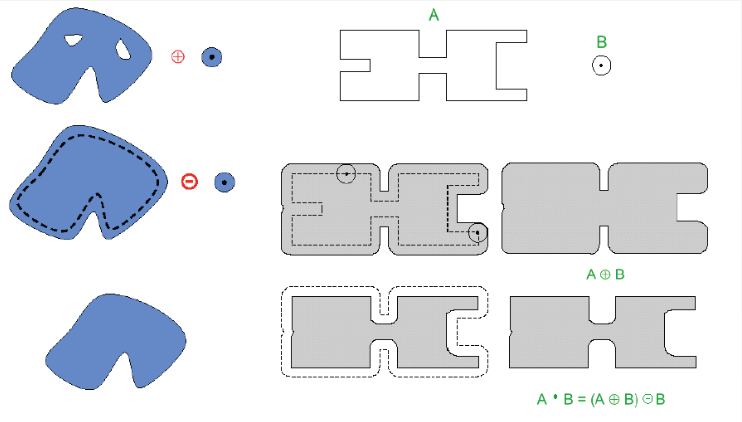
Ouverture

$$A \circ B = (A \ominus B) \oplus B$$

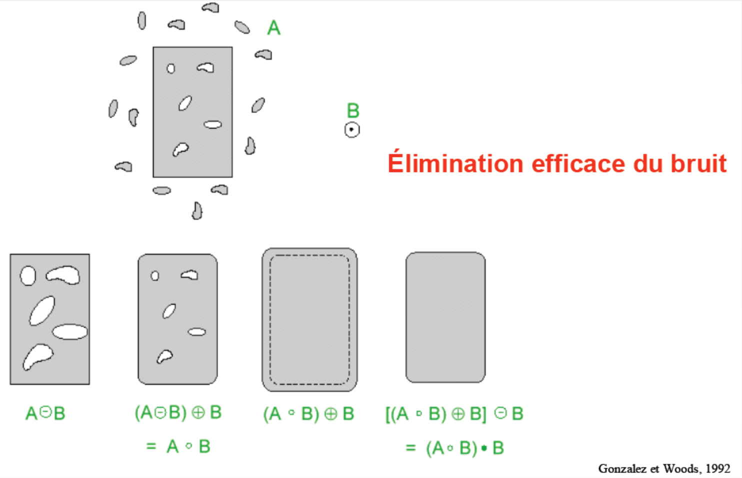


Fermeture

$$A \bullet B = (A \oplus B) \ominus B$$

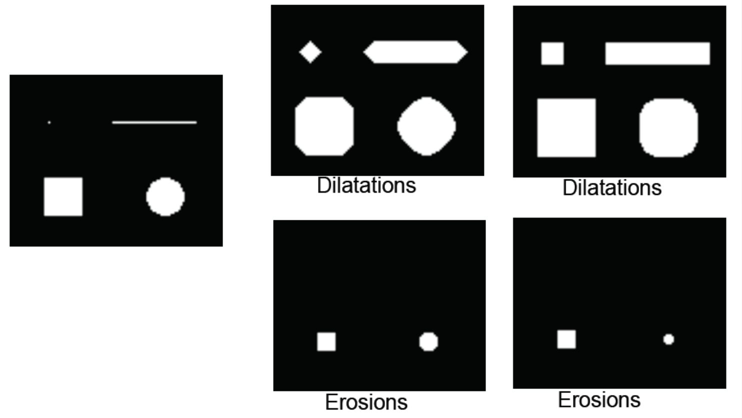


Elimination du bruit

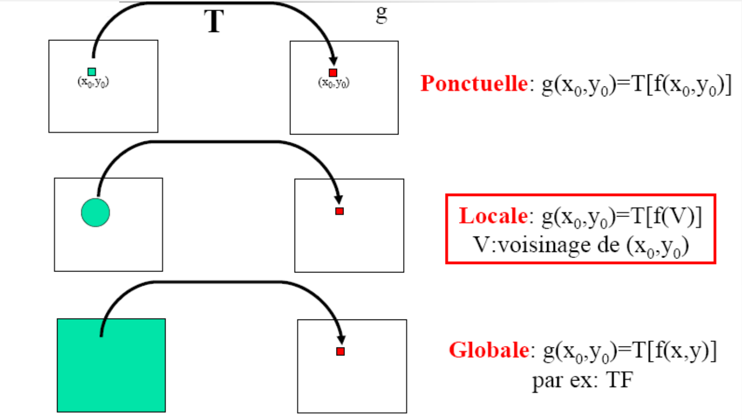


Element structurant

Importance de l'élément structurant :



Modification des valeurs d'une image



Convolution exemple

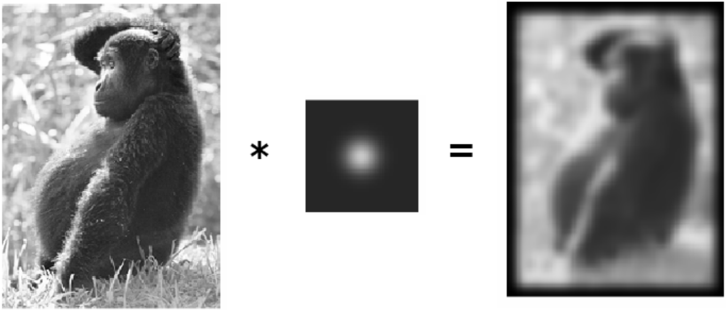


Image d'origine

Filtre de convolution (masque)

Image convoluée (résultat)

Convolution exemple

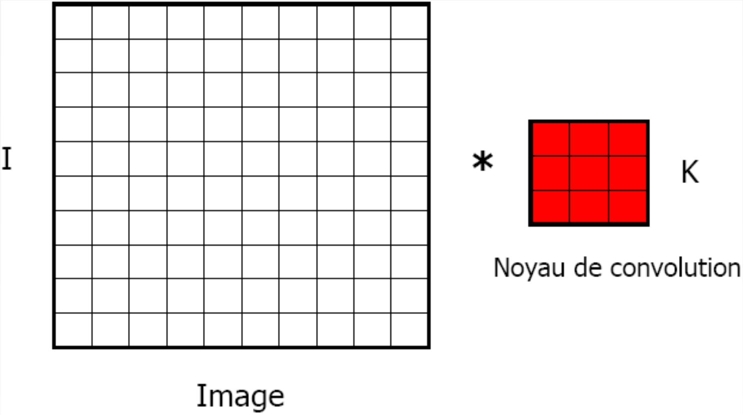
- En pratique (cas discret), la convolution numérique d'une image se fera par une somme de produits.
- Un filtre de convolution est une matrice généralement (mais pas toujours) de taille impaire et symétrique (mais pas toujours).

Convolution d'une image par un filtre 2D :

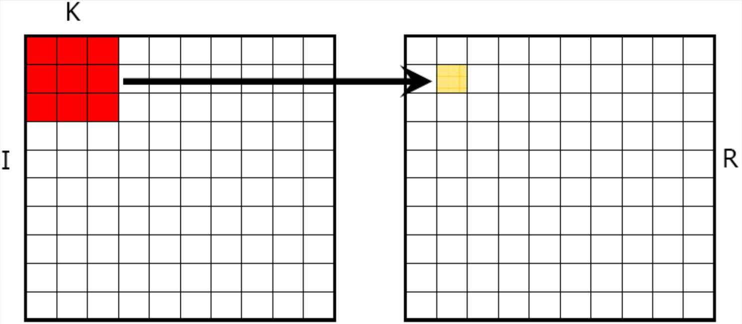
$$I'(i,j) = I(i,j).filtre(i,j)$$

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

Convolution détail

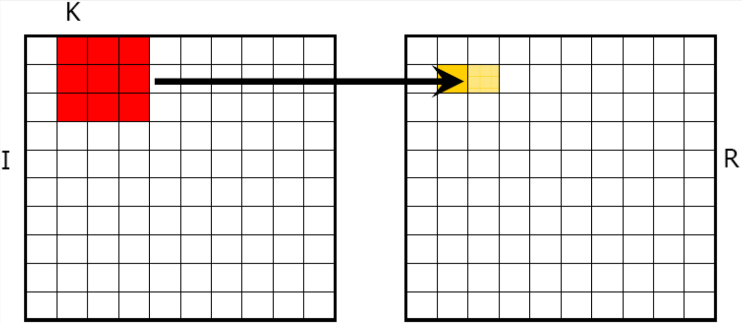


Convolution détail



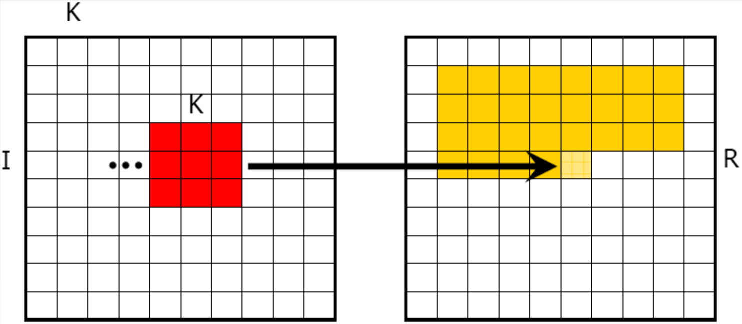
$$
\begin{aligned}
R(1,1) = {} & I(0,0)\,K(0,0) + I(1,0)\,K(1,0) + I(2,0)\,K(2,0) \\
& + I(0,1)\,K(0,1) + I(1,1)\,K(1,1) + I(2,1)\,K(2,1) \\
& + I(0,2)\,K(0,2) + I(1,2)\,K(1,2) + I(2,2)\,K(2,2)
\end{aligned}
$$

Convolution détail



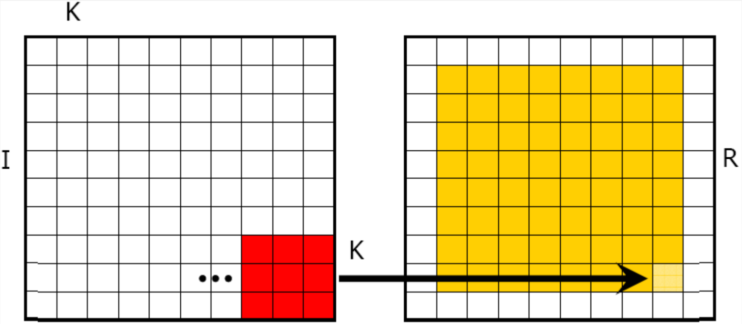
$$
\begin{aligned}
R(2,1) = {} & I(1,0)\,K(0,0) + I(2,0)\,K(1,0) + I(3,0)\,K(2,0) \\
& + I(1,1)\,K(0,1) + I(2,1)\,K(1,1) + I(3,1)\,K(2,1) \\
& + I(1,2)\,K(0,2) + I(2,2)\,K(1,2) + I(3,2)\,K(2,2)
\end{aligned}
$$

Convolution détail



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

Convolution détail

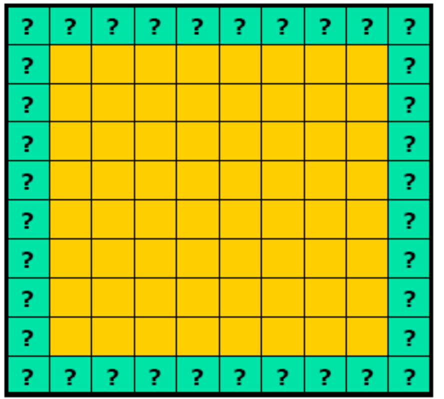


R(N-2,M-2) = I(N-3,M-3) K(0,0) + I(N-2,M-3) K(0,1) + I(N-1,M-3) K(0,3)
+ I(N-3,M-2) K(1,0) + I(N-2,M-2) K(1,1) + I(N-1,M-2) K(1,2)
+ I(N-3,M-1) K(2,0) + I(N-2,M-1)K(2,1) + I(N-1,M-1) K(2,2)

Que fait on des bords ?

Problème : Que faire avec les bords de l'image ?

- Mettre à zéro
- Convolution partielle
- Sur une portion du noyau
- Miroir de l'image
- f(-x,y) = f(x,y)
- pas de solution miracle



Masque de convolution

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

Exemples de filtres

- Filtres passe-bas : Atténue le bruit et les détails (impression de lissage)
- Filtres passe-haut : Accentue les détails et les contours (impression d'accentuation)

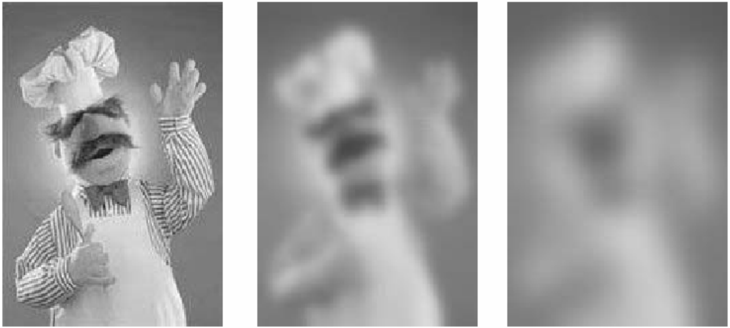
Le filtre moyenneur

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

$$\begin{array}{|c|c|c|}\hline 1/9 & 1/9 & 1/9 \\ \hline 1/9 & 1/9 & 1/9 \\ \hline 1/9 & 1/9 & 1/9 \\ \hline \end{array} \quad \text{ou} \quad 1/9 \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad\qquad \begin{array}{|c|c|c|c|c|}\hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

3x3   *5x5*

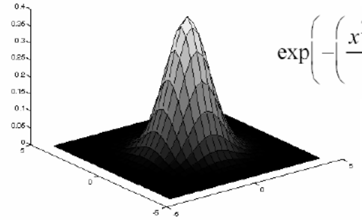
Filtre moyenneur : exemple



Original | Moyenne 5x5 | Moyenne 11x11

Le filtre Gaussien

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



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

Fonction gaussienne en 3D

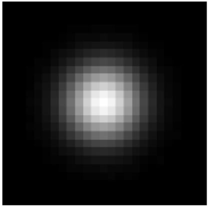


Image d'une gaussienne

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

Filtre Gaussien : exemple

Original

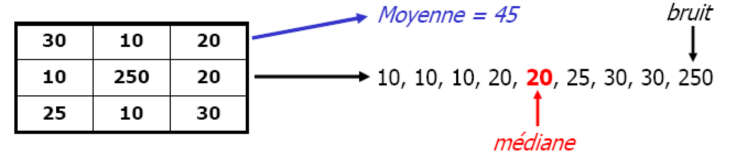
Gauss 5x5

Gauss 11x11

Filtre médian (non linéaire)

Pour nettoyer le bruit dans une image, il existe mieux que le filtre moyenneur ou le filtre gaussien

- Il s'agit du filtre médian
- C'est un filtre non-linéaire, qui ne peut pas s'implémenter comme un produit de convolution
- On remplace la valeur d'un pixel par la valeur médiane dans son voisinage *NxN*



Filtre median : exemple

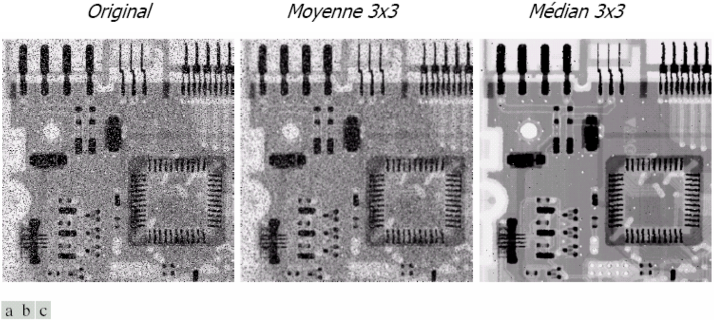


FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3 × 3 averaging mask. (c) Noise reduction with a 3 × 3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Nettoyage du bruit dans une image