

Université Echahid Hamma Lakhder El Oued.

Module : Image, son, vidéo codage et transmission

TP5

Transformée de Fourier d'une image et filtrage fréquentiel

Objectifs du TP5 :

Le but de l'application de la transformée de Fourier sur une image est la mise en évidence des caractéristiques fréquentielles de cette dernière. Nous nous intéresserons au spectre de Fourier (i.e le module de la transformée de Fourier) de l'image. En effet, le spectre permet de rendre compte de la distribution énergétique de l'image. Vu que la transformée de Fourier permet d'explorer la composition fréquentielle de l'image, nous pouvons tirer profit de ses propriétés afin d'appliquer des opérateurs de filtrage à une image dégradée dans le but d'atténuer le bruit ou de le supprimer. Lors de cette transformation, le signal est décomposé sur un ensemble de signaux de base qui sont cosinus, sinus ou l'exponentielle imaginaire et forment une base orthogonale (qui permet de supprimer les redondances d'informations).

Nous pouvons utiliser les fonctions Matlab suivantes :

fft2 : fonction Matlab pour le calcul de la transformée de Fourier discrète 2D.

L'origine du spectre (composante continue) se trouve en haut à gauche, il est plus naturel de voir l'origine des fréquences au centre du spectre. En effet, nous n'allons pas changer le contenu dans le spectre mais juste son agencement.

Centrage du spectre : nous multiplions chaque pixel de coordonnées (x, y) par $(-1)^{x+y}$. cela met en avant la propriété de symétrie des spectres par rapport à $F(0, 0)$.

fftshift : fonction Matlab pour le centrage du spectre.

1- Transformée de Fourier des images synthétiques

Le code suivant définit un rectangle discret et affiche sa transformée de Fourier discrète :

```
N = 256; r1 = 10; r2 = 22;
% Création du carré
tmp1 = [zeros(N/2 - r1,1) ; ones(2*r1,1) ; zeros(N/2-r1,1)] ;
tmp2 = [zeros(N/2 - r2,1) ; ones(2*r2,1) ; zeros(N/2-r2,1)] ;
f = tmp1 * tmp2';
figure(1) ; imshow(f,[ ]);
% Transformée de Fourier
ft = fftshift(fft2(f)) ;
% Représentation de la TF en échelle normale
figure(2) ; imshow(abs(ft),[ ] ) ;
figure(3) ; imshow(log(1+abs(ft)),[ ] ) ;
```

2- Transformée de Fourier des images en niveaux de gris

Le code suivant définit la transformée de Fourier de l'image 'cameraman.tif' :

```
x = imread('cameraman.tif');
figure(1); imshow(x);
y=fft2(x);
ymod = abs(fftshift(y));
figure(2); imshow(log(ymod),[]);
```

Pour afficher la transformée de Fourier de l'image. Il est intéressant de représenter la norme de la transformée de Fourier, qui représente l'amplitude des fréquences spatiales de l'image. Nous avons utilisé la fonction **fftshift** pour recentrer **y** afin de visualiser l'amplitude du contenu fréquentiel avec les basses fréquences au centre de l'image (domaine principal).

Etant donné la grande variabilité des amplitudes possible, il est plus informatif de les afficher en échelle logarithmique, i.e. **log(ymod)**.

Enfin, comme nous ne connaissons pas à priori la gamme des valeurs que peut prendre **log(ymod)**, nous donnons l'intervalle vide [] comme argument à **imshow**, qui l'ajuste automatiquement aux valeurs minimales et maximales de l'image à afficher.

Notons que de manière alternative, nous aurions pu utiliser la fonction **imagesc** qui elle aussi utilise toute la gamme dynamique de l'image pour la visualisation, et permet en outre de choisir une carte de couleur quelconque.

Nous allons afficher l'image en niveau de gris :

```
figure(2); imagesc(log(ymod)); colormap(gray); truesize
```

3- Filtrage d'une image en utilisant la transformée de Fourier

Le code suivant définit le filtrage d'une image bruitée avec un bruit 'poivre et sel' en utilisant un filtrage fréquentiel basé sur la transformée de Fourier

```
im=imread('cameraman.tif');
im=imnoise(im,'salt & pepper',0.2);% Ajouter à l'image un bruit de type 'poivre et sel'
% Illustration du filtrage d'une image en passant
% par la transformée de Fourier
% Explications sur le programme:
disp('Filtrage d'une image en utilisant la transformée de Fourier')
% lecture de l'image
I = double(im)/255;
% TFD de l'image et visualisation du spectre
% (le zero est placé au centre par fftshift)
Itfd = fft2(I);
logItfd = log10(abs(fftshift(Itfd))+eps);
figure, imagesc(logItfd), colorbar
title('spectre image source (échelle log)');
colormap(gray)
% Mémorisation des bornes de l'échelle
smax = max(max(logItfd));
smin = min(min(logItfd));
% TFD du filtre et visualisation du spectre
a = 2; % Taille du filtre: (2*a+1)x(2*a+1)
h = ones(2*a+1,2*a+1)/((2*a+1)*(2*a+1)); % Le filtre
```

```
hh = zeros(size(I)); % Filtre étendu
centre = ceil(0.5+size(I)/2); % Centre filtre étendu
hh(centre(1)-a:centre(1)+a, ...
centre(2)-a:centre(2)+a) = h; % hh(129-2 : 129+2, 129-2 : 129+2) → hh(127:131,127:131)=h
hh = fftshift(hh);
H = fft2(hh);
figure, imagesc(abs(fftshift(H))), colorbar
title('spectre du filtre (echelle lineaire)');
colormap(gray)
% Calcul de l'image filtrée
Ftfd = conj(H) .* Itfd;%transformée de Fourier de l'image * transformée de Fourier du filtre
étendu
logFtfd = log10(abs(fftshift(Ftfd))+eps);
figure, imagesc( min(max(logFtfd,smin),smax) )
colorbar
title('spectre image filtrée (echelle log)');
colormap(gray)
F = real(iff2(Ftfd));
figure,
subplot(1,2,1), imshow(I)
title('image source')
subplot(1,2,2), imshow(F)
title('image filtrée, en utilisant la TFD');
```

Quelle est la différence entre un filtrage spatial et un filtrage fréquentiel ?