Traitement Numérique des Données

M1 – INF 2163

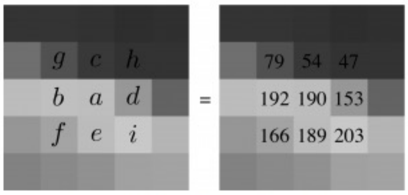
AIDN: Applications Interactives et Données Numériques

Sylvie Gibet

Filtrage des images

Filtrage par moyennage

- Enlever le bruit par des moyennes locales :
- remplacer la valeur *a* de chaque pixel par la moyenne de *a* et des 8 valeurs *b*,*c*,*d*,*e*,*f*,*g*,*h*,*i* des 8 pixels voisins de *a*.



Exemple d'un voisinage 9 pixels

Image originale bruitée

- On remplace a par (b+c+d+e+f+g+h+i)/9

3

3

Filtrage par moyennage

Image bruitée

Moyenne sur 9 pixels

- Introduit du flou dans l'image

4

4

Filtrage par convolution

- Convolution : qu'est-ce que c'est ?

  – Filtrage : application d'un produit de convolution à un signal d'entrée *x*, en sortie on a *y* :

  y = x * h

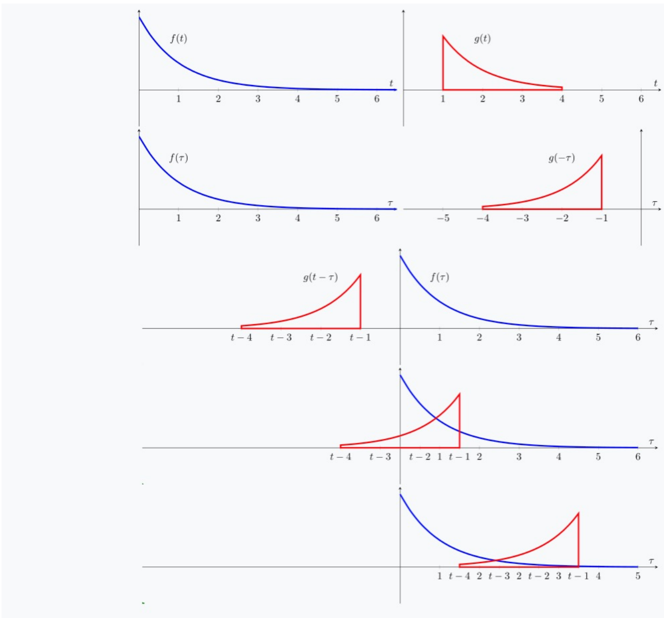
  avec * : produit de convolution

  h : fonction qui effectue le filtrage

$$y(n) = \sum_{k=-\infty}^{+\infty} h(k).x(n-k)$$

5

5



https://en.wikipedia.org/wiki/Convolution

6

6

Filtrage par convolution

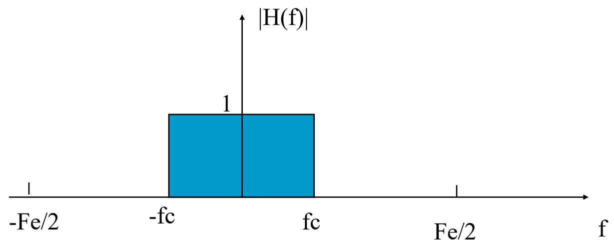
- Convolution : à quoi ça sert ?
  - Le produit de convolution (filtrage) est utilisé en traitement du signal.
  - On peut ainsi construire des filtres passe-bas, passe-bande, passe-haut

7

7

Filtre passe-bas

- Filtre qui élimine les fréquences supérieures à la fréquence de coupure f_c du filtre.

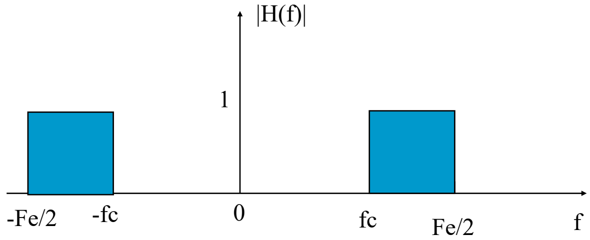


S. Gibet 8

8

Filtre passe-haut

- Filtre qui ne laisse passer que les fréquences inférieures à la fréquence de coupure f_c du filtre.

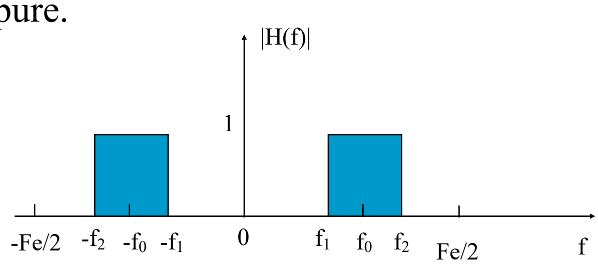


S. Gibet 9

9

Filtre passe-bande

- Pour ce type de filtre, il est nécessaire de définir une fréquence de coupure basse f_{c1} et une fréquence de coupure haute f_{c2} . Ce filtre conserve toutes les fréquences situées entre les deux fréquences de coupure.

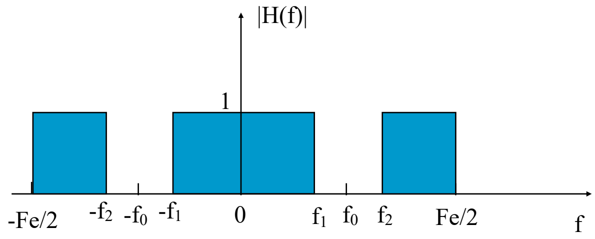


S. Gibet 10

10

Filtre coupe-bande

- ou réjecteur de bande ; c'est le filtre complémentaire du passe-bande : il élimine les fréquences intermédiaires entre les fréquences de coupure basse et haute.



S. Gibet 11

11

Filtrage par convolution

- Compléments de cours sur les filtres linéaires sur Moodle (LTI ou Linear Time Invariant filters)
- Application d'un filtre à une image :
  produit de convolution : $f * h$
- On dit que l'on convolue l'image *f* et le noyau *h* (kernel)

12

12

Filtrage d’image par convolution

- Exemple :

$$h = \begin{pmatrix} 1 & 2 & 1 \\ 2 & \boxed{1} & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

$$f = \begin{pmatrix} 2 & -1 & 0 \\ 4 & -2 & -1 \\ 1 & 2 & 3 \end{pmatrix}$$

1*2+2*(-1) +1*0 +
2*4+1*(-2)+2*(-1) +
1*1+2*2+1*3 = ?

13

Filtrage d’image par convolution

- Exemple :

$$h = \begin{pmatrix} 1 & 2 & 1 \\ 2 & \boxed{1} & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

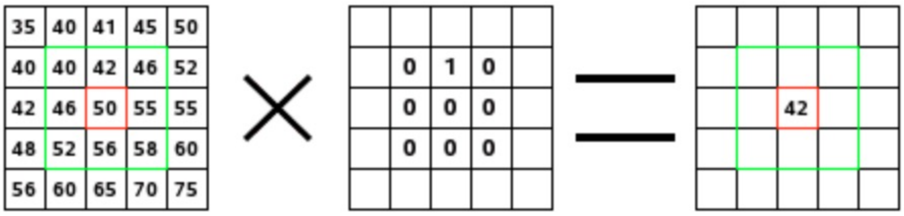
$$f = \begin{pmatrix} 2 & -1 & 0 \\ 4 & -2 & -1 \\ 1 & 2 & 3 \end{pmatrix}$$

$\boxed{12}$

1*2+2*(-1) +1*0 +
2*4+1*(-2)+2*(-1) +
1*1+2*2+1*3 = 12

14

Filtrage d'image par convolution

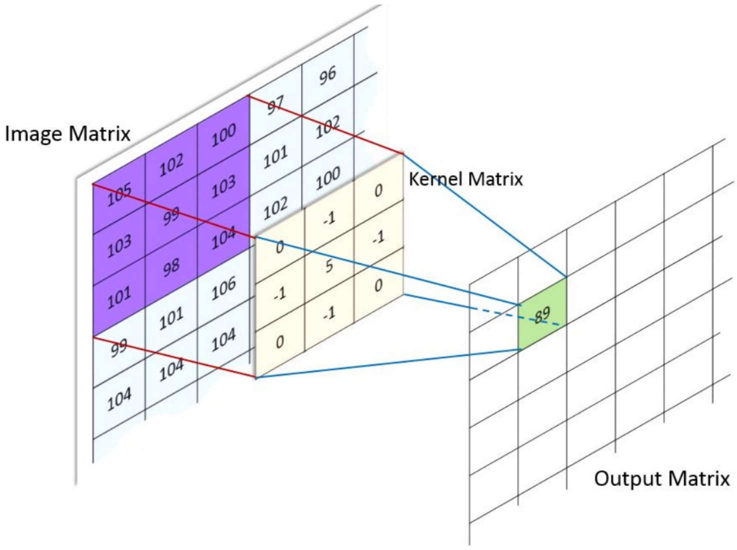


Filtrage d'image par convolution

$$g(x,y) = \omega * f(x,y) = \sum_{dx=-a}^{a} \sum_{dy=-b}^{b} \omega(dx,dy) f(x+dx, y+dy),$$

- g : image filtrée
- f(x,y) : image originale
- ω est le filtre (kernel)

Filtrage d’image par convolution

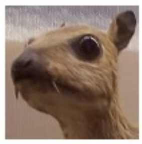


17

17

Filtrage d’image par convolution

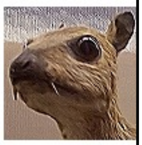
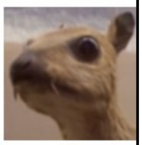
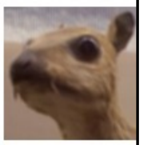
- En fonction des éléments de h, on réalise des opérations différentes : f * h
- Remarque : la taille de h peut varier : on prend plus ou moins de points au voisinage de chaque pixel de l’image à transformer.
- Détection de contour

	h	
Original	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
Edge-Detect	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	

18

Filtrage d'image par convolution

- En fonction des éléments de h, on réalise des opérations différentes : f * h
- Augmente le contraste
- Flouté

	h	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Blur*	$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	

19

19

Convolution : traitement des bords

- **Plusieurs méthodes**
  - Agrandissement : les pixels de bordure sont étendus conceptuellement aussi loin que nécessaire pour permettre la convolution.
  - Coupure : tout pixel qui demanderait des valeurs en dehors de l'image est enlevé -> peut résulter en une image de sortie plus petite.
  - Les valeurs de bord de l'image sont prises du coin opposé de l'image.
- **Normalisation**
  - Revient à normaliser chaque élément du noyau par la somme de tous les éléments du noyau. Cela permet d'assurer que les valeurs des pixels de l'image de sortie sont de la même amplitude relative que celles de l'image d'entrée.

20

20

Convolution – lien avec la TFD

- Retour sur la Transformée de Fourier (TFD)

$$y[n] = x[n] * h[n] \Leftrightarrow Y[k] = X[k].\ H[k]$$

avec X = fft(x), H = fft(h), Y == fft(y)

- Pourquoi passer dans le domaine fréquentiel ?

21

21

Quand utiliser la transformée de Fourier ?

- Considérations liées à la complexité algorithmique !

Soit une image de taille nxn = n^2

- Si le filtre est en m^2 (taille du kernel h), alors le produit de convolution s'effectue en : $m^2.n^2$ opérations

- Algorithmes fft rapides :
  complexité = t log t avec t : taille de l'image
  Si t = n^2, alors : complexité = $n^2.\log(n^2)$ -> $2.n^2.\log(n)$
  algo en $n^2.\log(n)$

22

22

Quand utiliser la transformée de Fourier ?

- Comparaisons convolution versus Fourier :
  - Si le filtre est petit : taille $m^2 < \log n$

    $m^2.n^2 < n^2 .\log n$

Dans ce cas, il est préférable de choisir la convolution et pas Fourier

  - Si le filtre est grand : taille $m^2 > \log n$

    $m^2.n^2 > n^2 .\log n$

  passer par Fourier !

23

23

Quand utiliser la transformée de Fourier ?

- Exemple :
  - image 1024 x 1024, kernel h de taille 3x3
  - Kernel h de taille 10x10

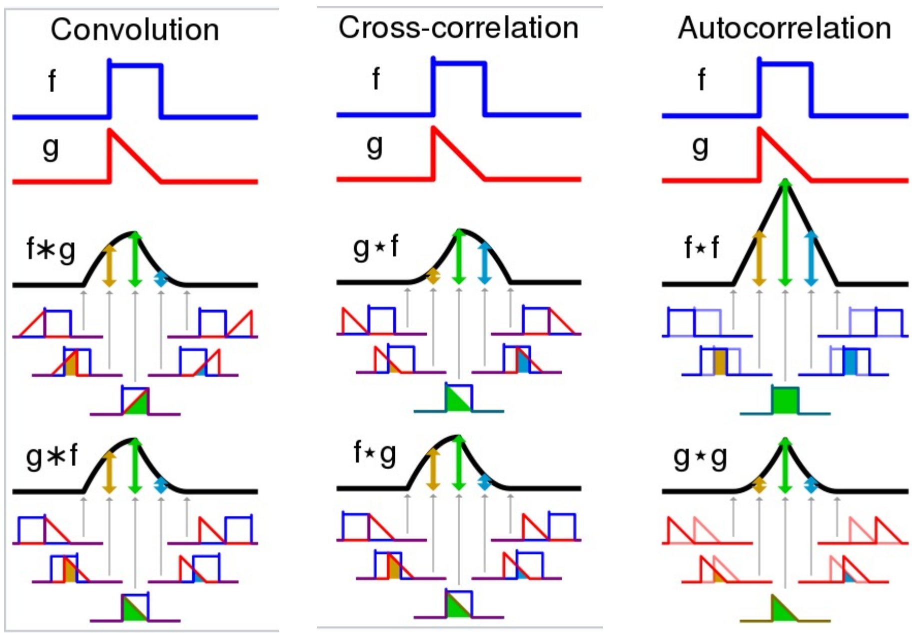
24

24

Compléments

25

25



26

26

Autres filtres : les filtres Gaussiens

- Filtres non linéaires définis par :

$$g_\sigma(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

- On pourrait définir à partir de là le masque, la convolution…

$$\begin{aligned} \mathcal{F}(g_\sigma)(u,v) &= \frac{1}{2\pi\sigma^2} \iint_{\mathbb{R}^2} exp\left(\frac{i\pi}{\sigma^2}(x^2+y^2)(ux+uv)\right) dx\, dy \\ &= exp\left(\frac{-\sigma^2(u^2+v^2)}{2}\right) \end{aligned}$$

27

27

Filtres gaussiens

- Donc on peut directement appliquer la matrice de convolution :

$$(g_\sigma * f)(x,y) = \mathcal{F}^{-1}\left(\mathcal{F}(f) \cdot \mathcal{F}(g_\sigma)\right)$$

- Eventuellement, il faudra compléter f (image)

28

28