



Traitement d'images

Convolution spatiale

Filtre spatial

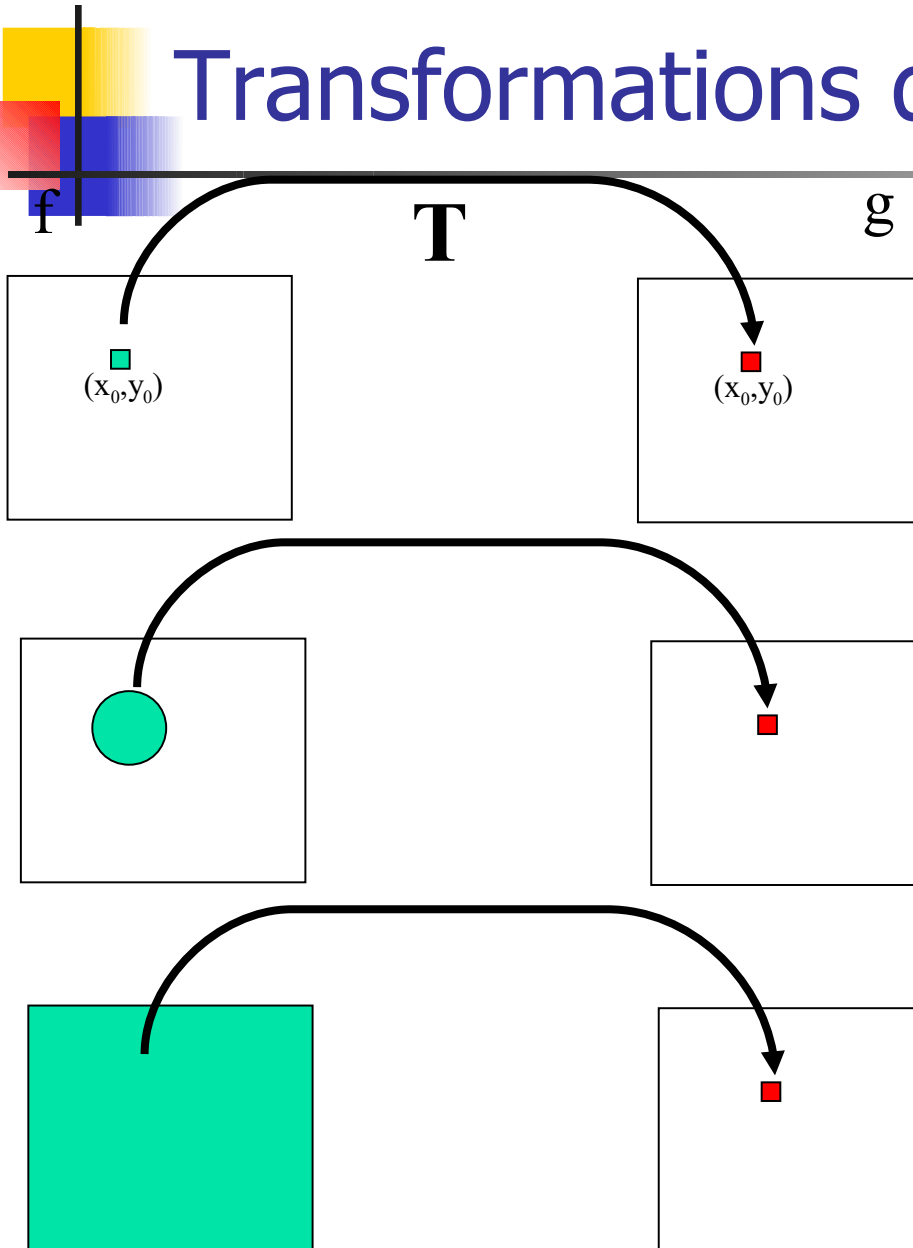
NGUYỄN Thị Oanh – IPH
oanhnt@soict.hust.edu.vn



Modification des valeurs d'une image

- Pour l'instant, nous avons vu surtout des **transformations ponctuelles** des pixels d'une image
 - *Lire la valeur d'un pixel → la remplacer par une autre*
- Il existe aussi des **transformations locales**
 - *Lire la valeur de quelques pixels voisins → calculer une nouvelle valeur pour un pixel*
- ...et des **transformations globales**
 - *Lire la valeur de tous les pixels de l'image → calculer une nouvelle valeur pour un seul pixel*

Transformations des pixels



Ponctuelle: $g(x_0, y_0) = T[f(x_0, y_0)]$

Locale: $g(x_0, y_0) = T[f(V)]$
 V : voisinage de (x_0, y_0)

Globale: $g(x_0, y_0) = T[f(x, y)]$
par ex: TF



Convolution numérique

- *filtres linéaires*



Convolution numérique

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

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

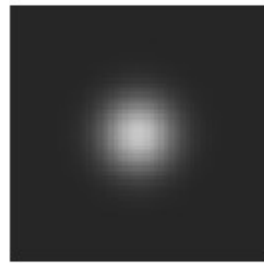
- **f(x)** est la fonction d'origine et **g(x)** la fonction convoluée (résultat de la convolution)
 - Dans notre cas, une image est vue comme une fonction mathématique
- **h(x)** est appelé *masque de convolution*, *noyau de convolution*, *filtre*, *fenêtre*, *kernel*, ...

Exemple de convolution 2D



Image d'origine

*



=



Image convoluée
(résultat)

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



Convolution numérique discrète

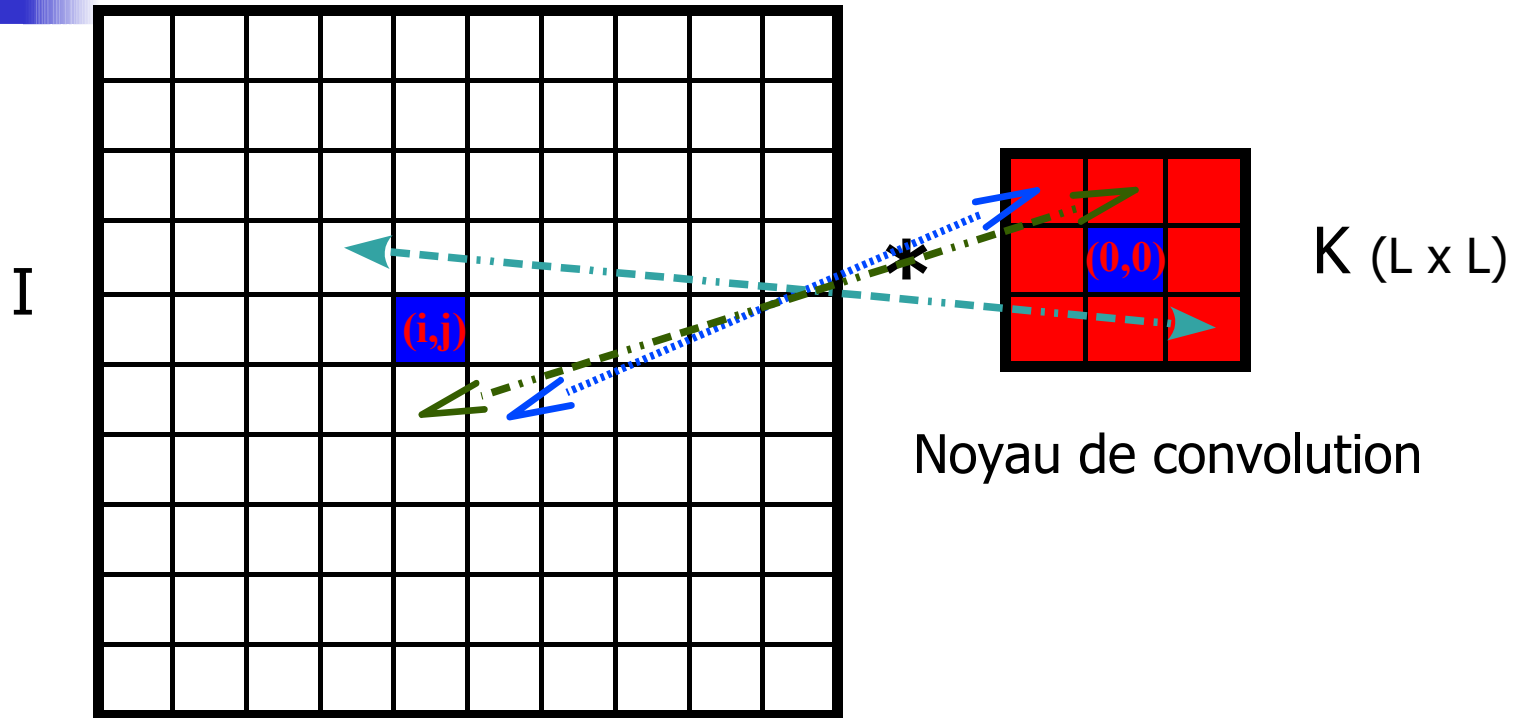
- En pratique, la convolution numérique d'une image se fera par une **sommation de multiplications**
- Un filtre de convolution est une matrice (image) généralement (*mais pas toujours*) de **taille impaire et symétrique**
 - 3x3, 5x5, 7x7, ...

Convolution d'une image par un filtre 2D :

$$I'(i, j) = I(i, j) * \text{filtre}(i, j)$$

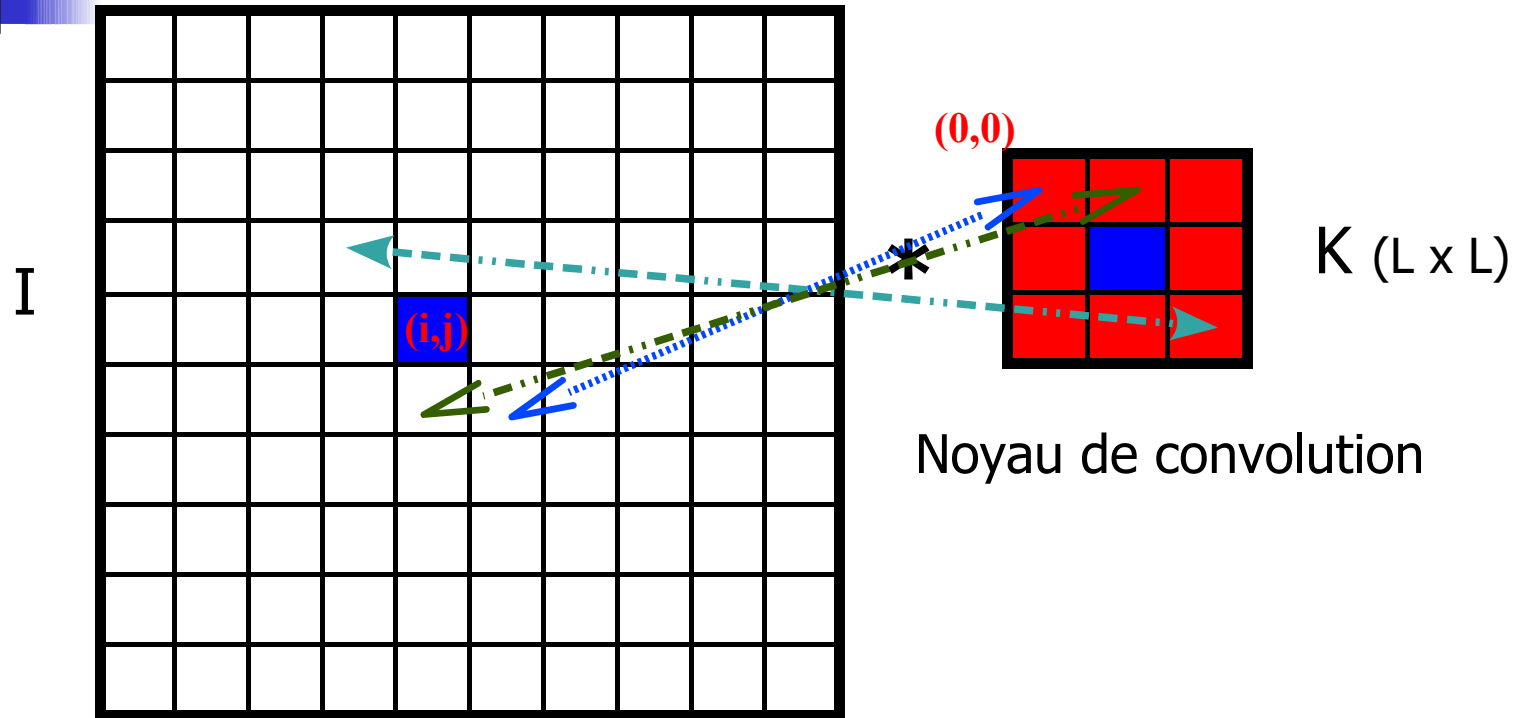
$$I'(i, j) = \sum_u \sum_v I(i-u, j-v) \cdot \text{filtre}(u, v)$$

Convolution numérique



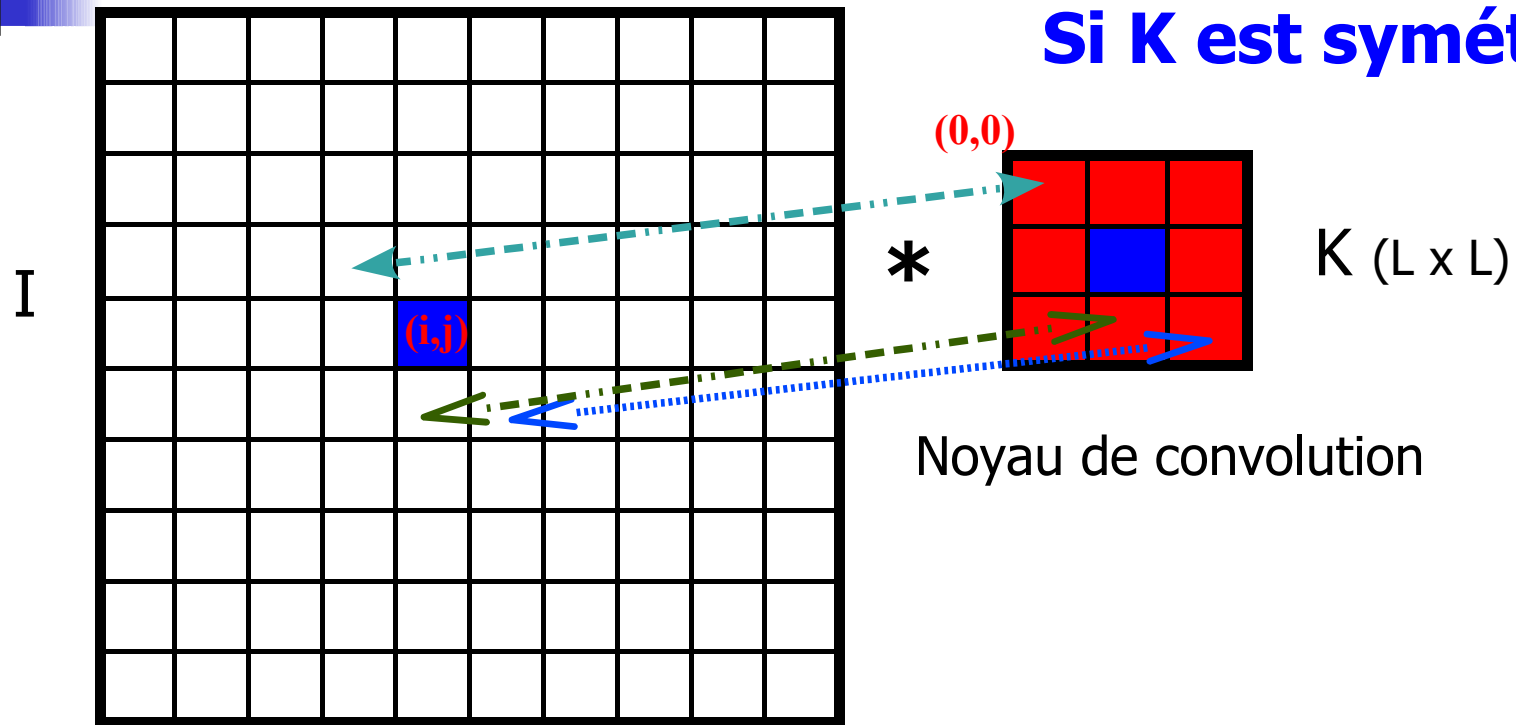
$$I'(i, j) = \sum_{u=-(L-1)/2}^{(L-1)/2} \sum_{v=-(L-1)/2}^{(L-1)/2} I(i-u, j-v) \cdot K(u, v)$$

Convolution numérique - Implémentation



$$I'(i, j) = \sum_{u=0}^{L-1} \sum_{v=0}^{L-1} I\left(i-u+\frac{L-1}{2}, j-v+\frac{L-1}{2}\right) \cdot K(u, v)$$

Convolution numérique - Implémentation



$$I'(i, j) = \sum_{u=0}^{L-1} \sum_{v=0}^{L-1} I\left(i+u-\frac{L-1}{2}, j+v-\frac{L-1}{2}\right) \cdot K(u, v)$$

Convolution numérique

105	102	100	97	96
103	99	103	101	102
101	98	104	102	100
99	101	106	104	99
104	104	104	100	98

Image Matrix

Kernel Matrix

0	-1	0
-1	5	-1
0	-1	0

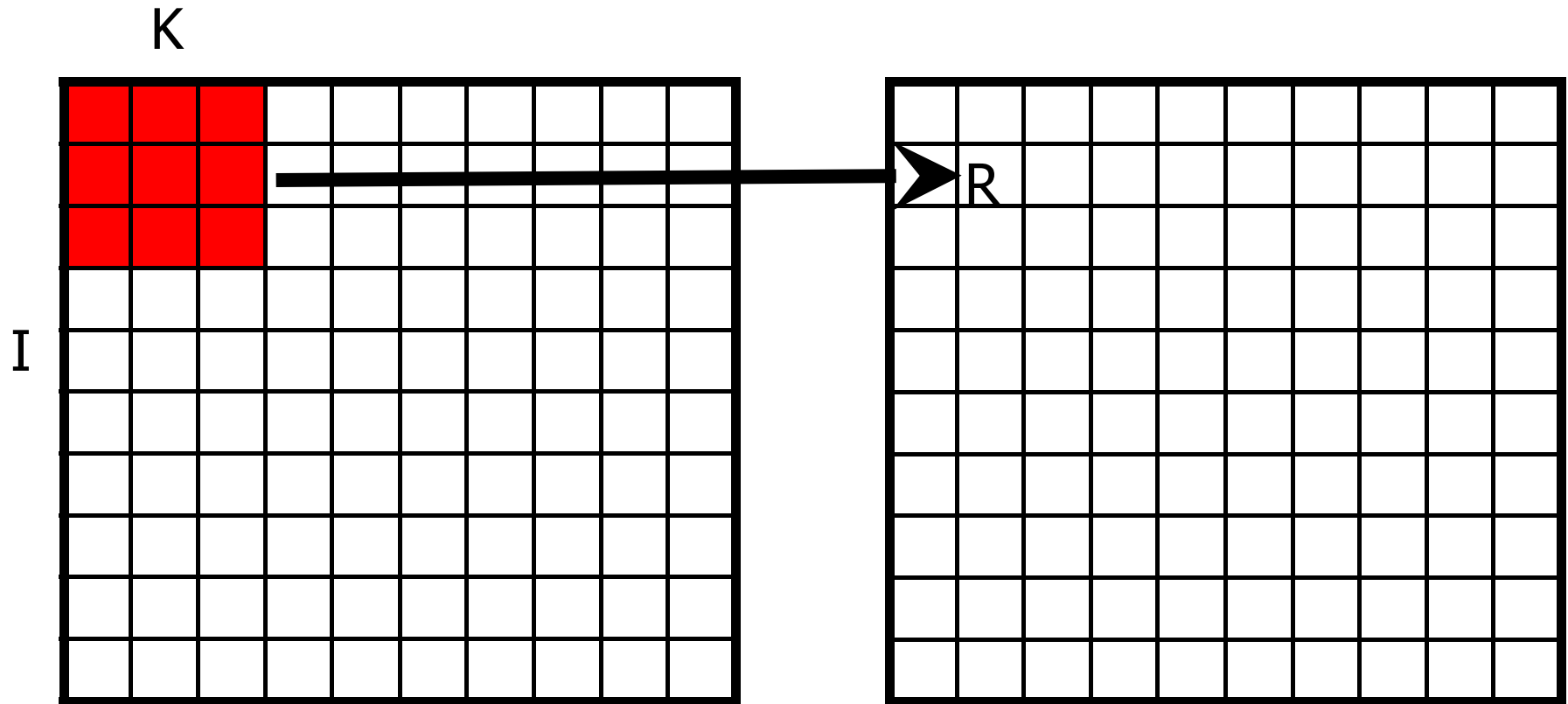
	89			

Output Matrix

$$\begin{aligned} &105 * 0 + 102 * -1 + 100 * 0 \\ &+ 103 * -1 + 99 * 5 + 103 * -1 \\ &+ 101 * 0 + 98 * -1 + 104 * 0 = 89 \end{aligned}$$

Source: <http://machinelearningguru.com>

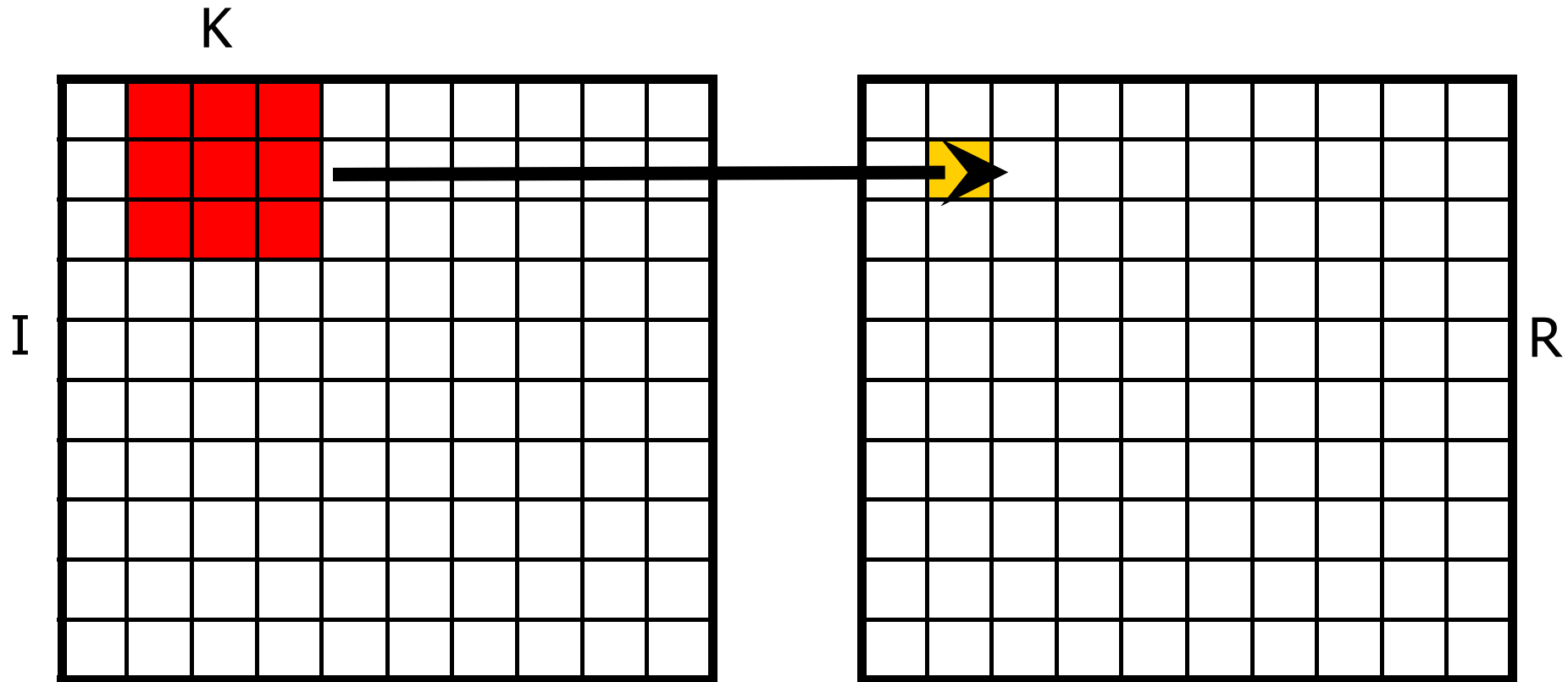
Convolution numérique $R = I * K$



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

Si K est symétrique

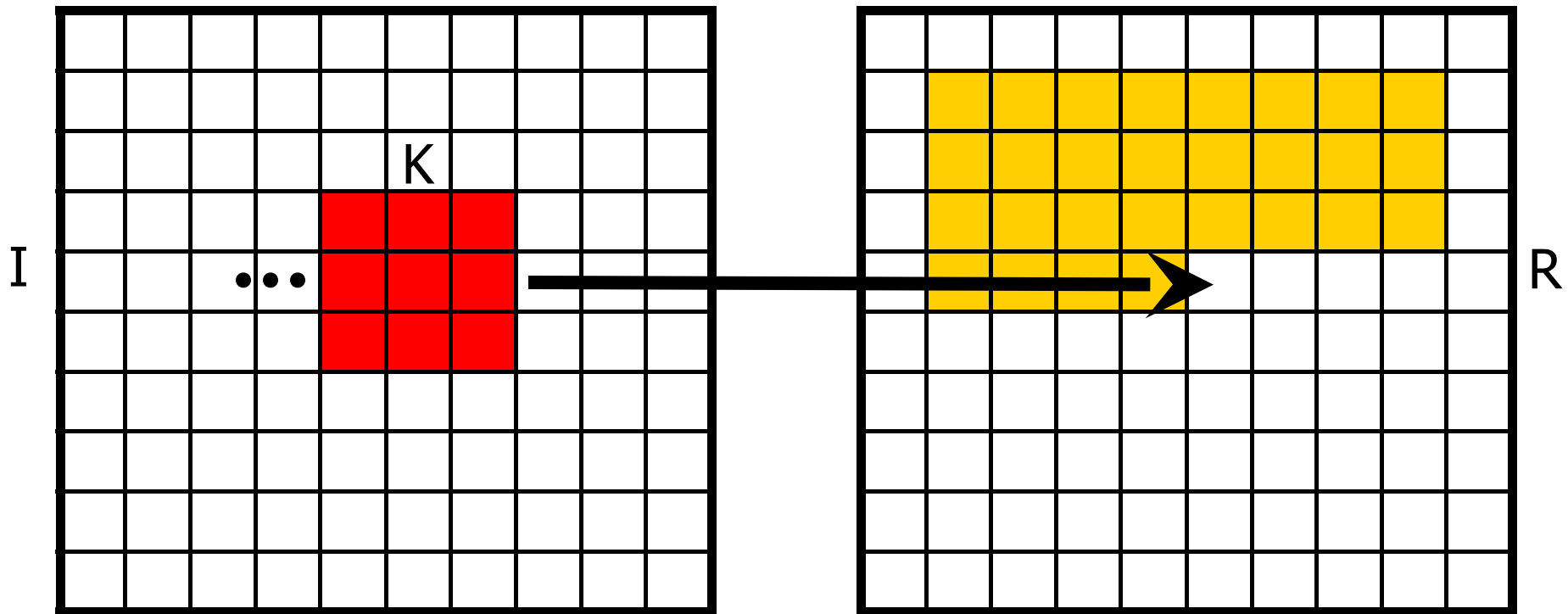
Convolution numérique $R = I * K$



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

Si K est symétrique

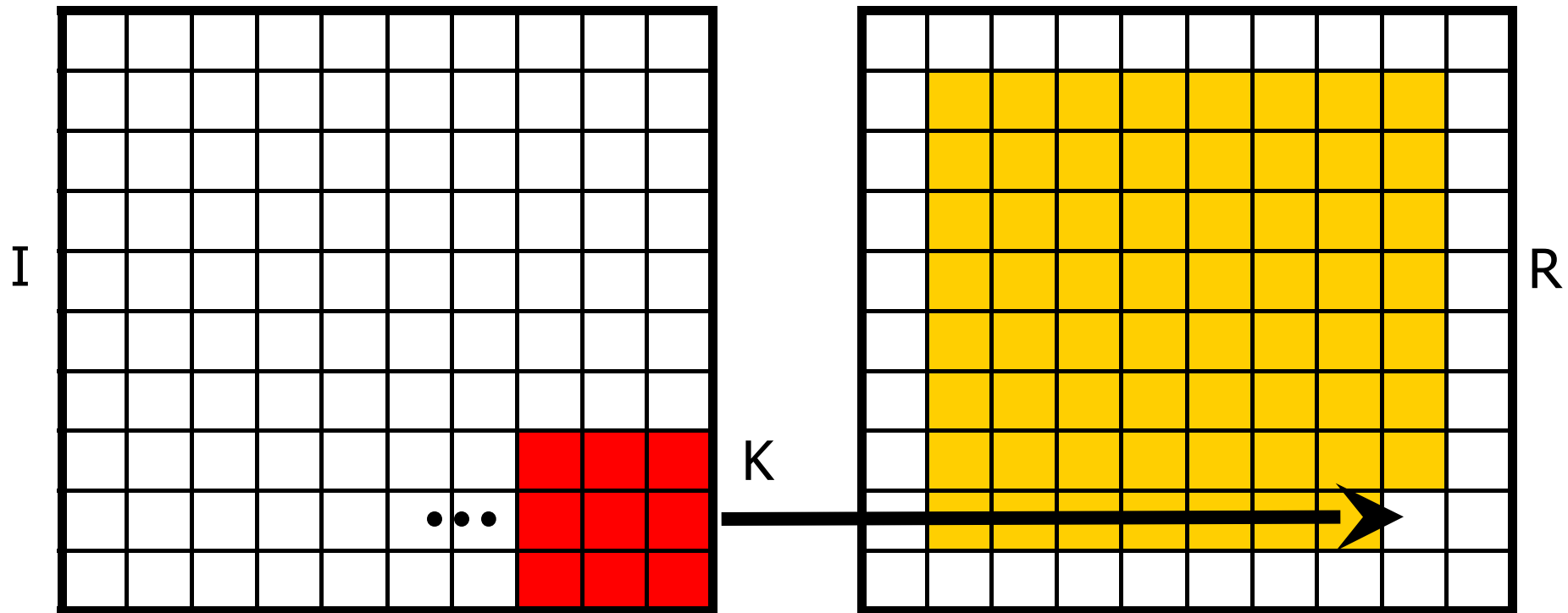
Convolution numérique $R = I * K$



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

Si K est symétrique

Convolution numérique $R = I * K$



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

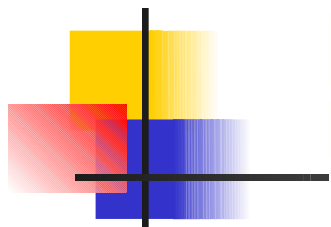
Si K est symétrique



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
 - $f(-x,y) = f(x,y)$
 - ... (pas de solution miracle)

?	?	?	?	?	?	?	?	?	?
?									?
?									?
?									?
?									?
?									?
?									?
?									?
?									?
?	?	?	?	?	?	?	?	?	?



0	0	0	0	0	0
0	105	102	100	97	96
0	103	99	103	101	102
0	101	98	104	102	100
0	99	101	106	104	99
0	104	104	104	100	98

Image Matrix

Kernel Matrix

0	-1	0
-1	5	-1
0	-1	0

320				
210	89	111		

Output Matrix

$$\begin{aligned}
 &0 * 0 + 0 * -1 + 0 * 0 \\
 &+ 0 * -1 + 105 * 5 + 102 * -1 \\
 &+ 0 * 0 + 103 * -1 + 99 * 0 = 320
 \end{aligned}$$

105	105	102	100	97	96
105	105	102	100	97	96
103	103	99	103	101	102
101	101	98	104	102	100
99	99	101	106	104	99
104	104	104	104	100	98

Image Matrix

Kernel Matrix

0	-1	0
-1	5	-1
0	-1	0

110				
	89	111		

Output Matrix





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
 - Pour éviter de modifier la luminance globale de l'image, la somme des coefficients doit être égale à 1

Examples



Original image

*

0	0	0
0	1	0
0	0	0



Filtered image
(no change)



Original image

*

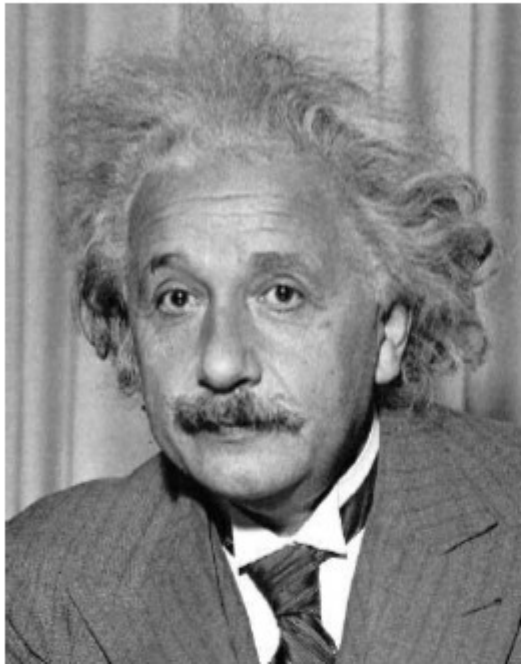
0	0	0
1	0	0
0	0	0



Filtered image
(shifted left by 1 pixel)

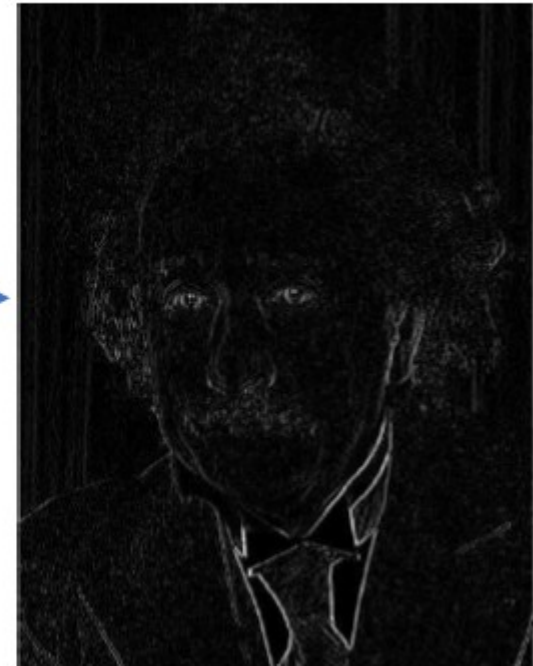
Examples

- Sobel



*

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



Vertical Edge
(absolute value)

Deux types pour le filtrage spatial

- **Filtres passe-bas**

- Atténue le bruit et les détails (basses fréquences)
→ lissage



- **Filtres passe-haut**

- Accentue les détails et les contours (hautes fréquences)
→ accentuation

Le filtre moyeneur

- Le filtre moyeneur
 - 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 moyeneurs :

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

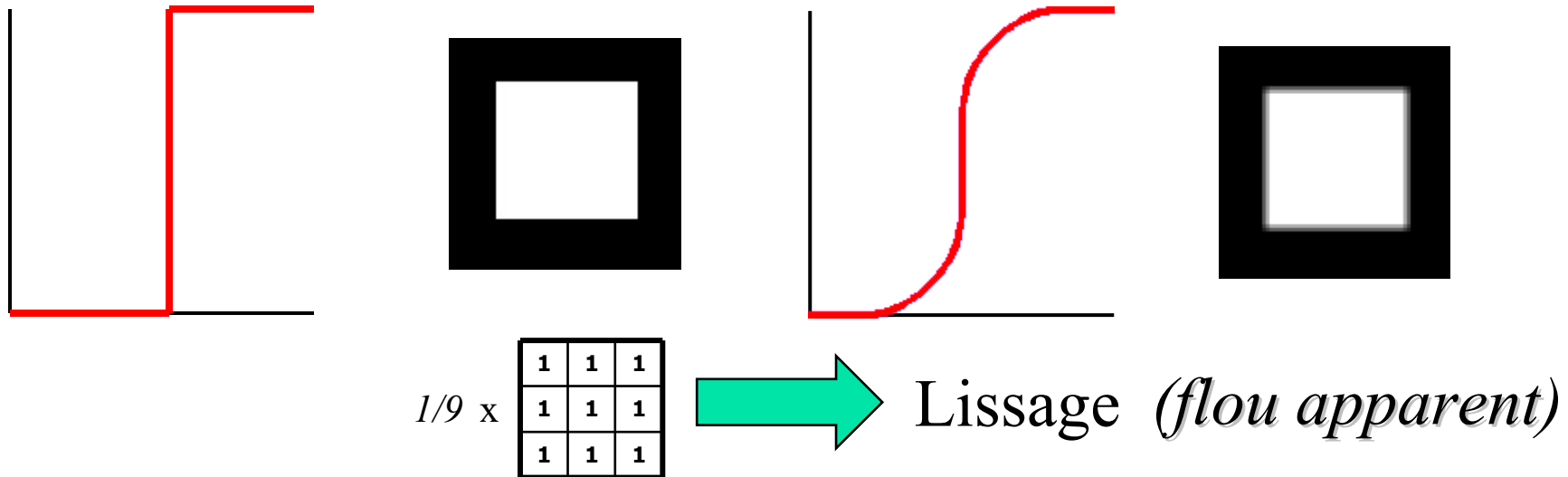
1/25

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

3x3

5x5

Le filtre moyeneur



Plus le filtre grossit , plus le lissage devient important et plus le flou s'accroît !

Exemples de filtres moyennneurs



Original

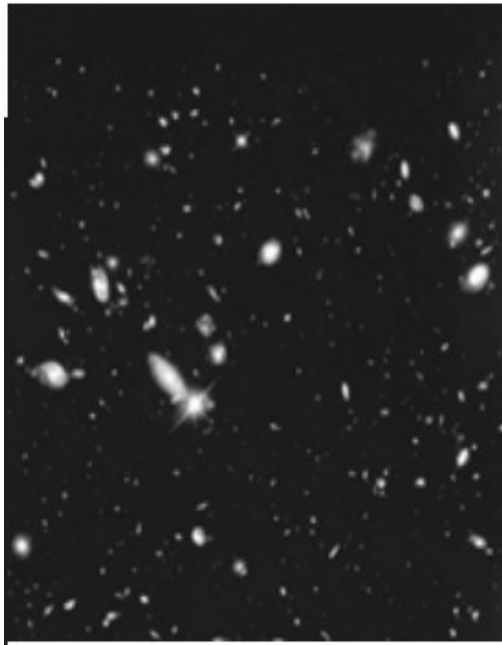


Moyenne 5x5

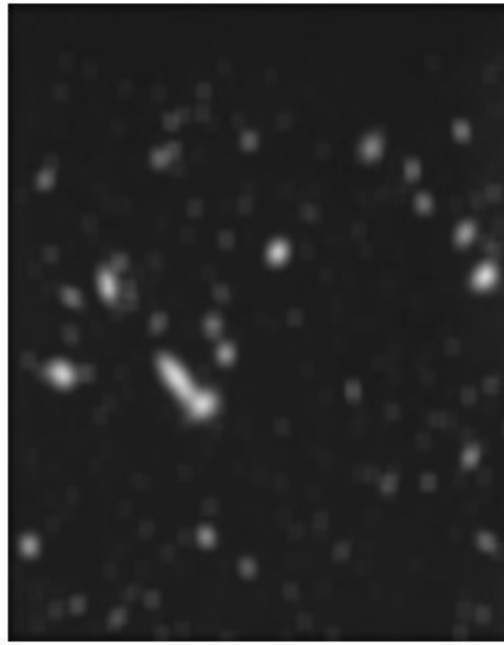


Moyenne 11x11

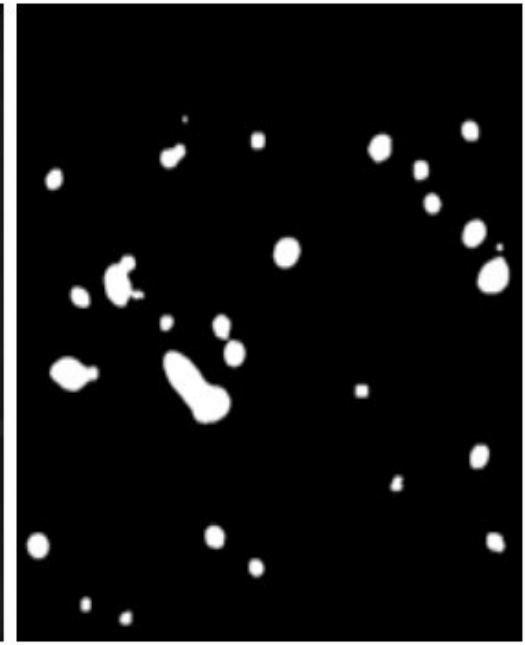
Filter moyenneurs



Original

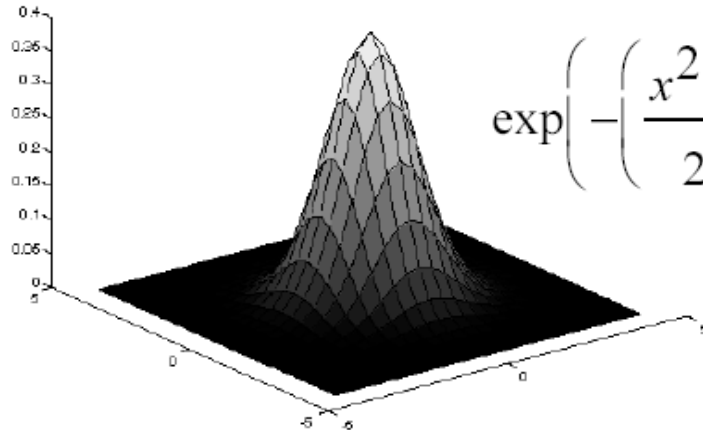


Moyenne 15x15



Seuillage

Le filtre Gaussien



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

Fonction gaussienne en 3D

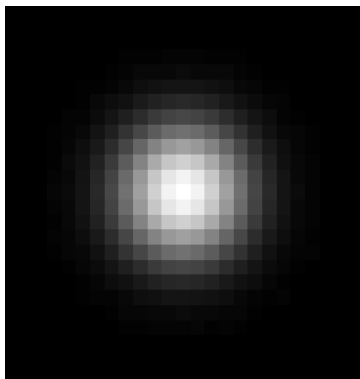


Image d'une gaussienne

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

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

Rule for Gaussian filter:
set **filter half-width to about 3σ**

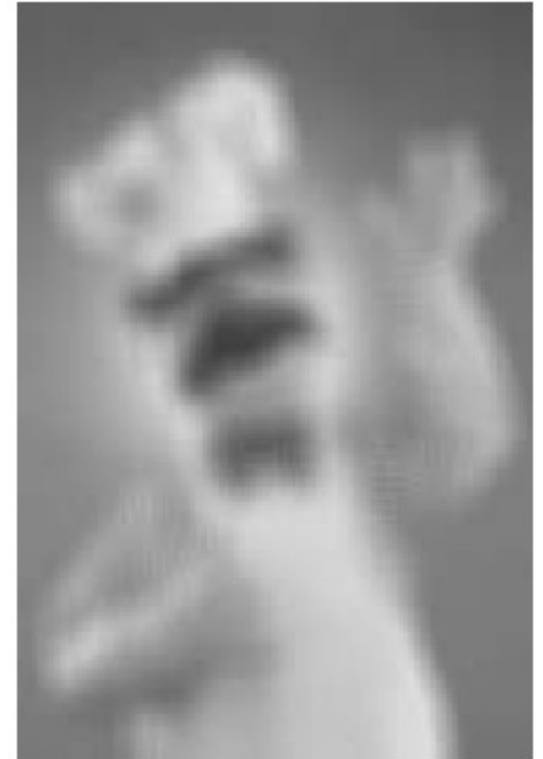
Exemples de filtres gaussiens



Original



Gauss 5x5



Gauss 11x11

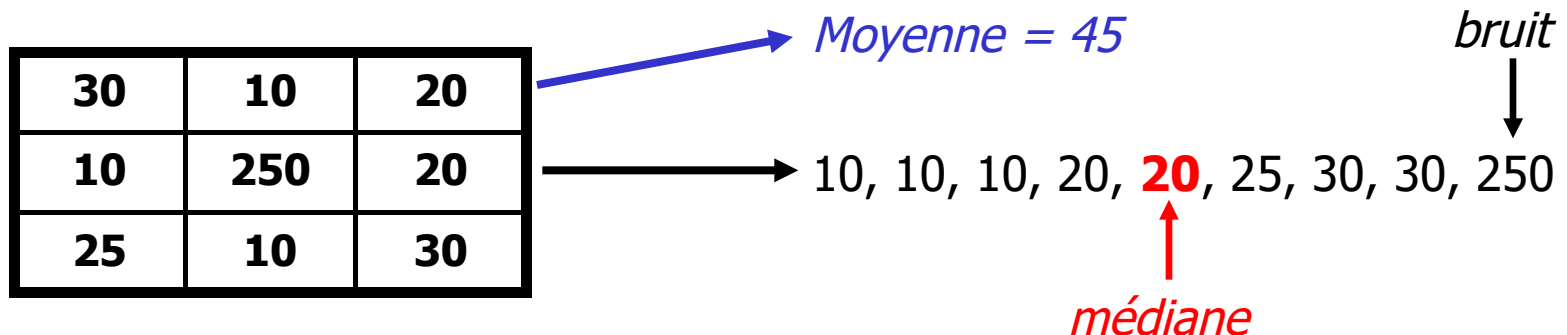


Filtres non-linéaires

(autre que convolution)

Filtre médian (*non-linéaire*)

- Pour nettoyer le bruit dans une image, il existe mieux que le **filtre moyennneur** 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

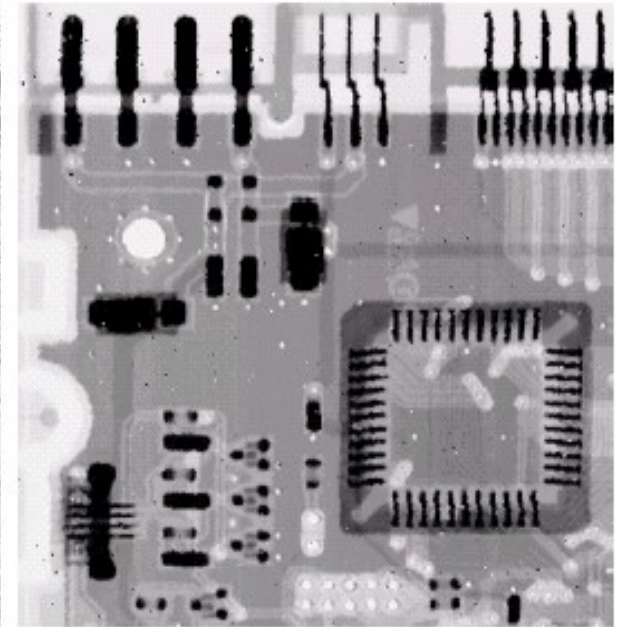
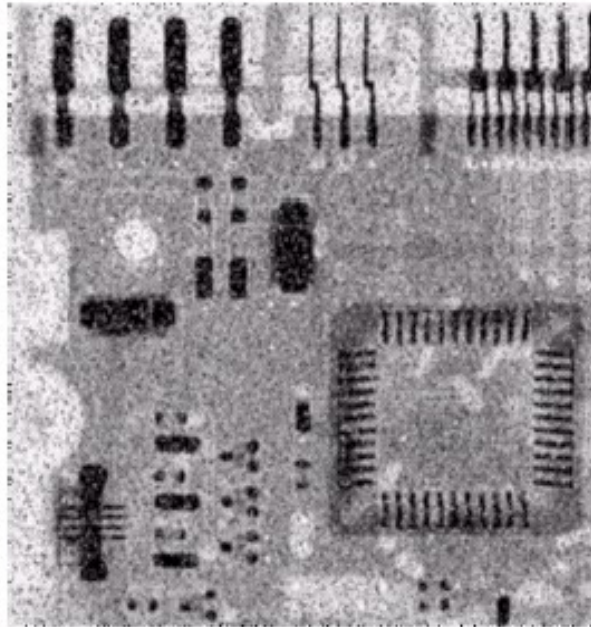
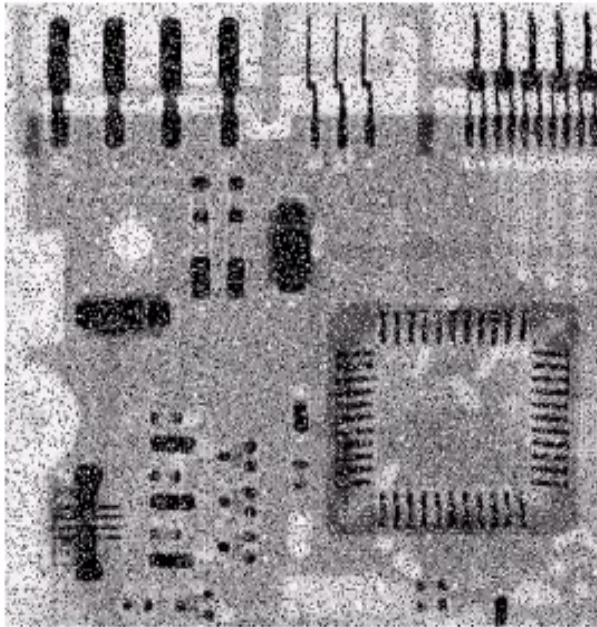


Exemple de filtre médian

Original

Moyenne 3x3

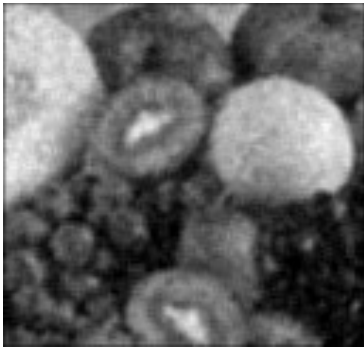
Médian 3x3



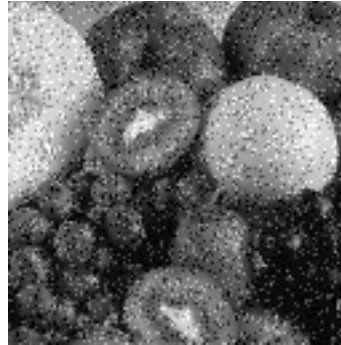
a b c

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.)

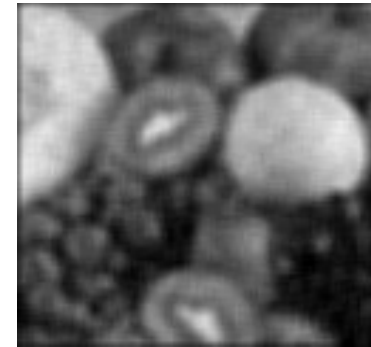
Nettoyage du bruit dans une image



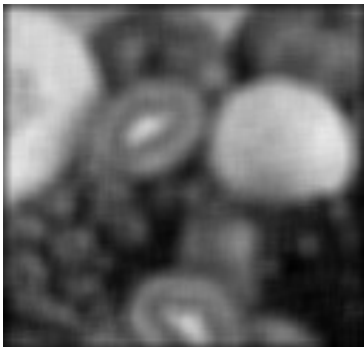
3 X 3 Moyenne



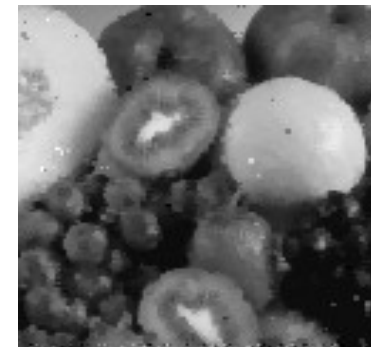
Bruit "poivre et sel"



5 X 5 Moyenne



7 X 7 Moyenne



Filtre médian



Image initiale



Bruit Poivre & Sel



Moyenne V8



Min V8



Max V8



Médian V8

Filtre median: attention !

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
0	0	100	100	0	100	150	150	100	0	100	100	100	0	0
0	0	100	100	100	0	0	150			100	100	100	0	0
0	0	100	100	100	100	0	0	0	0	100	100	100	0	0
0	0	100	100	100	100	100	0	0	0	100	100	100	0	0
0	0	100	100	100	100	100	100	0	0	100	100	100	0	0
0	0	100	100	100	0	100	100	0	0	100	100	100	0	0
0	0	100	100	100	0	0	100	100	0	100	100	100	0	0
0	0	100	100	100	0	0	0	100	100	100	100	100	0	0
0	0	100	100	100	0	100	0	100	100	100	100	100	0	0
0	0	100	100	100	0	200	0	0	100	100	100	100	0	0
0	0	100	100	100	0	0	0	0	0	100	100	100	0	0
0	0	100	100	100	0	0	0	0	0	0	100	100	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Filtrez cette image avec le filtre median 3x3 et 5x5, 7x7!
Resultats attendus?



Références

- Livres:
 - Introduction au Traitement d'Images (Lingrand), sections 4.3 et 4.4
 - Digital Image Processing 2ed (Gonzalez & Woods), chapitres 3 et 4
- Cours d'Antoine Mazanera sur le filtrage d'images (spatial et fréquentiel)
http://www.ensta-paristech.fr/~manzaner/Cours/D9_2/d92_Intro.pdf
- Caroline Rougier. Cours de Traitement d'images (IFT2730). Université de Montréal (Canada) (liens ne plus disponibles)
 - <http://www-etud.iro.umontreal.ca/~rougierc/ift2730/>