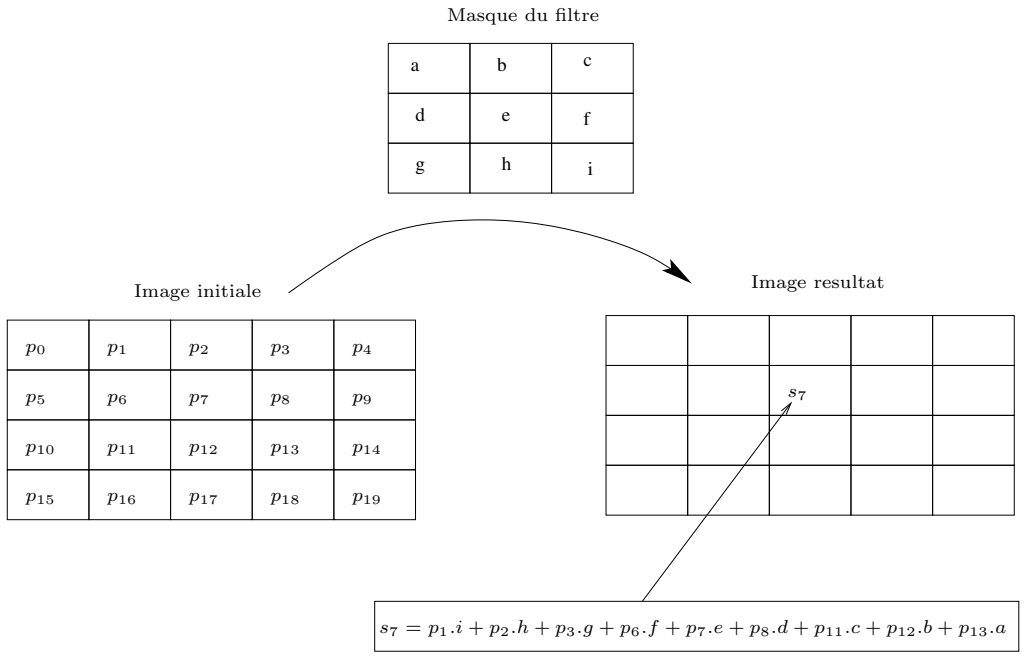
Traitement de l'image : Filtrage I

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 :

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

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

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 Fonctions scilab

Une image ouverte avec la fonction imread de Scilab renvoie une variable de type uint8 (entier codé sur huit bits). Avant de traiter cette image, il est nécessaire de la transformer en double (réel à virgule flottante codé sur 64 bits) à l'aide de la fonction double. Les fonctions suivantes peuvent être utilisées dans votre code. Reportez-vous à l'annexe Scilab pour des fonctions plus classiques.

imread charger une image.

double conversion en réel à virgule flottante (sur 64 bits).

imshow(I) afficher une image I (attention au type d'entrée)

fftshift déplace la composante de fréquence nulle au centre du spectre.

fft2 transformée de Fourier 2D.

round, ceil, floor arrondis.

rand('normal') bruit gaussien centré d'écart-type (standard deviation) 1.

im2bw convertit une image quelconque en image binaire.

find cherche les coordonnées des éléments non-nuls d'une matrice.

1.3 Implémentation

1. **Pour une image en niveau de gris, traduire en français la commande suivante** `(log(1+abs(fftshift(fft2(Im)))))`, puis afficher le résultat.
2. **expliquer le rôle de chaque fonction.**
3. **en faire une fonction** `fouriervisu`**,**
4. **Pour chacune de ces images réelles crepis, laine, CryoEM, afficher leur TFD et interpréter.**
5. **Tester la transformée de Fourier inverse avec des images (de fréquences) simples.**

1.4 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. On modélise les aberrations comme un filtre linéaire F

5. Proposer une manière simple d'estimer le profil du filtre F en tenant compte du bruit (m'appeler).
6. **calculer et appliquer le filtre correspondant.**

2 Filtre « lisseur »

2.1 Moyenne locale

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 :

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

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

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

```
plot2d(bois(70,:)); plot2d(res(70,:));
```

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

10. **Créer l'image *boisbruit* avec la commande** `imnoise`**, paramètres : moyenne nulle et variance à 0,01. Visualiser l'image et son spectre.**
11. **Appliquer les 2 filtres sur l'image *boisbruit*.**
12. **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).

13. **Créer l'image *boisbruit2*. Cette image correspond à l'image *bois* corrompue par un bruit impulsionnel** (`poivre et sel`) **avec** $\alpha = 0.001$.
14. **Visualiser l'image.**
15. **Programmer le filtre médian dans un fonction avec comme paramètres l'image et la taille de la fenêtre (attention à la gestion du bord).**
16. **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*.**
17. **Visualiser et comparer les 2 résultats.**
18. **Commenter les principales différences de ces filtres face au bruit impulsionnel.**

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.**

Références

[1] H.P. Kramer et J.B. Bruckner, *Iterations of a non-linear transformation for enhancement of digital images*, Pattern Recognition,1975,**7**,pp 53–58,

[2] P. Perona et J. Malik, *Scale-space and edge detection using anisotropic diffusion*, IEEE Trans. Pat. Anal. and Machine Intel., 1990, **12**, pp. 629–639,