

TP2 : Transformée de Fourier et Traitement Fréquentiel

1 Introduction

Ce rapport fait suite au projet 2 du module traitement d'image sur la transformée de Fourier et du traitement fréquentiel. Il fait la description du travail effectué et de l'analyse des résultats issue de ce travail. Dans ce projet nous avons conçu deux programmes l'un qui nous permet de déterminer la transformée de Fourier et son inverse et l'autre nous permet d'appliquer un filtre (Passe-Bas ou Passe-Haut) à une image d'entrée et d'interpréter le résultat pour réduire les bruits ou faire ressortir les contours de l'image. Ces deux programmes sont tous fonctionnels, leurs utilisations sont expliquées dans les différentes parties où ils sont utilisés.

2 partie 1 : Transformée de Fourier

Pour cette partie du TP le programme conçu est la (`transf_fourier.cpp`) et qui prend en paramètre l'image qui est entrée par l'utilisateur. ainsi ci dessous les étapes de son utilisation :

- Compilation : il faut se placer dans le dossier du TP au moyen de la commande `cd` puis exécuter la commande `make` pour compiler le programme après cette opération un fichier sera généré qui va vous permettre d'exécuter le programme.
- Exécution : le programme est exécuté via la commande `./Transf_fourier` suivi du nom de l'image.

Interprétations des résultats :

1. Exemple 1 : *cas d'une image normale* Remarquer ci-dessous les résultats des tests que nous avons réalisé avec le programme :

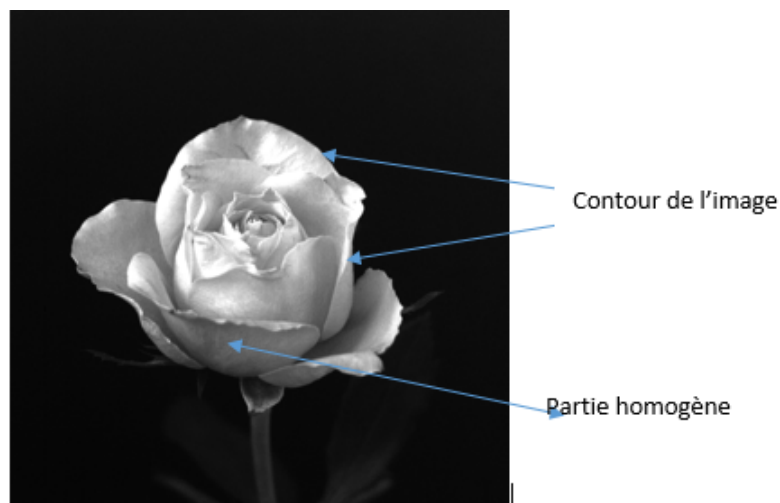


FIGURE 1 – Image Originale

Grâce aux images ci-dessus, nous constatons que l'image *originale* et la *transformée inverse de Fourier* sont les mêmes. Cela s'explique alors par le fait qu'il n'y a pas eu de traitement sur le spectre qui est obtenue après application de la transformée de Fourier.

L'image de la transformée de Fourier nous montre que l'image originale contient des composantes de toutes les fréquences (Basse et Haute fréquences), mais que leur amplitude est plus petite pour des fréquences plus élevées. Par conséquent les basses fréquences contiennent



FIGURE 2 – Transformée inverse de Fourier

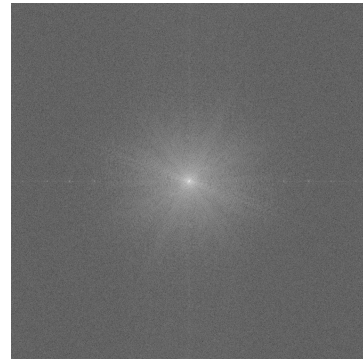


FIGURE 3 – Image de la Transformée de Fourier

plus d'informations de l'image que les plus élevées. L'image transformée nous dit aussi qu'il y'a plusieurs directions dans l'image de Fourier.

La zone lumineuse au centre de l'image est la zone qui renferme les informations importantes de l'image, cette zone correspond alors à la zone homogène de l'image qui correspond à la partie basse fréquence. Quant au point lumineux centre il correspond à l' origine du repère.

2. Exemple 2 : cas d'une image bruitée :

Dans ce second cas notre image d'entrée contient des bruits sinusoïdaux ce qui rend sa transformée de Fourier un peu particulière par rapport à celle des images simples.

Nous remarquons trois pics sur la transformée de Fourier celui du milieu correspond à l' origine du repère et puisque l'image de Fourier est symétrique à son centre, les deux points correspondant à la fréquence des bandes dans l'image originale. Notez que les deux points sont situés sur une ligne verticale à travers le centre de l'image, parce que l'intensité de l'image dans le domaine spatial change .

Nous remarquons aussi des traces verticales et obliques, mais ces derniers sont plus nombreuses ce qui est logique vu qu'ils vont dans le même sens que les lignes de l'image originale.



FIGURE 4 – Image Bruitée Originale

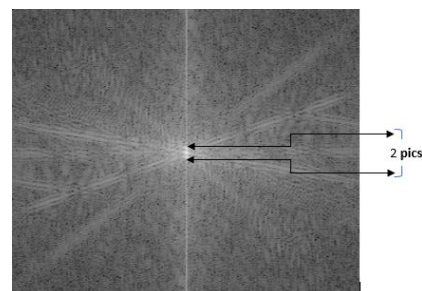


FIGURE 5 – Transformée de Fourier

Nous remarquons aussi comme dans notre 1 er Exemple la densité des informations au centre qui correspond à la partie basse fréquence et une fluidité d'information à la périphérie qui correspond à la partie haute fréquence. De plus nous remarquons que l'image de la transformée inverse et celle de l'image originale est la même .



FIGURE 6 – Transformée Inverse de Fourier

3 partie 2 : Traitement Fréquentiel

Pour les besoins de cette partie qui est le traitement fréquentiel nous avons implémenté un programme du nom de `fontion_filtres.cpp` qui nous permettra de faire les filtres passe-bas et passe-haut à une image bien donnée. l'utilisation de cette dernière se fait de la manière suivant :

- Compilation : se placer dans le dossier du TP au moyen de la commande `cd`, puis exécuter la commande `make`.
- Exécution : l'étape précédente nous permet de généré un fichier exécutable qui va nous permettre d'exécuter le programme. Pour ce faire exécuter la commande `./fonction_filtres nom de l'image` et suivre les instructions du programme.

Interprétation des résultats :

- 1er choix : **filtre passe_bas** (prenons une fréquence de coupure égale à 0,15)



FIGURE 7 – Image Entrée

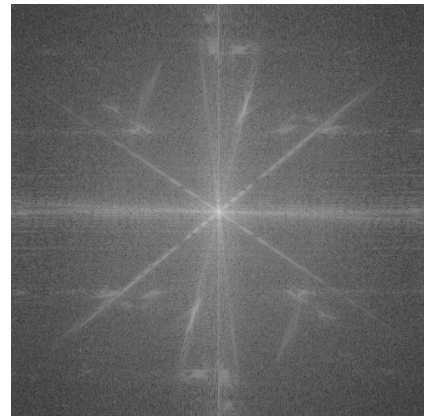


FIGURE 8 – Spectre transformée de Fourier

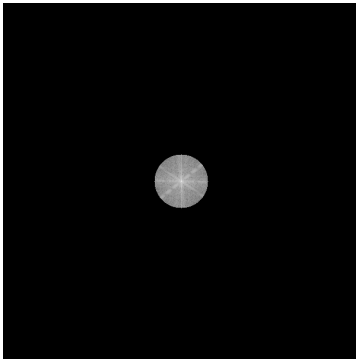


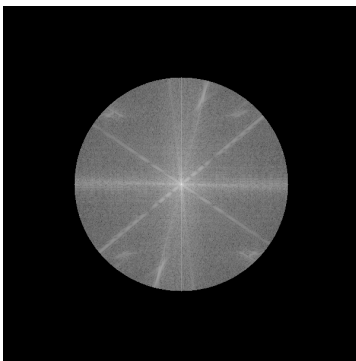
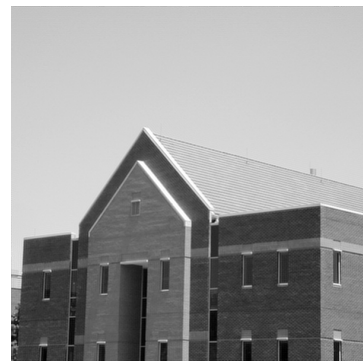
FIGURE 9 – Spectre Traité



FIGURE 10 – Image résultante

Les images ci-dessus sont en effet les résultats de l'exécution de notre programme avec un choix du filtre Passe-bas et une fréquence de coupure de 0.15. L'image entrée a subi trois étapes qui sont : la transformée de Fourier qui nous a permis de déterminer le spectre de l'image puis s'ensuit l'application du filtre sur le spectre pour pouvoir supprimer les hautes fréquences et enfin l'inverse de la transformée de Fourier pour pouvoir restituer l'image.

Après toutes ces étapes on obtiens une image flouée avec des contours moins nets et des petites granulations au tour de l'image tout ceci est dû à la suppression de la partie haute fréquence qui gère les contours. **Filtre Passe-Bas avec fréquence de coupe de 0.60**

FIGURE 11 – Spectre Traité avec $f=0.60$ FIGURE 12 – Image résultante avec $f=0.60$

En appliquant le même filtre avec une fréquence de coupure de 0.60 nous obtenons un meilleur rendement par rapport au précédent. Ceci résulte du fait que la fréquence de coupure s'est élargit en prenant des parties de la haute fréquence raison pour laquelle nous avons obtenu une image résultante proche de l'originale. Étant donné que ce filtre conserve un plus grand nombre de fréquences, plus de détails restent dans l'image de sortie. L'image est moins flouée, mais contient aussi de bruit. La granulation est moins élevée, mais existe toujours.

Filtres Passe-Haut avec une fréquence de coupure de 0.15



FIGURE 13 – Image Originale

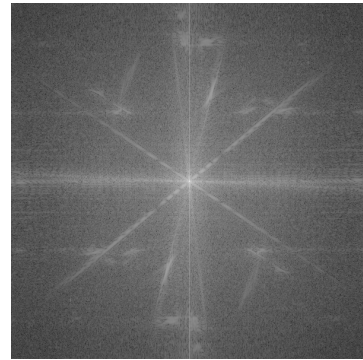


FIGURE 14 – Spectre Fourier

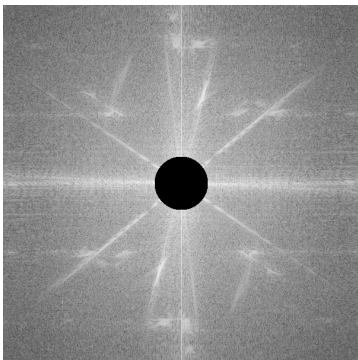


FIGURE 15 – Image Spectre Traité

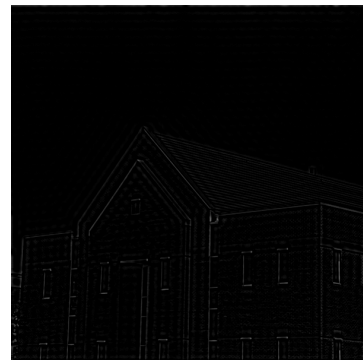


FIGURE 16 – Image Restaurée

Filtres Passe-Haut avec une fréquence de coupure de 0.8

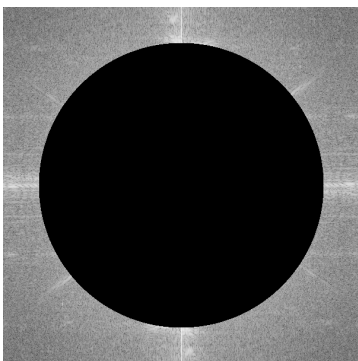
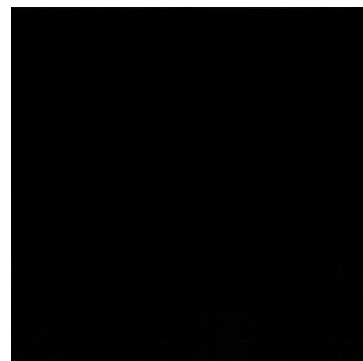
FIGURE 17 – Image Spectre Traité ($f=0.8$)

FIGURE 18 – Image résultante

Pour cette partie nous appliquons le filtre passe-haut. Nous remarquons alors que pour une fréquence de coupure à 0,15 l'image de la transformée inverse ne fait ressortir que les contours de l'image originale mais nous distinguons une fois encore assez bien les informations qui y sont contenues. Malgré que cette fréquence est faible nous distinguons une partie homogène de l'image, ce qui veut dire qu'elle prend en compte une partie de la partie basse fréquence, mais beaucoup d'information de l'image sont perdues. Avec une fréquence de coupure de 0.8 nous constatons que l'image est plus sombre que la précédente, car moins de basses fréquences sont prises en comptes. On peut aussi faire un constat plus la fréquence de coupure est élevée plus l'image devient encore plus sombre (méconnaissable) et plus il tend vers 0 l'image est très visible, car en tendant vers zéro elle tend vers l'homogénéité de l'image. Nous pouvons enfin dire que ce filtre permet à la détection de contours.

4 Conclusion

Au regard de tout ce qui précède nous pouvons dire alors que ce travaux pratique nous a sans doute permis de mieux comprendre les enjeux du fonctionnement de Fourier qui permet de déterminer le spectre d'une image sur un plan de fréquence et sa reconstitution a partir de son spectre.

Ainsi nous avons compris le principe de fonctionnement des filtres passe-bas et passe-haut qui permettent de mettre en exergue la partie homogène de l'image et les contours de l'image, les basses fréquences sont aussi associées aux parties homogène de l'image et les hautes fréquences les contours de l'image.