M4201Cin TP 5

Transformée de Fourier

16 mars 2018 – B. Colombel

À rendre

- Un fichier contenant les fonctions et les réponses aux questions des exercices 1 à 6 ecrit dans l'éditeur Scinote et nommé

  `Nom_Prenom-TP4.sci`
- L'image `Bali_marquee.pgm` construite à l'exercice 4.

Le tout sera déposé dans le dossier sur Ametice prévu à cet effet (M4201Cin, TP4).

Si le module SIVP de **Scilab** est bien installé, vous verrez au démarrage :

```
Initialisation :
  Chargement de l'environnement de travail

SIVP - Scilab Image and Video Processing Toolbox
        load macros
        load gateways
        load help
        load demos
```

Sinon, installez le :

Applications ⟩ Gestionnaire de modules - ATOMS ⟩ Traitements des images ⟩ SIVP

Cliquer sur installer puis redémarrer **Scilab**.

1 Transformée de Fourier d'une image

La transformation de Fourier est codée en **Scilab** avec les fonction **`fft`** (pour les fonctions à une variable) et **`fft2`** (pour les fonctions à deux variables).

La transformée de Fourier transforme l'image en une matrice de nombres complexes, pour la visualiser il faut se limiter à la partie réelle (ou imaginaire) ou au module.

Charger l'image `lena.pgm` et la stocker dans la variable `image0`.

```
--> im = double(image0);
--> im1a = fft2(im);
```

Pour pouvoir visualiser le module de la transformée on utilise l'image *rehaussée* du spectre de l'image :

```
--> im1b = 1 + log10(abs(im1a));
--> im1b = 255*im1b./max(im1b) // on normalise les valeurs entre 0 et 255
--> imshow(uint8(im1b));
```

Pour vous convaincre que c'est nécessaire, taper :

```
--> imshow(uint8(255*abs(im1a)./max(abs(im1a))));
```

Les zones où le niveau de gris est le plus élevé correspond aux « basses fréquences » , la fonction **`fftshift`** permet de ramener les basses fréquences au centre de l'image :

```
--> im1c = fftshift(im1b);
--> imshow(uint8(im1c));
```

On peut inverser la transformation de Fourier en appliquant la fonction `ifft`.

```
--> im2a = ifft(im1a);
--> im2b = real(im2a);
--> imshow(uint8(im2b);
```

2 filtrage fréquentiel

2.1 Masques pour le filtrage

Supposons que `im2` représente la transformée de Fourier de `image0` avec les basses fréquences au centre de l'image.
Pour supprimer les basses-fréquences ou les hautes fréquences de l'image, il suffit de multiplier (terme à terme) la matrice représentant `im2` par une image de même taille ayant deux niveaux :
— 0 (pour les fréquences à supprimer) et
— 1 (pour les fréquences à conserver).
Pour cela nous avons besoin :
— des normes vues en cours :

$$\|(x;y)\|_1 = |x| + |y| \qquad \|(x;y)\|_2 = \sqrt{x^2+y^2} \qquad \|(x;y)\|_\infty = \max(|x|;|y|)$$

Ces normes permettent de quantifier les notions de *distances*.
— des fonctions **`bool2s`** et **`feval`**[1].

1. **Masque circulaire**
   (a) Recopier la fonction suivante dans **Scinote** puis charger cette fonction (F5).

```
// Norme euclidienne
function z = n2(x, y)
  z = sqrt((x).^2+(y).^2);
endfunction
```

   (b) Charger l'image `lena.pgm` et l'appeler `image0`.
   (c) Recopier les commandes suivantes dans la console **Scilab** :

```
--> [nx, ny] = size(image0);
--> x = [1:nx]; // liste des coordonnees x
--> y = [1:ny]; // liste des coordonnees y
--> xc = nx/2; yc = ny/2; // centre de l'image
--> masque = bool2s(feval(x-xc, y-yc, n2) > 100); //masque de rayon 100
pixels autour de (xc,yc)
--> imshow(uint8(255*masque)); //affichage du masque
--> image = image0.*masque;
--> imshow(image);
```

1. à chercher dans l'aide

Exercice 1 :

1. Écrire deux fonctions `z = n1(x, y)` et `z = ninf(x, y)` qui retourne les normes $\|\cdot\|_1$ et $\|\cdot\|_\infty$ définies précédemment.
2. Écrire une fonction `res = masque(image, xc, yc, r, n)` qui renvoie un masque centré au pixel `(xc;yc)` de la même dimension que l'image `image` de rayon r en utilisant la norme n.
3. Quelles sont les différentes formes obtenues pour le masque suivant les normes ?

2.2 Filtrage fréquentiel

Exercice 2 :

1. Écrire une fonction `res = PasseBas(image0, r)` qui retourne l'image filtrée par un filtre passe-bas, c'est-à-dire après élimination des hautes-fréquences.

$$\begin{tikzcd}
\text{Domaine spatial} & & \text{Domaine fréquentiel} \\
\boxed{\text{Image originale}} \arrow[rr, "\mathrm{TF}"] & & \boxed{\text{Image spectre}} \arrow[d, "\text{filtre passe-bas}"] \\
\boxed{\text{Image transformée}} & & \boxed{\text{Image spectre transformée}} \arrow[ll, "\mathrm{TF}^{-1}"']
\end{tikzcd}$$

2. Tester la fonction avec l'image `lena_bruit.pgm`.
3. Comparer le résultat obtenu avec un filtre gaussien par exemple.

Exercice 3 :

1. Écrire une fonction `res = PasseHaut(image0, r)` qui retourne l'image filtrée par un filtre passe-haut, c'est-à-dire après élimination des basses-fréquences.
2. Tester avec l'image `lena.pgm` et comparer avec un filtre détecteur de contour comme l'opérateur laplacien.

3 Marquage de photo en Hautes Fréquences

Exercice 4 :

1. Ouvrir les images `Bali.pgm` et `Depardieu.pgm` puis les visualiser.

On veut *cacher* l'image `Depardieu.pgm` dans l'image `Bali.pgm`. Pour cela, on procède comme suit :

— éliminer les hautes-fréquences et garder les basses-fréquences dans un rappport de $^1/_{3,5}$ dans l'image `Bali.pgm` ;
— éliminer les basses fréquences dans les mêmes proportions de $^1/_{3,5}$ de l'image `Depardieu.pgm` ;
— additionner les 2 matrices de Fourier correspondantes ;
— reconstruire de l'image composite.

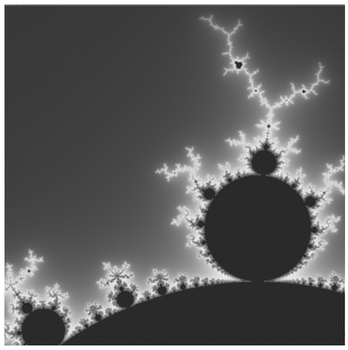
2. Suivre la méthode décrite ci-dessus et sauver l'image obtenue sous le nom `Bali_marquee.pgm`.
3. Á partir de cette image, retrouver le nom du célèbre GÉRARD.

(a) image originale (512 × 512) (b) image sous échantillonée par 2 (256 × 256) (c) image sous-échantillonée par 4 (126 × 126)

Figure 1 – Phénomène de moiré lors d'un sous-échantillonnage

Exercice 5 : Dans l'image ci-contre, le nom d'un assassin célèbre est caché. Le but de l'exercice est de découvrir ce nom.

Ouvrir l'image `Mandelbrot.pgm` puis en s'inspirant de l'exercice précédent, retrouver ce célèbre assassin.



4 Théorème d'échantillonnage de Nyquist—Shanon

Lorsque image est sous-échantillonée on voit apparaître un effet de moiré pu aliasing comme l'illuste la figure 1 page 4.

Cela est du au phénomène de *repliement spectral* lorsque la fréquence d'échantillonnage est trop faible par rapport à la fréquence maximale présente dans l'image.

Nous avons tous observé ce phénomène :
- À la télévision, une chemise aux rayures trop serrées fera apparaître des motifs erronés ;
- dans les Western, les roues des roulottes paraissent souvent tourner dans le mauvais sens.

Atténuation du phénomène de moiré par étalement du spectre

> Le théorème de Nyquist et Shannon précise les conditions de validité de la reconstruction afin de s'affranchir des problèmes de repliement spectral :
> - La transformée de Fourier est bornée ;
> - La fréquence d'échantillonnage doit être au moins égale au double de la fréquence maximale de l'image.

Pour respecter les conditions de ce théorème, on peut :

1. effectuer un filtrage passe-bas de l'image afin d'éliminer les hautes fréquences et diminuer sa fréquence maximale.

   Cependant, des détails seront atténués (perte d'information).

2. augmenter le pas d'échantillonnage. Le *suréchantillonnage* consiste à avoir plus d'échantillons que de pixels.

Exercice 6 :

1. On cherche à éliminer l'effet de moiré obtenu avec le sous-échantillonnage avec `f=mean` et $l = 8 = k$ en appliquant un filtre passe-bas à l'image avant sous-échantillonnage de l'image `ploop.pgm`.

   Éliminer les 5 % fréquences les plus grandes dans l'image `ploop.pgm`. Est-ce suffisant ? Si non, augmenter ce pourcentage de 5 en 5 et trouver le pourcentage adéquat.
2. Écrire une fonction enchaine filtre passe-bas puis sous-échantillonnage.