L3 SI

Traitement d'image

TP N° 3 : Transformée de Fourier

1. Introduction

Soit I une image en niveaux de gris composée de pixels p(i,j). La transformée de Fourier 2-D discrète d'une telle image I est une image complexe composée d'éléments X(f,g) tels que :

$$X(f,g) = \sum_i \sum_j p(i,j).e^{-2j\Pi(if+jg)}$$

Les X(f,g) sont des valeurs complexes possédant un module et une phase.

La transformée de Fourier permet de passer d'une représentation spatiale de l'image à une représentation fréquentielle. Les modules des composantes complexes représentent notamment l'énergie de la fréquence correspondante.

2. Manipulation

Pour les besoins de ce TP, il est conseillé de récupérer et de modifier le fichier **`tp3_widget.py`** sur moodle. Il est de plus nécessaire d'importer les modules *numpy* et *opencv*.

2.1. Transformée de Fourier d'une image sinusoïdale

Une image de sinus en 256 niveaux de gris est obtenue par l'équation :

$$I(x,y) = 128 \times (\sin(2 \times \pi \times (A \times x/256 + B \times y/256)) + 1)$$

A et B sont les fréquences respectives (le nombre d'oscillations) dans les directions *x* et *y*.

Pour réaliser cette fonction il est nécessaire au préalable de créer les images *x* et *y* correspondant aux coordonnées des pixels dans l'image :

>> **`x, y = np.meshgrid(range(0, 256), range(0, 256))`**

A partir des images x et y il est maintenant possible de créer l'image I (il est nécessaire au préalable de définir A et B) :

```
>> i = 128*(np.sin(2*np.pi*(A*x + B*y)) + 1)
```

a. Afficher l'image **`I`** pour différentes valeurs de **`A`** et **`B`**. Conclusion.

b. Déterminer la transformée de Fourier de l'image **`I`**. Pour cela nous allons utiliser la fonction numpy **`fft2`** :

```
>> fi = np.fft.fft2(i)
```

La transformée de Fourier d'une image est constituée de pixels dont les valeurs sont des complexes. Sans entrer dans les considérations théoriques, nous nous intéresserons uniquement aux modules de ces valeurs qui représentent les énergies des composantes fréquentielles correspondantes. Le module de l'image **`fi`** est calculé à l'aide de la fonction **`abs()`** :

```
>> m_fi = abs(fi)
```

Une dernière opération va nous permettre de représenter l'image des fréquences dans un repère dont l'origine (qui correspond à la fréquence nulle c'est à dire la composante continue) est située au centre de l'image :

```
>> m_fi = np.fft.fftshift(m_fi) ;
```

Afficher maintenant l'image **`m_fi`**. Qu'observe-t-on ? Conclusion.

c. Réaliser la manipulation pour différentes valeurs de A et B. Observer l'évolution des composantes fréquentielles. Conclusion.

2.2. Linéarité de la transformée de Fourier

a. Créer deux images sinusoïdales **i1** et **i2** avec des valeurs différentes de A et B comme précédemment. Créer ensuite l'image somme **i = i1 + i2** (normalisée entre 0 et 255) de ces deux images.

b. Calculer le module de la transformée de Fourier de **i** en appliquant les différentes étapes précédentes. Afficher l'image résultat. Conclusion.

c. Réaliser la même opération en effectuant la somme de plus de 2 images constituées de diffiérentes composantes fréquentielles (différentes valeurs de A et B). Observer le résultat sur le module de la transformée de Fourier. Conclusion.

2.3. Filtrage dans l'espace des fréquences

Nous allons maintenant réaliser une image perturbée artificiellement par une composante fréquentielle indésirable. Le but de cette partir et de réaliser le filtrage de cette composante dans l'espace des fréquences. Ce traitement est en réalité équivalent à une convolution des le domaine spatial.

L'image de la composante fréquentielle sera constitué d'une sinusoïde pure, sans composante continue, réalisée à l'aide de la commande :

```
>> i = 128*np.sin(2*np.pi*(A*x + B*y))
```

L'image à perturber sera constituée du photophore des tps précédents.

```
>> im = cv.imread('photophore.tif', 0)
```

a. Réaliser l'image perturbée en ajoutant à l'image **im** l'image **i** de la sinusoïde. Afficher l'image résultat. Conclusion.

b. Pour filtrer l'image il est nécessaire dans un premier temps de localiser les pics fréquentiels de **i**. Réaliser les différentes étapes permettant de calculer le module de la transformée de Fourier de **i**. Afficher l'image du module et localiser les coordonnées des pics. Par exemple, pour A=50 et B=10 les pics sont localisés en (119,79) et (139,179).

c. Calculer l'image de la transformée de Fourier de **im** + **i**. Afin de pouvoir calculer la transformation inverse, il est nécessaire de conserver les composantes complexes de la transformée de Fourier. **L'étape du calcul du module à l'aide de la fonction `abs()` (et uniquement celle là) ne sera donc pas effectuée.**

d. Réaliser le filtrage dans l'espace des fréquences. Pour cela, il faut simplement annuler les composantes correspondant aux fréquences de **i**.

Dans l'exemple précédent cela donne :

```
>> FI(119,79) = 0
>> FI(139,179) = 0
```

fi étant l'image de la transformée de Fourier shiftée de **im + i**

Réaliser à nouveau le changement de repère de l'image (afin de retrouver le repère initial) :

```
>> fi = np.fft.fftshift(fi)
```

puis calculer la transformée de Fourier inverse :

```
>> im2 = np.fft.ifft2(fi)
```

Afficher l'image résultat. Conclusion.