

UNIVERSITÉ NATIONALE DU VIETNAM, HANOÏ

INSTITUT DE LA FRANCOPHONIE POUR L'INNOVATION

Option : Systèmes Intelligents et Multimédia (SIM)

Classe : Master 1

Module : Traitement d'images.

TP2: TRANSFORMÉE DE FOURIER ET TRAITEMENTS FRÉQUENTIELS

Rédigé par : **OUBDA Raphaël Nicolas Wendyam**

Promotion 22 .

Enseignant : NGUYEN Thi Oanh – IPH / oanhnt@soict.hust.edu.vn

INTRODUCTION

Le présent rapport présente l'analyse des résultats de traitement d'images sur les travaux pratiques N°2. En effet, nous évoquons la transformée de Fourier qui est une méthode qui permet de transformer l'information temporelle d'une image en une information fréquentielle et vice versa. Nous évoquons aussi le traitements des filtres passe-bas et des filtres passe-haut.

Tous les travaux pratiques ont été réalisés sur le système d'exploitation Linux 16.0.4 avec l'éditeur de texte gedit. Les programmes ont été écrits en c++ et en utilisant la librairie opencv.

En effet notre rapport comporte deux grandes parties. Dans la première partie nous aborderons la transformée de Fourier et dans la deuxième partie les traitements fréquentiels.

Fonctionnement du programme : Pour lancer le programme, nous ouvrons un terminal puis à l'aide de la commande `cd` nous allons dans le dossier de notre projet. Ensuite, nous compilons le programme avec la commande **`cmake .`**, et nous tapons la commande **`make`** pour créer l'exécutable. Enfin nous lançons le programme avec la commande **`./run`**. Notre programme prend en entrée le chemin d'une image pour son traitement. En effet nous saisissons **1** pour la transformée de Fourier et **2** pour le traitements fréquentiels.

```
*****
*
*   TRANSFORMEE DE FOURIER ET TRAITEMENTS FREQUENCIAIS   *
*
*****

*****1---Pour la Transformée de Fourier*****
*****2---Pour les traitements fréquentiels *****
VOTRE CHOIX: █
```

PARTIE 1: TRANSFORMÉE DE FOURIER.

Fonctionnement : Cette partie de notre programme prend en entrée le chemin d'une image à niveau de gris ou couleur et son extension.

Exemple: "home/utilisateur/image/image1.tif".

Il fournit en sortie, trois nouvelles images notamment l'image originale convertie à niveau de gris, l'image du spectre de la transformée de Fourier et l'image restaurée à partir du spectre. En effet, les trois images ont été créées et enregistrées automatiquement dans le répertoire en cours. La première image représente l'image initiale (**ImageInitiale.png**), la deuxième image représente le spectre de la transformée de Fourier (**fourier.png**) et la troisième représente la transformée inverse de Fourier ou restitution de l'image originale (**ImageRestituee.png**).

Pour la phase d'expérimentation, nous allons prendre dans un premier temps le cas d'une image sans bruit et le deuxième cas nous prendrons une image avec bruit.

■ 1e cas : image sans bruit.

Nous avons choisi l'image sans bruit à niveau de gris {z.png}.

Résultats obtenus

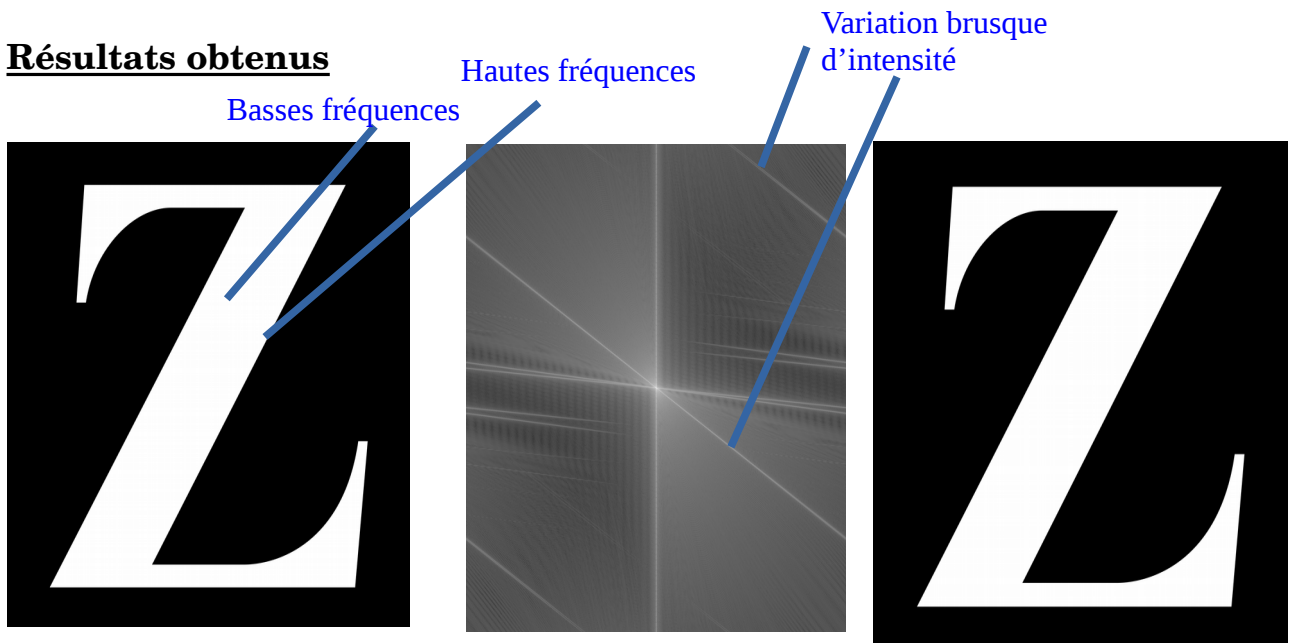


Fig 1: Image originale

Fig 2: Spectre de Fourier

Fig 3: Image de la transformée inverse de Fourier

Analyse du résultat

Analyse du spectre de Fourier:

Le spectre contient les basses et hautes fréquences.

- ◆ Le point lumineux au centre de l'image correspond à l'origine du spectre et représente aussi la valeur moyenne des pixels de l'image d'origine.
- ◆ la zone autour du centre est celle où se concentre la majeure partie de l'information véhiculée par l'image d'origine. Elle représente les basses fréquences qui correspondent aux régions homogènes de l'image originale ;
- ◆ les régions éloignées du centre du spectre qui sont sous forme de lignes correspondent aux hautes fréquences qui représentent les variations brusques d'intensité des pixels de l'image originale. Sur l'image originale, cette partie représente les contours, les frontières entre les couleurs noire et blanche.
- ◆ Nous constatons aussi des lignes(5) sur le spectre qui représentent les différentes variations brusque d'intensités lumineuses. Ces lignes correspondent aux différentes parties qui constituent le « Z » de notre image qui est composée de 5 parties aussi.

Analyse de la transformée inverse :

Nous constatons que la transformée de Fourier inverse donne la même image que l'image originale.

En conclusion, des résultats obtenus, nous remarquons que l'image de la transformée de Fourier inverse correspond exactement à l'image d'origine donnée en entrée. Donc nous pouvons dire que la transformée de Fourier n'effectue aucun traitement sur l'image originale elle donne uniquement son spectre.

■ 2^e Cas : Image avec bruit.

Pour ce cas nous avons choisi l'image(**house.png**) avec bruit .

Résultats obtenus



Fig 4: Image originale avec bruit

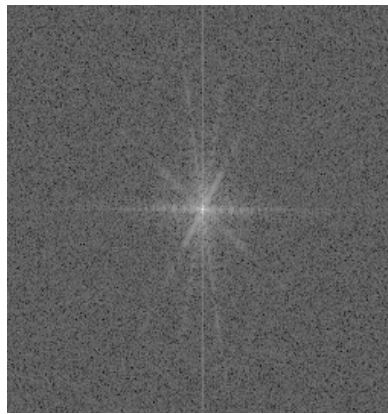


Fig 5: Spectre de la TF de l'image avec bruit



Fig 6: Transformée inverse de Fourier

Analyse des résultats

Nous constatons que l'image avec bruit présente un centre lumineux qui représente la moyenne de l'intensité des pixels. Les régions à proximités de ce point représentent les fréquences basses et les régions éloignées les fréquences hautes. Nous constatons qu'il n'y a pas trop de variations brusques de l'intensité sur le spectre. Donc les contours de l'image ne sont pas bien clairs sur le spectre, cela est dû aux bruits (points sur l'image) qui empêchent la détection des variations brusques de l'image qui constituent les contours (frontières) de l'image.

Pour la transformée inverse de Fourier, nous constatons que nous obtenons la même image que celle de l'origine, donc une image bruitée. Donc, la transformée de Fourier ne modifie pas l'image bruitée, aucun traitement n'est effectué sur l'image.

Conclusion de la première partie

La transformée de Fourier ne fait aucun traitement sur l'image, elle ne modifie pas l'image (avec ou sans bruit). Elle donne juste son spectre, c'est à dire elle transforme l'image de son domaine spatiale dans un domaine fréquentielle. Les hautes fréquences du spectre représentent les frontières, les contours de l'image tandis que les basses fréquences représentent les régions homogènes, flou.

La transformée inverse de Fourier donne exactement l'image d'origine, ce qui vient confirmer que la transformée de Fourier n'effectue aucun traitement sur les images.

PARTIE 2: TRAITEMENTS FRÉQUENTIELS

Pour cette partie du programme, il faut choisir (2) dans le programme principal comme le montre la figure ci-dessous. Cette partie permet de traiter fréquentiellement les différentes images. En effet, nous avons implémenté les filtres passe-bas et les filtres passe-haut. Le programme prend en entrée le chemin d'une image et son extension.

Exemple : /home/nicolas/Musique/TP_TI/TP2/Fourier/image.png

```

*****
*                                     *
*   TRANSFORMEE DE FOURIER ET TRAITEMENTS FREQUENCIAELS   *
*                                     *
*****

*****1---Pour la Tansformée de Fourier*****
*****2---Pour les traitements fréquentiels *****
VOTRE CHOIX: 2
Donner le nom de votre image y compris avec l'extension (.png,.tif,.jpg...) et s
on chemin d'accès
$ /home/nicolas/Musique/TP TI/TP2/Fourier/image.png

```

■ 1e Cas : Filtre Passe-bas.

Pour les filtres Passe-bas , après avoir renseigné l'extension et le nom de l'image, vous devez choisir le type de filtre (**1 Pour le filtre Passe-bas et 2 Pour Passe-haut**). Donc pour ce cas nous choisissons 1. Enfin nous choisissons la fréquence de coupure(f_c) qui est compris entre 0 et 1 ([0 ;1]).

✓ Expérimentation avec une image sans bruit.

Nous choisissons une image sans bruit . png pour nos tests.

Résultats obtenus.



Fig 7: Image originale

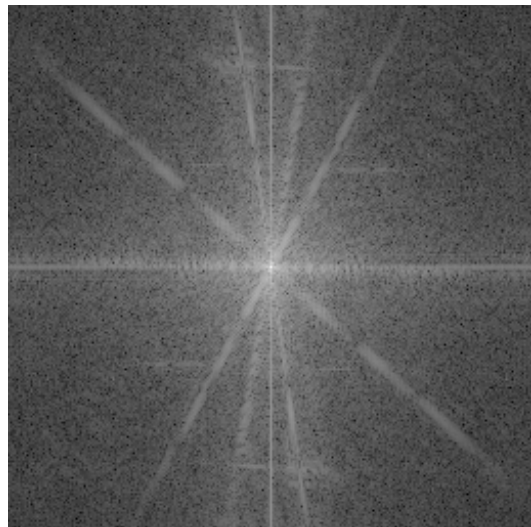


Fig 8: Spectre de l'image originale

a) Fréquence à 0.25

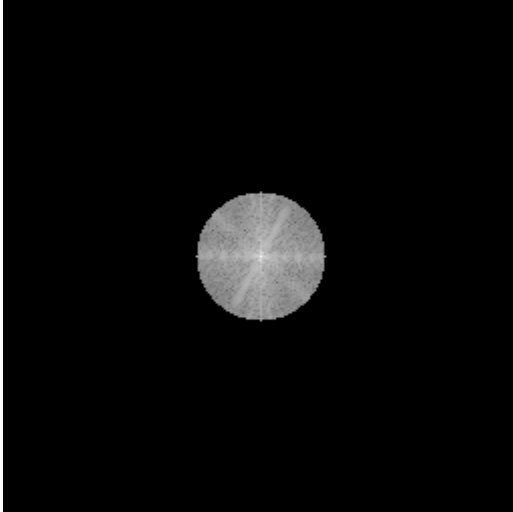


Fig 9: Spectre filtrée de l'image originale avec $f_c=0.25$



Fig 10: Image Filtrée avec $f_c= 0.25$

b) Fréquence à 0.75

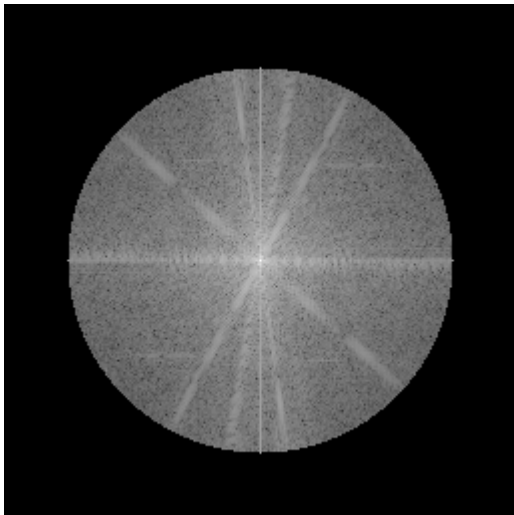


Fig 11: Spectre filtrée de l'image originale avec $f_c= 0.75$



Fig 12: Image Filtrée avec $f_c=0.75$

Analyse des résultats obtenus.

Dans cette partie nous avons appliqué le filtre passe-bas avec deux fréquences de coupure différentes. Nous constatons que:

Pour **$f_c=0.25$** (Fig 9 et Fig10), l'image devient flou et les contours deviennent moins nettes. En effet, le filtre laisse passer les basses fréquences les plus proches de l'origine du spectre (qui correspondent aux parties homogènes de l'image) et met à 0 les valeurs des pixels en dehors du rayon de coupure .

Pour **$f_c=0.75$** (Fig 11 et Fig12), nous constatons une amélioration de la qualité de l'image par rapport à **$f_c=0.25$** . L'image filtrée ressemble à l'image originale. En

effet, la fréquence de coupure est élevée donc les hautes fréquences (Variations brusques d'intensité) sont prises en compte c'est pour cela que l'image filtrée est claire et les contours aussi sont présentes.

Nous pouvons dire que dans le cas des filtres passe-bas plus la fréquence de coupure est proche de 1 plus l'image filtrée ressemble à l'image d'origine car les hautes fréquences sont prises en comptes. Ainsi, les fréquences basses contiennent la plus grande partie de l'énergie d'une image. Donc nous pouvons conclure que les informations importantes d'une image se retrouvent dans les basses fréquences et les filtres passe-bas laissent passer les basses fréquences et bloquent les hautes fréquences.

■ 2^e Cas : Filtre Passe-haut.

Pour les filtres Passe-haut, après avoir choisit 2 pour le traitement fréquentiels, renseigner l'extension et le nom de l'image, vous devez choisir le type de filtre (1 Pour le filtre Passe-bas et 2 Pour Passe-haut). Donc pour ce cas nous choisissons 2. Enfin nous choisissons la fréquence de coupure(f_c) qui est compris entre 0 et 1 ($[0 ; 1]$).



Fig 1: Image originale

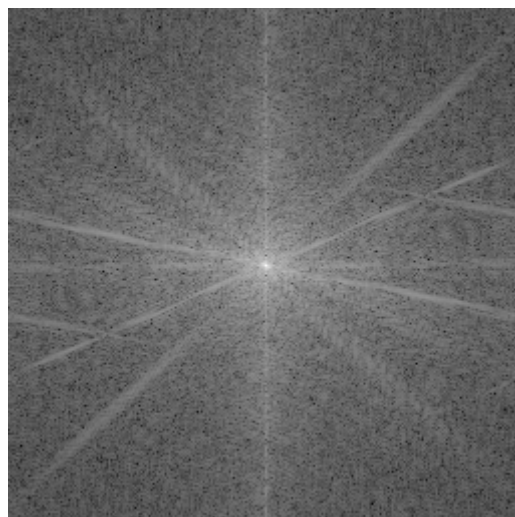


Fig 2: Spectre de l'image originale

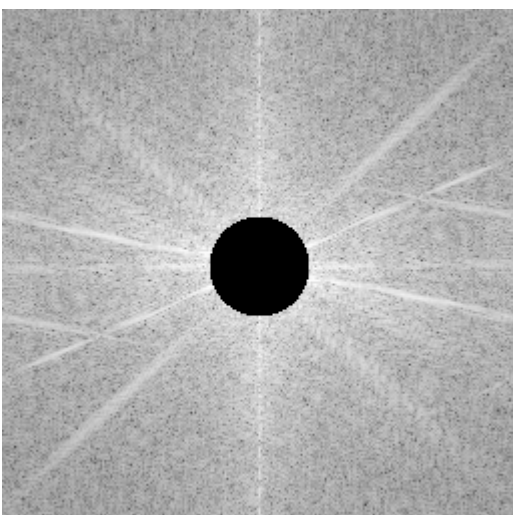


Fig 4: Spectre filtrée de l'image originelle avec $f_c=0.2$



Fig 3: Image de la transformée inverse du spectre filtrée avec $f_c=0.2$

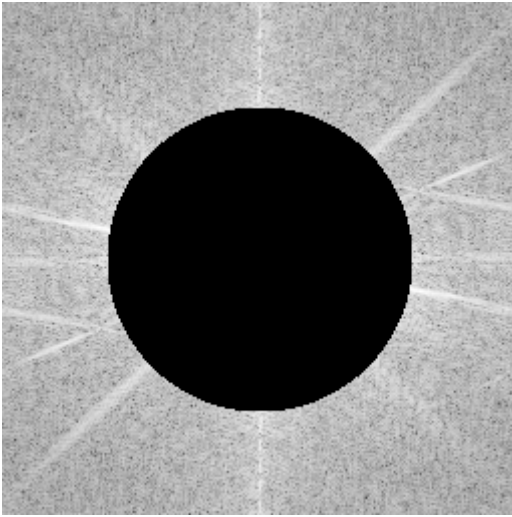


Fig 5: Spectre filtrée de l'image originelle avec $f_c=0.6$



Fig 6: Image de la transformée inverse du spectre filtrée avec $f_c=0.6$

Analyse des résultats.

Nous constatons selon les résultats que:

Pour **$f_c=0.2$** , nous constatons que l'image devient noire, mais nous apercevons toujours les contours et quelques régions homogènes (basses fréquences). En effet, sur le spectre la région noire (basses fréquences) est mise à 0, seule la région grise du spectre est représentée sur la transformée inverse. De ce fait, cette fréquence de coupure laisse passer toutes les hautes fréquences et quelques basses fréquences.

Pour **$f_c=0.6$** , nous constatons qu'aucune région homogène (basses fréquences) de l'image n'est représentée. L'image est noire et nous n'apercevons que les contours de l'image. En effet, toutes les basses fréquences sont mises à 0, seule les hautes fréquences sont présentes ce qui explique la présence des contours.

En somme, dans le cas des filtres passe-haut, seules les hautes fréquences sont présentes, les basses fréquences sont bloquées. Plus la fréquence de coupure tend vers 0, plus l'image de la transformée inverse ressemble à l'image originale. Mais lorsque f_c tend vers 1, les basses fréquences sont bloquées, l'image devient noire et on ne voit que les contours de l'image.

Conclusion de la deuxième partie .

Les filtres passe-bas bloquent les hautes fréquences et laissent passer les basses fréquences. Les basses fréquences renferment l'information importante d'une image, ils représentent les régions homogènes de l'image.

Les filtres passe-haut bloquent les basses fréquences et laissent passer les hautes fréquences. Les hautes fréquences représentent les contours de l'image, le changement brusque d'intensité de l'image.

CONCLUSION GÉNÉRALE

Durant notre TP nous avons créé un programme en C++ qui fonctionne. Le programme permet d'effectuer la transformée de Fourier et le traitement fréquentiel des images. En effet , selon les résultats obtenus lors des phases d'expérimentations, nous pouvons dire que la transformée de Fourier n'effectue aucun traitement sur les images elle donne uniquement le spectre de l'image pour un éventuel traitement(filtrage). Nous pouvons ajouter aussi que les basses fréquences contiennent la partie essentielle de l'information d'une image tandis-que les hautes fréquences représentent les contours d'une image.

Références.

http://docs.opencv.org/doc/tutorials/core/discrete_fourier_transform/discrete_fourier_transform.html
https://docs.opencv.org/3.4.0/df/d65/tutorial_table_of_content_introduction.html
https://docs.opencv.org/3.4.0/d9/df8/tutorial_root.html