De la photographie numérique
à la photographie computationnelle

CES8AH - École des Mines de Nancy

Séance 4

Frédéric Sur

L'objectif de cette séance est d'expérimenter le phénomène d'aliasing.

1 Vérification expérimentale du phénomène d'aliasing

On considère le signal analogique (continu) dépendant du temps suivant :

$$y(t) = \sin(2\pi t) + 0.3\sin(40\pi t).$$

Quelle est la plus haute fréquence présente dans ce signal ? Quel est la fréquence d'échantillonnage minimale pour satisfaire la condition de Nyquist ?

La fonction Matlab (à sauvegarder dans un fichier `signal.m` dédié) :

```
function [T,Y]=signal(fe)
T=[0:1/fe:5];
Y=sin(2*pi*T)+0.3*sin(40*pi*T);
```

calcule le signal `Y` échantillonné à la fréquence `fe` ainsi que les instants d'échantillonnage `T`, sur 5 périodes.

Tracez le graphe du signal échantillonné pour différentes valeurs de `fe`, ainsi que le spectre du signal échantillonné en portant la fréquence de la composante en abscisse. Diminuez la valeur de la fréquence d'échantillonnage, mais en la gardant plus grande que la fréquence de Nyquist. Observez que vous pouvez vérifier la fréquence de la composante $0.3\sin(40\pi t)$ du signal y en comptant le nombre de maxima locaux sur une période de la composante principale $\sin(2\pi t)$.

Que se passe-t-il lorsque la fréquence d'échantillonnage est égale à la fréquence de Nyquist ?

Et lorsque la fréquence d'échantillonnage est inférieure à la fréquence de Nyquist ? Comptez les maxima locaux : que pensez-vous du résultat ?

2 Sous-échantillonnage d'images

Question préliminaire : charger l'image `synthetique.tif` et visualisez-là avec `imshow`. Observez l'aspect de l'affichage lorsque vous redimensionnez la fenêtre à la souris.

On suppose disposer d'images de taille $N \times N$ pixels, que l'on veut réduire à une taille $N/2 \times N/2$ pixels. On parle de sous-échantillonnage de facteur 2. Comme le sur-échantillonnage, le sous-échantillonnage est une opération classique pour la diffusion de contenus sur les canaux numériques type TNT, Blu-Ray, DVD... et des artefacts sont visibles lorsque l'opération est mal faite !

La méthode naïve consiste à procéder à une décimation : on garde un pixel sur deux (resp. quatre) par ligne et colonne. La fréquence d'échantillonnage dans chaque direction est alors $N/2$ *cycles par image*.

Pour être en accord avec la condition de Shannon-Nyquist, cette fréquence d'échantillonnage doit être au moins le double de la plus haute fréquence f (également exprimée en *cycles par image* présente dans l'image : $N/2 > 2f$.

Rappel pour ce qui suit : le coefficient de Fourier discret $X_{n,m}$ $(-N/2+1 \leqslant n, m \leqslant N/2)$ est une approximation du coefficient de Fourier de l'image continue correspondant aux fréquences n cycles par image (horizontalement) et m cycles par image (verticalement), cf séance 2.

Commencez par visualiser les images fournies (dans l'ordre : `WTC.tif`, `boat.tif`, `synthetique.tif`) et leur spectre d'amplitude.

2.1 Méthode 1 : décimation brutale

Testez la méthode par décimation pour diviser par 2 la taille des images.
Indication : que fait `Ima2=Ima(1:2:end,1:2:end)` ?

D'après la théorie, comme les images originales ont des composantes de fréquences plus grandes que $N/4$, on doit observer de l'aliasing.
Comment se traduit cet aliasing sur les images décimées ? Et sur les spectres ?

2.2 Méthode 2 : filtrage passe-bas idéal

Une première idée pour éliminer l'aliasing serait de mettre à zéros les coefficients des fréquences plus grandes que $N/4$, puis de procéder à la décimation sur la transformée de Fourier inverse.

Indication : On créera le « masque » suivant à appliquer à la transformée de Fourier de l'image originale pour mettre à zéro les hautes fréquences

```
>> masque=zeros(512,512);
>> masque(129:385,129:385)=1;
```

On définit alors la transformée de Fourier filtrée par :

```
>> fftimage_m=ifftshift(fftshift(fftImage).*masque);
```

Constatez l'amélioration au niveau de l'aliasing. Constatez également l'apparition du phénomène de Gibbs le long des contours des images, consécutif à l'annulation brutale des hautes fréquences.

2.3 Méthode 3 : filtrage passe-bas « doux »

La méthode précédente consiste à multiplier la transformée de Fourier par un échelon (le masque). Cela revient à effectuer la convolution de l'image originale par la transformée de Fourier inverse du masque (c'est un sinus-cardinal). Observez cette transformée de Fourier inverse :

```
>> figure, imagesc(fftshift(ifft2(ifftshift(masque),'symmetric')))
>> colorbar
```

Vous constatez (en zoomant avec la loupe de la figure) qu'elle décroît très lentement (ce qui est logique, car le masque n'est pas continu), donc a des effets "à longue distance" qui se traduisent par des effets de Gibbs sur les contours dans l'image.

Pour améliorer la situation, il faudrait utiliser un masque annulant les hautes fréquences, mais dont la transformée de Fourier inverse décroît plus rapidement (et par conséquent ce masque doit être assez régulier). On va lisser le masque précédent par convolution avec une fonction Gaussienne 2D :

```
>> g=fspecial('gaussian',[96,96],16);
>> masque_b=conv2(masque,g,'same');
```

Comparez la forme du masque de la méthode précédente et celle de `H`.

Vérifiez que le filtre obtenu a un effet plus localisé que le filtre « idéal ».

Changez le masque précédent par `masque_b` et relancez les expériences. Que constatez-vous concernant l'effet de Gibbs ?