UNIVERSITÉ DE RENNES 1

Module VIS - Master MITIC
TP4 : filtrage d'images dans le domaine fréquentiel

O. LE MEUR
olemeur@irisa.fr

24 janvier 2016

But du TP :
- Compréhension et application de la transformée de Fourier bi-dimensionnelle
- Filtrage fréquentiel.

Travail à rendre :
Le travail est à rendre pour la semaine suivante (voir avec votre encadrant pour définir une date). Pour chaque question, on donnera les commandes et traitements effectués, les images et valeurs obtenues et les commentaires correspondants.

L'ensemble doit être rendu *sous forme électronique*(au format PDF, Word ou Open Office). Ce compte rendu est à envoyer à votre encadrant : `olemeur@irisa.fr`. Si vous travaillez à deux, le nom du fichier doit inclure les noms des binômes.
Vous trouverez le sujet et les ressources associées à ce tp sur le lien suivant :
`http://people.irisa.fr/Olivier.Le_Meur/teaching/MITIC/TP4_ressources.zip`

1 Transformée de Fourier

Cette section commence par donner le code nécessaire pour effectuer une transformée de Fourier directe et inverse sur une image. Tester ce code, et essayer de comprendre les différentes étapes.

1.1 Code Matlab pour appliquer la transformée de Fourier directe une image

```
clear
close all
% ================================================
% Transformée de Fourier Directe
% ================================================
figure;
I=double(imread('street.pgm'));
imfft = fft2(I);
imfft = fftshift(imfft);
mag = abs(imfft);
phase = angle(imfft);
imagesc(log(1+mag));
figure;
contour(log(1+mag))
```

1.2 Code Matlab pour appliquer la transformée de Fourier inverse

```
% ================================================
% Transformée de Fourier Inverse
% ================================================
%real = mag.*cos(phase);
%imaginary = mag.*sin(phase);
```

```
%newfft = (real + 1i .*imaginary);
%im = abs(ifft2(fftshift(newfft)));
%imshow(round(im)/255);
```

1.3 Rayures blanches sur fond noir...

1.3.1 Construction d'images binaires

On propose de construire deux images binaires A et B de taille 30×30. Le résultat doit être proche de celui présenté à la figure 1.3.1.

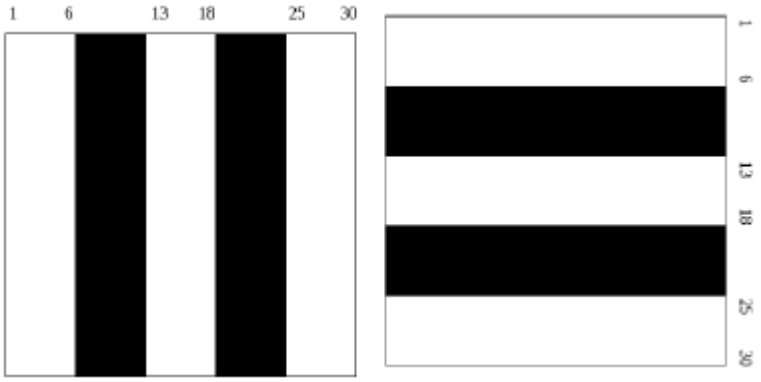

FIGURE 1 – Exemples d'images binaires.

Quelques pistes et remarques...
- On peut commencer par créer une matrice noire (d'intensité 0) à l'aide de la fonction `zeros`. Ensuite des rayures blanches d'intensité 1 peuvent être ajoutées.
- Comme les images créées sont petites, on peut ajouter l'option `'InitialMagnification','fit'` à la fonction `imshow`.
- La fonction `fft2` (et son inverse `ifft2` qu'on utilisera) prend des valeurs de type `double` en paramètre d'entrée.

1.3.2 Calcul et visualisation de la transformée de Fourier

Pour chacune des deux images A et B :
- calculer la transformée de Fourier et centrer son spectre.
- visualiser leurs modules.
- Comparer et expliquer les allures des transformées de Fourier centrées des images A et B.

Commandes et remarques utiles

`fft2`, `fftshift`

La transformée de Fourier a une valeur complexe : on peut donc visualiser le module (fonction `abs`) et la phase (fonction `angle`) des coefficients.

1.4 Image réelle

Charger les images suivantes, visualiser leurs spectres et commenter :
- Image *street.pgm*
- Image *desert.pgm*
- Image *mountain.pgm*

Afin d'afficher le module du spectre de Fourier, il est nécessaire de compresser la dynamique. Le module est donc modifié en appliquant une transformation log. Généralement, on utilise $log(1 + module)$. Vous visualiserz également le spectre en utilisant la commande `contour`.

2 Filtrage fréquentiel

2.1 Filtrage de Butterworth

La fonction de transfert du filtre de Butterworth est donnée ci-dessous :

$$f(u,v) = \frac{1}{1 + (\frac{w}{D_0})^{2n}}$$

avec, n l'ordre du filtre, D_0 la fréquence de coupure et w la fréquence radiale (distance Euclidienne par rapport au centre de l'image).

- Construiser ce filtre (voir remarque ci-dessous)
- Appliquer ce filtre à l'image *lena.pgm*. Faites varier l'ordre et la fréquence de coupure. Commenter.

Une façon élégante (en évitant les boucles FOR) de calculer le rayon pour tout point d'une image et en prenant le centre comme origine est donnée ci-dessous :

```
xrange = [-cols/2:(cols/2-1)];
yrange = [-rows/2:(rows/2-1)];
[x,y] = meshgrid(xrange, yrange);
radius = sqrt(x.^2 + y.^2);        % A matrix with every pixel = radius relative to centre.
```

2.2 Débruitage

L'idée de cette partie est de débruiter une image. Dans un premier temps, charger et afficher l'image originale *lena.pgm* et sa version bruitée *lenabruitee.pgm*.

- Visualiser le module du spectre de Fourier des deux images. Commenter.
- Proposer une solution pour débruiter l'image.
- On dispose d'un filtre prédéfini dans le fichier *masque.mat* fourni avec le tp. Charger le masque ($MasqueStruct = load('masque.mat'); masque = MasqueStruct.F_m;$) et le visualiser. Comment peut-on l'utiliser pour éliminer le bruit ?
- Réaliser le débruitage et visualiser l'image débruitée dans le domaine spatial.