ECHANTILLONNAGE

1• Objectif de ce travail :

Le travail proposé ici consiste à manipuler des images pour observer le résultat de l'étape d'échantillonnage.

2• Petite histoire : Harry NYQUIST & Claude Elwood SHANNON

Harry Nyquist

Né le 7 février 1889 à Nilsby (Suède) – mort le 4 avril 1976.à Harlingen (Texas)

Il émigra vers les Etats-Unis en 1907 et entra à l'université du Dakota du nord en 1912. Cinq ans plus tard, il fut reçu comme docteur en physique à l'université de Yale Après avoir travaillé de 1917 à 1934 chez AT&T, il partit pour les laboratoires Bell.

Aux Bell Labs, il fit des recherches sur le bruit thermique appelé également «bruit de Johnson-Nyquist» et sur la stabilité des amplificateurs bouclés (Diagramme de Nyquist).

Ses travaux théoriques sur la détermination de la bande passante nécessaire à la transmission d'information, publiés dans l'article « *Certain factors affecting telegraph speed* » posent les bases pour les recherches de Claude Shannon qui amèneront la théorie de l'information.

En 1927 Nyquist détermine qu'**un signal analogique doit être échantillonné à au moins deux fois la plus haute fréquence qui le constitue si l'on veut le convertir en un signal numérique**. Ce résultat, connu sous le nom de théorème d'échantillonnage de Nyquist-Shannon, a été publié dans l'article « *Certain topics in Telegraph Transmission Theory »* en 1928 (voir la publication en fin de fichier).

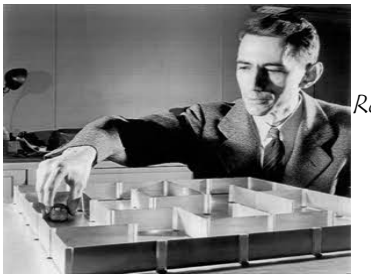
Harry Nyquist a eu un rôle important dans la théorie de l'information et de l'automatique.

Référence Site web de :

- ✔ *http://www.britannica.com/eb/article-9125110/Harry-Nyquist*
- ✔ *http://fr.wikipedia.org/wiki/Harry_Nyquist*

Claude Elwood SHANNON :

Né le 30 avril 1916 à Gaylord (Michigan) – mort le 26 février 2001.



Référence Site web de :

- *http://www.bell-labs.com/news/2001/february/26/1.html*
- *http://www.nyu.edu/pages/linguistics/courses/v610003/shan.html*

3 . MANIPULATIONS :

3.1 . Échantillonnage :

Dans cette première partie, on vous propose de revoir rapidement quelles sont les règles d'échantillonnage d'un signal.

Reprenez le TP « Transformée de Fourier », et utilisez la question « *3.3 . Animez votre Transformée de Fourier 2D !* » pour générer une image comportant une sinusoïde, faites augmenter la fréquence de cette sinusoïde d'une valeur nulle à $2.F_{ech}$. Que se passe-t-il dans le plan de Fourier ?

```
% Génération d'un Cosinus 2D
clear all;

Tox=512;
Toy=512;
while(Tox>1)
    for i=1:256
        for j=1:256
        mon_cos(i,j)=double(128*(1+cos(2*pi*(i/Tox+j/Toy))));
        end
    end
    figure (1);
    image(mon_cos);
    drawnow;
    colormap(pink(256));
    axis('image');

    figure (2);
    imagesc(log10(abs(fftshift(fft2(mon_cos)))));
    colormap(gray);
    axis('image');

    Tox=Tox/1.05;
    Toy=Toy/1.05;
end
```

3.2 . Échantillonnage d'une image réelle :

Maintenant, ouvrez l'image « Echantillonnage.bmp »

```
MonImage=imread('Echantillonnage.bmp');
```

L'image qui vous est proposée, est codées sur 8 bits. Pour certaines opérations, il est

important de noter que le résultat obtenu est plus précis, voire très différents, si les calculs sont réalisés sur des valeurs réelles. Pour convertir l'image en « flottant » :

```
MonImage = double(MonImage) ;
```

L'image en niveau de gris est formée de Nlin lignes, Ncol colonnes et 1 seul plan. Ces informations sont récupérables facilement grâce à :

```
[Nlin, Ncol] = size(MonImage) ;
```

Pour afficher l'image :

```
figure(1);
image(MonImage);
axis('image');
colormap(gray(256));
```

Vous pouvez calculer la FFT de l'image :

```
Freq_MonImage = fft2(MonImage);
```

Rappel : votre image est un signal 2D réel et sa fft est complexe (la fonction abs() donne le module).

Échantillonnez cette image en prélevant :

- 1 point sur 2
- 1 point sur 3
- 1 point sur 4

Expliquez le résultat (regarder aussi les spectres).

Attention : agrandissez les fenêtres qui contiennent les images pour qu'elles occupent tout l'écran.

3.3 . Interpolation (effet de zoom) :

L'objectif de cette partie du TP est d'essayer de réaliser un agrandissement de l'image d'origine ('Echantillonnage.bmp'). Pour cela vous essayerez de réaliser une interpolation linéaire, puis par une fonction polynomiale (cubique).

3.3.1 . Principe :

Le principe est assez simple. Si l'on considère que l'image qui vous est donnée respecte le théorème de Nyquist-Shannon :

- On reconstruit la fonction continue (objet d'origine) qui par échantillonnage a permis d'obtenir l'image dont vous disposez.
- Cette fonction continue est alors filtrée pour obtenir un spectre compatible avec l'échantillonnage que l'on souhaite réaliser (limitation du spectre pour éviter le repliement de spectre au cours de l'échantillonnage).

- Pour terminer, on échantillonne l'image « continue filtrée » aux points que l'on veut calculer pour la nouvelle image.

Dans la pratique, la solution consiste à combiner la reconstruction et le filtrage en une seule étape, mais aussi pour améliorer le temps de calcul, à remplacer la fonction sinus cardinal, classiquement utilisée pour la reconstruction, par une forme polynomiale (une somme finie de termes remplace alors une somme infinie dans le cas du sinus cardinal).

$$Image\_reconstituée(i,j) = Image\_échantillonnée(i,j) \otimes h(i,j) \qquad (1)$$

h(i,j) est la fonction choisie pour la reconstitution. Classiquement h(i,j) = sin_c(i,j), car cette fonction correspond à une fonction porte dans l'espace des fréquences H(u,v)=rect(u,v).

Remarque : rect(u,v) est la fonction porte de largeur T.

$$rect(u,v) = \begin{matrix} 1 & si\ \sqrt{u^2+v^2} \le T \\ 0 & ailleurs \end{matrix}$$

la relation (1) s'écrit dans l'espace de Fourier :

$$Spectre\_Image\_reconstituée(u,v) = Spectre\_Image\_échantillonnée(u,v)\ .\ H(u,v)$$

- Pour une interpolation linéaire, on peut utiliser une fonction h(i,j) de la forme pour i et j compris entre -T et +T :

$$h(i,j) = \begin{matrix} (1-\frac{|\sqrt{i^2+j^2}|}{T}) & si\ \sqrt{i^2+j^2} \le T \\ 0 & ailleurs \end{matrix}$$

Dans le plan de Fourier :

$$H(u,v) = T^2 . \sin_c^2(\pi . u . v)$$

- Pour une interpolation cubique, on peut utiliser une fonction h(i,j) de la forme :

$$h(i,j) = \begin{matrix} (a+2).|x|^3-(a+3).|x|^2+1 & si\ |x|<1 \\ a.(|x|^3-5.|x|^2+8.|x|-4) & si\ 1 \leqslant |x| \leqslant 2 \\ 0 & sinon \end{matrix}$$

si a=-1/2, on obtient dans le plan de Fourier :

$$H(u,v)=\frac{18-24.\cos(2.\pi.u)+6.\cos(4.\pi.u)-\pi.u.[2.\sin(2.\pi.u)-\sin(4.\pi.u)]}{(2.\pi.u)^4}$$

3.3.2. Manipulation :

- Créez une image de dimension double (zoom = 2) qui ne contient pas encore les valeurs interpolées.

```
% création de l'image zoomée non interpolée
for i=1:zoom*Ncol
    for j=1:zoom*Nlin
     if ((uint16(j/zoom)*zoom==j) && (uint16(i/zoom)*zoom==i)
&& (uint16(j/zoom)+1<Ncol) && (uint16(i/zoom)+1<Nlin) )
                          MonImage1(j,i)=MonImage(uint16(j/zoom)
+1,uint16(i/zoom)+1);
        else MonImage1(j,i)=0;
        end
    end
end
```

- Créez le masque de la fonction qui servira à la l'interpolation :

```
% Calcul de la matrice h:
matrix=2;
for k1=-matrix:matrix
    for k2=-matrix:matrix
       h(k1+matrix+1,k2+matrix+1)= ....
    end
end
% affichage du noyau de convolution
figure();
imagesc(h);
axis('image');
colormap(pink(256));
```

- Calculez la convolution du noyau h(i,j) par l'image. Sous matlab, il existe la fonction « conv2() ».
- Affichez le résultat
- Effectuez la même opération dans le plan des fréquences.
- Essayez différentes fonctions pour le noyau de convolution.

3.3 . Quantification :

Quel est l'effet de l'opération de quantification sur les images ?

3.3.1 . Principe :

Vous allez répondre à cette question dans cette partie du TP. Cette quantification se

produit à plusieurs niveaux :

- Lors de l'acquisition de l'image, le capteur en fonction de ses qualités intrinsèques (rapport signal/bruit) réalise une quantification du signal reçu de l'objet.

- Outre les capacités du capteur, le nombre de bits réellement nécessaires pour coder les couleurs d'une image varie en fonction de l'image et de son « *contenu informationnel* ». Ce nombre dépend de l'*entropie*, définie à partir de la distribution des niveaux de gris de l'image.

$$E = \sum_{i=0}^{N} - p_i . \log_2(p_i)$$

N est le nombre de niveaux de gris présents,
p_i est la proportion ($0 < pi < 1$) de points de l'image ayant pour niveau de gris i.

E représente le nombre moyen de bits par pixel nécessaires pour coder toute l'information présente.

3.3.2. Manipulation :

Essayer de quantifier l'image « clown.bmp » sur 16 bits, 7 bits, 6 bits, 5 bits, 4 bits, 2 bits et enfin 1 bit.

Conclusions ?

4 . CHERCHEZ ...

Vous allez maintenant rechercher dans une image une information qui peut être différente suivant l'échantillonnage que vous choisissez.

4.1 . Principe :

- Échantillonnez l'image 'Surprise.bmp' avec une fréquence égale à 1/3 de la fréquence actuelle et sans introduire de déphasage.
- Échantillonnez l'image 'Surprise.bmp' avec une fréquence égale à 1/3 de la fréquence actuelle et mais avec un déphasage de 1 pixel.
- Échantillonnez l'image 'Surprise.bmp' avec une fréquence égale à 1/3 de la fréquence actuelle et mais avec un déphasage de 2 pixels.

4.2 . Manipulation :

Vous pourrez utilisez le code ci-dessous comme exemple :

```
clear all;
Im_Surprise=imread('Surprise.bmp');
[Nlin Ncol]=size(Im_Surprise)

for j=1:3:Nlin
   for i=1:3:Ncol
       Image1(round(i/3)+1,round(j/3)+1)=Im_Surprise(i,j);
      Image2(round(i/3)+1,round(j/3)+1)=...
      Image3(round(i/3)+1,round(j/3)+1)=...
   end;
end;
% Affichage
figure(3);
image(Image3);
colormap(pink(256));
figure(4);
image(Image1);
colormap(pink(256));
figure(5);
image(Image2);
colormap(pink(256))
```

Lectures ...

Publication Nyquist

Publication Shannon