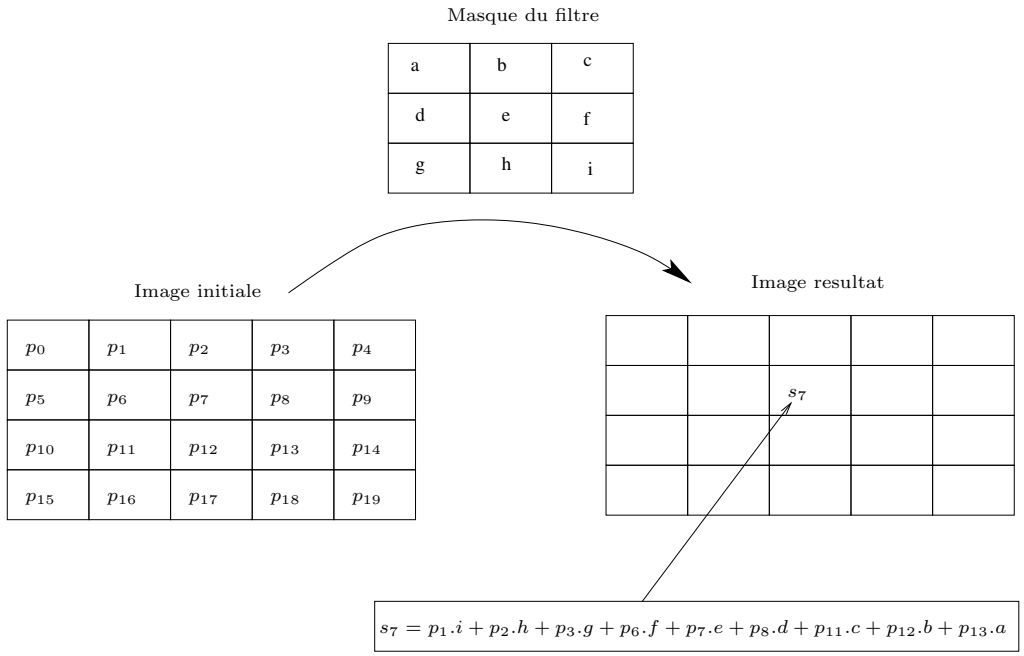
Traitement de l'image : Filtrage I

Introduction

Le principal objectif du filtrage est de pré-traiter une image afin que l'image filtrée soit plus adaptée que l'image originale pour une application spécifique. Nous allons étudier dans ce cadre les méthodes s'appliquant dans le domaine spatial (tout en illustrant, lorsque c'est possible, le lien avec le domaine fréquentiel). Les méthodes spatiales sont des procédures qui opèrent directement sur les pixels de l'image. D'une manière générale ces techniques mettent en œuvre une convolution discrète 2D par un masque h de l'image de départ I, comme indiqué sur la figure ci-dessous.



L'opération qui, dans le domaine temporel, se traduit par un produit de convolution, s'exprime aussi par un simple produit entre le gain fréquentiel du filtre h et la transformée de Fourier de l'image e :

$$s(i,j) = e(i,j) * h(i,j) = \sum_{n=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} e(n,p)h(i-n,j-p) \quad (1)$$

$$= \mathcal{F}^{-1}\left(\mathcal{F}(e(i,j)) \cdot \mathcal{F}(h(i,j))\right) \quad (2)$$

C'est pourquoi l'étude du gain fréquentiel du masque utilisé (correspondant à la transformée de Fourier de h) permet souvent de caractériser la nature du filtre (filtre passe-bas, filtre passe-haut). Nous allons maintenant étudier différents filtres.

1 Transformée de Fourier

1.1 Rappel

L'image ayant un nombre fini de pixels (c'est un signal à support borné), on utilise la transformée de Fourier discrète (TFD) définie par

$$F(u,v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) e^{-i\frac{2\pi um}{M}} e^{-i\frac{2\pi vn}{N}}$$

où

— $F(u, v)$ est la transformée de Fourier de l'image $f(m, n)$;
— M et N sont les dimensions de l'image ;
— les deux variables u et v représentent les fréquences spatiales de l'image selon les directions x et y respectivement.

La fonction $F(u, v)$ est en général un nombre complexe, même si $f(m, n)$ est un nombre réel : elle possède donc un module et une phase et on peut choisir de représenter l'un ou l'autre. Nous ne nous intéresserons ici qu'au module.

1.2 Implémentation

Dans cette section, on applique la TF et on l'affiche. Ce dernier point demande quelques manipulations pour que le rendu visuel soit bon. Vous aller avoir besoin de l'import suivant :

```
# Visionneuse externe interactive
%matplotlib

#imports
import numpy as np              # Tableaux
import scipy.ndimage as scp     # Correlation et convolution
import matplotlib.pyplot as plt # Visionneur et tracés
from matplotlib.colors import LogNorm # echelle logarithmique
import skimage
from scipy import fftpack       # Fourier 2D
```

Deux commandes qui peuvent être utiles :
— Pour importer en niveaux de gris, option `as_gray`
— pour choisir l'échelle logarithmique lors de l'affichage, option `norm=LogNorm(vmin=5)`

1. **importer vos images.**
2. Peut-on appliquer la TF à une image couleur ? **Appliquer la TF avec** `fftpack.fft2`. Quel est le type du résultat ?
3. **Afficher bêtement le module et l'échelle des couleurs.** Que déduire de l'échelle des couleurs ?
4. **Afficher dans la bonne échelle.** D'après les valeurs, où sont les basses fréquences ?
5. **utiliser** `fftshift` **sur la transformée de Fourier et afficher**
6. **Pour chacune de ces images réelles crepis, laine, CryoEM, afficher leur TFD et interpréter.**

2 Filtre « lisseur »

On se place dans le cas où un bruit additif se superpose à l'image. Dans ce cas, une méthode simple pour « lisser » l'image consiste à calculer une moyenne locale en (n, m) : chaque pixel est remplacé par la moyenne pondérée de ses voisins. Un masque typique correspondant à cette opération se définit par :

$$h = \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Étudions tout d'abord l'influence de tels filtres sur des images non bruitées.

7. **Charger l'image *bois.jpg* . Visualiser son spectre .**
8. **Créer deux masques** `h1 et h2` **effectuant une moyenne locale sur un voisinage** 3×3 **et** 9×9 **(utiliser la commande** `numpy.ones`).
9. **Appliquer ces filtres sur l'image *bois* en tapant en utilisant** `scipy.signal.convolve2d`.
10. **Visualiser les résultats et comparer à l'image originale** (vous pourrez pour cela afficher une « vue en coupe »de l'image)

Nous appliquons maintenant ces 2 filtres sur une image bruitée par un bruit gaussien que vous allez fabriquer.

11. **Créer l'image *boisbruit* avec la commande** `skimage.util.random_noise`**, paramètres : moyenne nulle et variance à 0,01. Visualiser l'image et son spectre.**
12. **Appliquer les 2 filtres sur l'image *boisbruit*.**
13. **Visualiser les résultats.**

3 Filtre non linéaire

Le principal inconvénient des filtres linéaires (basés sur le produit de convolution) est que la réduction de bruit s'accompagne d'un étalement des transitions entre régions. Cette difficulté peut être surmontée par l'utilisation des filtres non linéaires, comme par exemple le filtre médian. Comme son nom l'indique, le filtre médian sélectionne le pixel de la fenêtre d'analyse ayant la valeur médiane. Ce filtre va être insensible à la présence de valeurs aberrantes si elles sont en nombre raisonnable, ce qui est le cas d'un bruit impulsionnel (de densité de probabilité $f(a) = Ce^{-\alpha|a|)}$ avec α petit).

14. **Créer l'image *boisbruit2*. Cette image correspond à l'image *bois* corrompue par un bruit impulsionnel** (`poivre et sel`) **avec** $\alpha = 0.001$.
15. **Visualiser l'image.**
16. **Programmer le filtre médian dans un fonction avec comme paramètres l'image et la taille de la fenêtre (attention à la gestion du bord).**
17. **Appliquer le filtre** $h = \frac{1}{9}\begin{pmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{pmatrix}$ **sur l'image *boisbruit 2*.**
18. **Visualiser et comparer les 2 résultats.**
19. **Commenter les principales différences de ces filtres face au bruit impulsionnel.**
20. **modifier le filtre médian pour que le pixel ne soit modifié que quand sa valeur est aberrante**

Notons que pour des bruits à distributions assez concentrées (Gaussien) les performances du filtre médian sont assez faibles.

19. **Recharger l'image *boisbruit* (bruit gaussien). Appliquer le filtre médian.**
20. **Commenter le résultat.**

4 Microscopie électronique

Le principe du microscope électronique est de faire passer un flux d'électrons au travers de lentilles magnétiques puis d'une fine tranche de l'échantillon à observer. On obtient alors une image I grossie de l'échantillon (image cryoEM).

Cependant les lentilles magnétiques ne sont pas parfaites et elles produisent des aberrations qui doivent être corrigées (en plus du bruit inhérent au flux électronique). La transformée de Fourier J de l'image I permet de visualiser les aberrations.

22. **Afficher la TdF de l'image cryoEM**. Quels sont les éléments notables de l'image ? Est-ce attendu ?

On modélise les aberrations comme un filtre linéaire F

23. **Afficher la colonne centrale de la TdF.** Peut-on récupérer le profil de filtre des aberrations magnétique à patir de cette ligne ?
24. **Appliquer une moyenne sur la TdF, puis réafficher la colonne**
25. **Calculer et appliquer le filtre correspondant.**
26. **Calculer une moyenne (par cercle) pour le profil.**
27. **Calculer un filtre correctif (attention au valeurs nulles, filtre à appliquer à la TdF complexe, pas seulement à son module)**