

TP 3

Bruit dans l'Image et Lissage

But

- Bruiter une image avec un bruit additif « gaussien » ou « poivre et sel », ou un bruit multiplicatif « speckle »
- Filtrer l'image dans le domaine spatial
 - Filtres moyennneur et gaussien
 - Filtre median
- Filtrer l'image dans le domaine Fréquentiel

I. Bruitage d'Image

On bruite l'image par :

- | | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> - Bruit additif <ul style="list-style-type: none"> o Gaussien o Poivre et sel | <ul style="list-style-type: none"> - Bruit multiplicatif (de Speckel) |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|

Commandes Matlab

Permettent de bruite l'image im par un :

`imB = imnoise(im, 'gaussian', m, var)` - Bruit gaussien de moyenne m et de variance var

`imB = imnoise(im, 'salt & pepper', d)` - Bruit « poivre et sel » de taux d

`imB = imnoise(im, 'speckle', v)` - Bruit multiplicatif « Speckel » de variance v
Le résultat est affecté à la variable imB

Manipulation

1. Lire l'image à niveau de gris 'cameraman.tif', affecter-la à la variable **im**

2. Bruit Gaussien

- a) Bruiter là avec un bruit Gaussien de moyenne **m** nulle et de variance **var= 0.001**. Utiliser la commande « **imnoise** ». Afficher le résultat.

- b) Faire varier la variance par ces valeurs : 0.002, 0.005, 0.01, 0.02, 0.05. Comparer.
- c) On peut lancer la commande « **imnoise** » sans les paramètres **m** et **var**. Tester et afficher. Aller vers le Help pour connaître les valeurs par défaut.
- d) Bruiter l'image autrement en utilisant les lignes de commande :

```
bruit=(sqrt(var)*randn(size(im))+m)*255;
imBB=double(im)+bruit;
```

- Afficher le résultat et comparer avec celui obtenu en question 2.a.
Pour l'affichage coder **imBB** sur 8bits via la commande : **imshow(uint8(imBB))**

3. Bruit Poivre et Sel

- a) Bruiter l'image **im** avec un bruit « Poivre et Sel » de taux **d=0.05** (5% des pixels de l'image sont des points de bruit). Utiliser la commande. Afficher le résultat.
- b) Faire varier le taux **d** par ces valeurs : 0.5% 1%, 5%, 8% 10% 15% 20%
. Comparer.
- c) On peut lancer la commande « **imnoise** » sans le paramètre **d**. Tester et afficher. Aller vers le Help pour connaître la valeur par défaut prise pour **d**.
- d) Bruiter l'image autrement en utilisant les lignes de commande :

```
imBB=im;
x = rand(size(im)); % matrice aléatoire de distribution uniforme
imBB(x < d/2) = 0; % les points 0 < x < d/2 : sont mis à zéro (poivre)
imBB(x >= d/2 & x<d) =255;% les points d/2 < x < d :sont mis à 255 (sel)
```

- Afficher le résultat et comparer avec celui obtenu en question 3.a.
Pour l'affichage coder **imBB** sur 8bits via la commande : **imshow(uint8(imBB))**

4. Bruit Multiplicatif de Speckle

- a) Bruiter l'image **im** avec un bruit « Speckle » de variance **v=0.01**. Utiliser la commande « **imnoise** ». Afficher le résultat.
- b) Bruiter l'image autrement en utilisant les lignes de commande :

```
imBB = double(im) + sqrt(12*v)*double(im).*(rand(size(im))- .5);
```

En effet, le bruit Speckle est tel un bruit de quantification :

- de puissance $= \frac{q^2}{12}$, donc le pas serait $q = \sqrt{12 \cdot p}$,
- et d'amplitude variant entre les valeurs extrêmes $\frac{-q}{2}$ et $\frac{+q}{2}$, autrement dans la plage : $q \times [\frac{-1}{2} \dots \frac{+1}{2}]$
- Afficher le résultat et comparer avec celui obtenu en question 4.a. (na pas oublier de coder sur 8bits avant l'affichage).

2. Filtrage du Bruit dans le Domaine Spatial

- Pour dé-bruiter l'image, on utilisera les filtres :
 - Moyenneur
 - Gaussien
 - Median.

Commandes Matlab

<code>h=fspecial('average',N);</code>	Pour créer un filtre de lissage : <ul style="list-style-type: none">- Moyenneur de taille $N \times N$
<code>h=fspecial('gaussian',N, sigma);</code>	Gaussien de variance sigma et de taille $N \times N$
<code>B =conv2(A, h,'same');</code>	Pour filtrer l'image A avec le filtre h. Il s'agit en fait d'une opération de convolution 2D.
<code>B =filter2(h, A,'same');</code>	<ul style="list-style-type: none">- Dans les 2 premières, A et B sont de type réel- Alors que dans la 3^{ème}, A et B sont entier.
<code>B =imfilter(A,h,'same');</code>	(attention à l'ordre des paramètres d'entrée A et h)
<code>B =medfilt2(A,[n m]);</code>	Pour effectuer un filtrage Median sur un voisinage de taille $n \times m$ dans l'image A

Manipulation

1. Lire l'image à niveau de gris '`cameraman.tif`', affecter-la à la variable `im`
2. Bruiter la avec un bruit gaussien de moyenne nulle et de variance 0.01. Appeler l'image bruitée `imB`.

3. Filtre Moyenneur

- a) Créer un filtre moyenneur `h` de taille 3×3 .
- b) Appliquer-le sur `imB`, par 3 façons : "`conv2`", "`filter2`" ou "`imfilter`". Afficher et comparer. Commenter également les effets de ce filtre sur l'image bruitée.
- c) Varier la taille du filtre moyenneur : 5×5 , 7×7 , 9×9 . Commenter.

4. Filtre Gaussien

- a) Créer un filtre Gaussien `h` de taille 3×3 et de variance 0.5.
- b) Appliquer-le sur `imB`.
 - Afficher et commenter les effets de ce filtre.
 - Commenter avec le résultat obtenu avec le filtre moyenneur.
- c) Pour une taille du filtre de 3×3 , varier les valeurs de la variance, `sigma=0.1`, `0.5`, `1`, `2`, `2.5`. Commenter.
- d) Pour une variance `sigma=0.5`, varier la taille du filtre Gaussien : 5×5 , 7×7 , 9×9 . Commenter.

5. Bruiter l'image `im` avec un bruit « Poivre et Sel » de taux 5%. Appeler l'image bruitée `imB`.

6. Filtre Median

- a) Créer un filtre Median `h` de taille 3×3 .
- b) Appliquer-le sur `imB`. Afficher et commenter les effets de ce filtre sur l'image bruitée.
- c) Varier la taille du filtre : 5×5 , 7×7 , 9×9 . Commenter.
- d) Pour un filtre `h` de taille 3×3 , varier le taux de bruit : 0.01, 0.05, 1%, 5%, 10%, 15%, 20%. Commenter.

3. Filtrage du Bruit dans le Domaine Fréquentiel

Commandes Matlab

Pour calculer la FFT en 2D, on utilisera les commandes :

<code>Xf=fft2(A)</code>	Permet de calculer la FFT en 2D de la matrice A
<code>XXf=fftshift(Xf)</code>	Permet de réarranger Xf de façon à le centrer autour de la fréquence <code>f=0</code> .
<code>module = abs(Xf)</code>	Permet de calculer le module de Xf.

Manipulation

1. Lire l'image à niveau de gris '`rice.tif`', affecter-la à la variable `im`
2. Bruiter la avec un bruit gaussien de moyenne nulle et de variance 0.008. Appeler l'image bruitée `imB`.
3. Calculer les FFTs des images originale et bruitée, et leurs modules puis afficher-les.
 - Commenter les spectres obtenus.

```
imf=fftshift(fft2(im));
imBf=fftshift(fft2(imB));
figure,
subplot(131); imshow(abs(imf/255),[0 255]);
title('fft de l''image originale')
subplot(132); imshow(abs(imBf/255),[0 255]);
title('fft de l''image bruitée')
colormap(1-gray),
```

4. Méthode 1 de Filtrage :

Idée : On admet que le bruit est de puissance faible / à l'image. Donc dans le spectre d'amplitude, les composantes fréquentielles de magnitudes faibles correspondraient au bruit.
→ Filtrer le bruit, revient à éliminer ces composantes. (il faudra établir une référence de comparaison, qu'on appellera seuil)

Procédure : Pour effectuer le filtrage en fréquence on suivra les étapes suivantes :

- Calculer la FFT de l'image bruitée imB $\rightarrow imBf$
- Calculer le module de $imBf$ $\rightarrow mod$
- Poser une valeur à un seuil de filtrage, $\rightarrow seuil$
 - on prendra : $seuil = 1.2 \times moyenne(mod)$;
 sous Matlab, on écrira : **`seuil=1.2*mean2(mod)`** ;
- Supprimer les composantes de $imBf$ dont les magnitudes sont inférieures à seuil $\rightarrow imDBf$
`imDBf=imBf` ;
`imDBf(mod<=seuil)=0` ;
- Reconstruire, par IFFT, l'image associée au spectre filtré = image débruitée $\rightarrow imDB$
`imDB=ifft2(fftshift(imDBf))` ;
- Afficher toutes les images (originale, bruitée, et débruitées ainsi que leurs spectres)
 pour l'image dé-bruitée, utiliser **`imshow(imDB/255)`** ;

- Observer les spectres obtenus et commenter
- Faire varier les valeurs du seuil : $moyenne(mod) * [1, 1.1, 1.15, 1.2, \dots 1.5]$.
 Commenter.

5. Méthode 2 de Filtrage :

Idée : On admet que le bruit se trouve dans les fréquences élevées

\rightarrow Filtrer le bruit, revient à éliminer les composantes du spectre se trouvant aux fréquences élevées (il faudra établir une référence de comparaison des fréquence, qu'on appellera **fseuil**)

Procédure : Pour effectuer le filtrage en fréquence on suivra les étapes suivantes :

- Calculer la FFT de l'image bruitée imB $\rightarrow imBf$
- Calculer l'axe des fréquences fx et fy
`[m,n]=size(im)` ;
`[fx,fy]=meshgrid(-m/2:m/2-1,-n/2:n/2-1)` ;
- Poser une valeur à un seuil de filtrage pour les fréquence, $\rightarrow fseuil$
`fseuilx=max(fx(:))` ; **`fseuily=max(fy(:))`** ;
`fseuil=max(fseuilx, fseuily)*0.5` ;
- Supprimer les composantes de $imBf$ au niveau des fréquences $> fseuil$ $\rightarrow imDBf$
`imDBf(abs(fx)>fseuil)=0` ;
`imDBf(abs(fy)>fseuil)=0` ;
- Reconstruire, par IFFT, l'image associée au spectre filtré = image débruitée $\rightarrow imDB$
`imDB=ifft2(fftshift(imDBf))` ;
- Afficher toutes les images (originale, bruitée, et débruitées ainsi que leurs spectres)
 pour l'image dé-bruitée, utiliser **`imshow(imDB/255)`** ;