3A IE

TP Image

CHAPUS Louka

Table des matières

[1. TP n°2 3](#_Toc135131325)

[1. Fourrier 2D 3](#_Toc135131326)

[a. Algorithme 3](#_Toc135131327)

[b. Harmoniques pures 3](#_Toc135131328)

[c. Contour 5](#_Toc135131329)

[d. Texture 7](#_Toc135131330)

[2. Le phénomène de repliement 9](#_Toc135131331)

# 1. TP n°2

## 1. Fourrier 2D

### Algorithme

Une solution pour calculer la transformée de Fourrier 2D est de calculer une première fois la transformée 1D sur les lignes puis de calculer la transformée 1D sur les colonnes de l’image précédente. Et on obtient comme cela la transformé 2D d’une image.

### Harmoniques pures

Pour commencer, on créait une image de 128 x 128 avec une fréquence d’oscillation de 0.1 selon n, 0 selon m et une fréquence fe à 1.

f\_1 = 0.1

f\_2 = 0

f\_e = 1

img=atom(128,128,f\_1,f\_2);

affiche\_image(img)

fourier2d(img,f\_e)

Figure [1] code utilisé

On appelle simplement les fonctions qui sont donné dans le sujet avec les bonnes valeurs et on obtient l’image suivante :

Une image contenant graphique

Description générée automatiquement

Figure [2], image avec

Une image contenant graphique

Description générée automatiquement

Figure [3], Spectre de la figure [2]

On remarque que lorsqu’on a une image avec des lignes verticales alors on obtient un spectre qui ne contient uniquement une ligne horizontale avec deux pics d’intensités à -0,1 et 0,1. On a donc ensuite essayé différentes fréquences d’oscillation mais sans changer la fréquence d’échantillonnage.

Une image contenant graphique

Description générée automatiquement

Figure [4], image avec ainsi que son spectre

Une image contenant graphique

Description générée automatiquement

Figure [5], image avec ainsi que son spectre

Une image contenant graphique

Description générée automatiquement

Figure [6], image avec ainsi que son spectre

On constate bien avec les figures [2] à [6] que lorsque l’image a des lignes verticales on obtient un spectre avec une ligne horizontale et inversement. De plus pour les images qui ont des lignes en diagonales on voit sur le spectre qu’il y a une ligne verticale et une ligne horizontale ce qui correspond bien à une diagonale.

### Contour

On se propose par la suite d’analyser les contours d’une image à l’aide de 3 images créés avec le code de la figure [7]. On va faire une image avec des lignes verticales, une avec des lignes horizontales et une avec des lignes en diagonales pour bien voir la différence dans les spectres des trois images.

f\_e = 1

rupt0=atom(128,128,0.05,0);

rupt1=atom(128,128,0,0.05);

rupt2=atom(128,128,0.05,0.05);

affiche\_image(rupt0)

affiche\_image(rupt1)

affiche\_image(rupt2)

fourier2d(rupt0,f\_e)

fourier2d(rupt1,f\_e)

fourier2d(rupt2,f\_e)

Figure [7] code utilisé pour générer les images et les spectres

Une image contenant graphique

Description générée automatiquement

Figure [8] Image avec le contour vertical

Une image contenant graphique

Description générée automatiquement

Figure [9] Image avec le contour horizontal

Une image contenant graphique

Description générée automatiquement

Figure [10] Image avec le contour oblique

On voit qu’a chaque fois pour le spectre il y a deux pics d’intensité qui correspondent à la fréquence des oscillations. De plus tous les spectres contiennent des lignes verticales, horizontales ou les deux, ce qui correspond à la rupture des contours sur les images de bases.

### Texture

On va maintenant analyser les transformées de Fourrier 2D de différentes textures et en déduire les paramètres les plus importants. On va donc utiliser le code de la Figure [11] juste en changeant le nom de l’image à chaque fois pour avoir l’image souhaiter.

#Ouvre une image

img = cv.imread("C:/Users/louka/Desktop/Cours/TP\_Image/imagesTP/Leaves0012GP.png")

img = cv.cvtColor(img,cv.COLOR\_RGB2GRAY)

affiche\_image(img)

fourier2d(img,f\_e)

Figure [11] code pour les textures

Une image contenant graphique

Description générée automatiquement

Figure [12] image Metal0007GUne image contenant graphique

Description générée automatiquement

Figure [13] image Water0000G

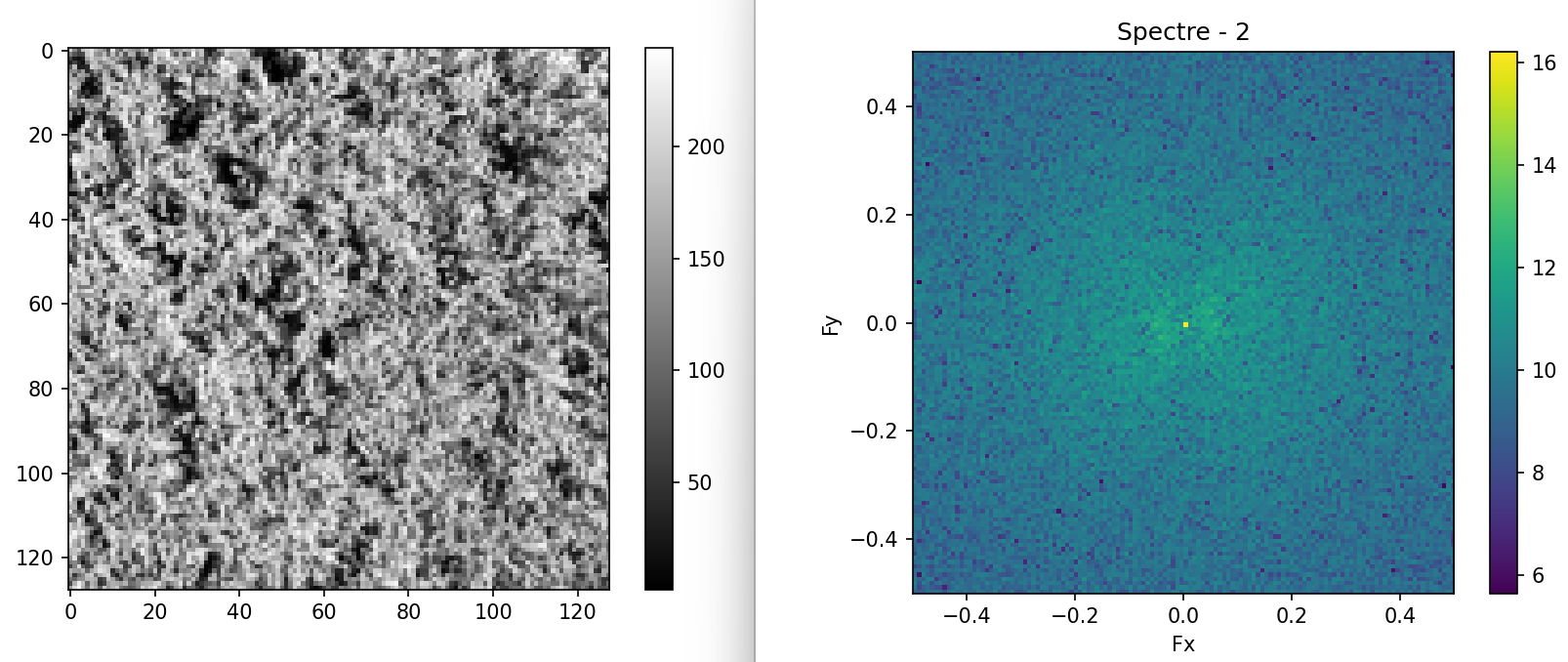


Figure [14] image Leaves0012G

On a testé différentes textures. On remarque bien que le spectre est totalement différent pour chaque image. Pour la Figure [12] on voit assez bien qu’il y a des lignes presque verticales et presque horizontales sur le spectre, ce qui indique que dans l’image de base il y a des lignes quasiment verticales et horizontales. Pour la Figure [13], on voit vaguement une ligne verticale se dessiner donc cela nous indique qu’il y a quelques lignes horizontales dans l’image de base mais vraiment pas beaucoup ou alors elles ne sont pas nettement dessinées. Enfin pour Figure [14], il n’y a aucune ligne ou forme vraiment claire donc pour cette image on ne peut pas conclure sur sa texture, on sait juste qu’il n’y a pas de ligne droite dedans.

Donc un paramètre important pour analyser les textures et de voir si le spectre contient des pics d’intensité et si le spectre contient des lignes assez nettes pour pouvoir conclure d’une certaine forme dans l’image de base.

## 2. Le phénomène de repliement

Pour cette dernière partie on se propose d’observer le phénomène de repliement spectrale dans le cadre d’une image. Pour ce faire on commence par créer une image comme indiquer dans le sujet à l’aide du code de la Figure [15].

img = atom(128,128,0.15,0.37);

affiche\_image(img)

fourier2d(img,f\_e)

#Utilise la fonction de binarisation du précédent TP

img\_binariser = binarisation\_manuelle(0.5,img)

affiche\_image(img\_binariser)

fourier2d(img\_binariser,f\_e)

image\_redimensionner = img[::2, ::2]

affiche\_image(image\_redimensionner)

fourier2d(image\_redimensionner,f\_e)

Figure [15] code utilisé pour générer les images

Une image contenant graphique

Description générée automatiquement

Figure [16] Image non binarisé

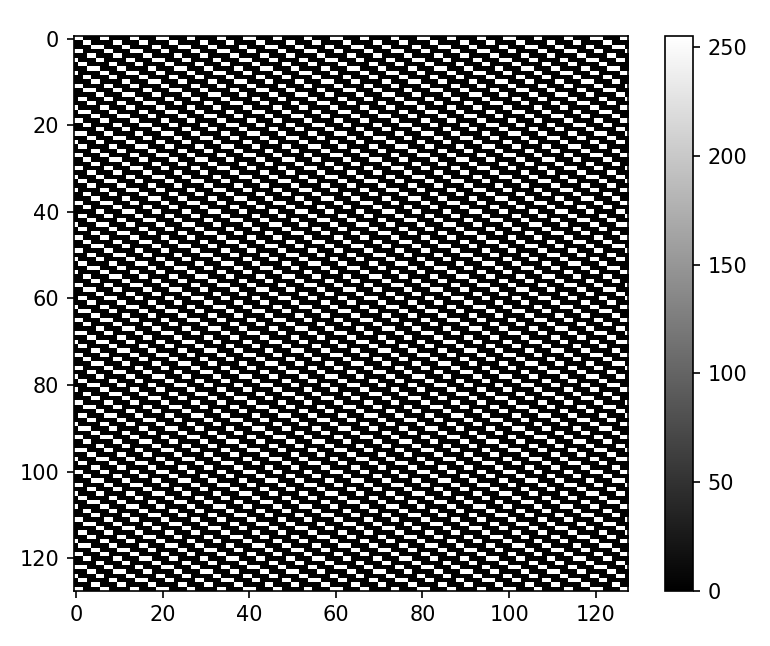


Figure [17] Image binarisé

Une image contenant graphique

Description générée automatiquement

Figure [18] Image non binarisé avec son spectre

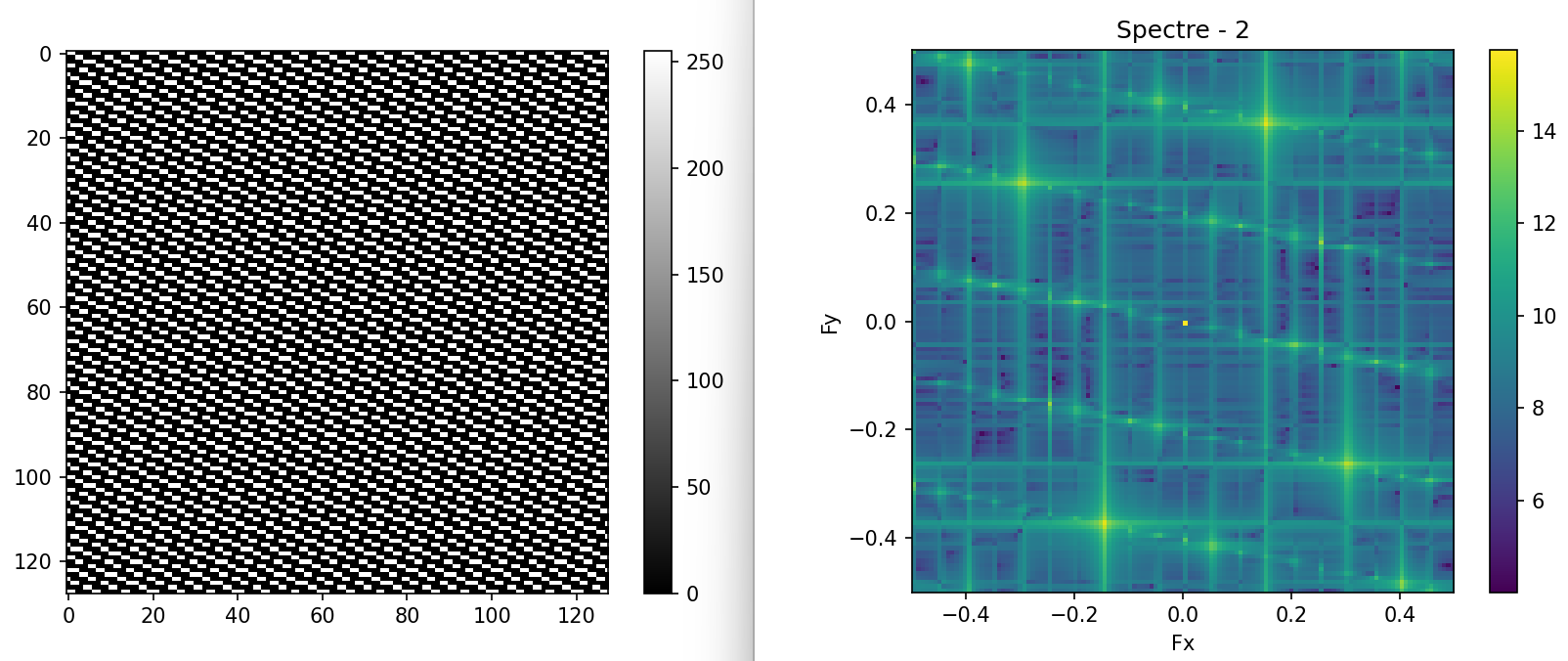


Figure [19] Image binarisé avec son spectre

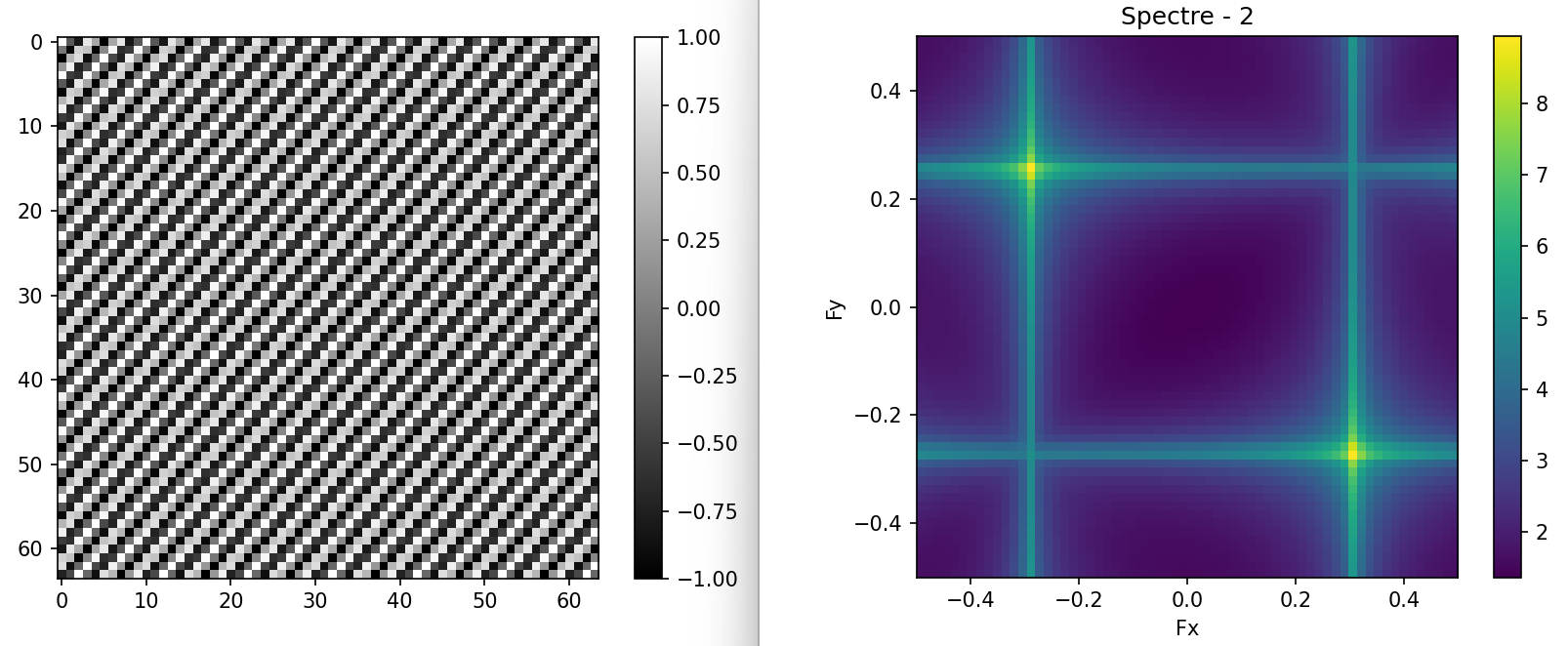


Figure [20] Image réduite avec son spectre