Traitement d'images

Transformations géométriques / Filtrage

Plan

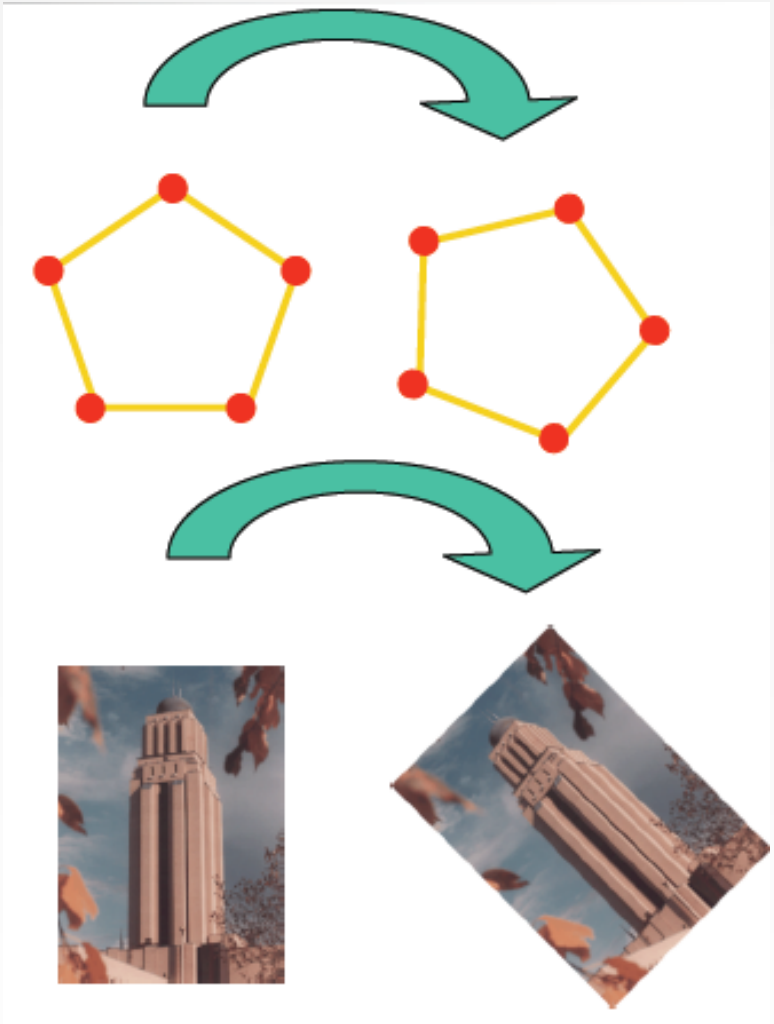
- Transformations géométriques
- Filtrage

Bibliographie

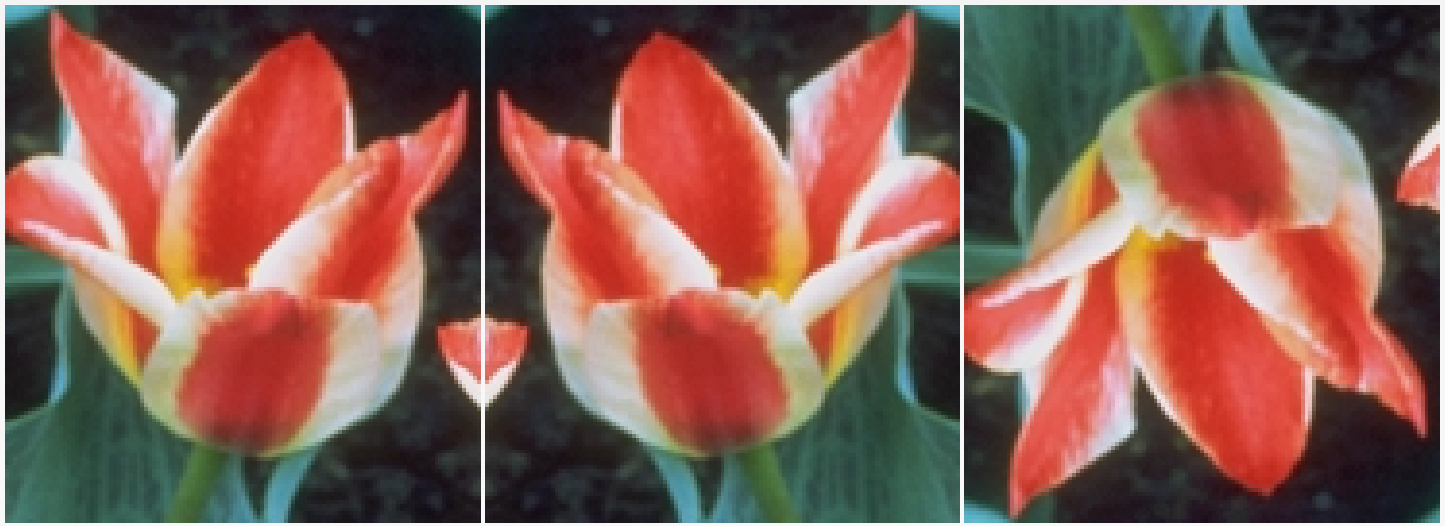
- Cours de traitement d'images Elise Arnaud - Edmond Boyer Université Joseph Fourier
- Cours de traitement d'images Alain Boucher
- Cours de traitement d'images T Guyer Université de Chambéry
- Cours de traitement d'images Caroline ROUGIER université de Montréal
- Analyse d'images : filtrage et segmentation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

Transformations géométriques - vectoriel/bitmap

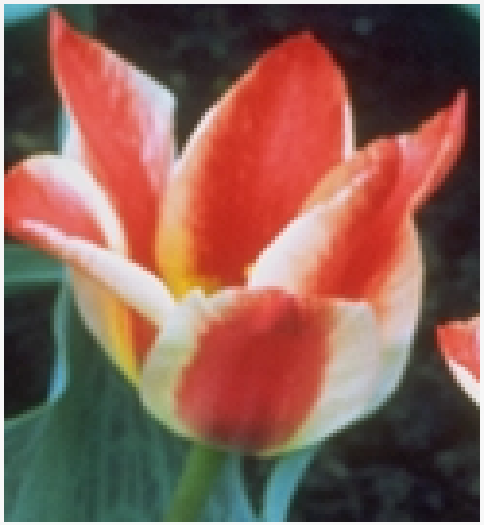
- Objet vectoriel : on transforme les sommets (ou points de contrôle) et on retrace
- Objet bitmap : calcul pour chaque pixel



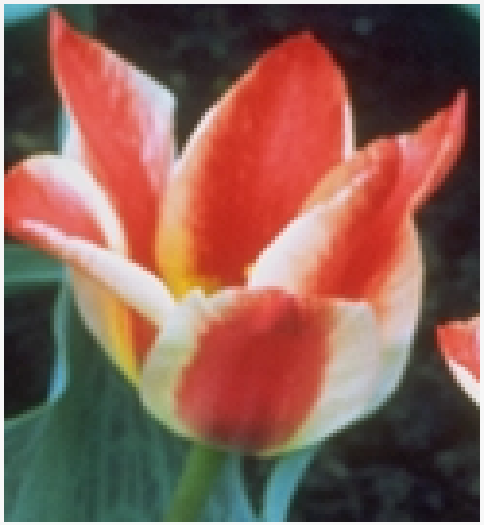
Flip



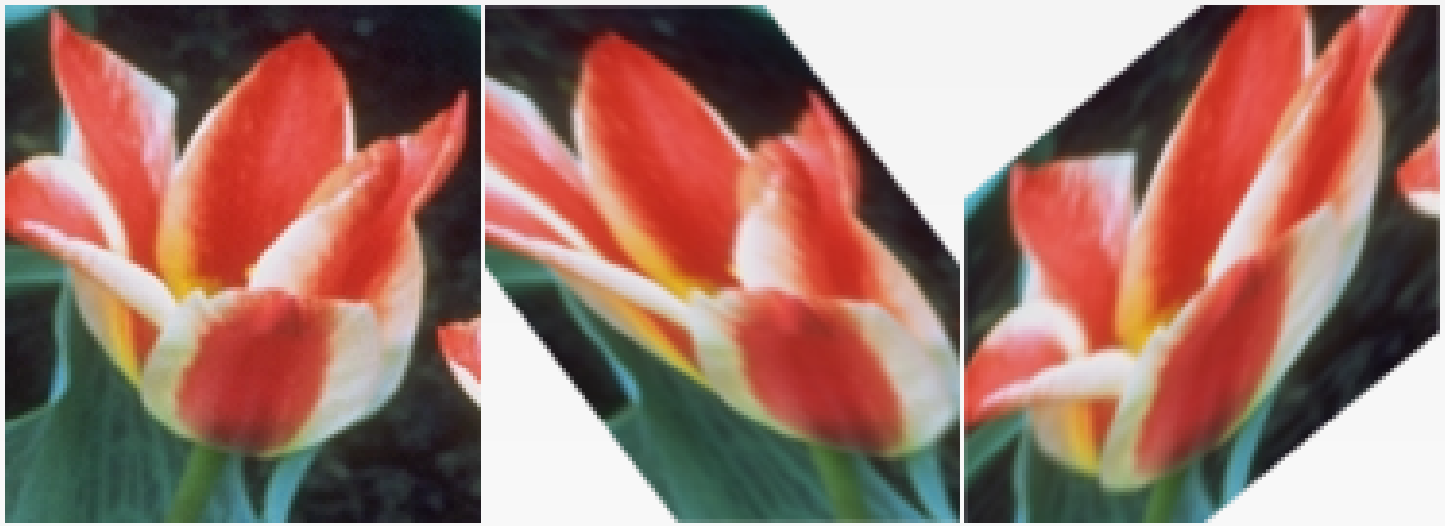
Changement d'échelle



Rotation



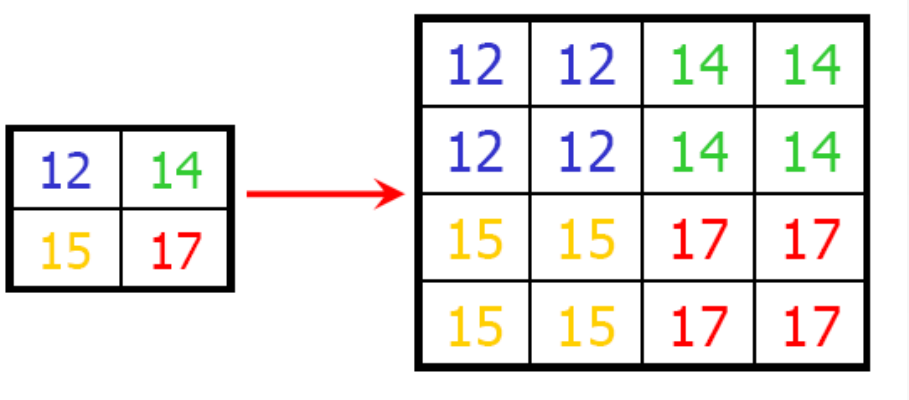
Cisaillement



Changement d'échelle

Première idée : agrandissement d'image par copie des pixels

Exemple : multiplication par 2 de la taille de l'image



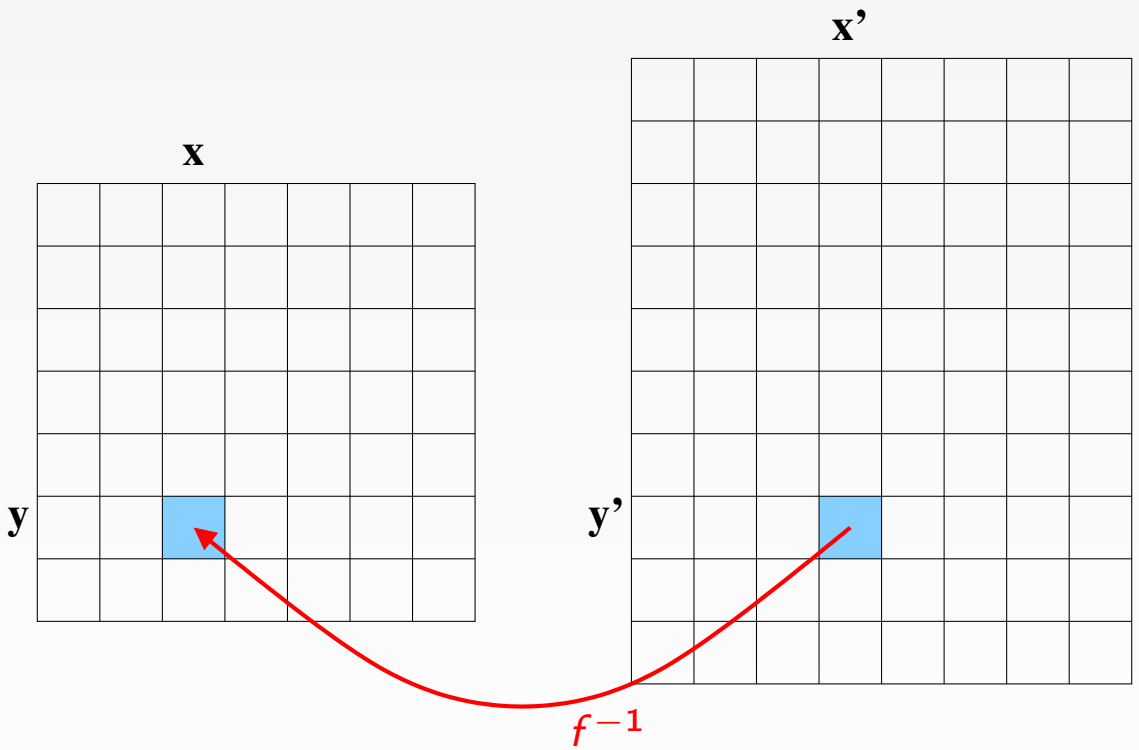
Limite de cette approche ?

Principe général

Recherche du pixel antécédent : pour chaque pixel de l'image résultat, on cherche le pixel correspondant dans l'image initiale.

Transformation géométrique $(x, y) \rightarrow (x', y') = f(x, y)$

Transformation inverse $(x', y') \rightarrow (x, y) = f^{-1}(x', y')$



Changement d'échelle : calcul

Le changement d'échelle est une homothétie de centre l'origine. On note S_x et S_y les facteurs d'échelle suivant chaque axe (agrandissement ou réduction).

$$x' = S_x.x \qquad x = \frac{1}{S_x}.x'$$

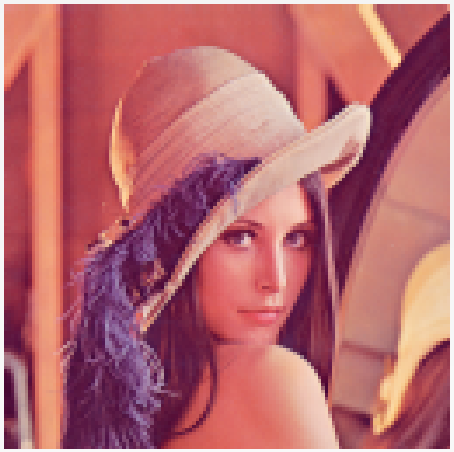
$$y' = S_y.y \qquad y = \frac{1}{S_y}.y'$$

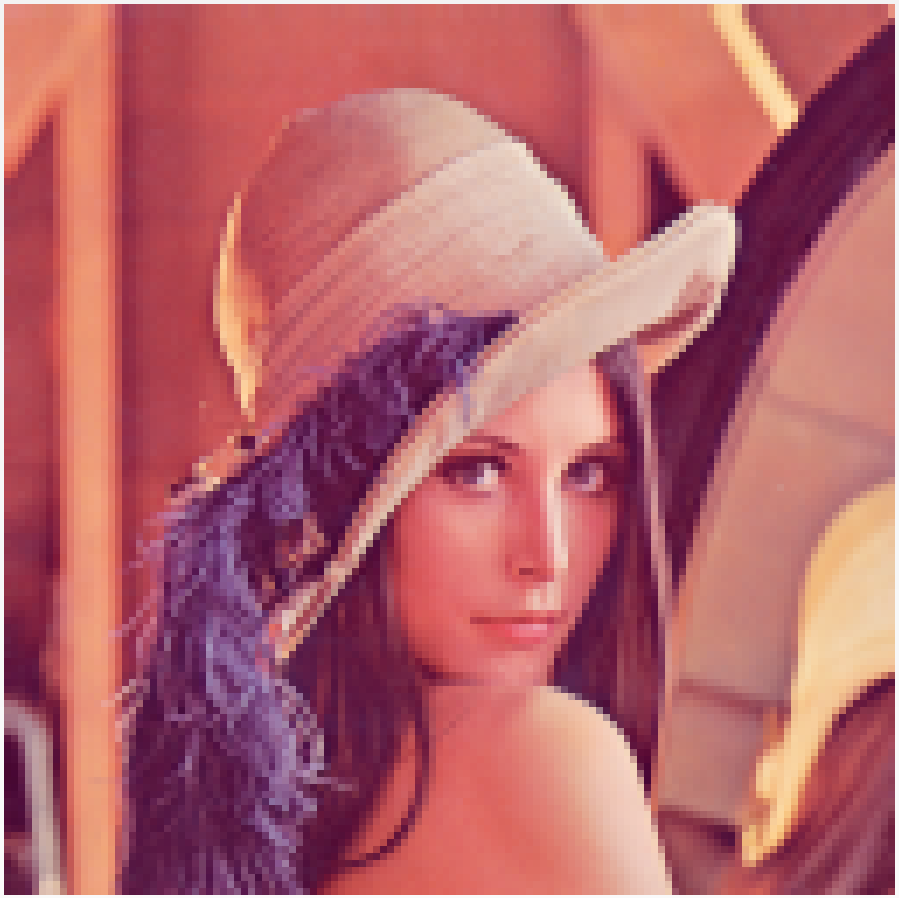
Algorithme

```
W' = Sx*W
H' = Sy*H
créer l'image résultat R de taille W', H'
for(y=0; y<H'; y++)
   for(x=0; x<W'; x++)
      R(x, y) = I(x/Sx, y/Sy)
```

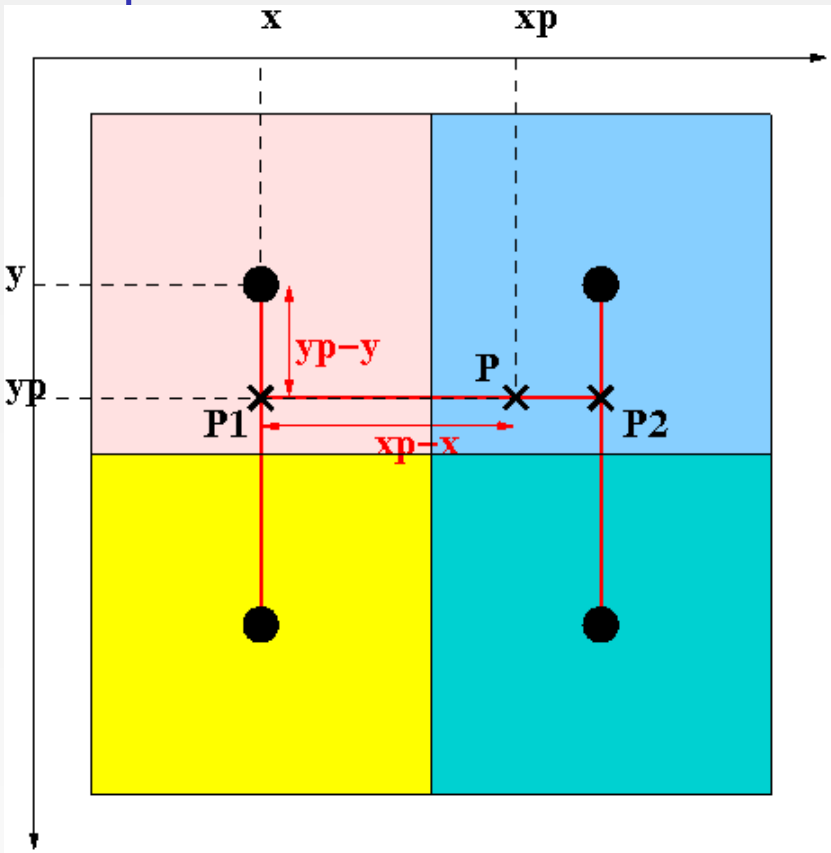
Remarque : arrondi entier des coordonnées de l'antécédent = "interpolation au plus proche voisin"

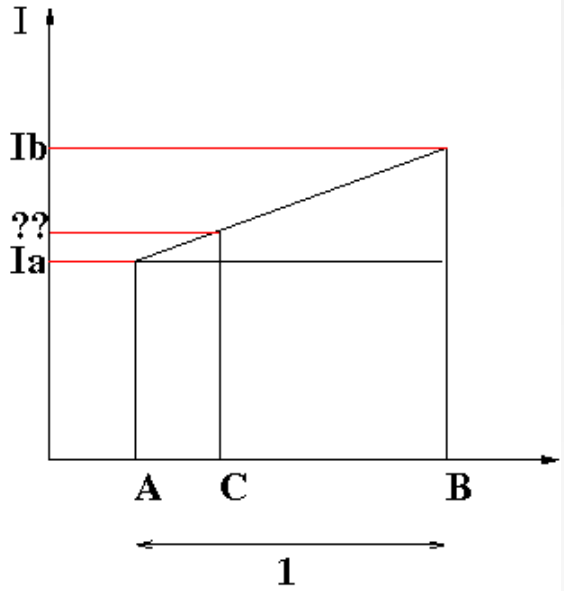
Changement d'échelle : aliasing





Interpolation bilinéaire





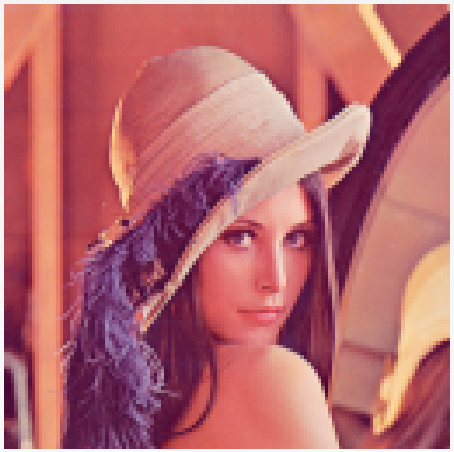
$$I_C = I_A + d(AC).(I_B - I_A)$$

$$I_{P1} = I(x,y) + (y_P - y).(I(x,y+1) - I(x,y))$$

$$I_{P2} = I(x+1,y) + (y_P - y).(I(x+1,y+1) - I(x+1,y))$$

$$I_P = I_{P1} + (x_P - x).(I_{P2} - I_{P1})$$

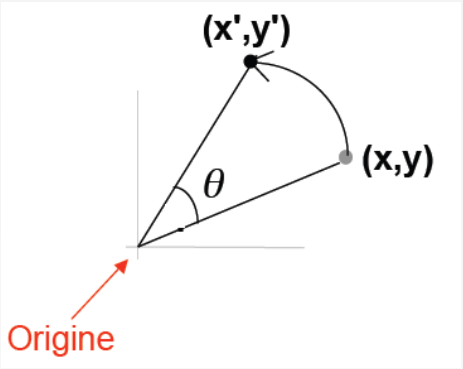
Interpolation bilinéaire : zoom $\times$ 2



Rotation autour de l'origine

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

avec θ angle de rotation



Démonstration

Coordonnées polaires : $x = r\cos(\alpha)$ et $y = r\sin(\alpha)$

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

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

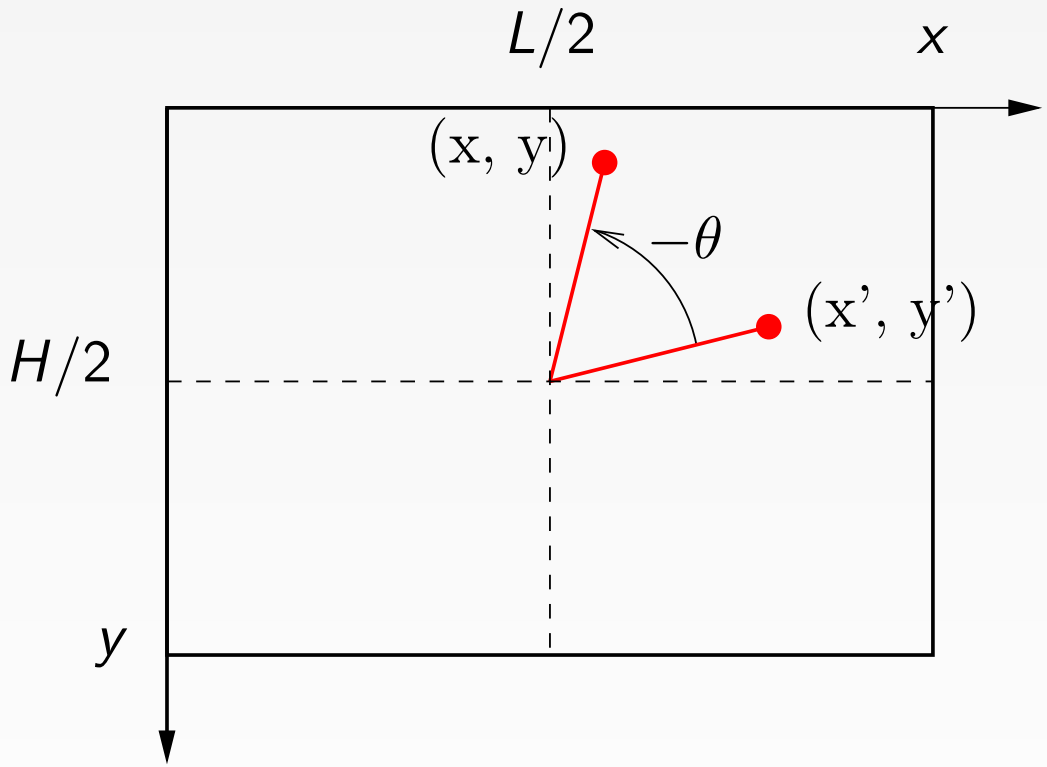
$$x' = x\cos(\theta) - y\sin(\theta)$$
$$y' = y\cos(\theta) + x\sin(\theta)$$

Rotation d'une image autour de son centre

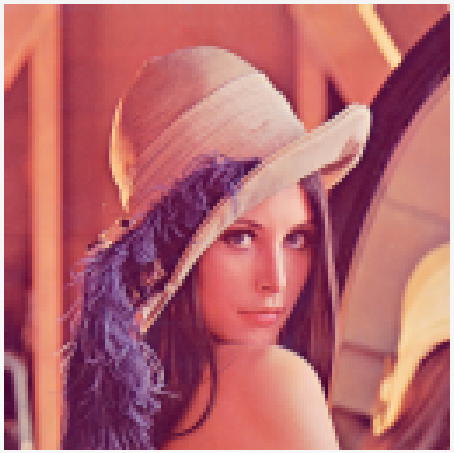
Recherche de l'antécédent :

$$x = W/2 + (x' - W/2)\cos(\theta) - (y' - H/2)\sin(\theta)$$

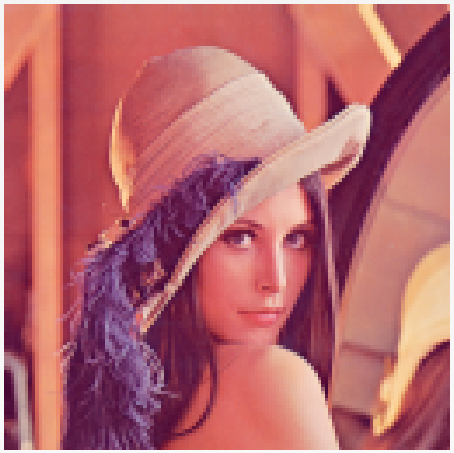
$$y = H/2 + (y' - H/2)\cos(\theta) + (x' - W/2)\sin(\theta)$$



Rotation avec interpolation au plus proche voisin

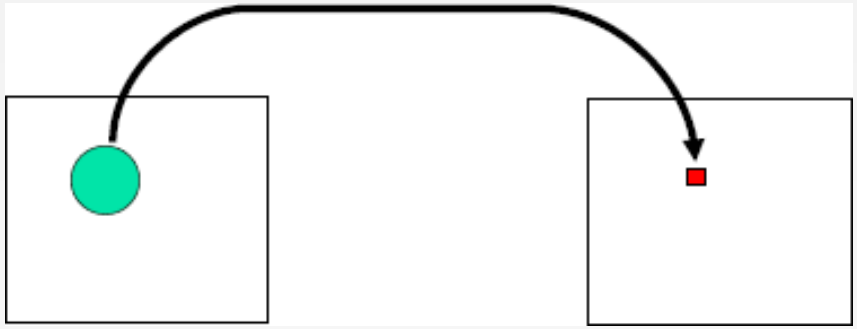


Rotation avec interpolation bilinéaire



Convolution

- Transformation locale : utilisation du voisinage de chaque pixel



- Un filtre de convolution (ou masque ou noyau) est généralement une matrice $2n+1 \times 2n+1$.
- Calcul : somme de produits, on parle de filtre linéaire

$$R(x,y) = \sum_{u=-n}^{u=n} \sum_{v=-n}^{v=n} I(x+u, y+v).K(u+n, v+n)$$

Pour éviter de modifier la luminance de l'image, la somme des coefficients du filtre doit être égale à 1.

Convolution : exemple

Image d'origine

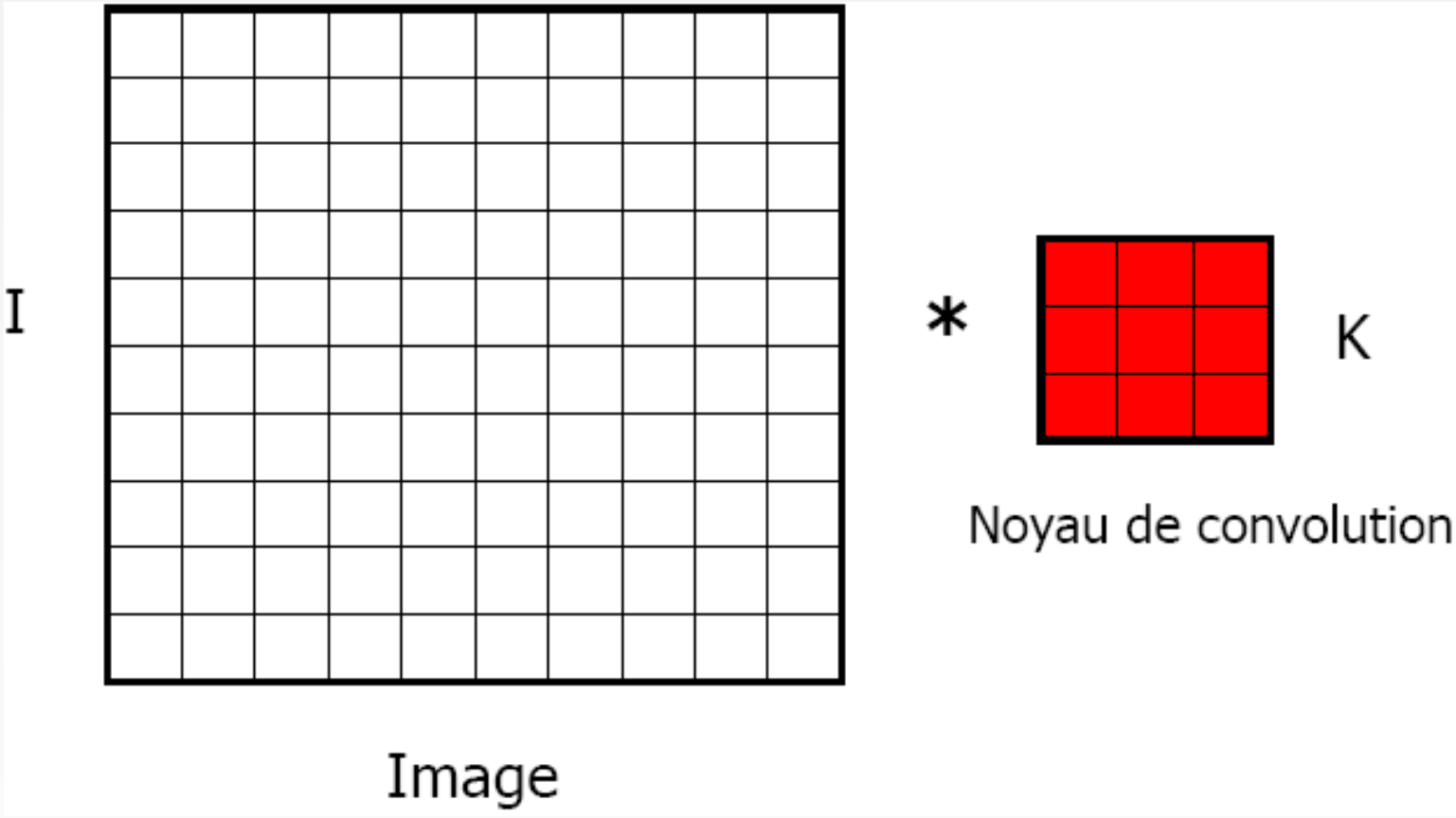
*

Filtre de convolution (masque)

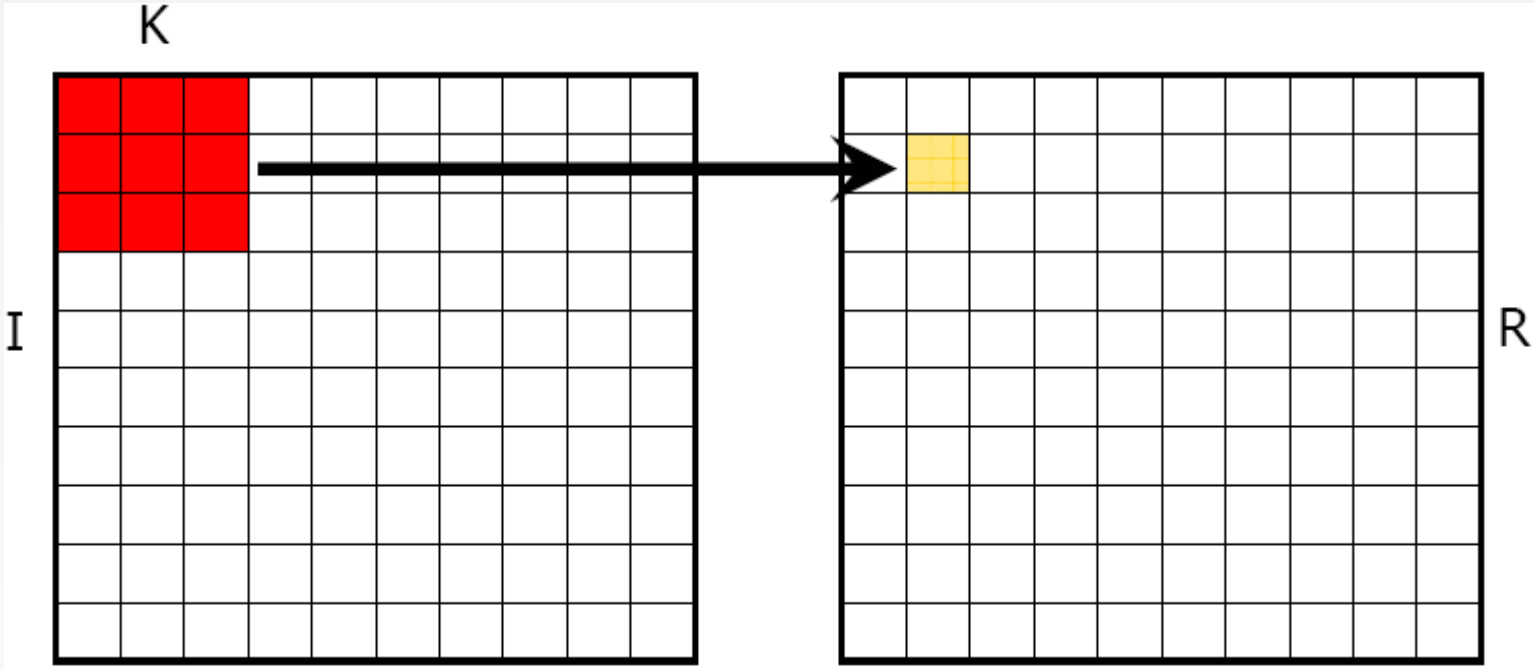
=

Image convoluée (résultat)

Convolution : détail 1/5

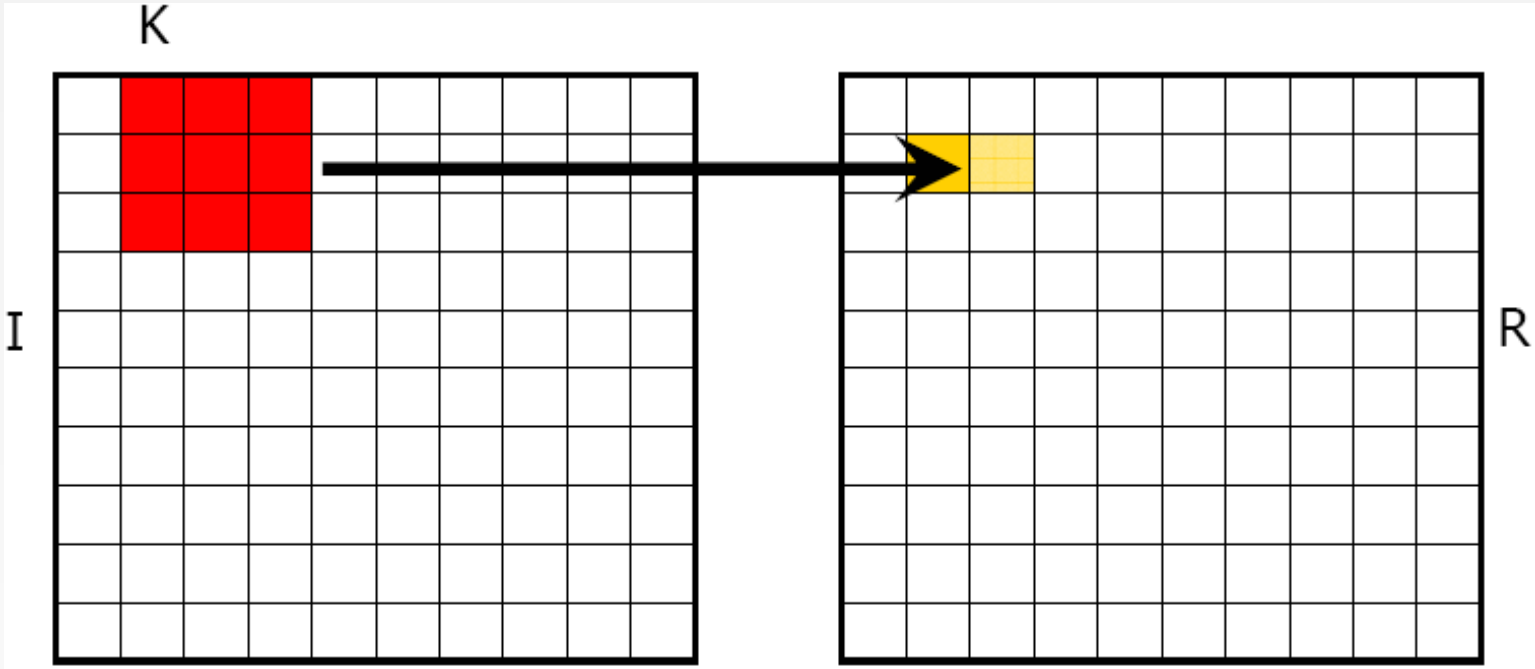


Convolution : détail 2/5



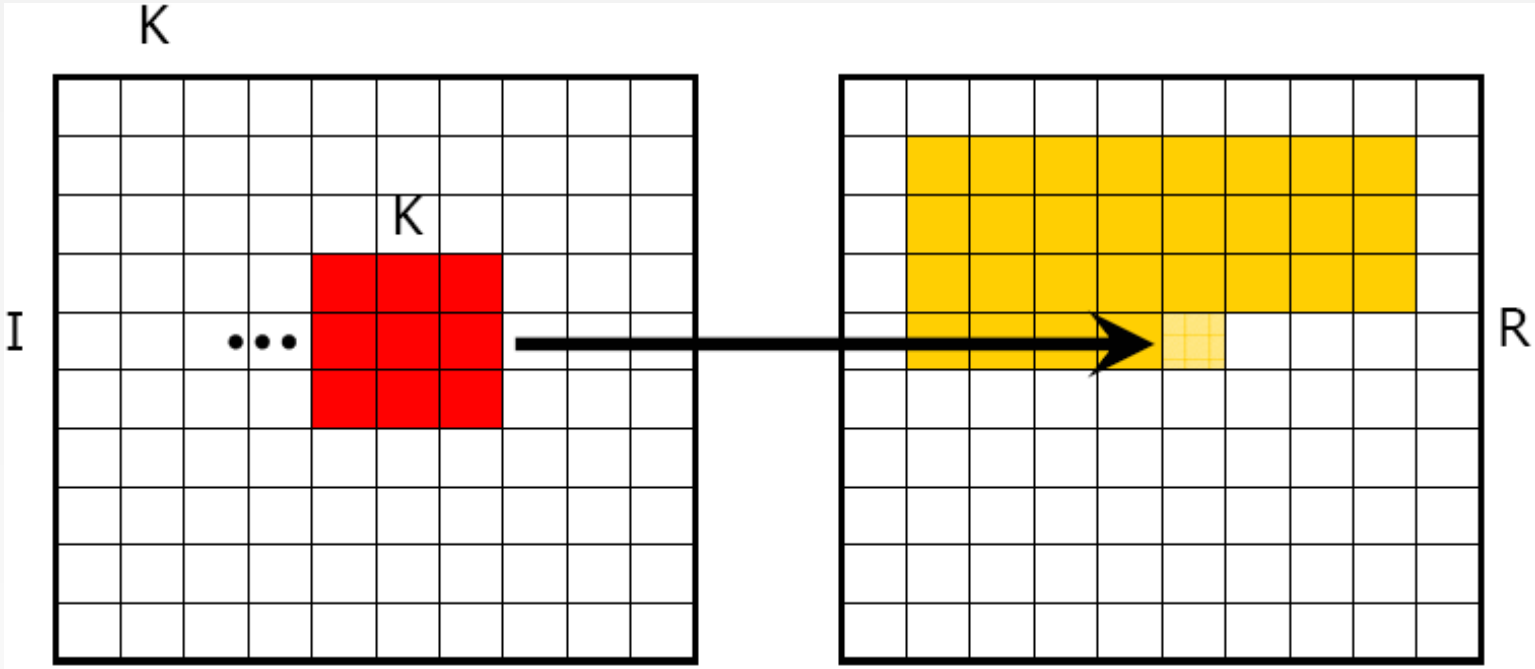
$$R(1,1) = I(0,0)\,K(0,0) + I(1,0)\,K(1,0) + I(2,0)\,K(2,0) + I(0,1)\,K(0,1) + I(1,1)\,K(1,1) + I(2,1)\,K(2,1) + I(0,2)\,K(0,2) + I(1,2)\,K(1,2) + I(2,2)\,K(2,2)$$

Convolution : détail 3/5



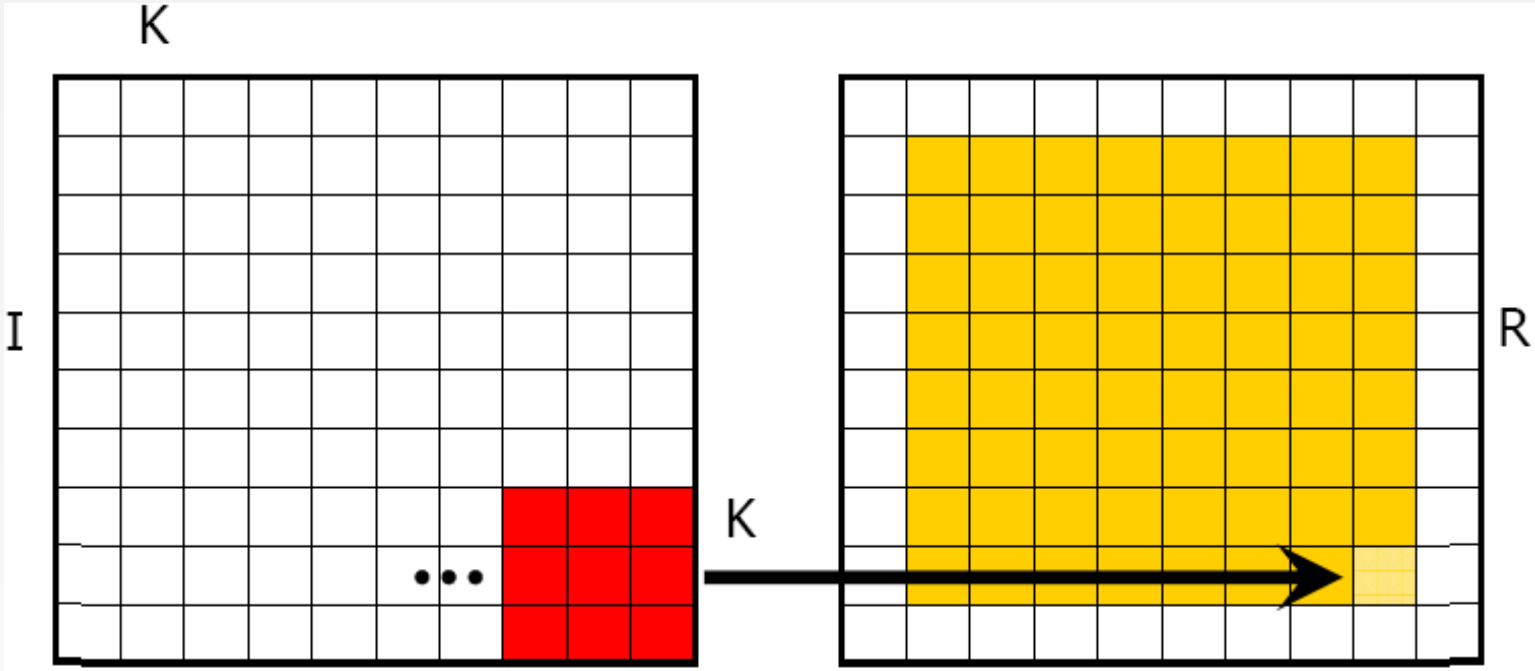
$$R(2,1) = I(1,0)\,K(0,0) + I(2,0)\,K(1,0) + I(3,0)\,K(2,0) + I(1,1)\,K(0,1) + I(2,1)\,K(1,1) + I(3,1)\,K(2,1) + I(1,2)\,K(0,2) + I(2,2)\,K(1,2) + I(3,2)\,K(2,2)$$

Convolution : détail 4/5



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

Convolution : détail 5/5

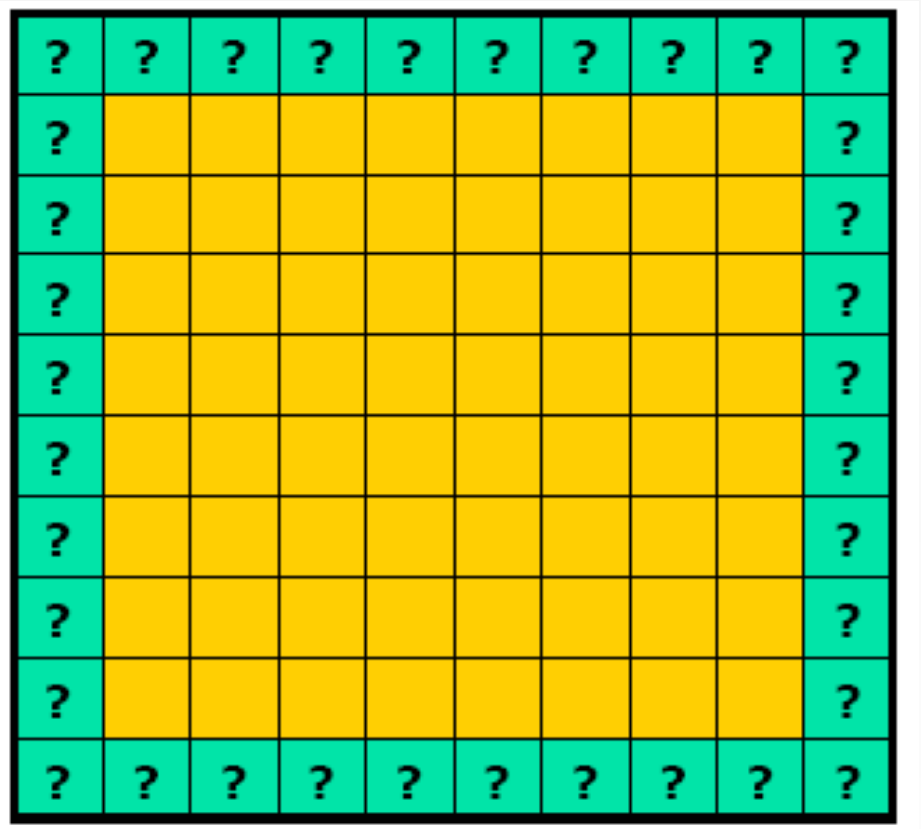


$$R(N-2,M-2) = I(N-3,M-3)\,K(0,0) + I(N-2,M-3)\,K(0,1) + I(N-1,M-3)\,K(0,3) + I(N-3,M-2)\,K(1,0) + I(N-2,M-2)\,K(1,1) + I(N-1,M-2)\,K(1,2) + I(N-3,M-1)\,K(2,0) + I(N-2,M-1)\,K(2,1) + I(N-1,M-1)\,K(2,2)$$

Calcul sur les bords de l'image

Plusieurs possibilités :

- Mettre à zéro
- Convolution partielle utilisant une portion du filtre
- Compléter les valeurs manquante en construisant le miroir de l'image



Familles de filtres

- Filtre passe-bas : atténue le bruit et les détails
- Filtre passe-haut : accentue les détails et les contours

Le filtre moyenneur

C'est un filtre passe-bas

- Lisse l'image (effet de flou)
- Réduit le bruit
- Réduit les détails

Filtre dont tous les coefficients sont égaux (chaque pixel est remplacé par la moyenne de ses voisins)

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

ou

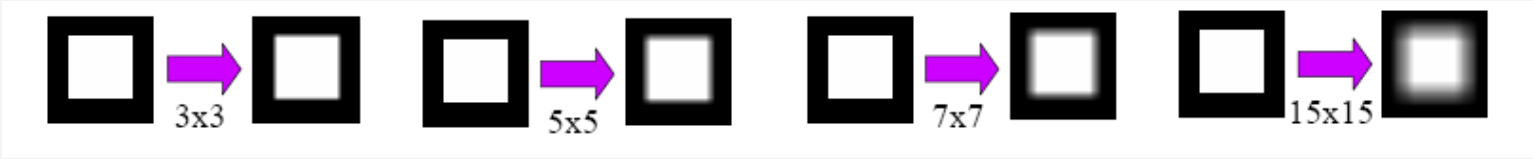
1/9

1	1	1
1	1	1
1	1	1

3x3

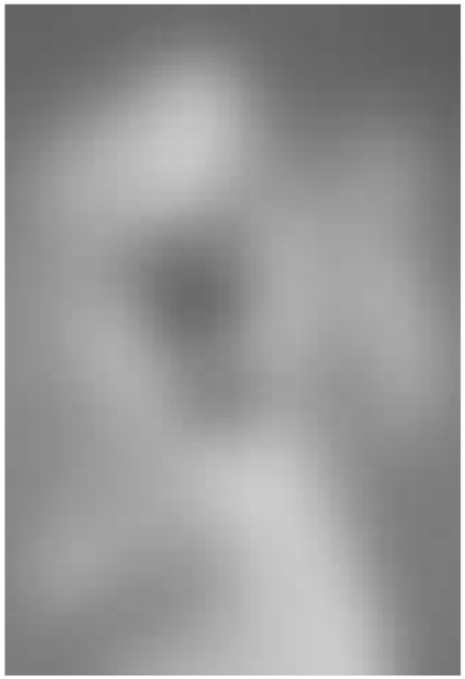
Filtre moyenneur : exemples

Plus le filtre grossit, plus le lissage devient important.



Original

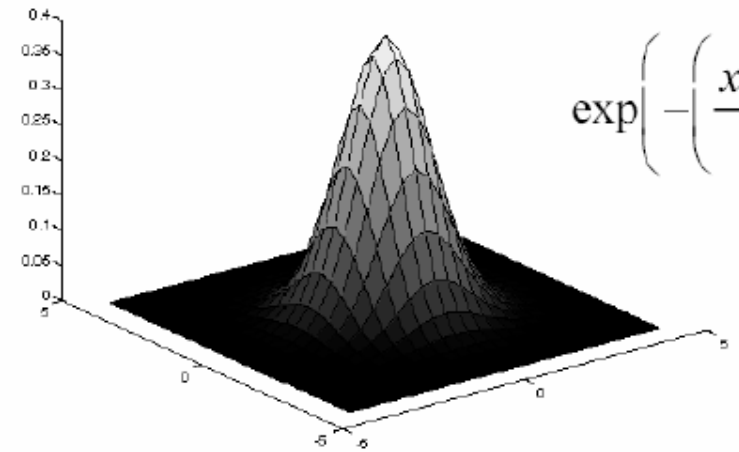
Moyenne 5x5



Moyenne 11x11

Le filtre Gaussien

Le filtre gaussien donne un meilleur 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

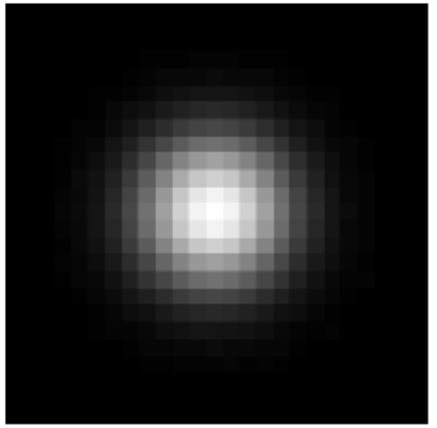


Image d'une gaussienne

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

Filtre Gaussien : exemple

Original

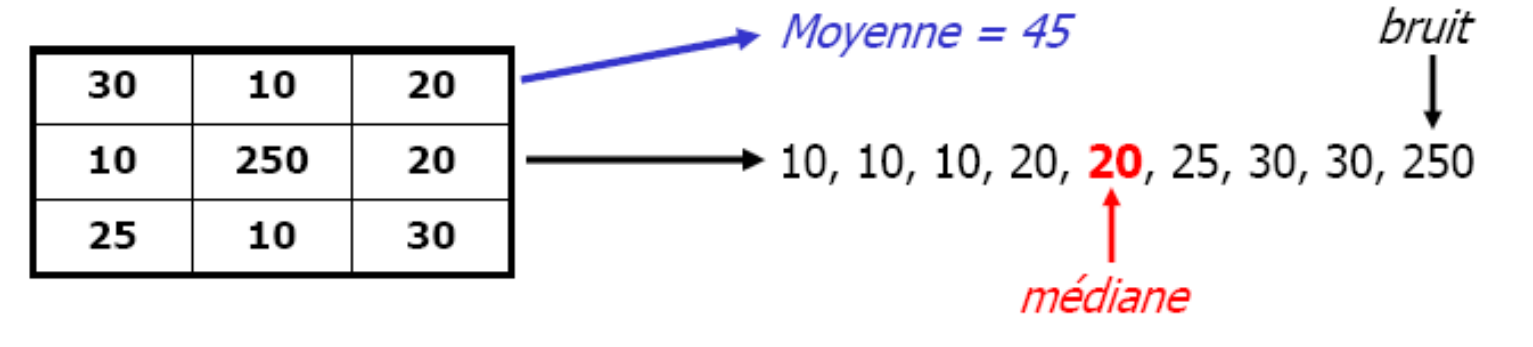
Gauss 5x5

Gauss 11x11

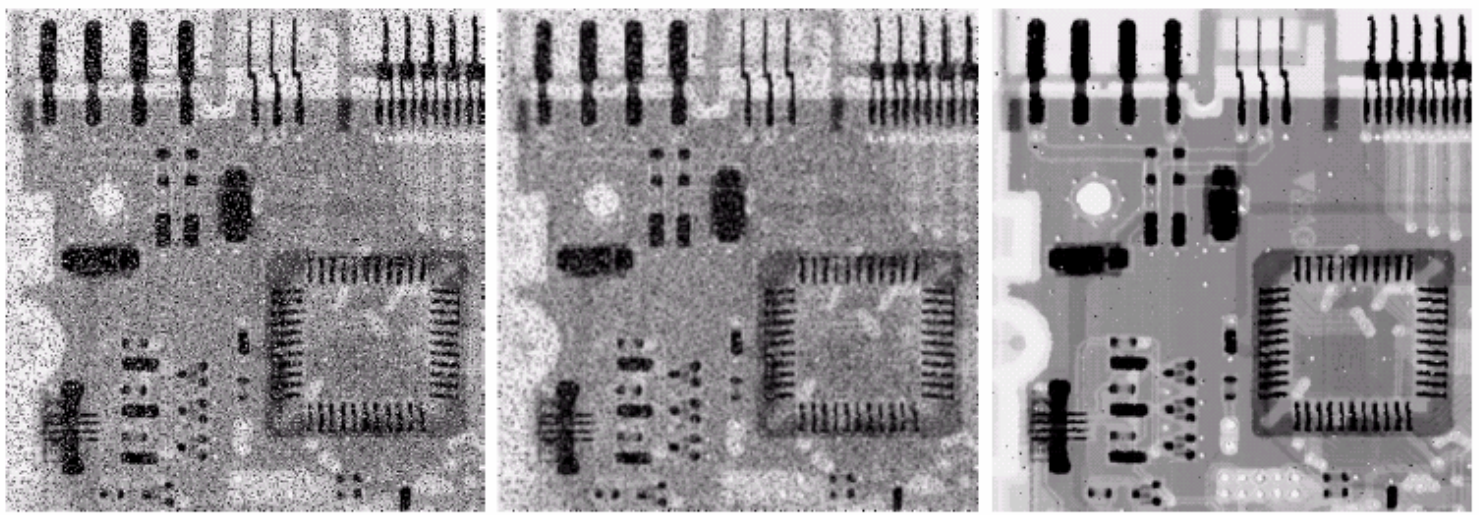
Filtre médian

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

- C'est un filtre non-linéaire, qui ne peut pas s'implémenter comme une convolution
- On remplace la valeur d'un pixel par la valeur médiane dans son voisinage $2n+1 \times 2n+1$



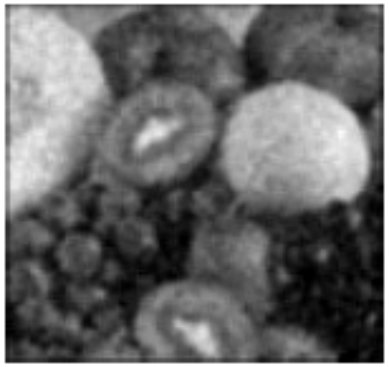
Filtre médian : exemple 1



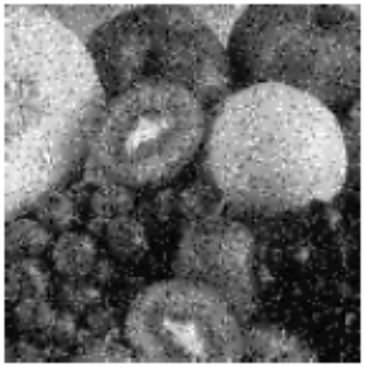
a b c

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.)

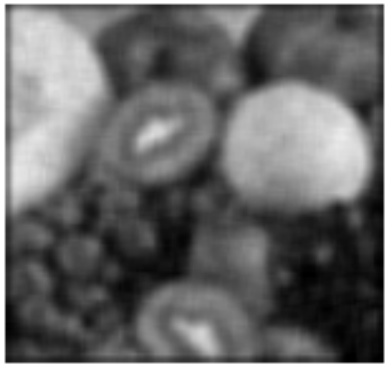
Filtre médian : exemple 2



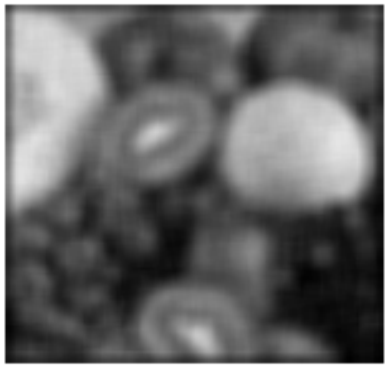
3 X 3 Moyenne



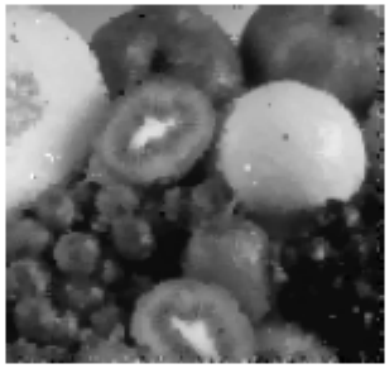
Bruit "poivre et sel"



5 X 5 Moyenne



7 X 7 Moyenne



Filtre médian

Filtre médian : exemple 3