

Plan

3 Traitement - analyse d'image

- Intro
- Traitement
- Transformations géométriques
- **Filtrage**
- Analyse - détection de contour
- Détection de droites - Transformée de Hough

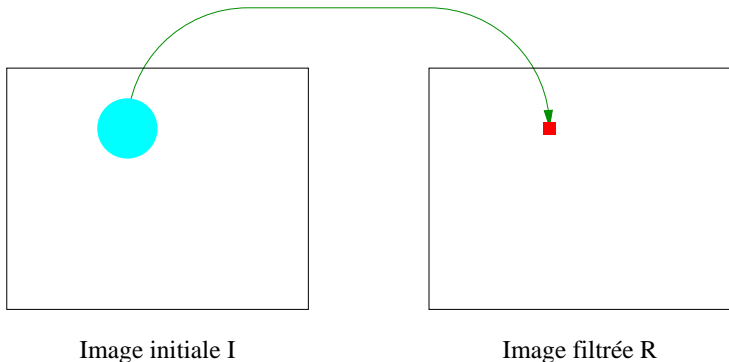
Filtrage linéaire

Filtrage linéaire

Image initiale $I(i,j)$ \longrightarrow Image filtrée $R(i,j)$

Filtrage linéaire

Image initiale $I(i,j)$ \longrightarrow Image filtrée $R(i,j)$



Filtrage linéaire

Image initiale $I(i,j)$ \longrightarrow Image filtrée $R(i,j)$

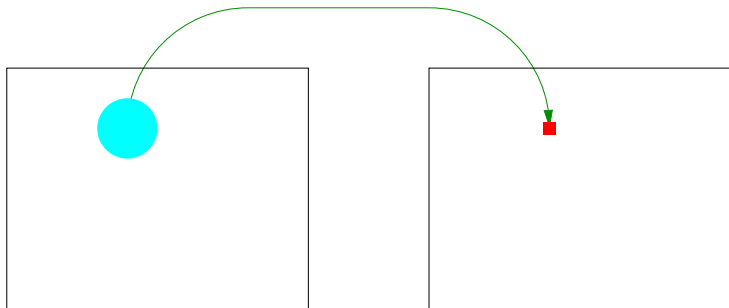


Image initiale I

Image filtrée R

Image $I \rightarrow$ image "filtrée" $R = I * K$
noyau de convolution K : tableau (matrice) de réels

Filtre linéaire - convolution

Filtre linéaire - convolution

Utilisation d'un noyau de convolution

$$K = (K(i, j))_{-n \leq i, j \leq n}$$

Filtre linéaire - convolution

Utilisation d'un noyau de convolution

$$K = (K(i, j))_{-n \leq i, j \leq n}$$

$$R(i, j) = \sum_{k=-n}^n \sum_{l=-n}^n K(k, l) I(i - k, j - l)$$

Filtre linéaire - convolution

Utilisation d'un noyau de convolution avec indices à partir de 0

$$K = (K(i, j))_{0 \leq i, j \leq 2n}$$

Filtre linéaire - convolution

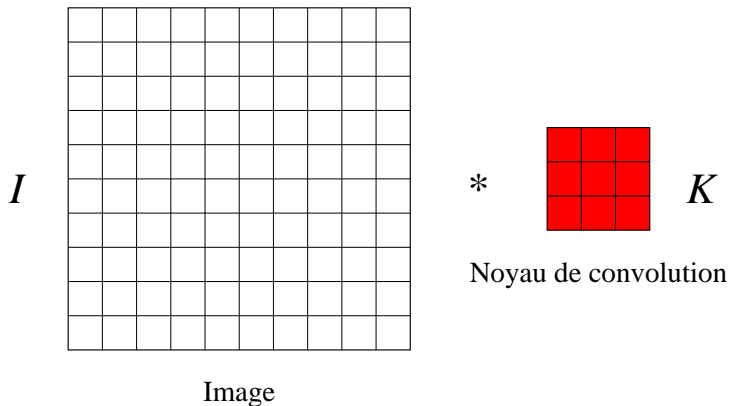
Utilisation d'un noyau de convolution avec indices à partir de 0

$$K = (K(i, j))_{0 \leq i, j \leq 2n}$$

$$R(i, j) = \sum_{k=-n}^n \sum_{l=-n}^n K(k+n, l+n) I(i+k, j+l)$$

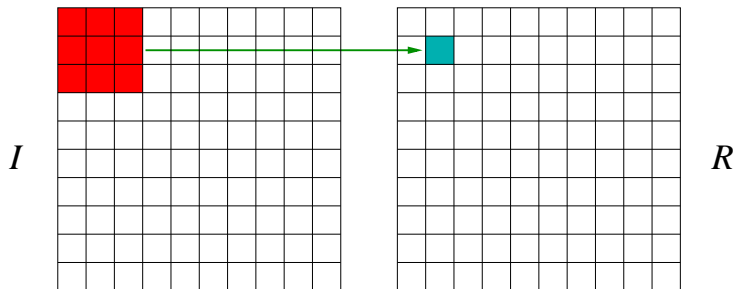
Filtre linéaire - convolution

Filtre linéaire - convolution



Le principe de calcul de la convolution $R = I * K$

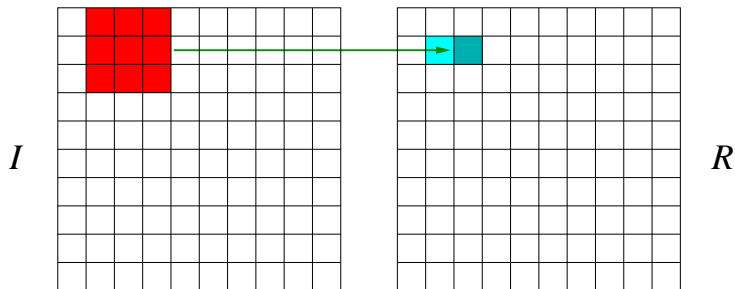
Filtre linéaire - convolution



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

Le principe de calcul de la convolution $R = I * K$

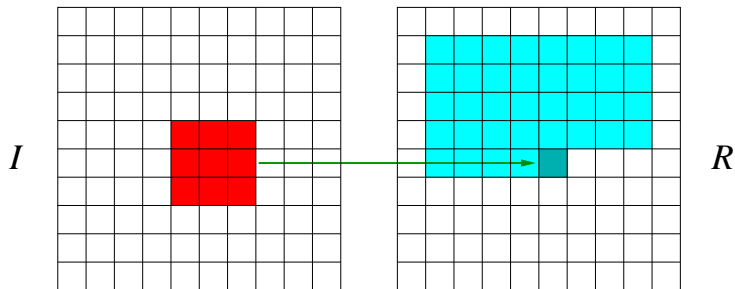
Filtre linéaire - convolution



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

Le principe de calcul de la convolution $R = I * K$

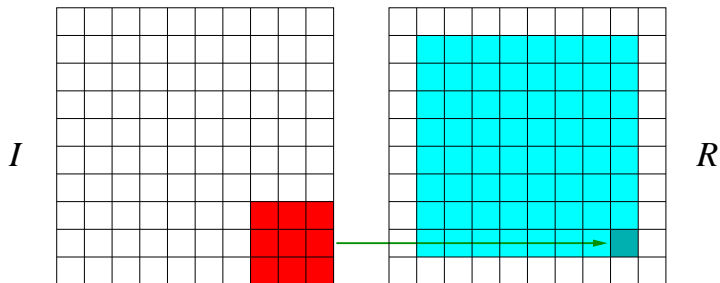
Filtre linéaire - convolution



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

Le principe de calcul de la convolution $R = I * K$

Filtre linéaire - convolution



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

Le principe de calcul de la convolution $R = I * K$

Filtre linéaire - convolution

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

La gestion du bord : différentes stratégies possibles

Filtre linéaire - exemple

Lissage (filtre passe-bas)

Filtre linéaire - exemple

Lissage (filtre passe-bas)



Image initiale

Filtre linéaire - exemple

Lissage (filtre passe-bas)

$$K = \frac{1}{3^2} \times \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \times (1 \ 1 \ 1)$$

Noyau du filtre *moyenne* ($n = 1$)

Filtre linéaire - exemple

Lissage (filtre passe-bas)

$$K = \frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Noyau du filtre *moyenne* ($n = 1$)

Filtre linéaire - exemple

Lissage (filtre passe-bas)



Application du filtre *moyenne* ($n = 1$)

Filtre linéaire - exemple

Lissage (filtre passe-bas)

$$K = \frac{1}{7^2} \times \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \times (1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1)$$

Noyau du filtre *moyenne* ($n = 3$)

Filtre linéaire - exemple

Lissage (filtre passe-bas)

$$K = \frac{1}{49} \times \begin{array}{|c|c|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

Noyau du filtre *moyenne* ($n = 3$)

Filtre linéaire - exemple

Lissage (filtre passe-bas)



Application du filtre *moyenne* ($n = 3$)

Filtre linéaire - exemple

Lissage (filtre passe-bas)

$$K = \frac{1}{4^2} \times \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \times (1 \ 2 \ 1)$$

Noyau du filtre *chapeau* ($n = 1$)

Filtre linéaire - exemple

Lissage (filtre passe-bas)

$$K = \frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Noyau du filtre *chapeau* ($n = 1$)

Filtre linéaire - exemple

Lissage (filtre passe-bas)



Application du filtre *chapeau* ($n = 1$)

Filtre linéaire - exemple

Lissage (filtre passe-bas)

$$K = \frac{1}{16^2} \times \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 3 \\ 2 \\ 1 \end{pmatrix} \times (1 \ 2 \ 3 \ 4 \ 3 \ 2 \ 1)$$

Noyau du filtre *chapeau* ($n = 3$)

Filtre linéaire - exemple

Lissage (filtre passe-bas)

$$K = \frac{1}{256} \times \begin{array}{|c|c|c|c|c|c|c|} \hline 1 & 2 & 3 & 4 & 3 & 2 & 1 \\ \hline 2 & 4 & 6 & 8 & 6 & 4 & 2 \\ \hline 3 & 6 & 9 & 12 & 9 & 6 & 3 \\ \hline 4 & 8 & 12 & 16 & 12 & 8 & 4 \\ \hline 3 & 6 & 9 & 12 & 9 & 6 & 3 \\ \hline 2 & 4 & 6 & 8 & 6 & 4 & 2 \\ \hline 1 & 2 & 3 & 4 & 3 & 2 & 1 \\ \hline \end{array}$$

Noyau du filtre *chapeau* ($n = 3$)

Filtre linéaire - exemple

Lissage (filtre passe-bas)



Application du filtre *chapeau* ($n = 3$)

Filtre non linéaire - exemple

Filtre médian

Pour chaque pixel $I(x, y)$ d'une image I de dimensions $L \times H$

Filtre non linéaire - exemple

Filtre médian

Pour chaque pixel $I(x, y)$ d'une image I de dimensions $L \times H$

considérer le voisinage de $I(x, y)$ à distance n ($n > 0$)

$$V_n = \{I(x + i, y + j) , -n \leq i, j \leq n , 0 \leq x + i < L , 0 \leq y + j < H\}$$

Filtre non linéaire - exemple

Filtre médian

Pour chaque pixel $I(x, y)$ d'une image I de dimensions $L \times H$

considérer le voisinage de $I(x, y)$ à distance n ($n > 0$)

$$V_n = \{I(x + i, y + j) \text{ , } -n \leq i, j \leq n \text{ , } 0 \leq x + i < L \text{ , } 0 \leq y + j < H\}$$

trier V_n et remplacer la valeur $I(x, y)$ par la médiane de V_n

Filtre non linéaire - exemple

Filtre médian

Pour chaque pixel $I(x, y)$ d'une image I de dimensions $L \times H$

considérer le voisinage de $I(x, y)$ à distance n ($n > 0$)

$$V_n = \{I(x + i, y + j) \text{ , } -n \leq i, j \leq n \text{ , } 0 \leq x + i < L \text{ , } 0 \leq y + j < H\}$$

trier V_n et remplacer la valeur $I(x, y)$ par la médiane de V_n

Filtre utilisé pour supprimer les pixels *parasites*

Filtre non linéaire - exemple

Filtre médian

Filtre non linéaire - exemple

Filtre médian



Image initiale

Filtre non linéaire - exemple

Filtre médian



Image bruitée

Filtre non linéaire - exemple

Filtre médian



Application du filtre *médian* ($n = 1$) sur l'image bruitée

Filtre non linéaire - exemple

Filtre médian



Application du filtre *moyenne* ($n = 1$) sur l'image bruitée