

Institut de la Francophonie pour l'informatique

Rapport de traitement d'images

TP2 : Transformée de Fourier et traitements fréquentiels

1^{re} PARTIE : Transformée de Fourier

La transformée de Fourier est un outil de traitement d'image important qui est utilisé pour décomposer une image dans ses composantes sinus et cosinus. La sortie de la transformation représente l'image dans le domaine de Fourier ou de fréquence, tandis que l'image d'entrée est l'équivalent du domaine spatial. Dans l'image du domaine de Fourier, chaque point représente une fréquence particulière contenue dans l'image du domaine spatial.

Cette première partie de notre TP traite de la transformée de fourier et de sa transformée inverse d'une image.

Afin de pouvoir mettre en œuvre cette méthode de traitement, nous avons procédé à son implémentation. Notre programme a été réalisé en c++ et le nom du fichier est « **Trans_fourier.cpp** ».

Pour son utilisation, se placer dans le répertoire du fichier « **Trans_fourier.cpp** » et le fichier makefile » puis suivre les instructions suivantes :

- 1) Copier les images que vous désirez traiter dans le même répertoire que le programme ;
- 2) Ouvrir un terminal et taper « **make** » pour compiler et créer un exécutable ;
- 3) Saisir « **./Trans_fourier nom_image** ».

Pour mieux comprendre le fonctionnement de notre programme, nous utiliserons une image de niveau gris non bruitée et une autre image de niveau de gris bruitée.

1^{er} Cas : Image sans bruit

Les résultats obtenus sur l'image nommée ‘image.tif’ ci-dessous sont obtenus par les commandes suivantes:

- **make**
- **./Trans_fourier image.tif**

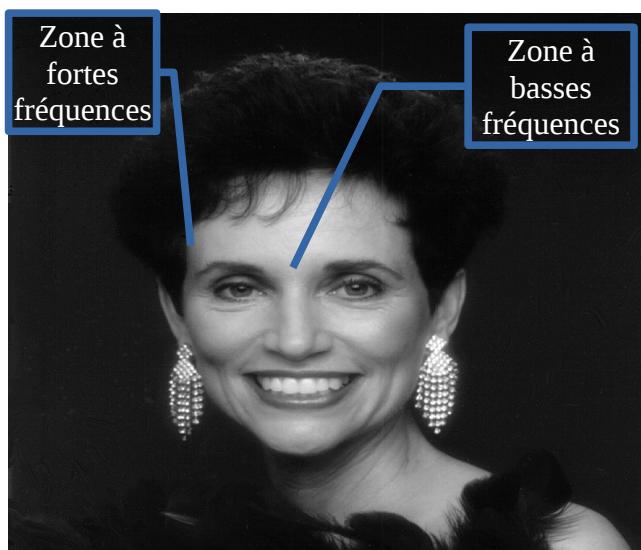


Illustration 2: image d'entrée

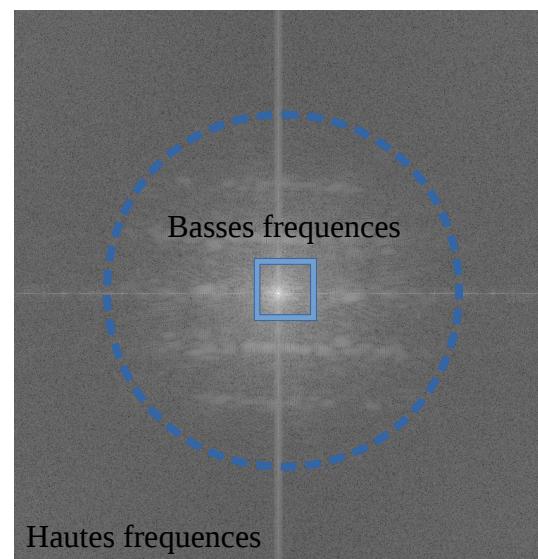


Illustration 1: Spectre de la transformée de fourier



Illustration 3: Transformée inverse de fourier

La transformée de fourrier nous montre un point lumineux au milieu du spectre. Ce point correspond à la composante continue ou encore la moyenne de l'intensité de l'image initiale. Le spectre nous montre également que l'image contient des composantes de toutes les fréquences, mais que leur amplitude est plus faible pour les fréquences plus élevées(zone éloignée du centre). Par conséquent, les basses fréquences (zone proche du centre) contiennent plus d'informations d'image que les plus élevées.

La ligne verticale et horizontale passant par le centre proviennent des motifs réguliers dans le fond de l'image originale.

Quand au résultat issu de la transformée inverse, nous constatons qu'il correspond à l'image d'entrée.

2^e Cas : image bruité(bruit sinusoïdal)

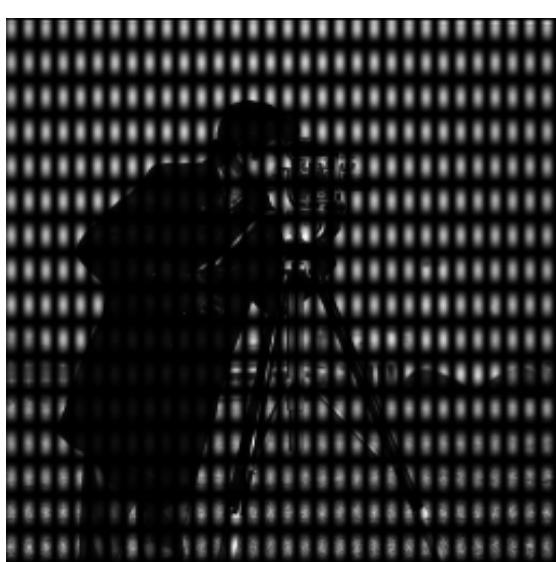


Illustration 5: Image d'origine

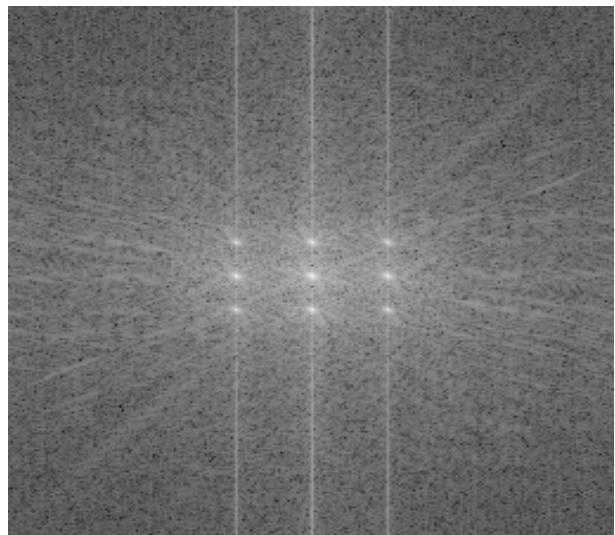


Illustration 4: Spectre de la transformée de fourrier



Illustration 6: Transformée inverse de fourrier

L'image d'origine est constituée de sinus dans les deux directions x et y. Dans la transformée de fourrier nous obtenons 9 points en dehors du point central. Ces points représentent la périodicité dans les sens horizontal, vertical et diagonal. Les autres points d'intensité plus faible sont dus à des effets de bord. La présence de plusieurs lignes dans presque tous les sens. Cela s'explique par le fait que l'image d'origine possède de les variations de motifs du fond même de l'image d'origine. De même que dans le 1^{er} cas, les basses fréquences sont celles situées à proximité du centre, et les hautes fréquences sont celles éloignées du centre de l'image.

2^e PARTIE : Traitements fréquentiels

Dans cette partie, notre travail consistera à implémenter et tester des traitements fréquentiels sur des images. Ce sont entre autres le filtre passe-bas, le filtre passe-haut et la correction de bruits sinusoïdaux.

Pour la mise en œuvre de ces différentes méthodes de traitements, nous avons implémenter un programme dont le nom du fichier est **fourier_filtres.cpp** et le **makefile**. Le programme fonctionne comme suite :

Tout d'abord, se placer dans le répertoire contenant les différents fichiers cités ci-dessus et suivre les instructions suivantes :

- 1) Copier les images que vous désiré traiter dans le même répertoire courant;
- 2) Ouvrir un terminal et taper « **make** » pour compiler et créer un exécutable;
- 3) Saisir « **./fourier_filtres nom_image** » ;
- 4) Choisir le type de filtre que vous désirer utiliser : 1 pour filtre passe-bas et 2 pour filtre passe-haut;
- 5) Entrer la valeur de la fréquence de coupure comprise entre 0 et 1.

1^{er} Cas : Filtre Passe-bas

Afin de mieux appréhender la notion de la fréquence de coupure, nous allons l'utiliser dans deux exemples avec de valeurs différentes.

Les résultats obtenus sont les suivants :



Illustration 7: image d'origine

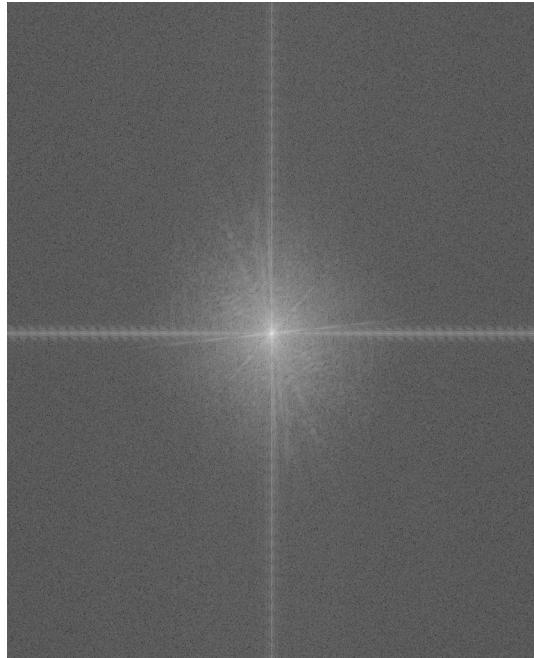
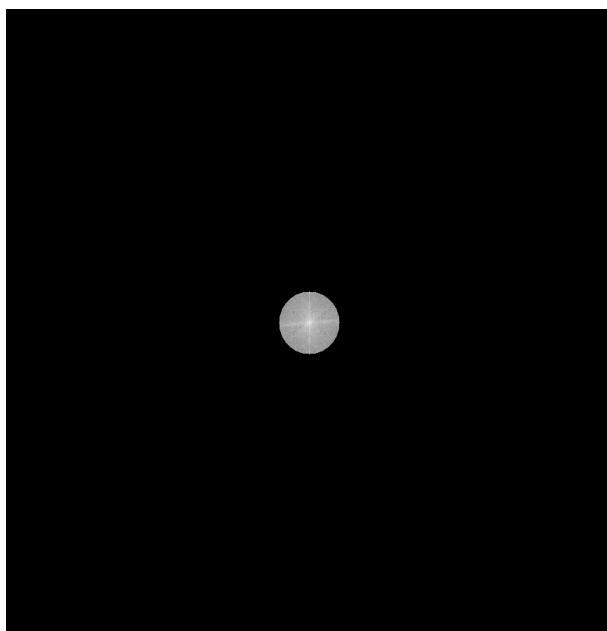


Illustration 8: spectre de la transformée de fourrier



*Illustration 9: mise à 0 des hautes fréquences,
 $FC=0,2$*



*Illustration 10: transformée inverse de
fourrier avec $fc=0,2$*

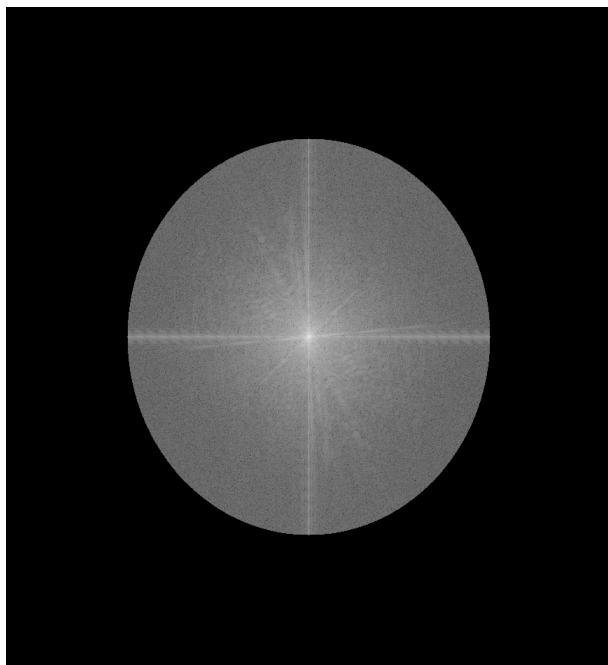


Illustration 11: mise à 0 des hautes fréquences, $FC=0,6$



Illustration 12: transformée inverse de fourrier avec $fc=0,6$

L'illustration 10 et 12 correspondent respectivement aux résultats obtenus par transformée inverse de fourrier par application d'une fréquence de coupure $fc=0,2$ et de $0,6$ sur le spectre. Visiblement, l'image obtenue pour $fc=0,6$ est bien visible avec des contours plus forts par rapport à celle obtenue avec $fc=0,2$ qui est peu lissée avec des contours faibles rendant ainsi l'image floue. Nous pouvons donc conclure que plus la valeur de fc est élevée, plus les basses fréquences sont prises en compte et plus on a tendance à entrer dans les hautes fréquences, ce qui explique la visibilité améliorée de l'image l'image.

2^e Cas: Filtre Passe-haut

Dans la même logique que précédemment, nous allons utiliser le filtre passe-haut avec des fréquences de coupures différents. Les résultats obtenus sont les suivants :

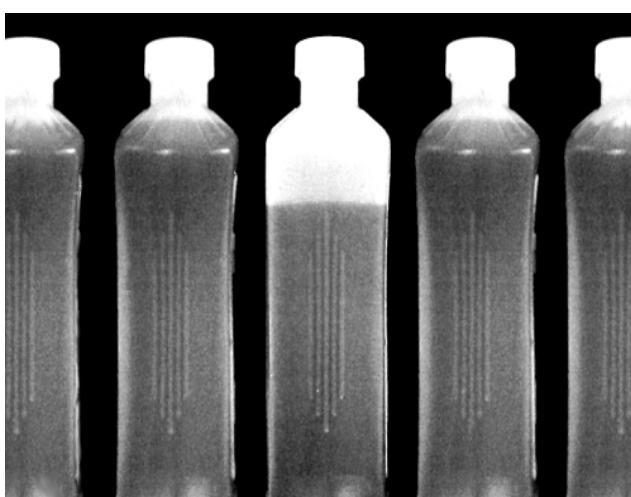


Illustration 13: image d'origine

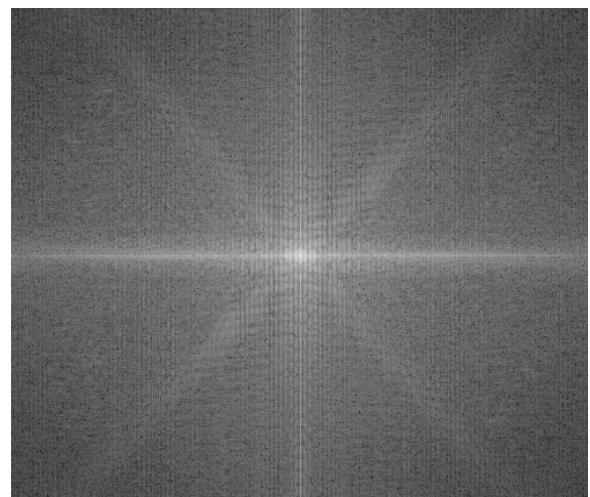


Illustration 14: spectre de la transformée de fourrier

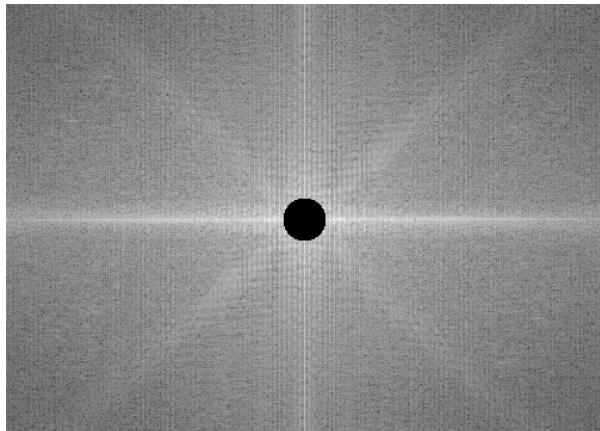


Illustration 15: mise à 0 des basses fréquences, $FC=0,1$



Illustration 16: transformée inverse de fourrier avec $fc=0,1$

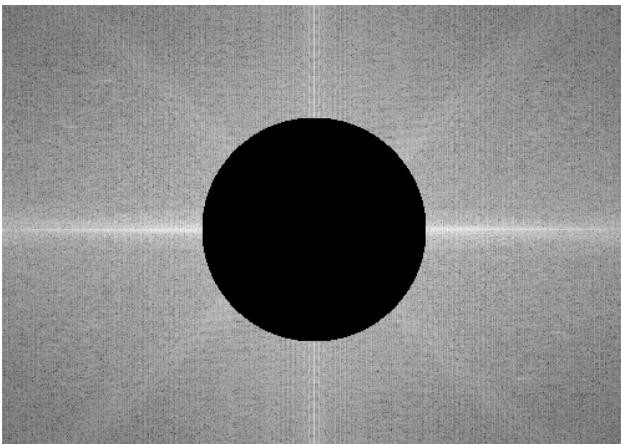


Illustration 17: mise à 0 des basses fréquences, $FC=0,5$

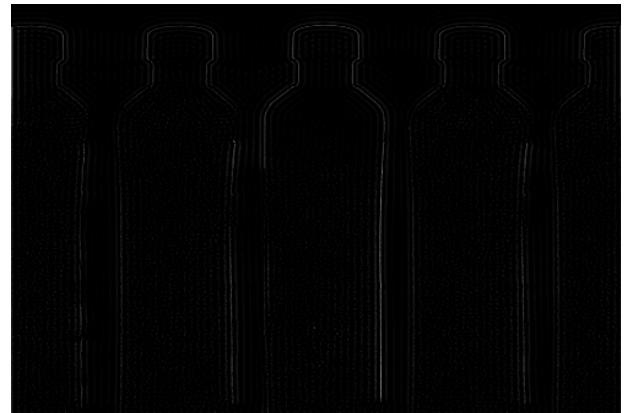


Illustration 18: transformée inverse de fourrier avec $fc=0,5$

En utilisant des fréquences de coupures $fc=0,1$ et $fc=0,5$, nous obtenons respectivement les illustrations 16 et 18 après la transformée inverse de fourrier. Nous constatons que plus la valeur de fc est élevée, plus nous avons tendance à perdre la majeure partie de l'information contenue dans l'image, seulement une partie des bordures demeurent visibles. Cela provient du fait que la partie du spectre contenant l'essentielle des informations sur l'image a été mise à 0. Il s'agit ici des basses fréquence, la zone proche du centre de l'image.

Plus la zone de basses fréquences est mise à 0, plus on perd l'information sur l'image.

En résumé, nous pouvons retenir que :

- Avec les filtres passe-bas, plus la fréquence de coupure fc est grande, plus on regroupe le maximum des basses fréquences de l'image. Ainsi, nous obtenons donc une image plus proche à celle d'origine.
- Avec les filtres passe-haut, lorsque la fréquence de coupure fc est élevée, nous quittons la zone de basses fréquences, c-à-d que qu'elles sont mises à 0. Ainsi nous perdons donc le maximum d'information contenues dans l'image sauf les contours que sont représentés par les hautes fréquences.

En conclusion, nous pouvons dire que ce travail nous a permis de mieux cerner le concept de la transformée de Fourier permettant d'explorer la composition fréquentielle d'une image, et de par ses propriétés, de lui appliquer des opérateurs de filtrage. L'utilisation des filtres nous permet d'obtenir des informations sur une image et leur application se basera sur les objectifs recherchés.

REFERENCES

- 1)http://docs.opencv.org/doc/tutorials/core/discrete_fourier_transform/discrete_fourier_transform.html
- 2) [Introduction to Fourier transforms for image processing](#)