

## TP2: Transformée de Fourier et Traitements fréquentiels

Préparé par: Mongetro GOINT

=====

À travers ce rapport, nous tenons à présenter les différents travaux effectués dans le cadre du TP2 du cours de traitement d'images.

Le travail touche deux aspects concernant les images: d'une part, la **transformée de Fourier**, notamment celle définie dans OpenCV avec la fonction `cvDFT` tout en utilisant comme modèle, les codes se trouvant sur la page: [https://docs.opencv.org/2.4/doc/tutorials/core/discrete\\_fourier\\_transform/discrete\\_fourier\\_transform.html](https://docs.opencv.org/2.4/doc/tutorials/core/discrete_fourier_transform/discrete_fourier_transform.html)

D'autre part, les traitements Fréquentiels.

### **PARTIE I : Transformée de Fourier**

Dans cette première partie, nous aurons à calculer et afficher la **transformée de fourier** d'une image prise en argument d'entrée, puis calculer sa transformée inverse et aussi sauvegarder l'image résultat.

#### - Fonctionnement du programme

Pour la compilation du programme, il suffit de se placer dans le répertoire **Transformee\_Fourier** contenant les fichiers relatifs à l'application, puis lancer la commande «**make**». Ensuite, pour passer à l'exécution, taper la commande: «**./TransformeeFourier nom de l'image**».

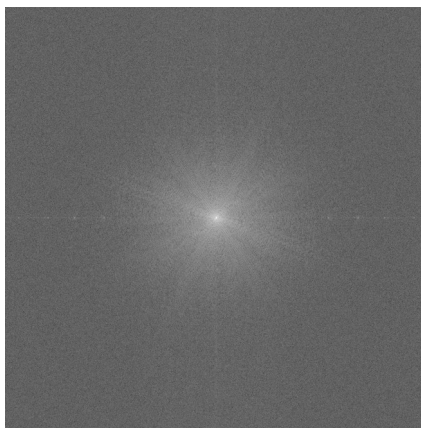
**Exemple** : Pour déterminer la Transformée de Fourier de l'image « Fleur.tif » il suffit d'exécuter la commande : «**./TransformeeFourier fleur.tif**».

### **Expérimentation**

Afin de vérifier le bon fonctionnement du programme, nous avons essayé de le tester avec des images en niveau de gris. En effet, les images ci-dessous montrent entre autre, l'une des résultats de notre expérience:



**Img1:** Image originale e à l'entrée



**Img2 :** Image de la transformée de fourier à la sortie



**Img 3:** Image de la transformée inverse à la sortie

### Analyse des résultats de l'expérience:

D'abord, en observant l'image de la transformée de Fourier obtenue à la sortie nous pouvons remarquer:

**a)** Un point central lumineux qui peut-être considéré comme l'origine du spectre. Celui-ci se présente comme la valeur moyenne des pixels de basse fréquence et de haute fréquence de l'image.

**b)** la zone autour du centre lumineux où est concentrée la plus grande partie de l'énergie de l'image. Là, on constate un changement d'intensité lent. Donc, c'est la partie de basse fréquence.

**c)** la zone la plus éloignée du centre lumineux, où on constate un changement d'intensité rapide. Donc, c'est la partie de haute fréquence.

Ensuite, en constatant l'image de la transformée de Fourier inverse, on peut remarquer qu'elle est identique à l'image d'origine. Ce qui explique qu'aucun changement n'a été effectué au spectre de l'image donnée à l'entrée.

### PARTIE II : Traitements fréquentiels

Dans cette deuxième partie, nous tenons à tester quelques traitements fréquentiels: D'une part, (*filtre passe-haut et filtre-passe-bas*) et d'autre part, la correction de bruit sinusoïdal.

#### - Fonctionnement du programme

Afin de compiler le programme, il suffit de se placer dans le répertoire **Traitements\_Frequentiels** contenant les fichiers relatifs à l'application, puis lancer la commande «**make**». Ensuite, pour passer à l'exécution, taper la commande: «**./Filtres nom de l'image**»

**Exemple** : Pour appliquer un filtre pour l'image « Einstein.tif » il suffit d'exécuter la commande : «**./Filtres Einstein.tif**».

Après l'exécution, le programme offre deux choix à l'utilisateur: 1) Possibilité d'appliquer un Filtre passe-bas; 2) Possibilité d'appliquer un Filtre passe-haut.

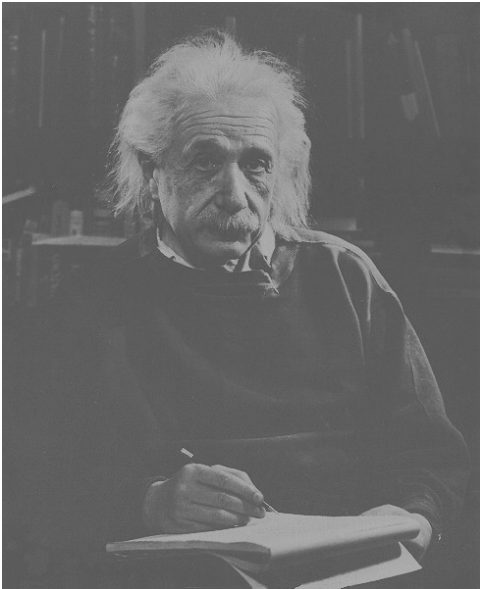
Après que l'utilisateur ait fini de faire son choix, il va devoir entrer la fréquence de coupure (une valeur flottante entre **0 et 1**) afin d'appliquer le filtre choisi.

Suite à l'application du filtre, on aura comme résultat quatre images à la sortie: D'abord, une image considérée comme l'image originale; ensuite, une image de spectre de l'image original sans application de filtre; en plus, une image de spectre de l'image original avec application de filtre; enfin une qui est issue après la détermination de la transformée de Fourier inverse du spectre filtré.

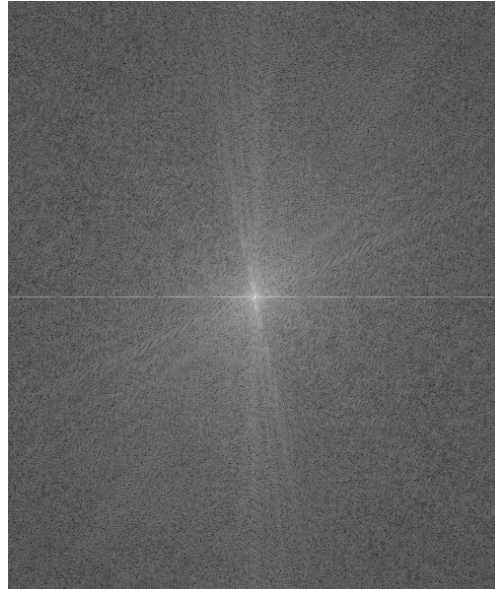
### Expérimentation

#### ➤ **Filtre passe-bas**

Pour une vérification du bon fonctionnement du programme, nous avons effectué des tests avec des images en niveau de gris, notamment l'image **Einstein.tif** pour laquelle nous avons appliqué un filtre passe-bas. En effet, nous avons obtenu les images ci-dessous comme résultat de notre expérience.

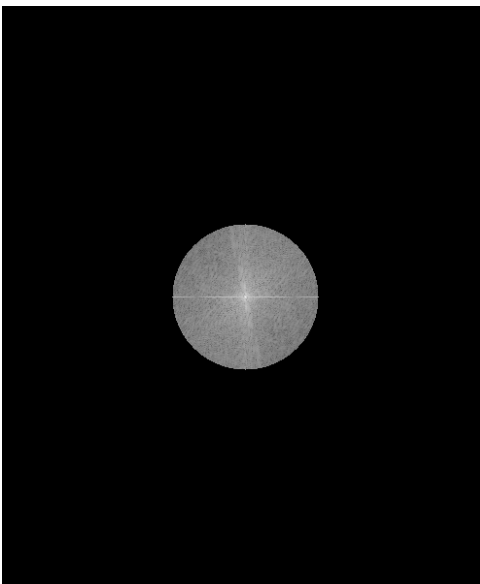


**Img4:** Image originale à l'entrée

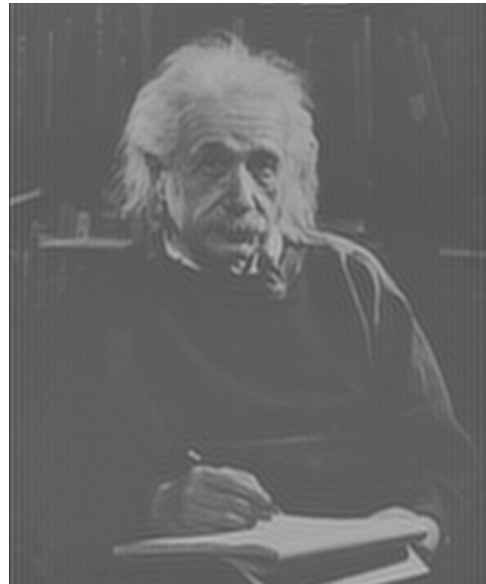


**Img5:** spectre de l'image original sans application de filtre

*1- Expérimentation avec une fréquence de coupure  $=0.30$*

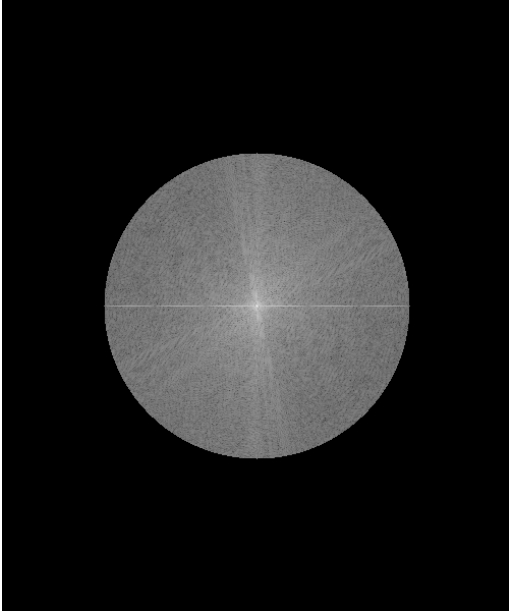


**Img6:** spectre de l'image original avec application de filtre

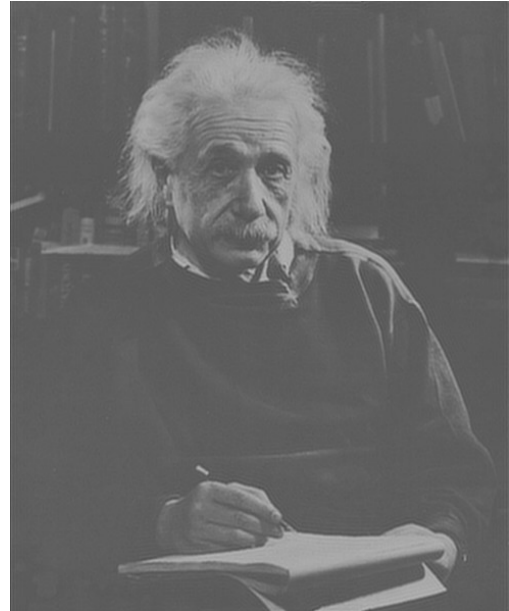


**Img7:** image issue après la détermination de la transformée de Fourier inverse du spectre filtré.

## 2- Expérimentation avec une fréquence de coupure = 0.60



**Img8:** spectre de l'image original avec application de filtre



**Img9:** image issue après la détermination de la transformée de Fourier inverse du spectre filtré.

### Analyse des résultats de l'expérience:

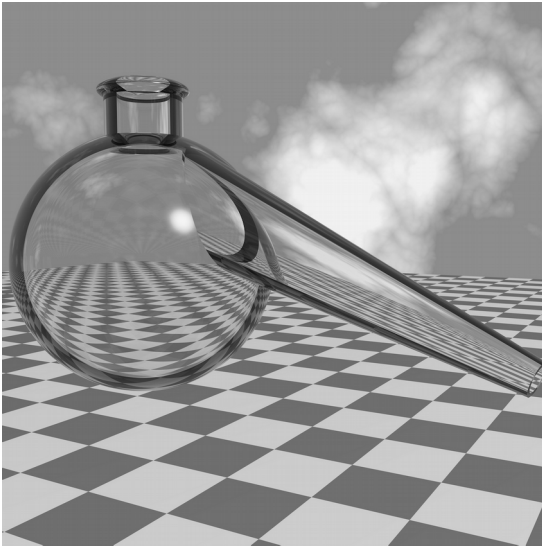
Au cours de l'expérience du **Filtre passe-bas**, nous avons appliqué deux coupures de valeurs différentes: d'abord avec une valeur=**0.30**, ensuite avec une valeur= **0.60**.

D'abord, pour la coupure 0.30, nous pouvons noter que l'image issue après la détermination de la transformée de Fourier inverse du spectre filtré présente une caractéristique un peu floue par rapport à l'image original à l'entrée. Ce qui explique que les basses fréquences qui sont très rapprochées du spectre ont été retenues, alors que celles qui sont plus éloignées ont été ignorées. La valeur des pixels se trouvant en dehors du cercle du spectre ont été négligée, donc égale à zéro.

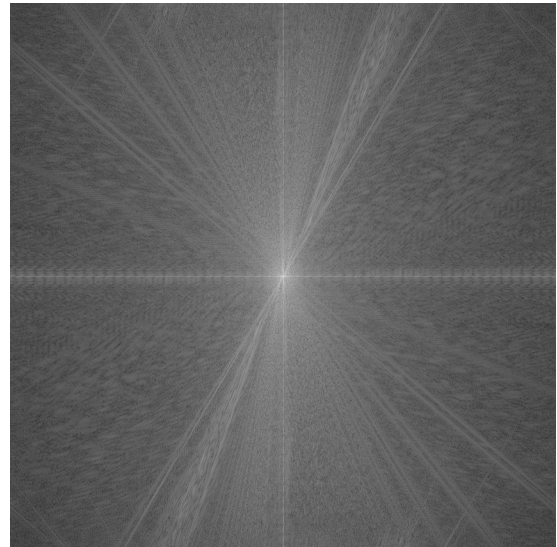
Ensuite, pour la coupure 0.60, nous pouvons souligner que l'image issue après la détermination de la transformée de Fourier inverse du spectre filtré paraît un peu plus nette par rapport au premier cas que nous venons de décrire. Nous observons une forte amélioration de l'image, en raison de que la fréquence de coupure qui prend en compte aussi une partie des hautes fréquences. Ce qui rend les caractéristiques de l'image plus proche de l'image originale.

#### ➤ **Filtre passe-haut**

Pour le filtre passe-haut, nous avons effectué des tests avec des images en niveau de gris. En effet, nous avons obtenu les images ci-dessous comme résultat de notre expérience pour l'une des cas.

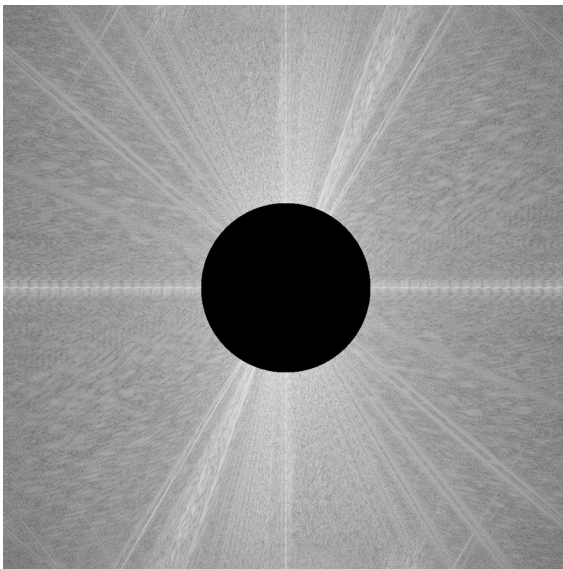


**Img10:** Image originale à l'entrée

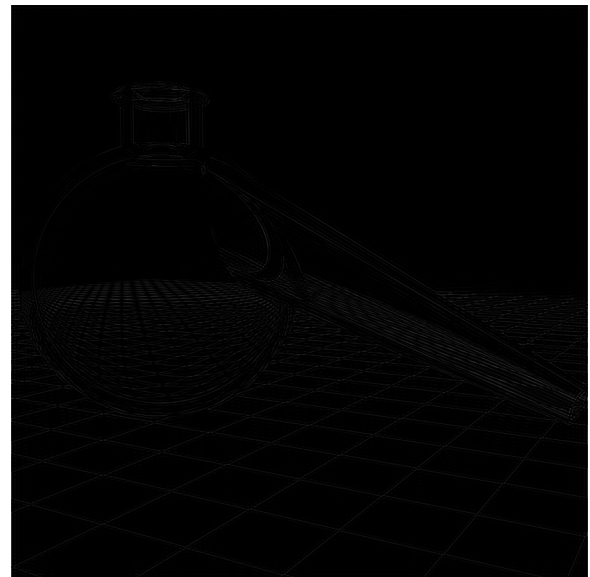


**Img11:** spectre de l'image original sans application de filtre

*1- Expérimentation avec une fréquence de coupure = 0.30*

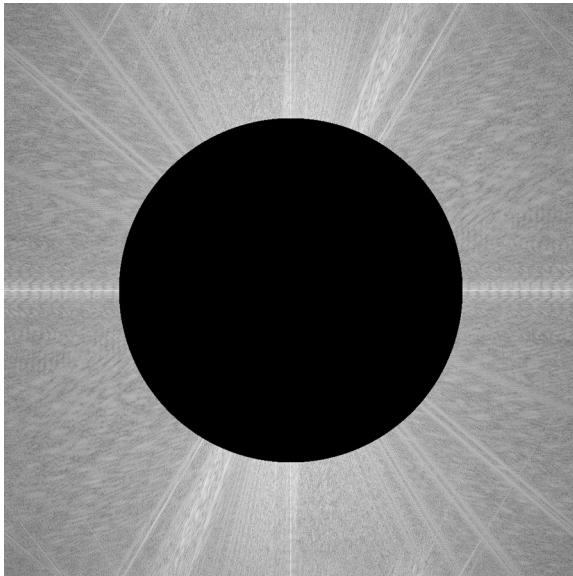


**Img12:** spectre de l'image original avec application de filtre

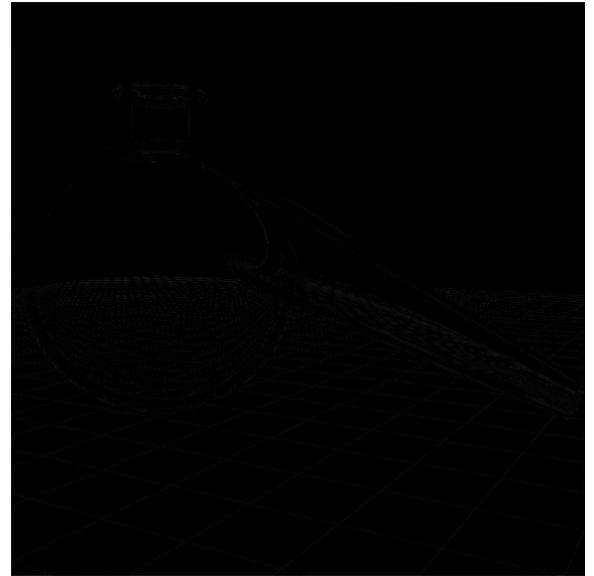


**Img13:** image issue après la détermination de la transformée de Fourier inverse du spectre filtré.

## 2- Expérimentation avec une fréquence de coupure =0.60



**Img14:** spectre de l'image original avec application de filtre



**Img15:** image issue après la détermination de la transformée de Fourier inverse du spectre filtré.

### Analyse des résultats de l'expérience:

Pour notre expérience avec le filtre passe-haut, nous avons appliqué une fréquence de coupure =0.30 et une autre fréquence de valeur 0.60.

Pour le premier cas, nous constatons que l'image issue après la détermination de la transformée de Fourier inverse du spectre filtré devient très noircie par rapport à l'image d'origine, mais, on peut toutefois distinguer plus ou moins bien les informations relatives à celle-ci. Ce qui explique que les informations les plus pertinentes véhiculées par l'image de départ sont perdues. Alors, cette coupure prend en compte la valeur des hautes fréquences et aussi une partie des valeurs de basses fréquences.

Avec le cas de la fréquence de coupure =0.60, l'image devient encore plus noircie par rapport à celle qui a subi la fréquence de coupure =0.30. Mais toutefois, les contours sont toujours présents.

### Conclusion

Enfin, après l'application du fourier et des traitements fréquentiels sur les images, nous avons fait des remarques assez pertinentes.

Pour la transformée de fourier inverse, l'image de la sortie reste identique à l'image d'entrée;

Pour le filtre passe-bas, autant que la valeur de la coupure est élevée, l'image paraît plus nette et plus identique à celle du départ. Pourtant, avec le filtre passe-haut, autant que la fréquence de coupure est proche de **0**, les informations concernant l'image sont présentes, alors qu'autant qu'elle est proche de **1**, l'image a tendance à perdre des informations.

## Références bibliographiques

1- Learning OpenCV (O'Reilly), Discrete Fourier Transform (DFT), pp. 177-180

2- Exemples d'utilisation de cvDFT avec OpenCV

3-

[http://docs.opencv.org/doc/tutorials/core/discrete\\_fourier\\_transform/discrete\\_fourier\\_transform.html](http://docs.opencv.org/doc/tutorials/core/discrete_fourier_transform/discrete_fourier_transform.html)

4- Introduction to Fourier transforms for image processing, par John M. Brayer (Université du Nouveau Mexique, USA)