

# Compte-rendu TP3 : Introduction à l'image numérique

Ilyass EL MANSOURI

10 avril 2016

## Préambule

On s'intéresse tout d'abord à la lecture d'image sur Matlab. On s'intéresse par exemple à l'image : *Papierpeint.tif*

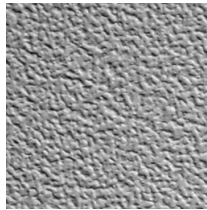


FIGURE 1 – papierpeint.tif

On visualise par la suite cette image comme une surface :

```
surf1(a) ; colormap gray ; shading interp ; axis equal ; view(-45+180,30) ;
```

On obtient ainsi :

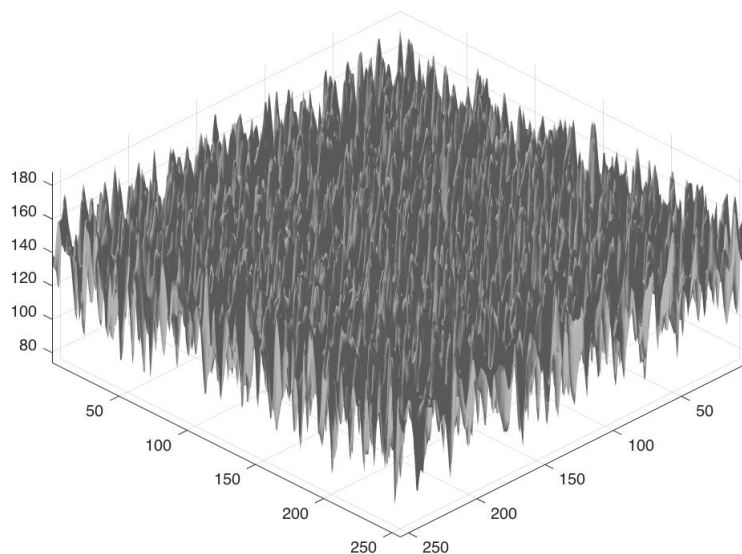


FIGURE 2 – Visualisation en surface de papierpeint.tif

# Observation du spectre d'images synthétiques

Dans cette partie, on observe et explique les spectres d'images synthétiques simples.

## — Sinusoïde horizontale

$$x = [0 : 255]'; y = 2 * pi * (8/256) * ones(1, 256); a = sin(x * y);$$

Visualisation du spectre : *Le spectre est sous la forme de deux points verticaux*

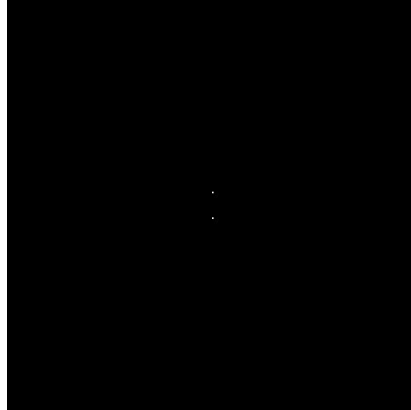


FIGURE 3 – Sinusoïde horizontale (TFD bi-dimensionnelle)

## — Sinusoïde verticale

$$xx = 2 * pi * (24/256) * ones(1, 256)'; yy = [0 : 255]; a = sin(xx * yy);$$

Visualisation du spectre : *Le spectre est sous la forme de deux points horizontaux plus espacé que la*

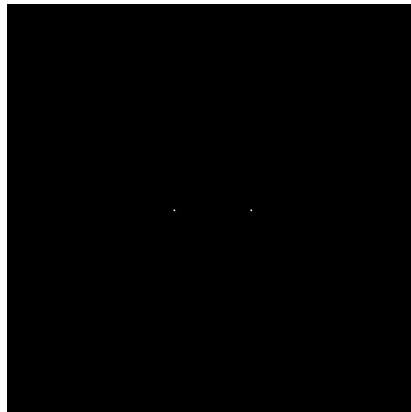


FIGURE 4 – Sinusoïde verticale (TFD bi-dimensionnelle)

*sinusoïde horizontale (fréquence plus élevée)*

— **Varitante**

$$x3 = 2 * pi * ((24 + 1/2)/256) * ones(1, 256)'; a = sin(x3 * yy);$$

Visualisation du spectre :

*Le décalage fréquentielle dans le sinus donne un spectre avec deux point plus étalés au niveau horizontale*

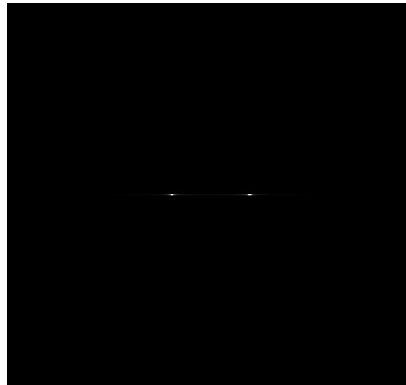


FIGURE 5 – Variante (TFD bi-dimensionnelle)

— **Sinusoïde diagonale**

$$a = sin(x * y + xx * yy);$$

Visualisation du spectre : *Le spectre est sous la forme de deux points en diagonale*

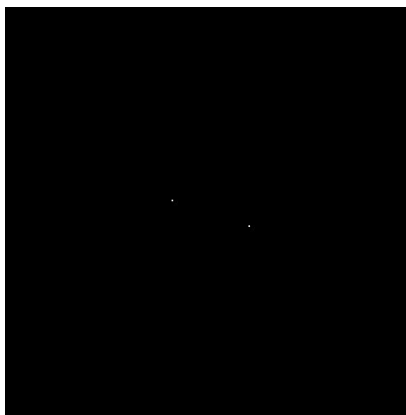


FIGURE 6 – Sinusoïde diagonale (TFD bi-dimensionnelle)

On s'intéresse ensuite à des images synthétiques contenant des contours :

— **Un carré**

$$a = zeros(256, 256); a(63 : 191, 63 : 191) = 1;$$

Visualisation de l'image et du spectre :

*On retrouve la transformée de fourrier d'une porte en 2D (un sinus cardinale sur les 4 flans du carré)*

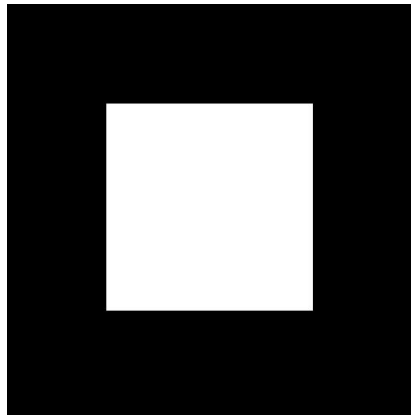


FIGURE 7 – Un carré

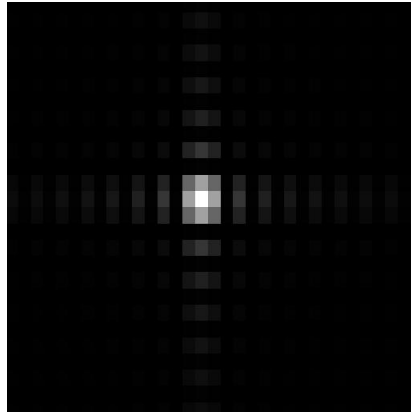


FIGURE 8 – Son spectre (TFD bi-dimensionnelle)

#### — Une sinusoïde fenêtrée

$$a = a. * \sin(x * y);$$

Visualisation de l'image et du spectre :

*Le spectre du produit entre un carré et une sinusoïde donne la convolution entre deux points et le spectre*



FIGURE 9 – Une sinusoïde fenêtrée

*précédent*

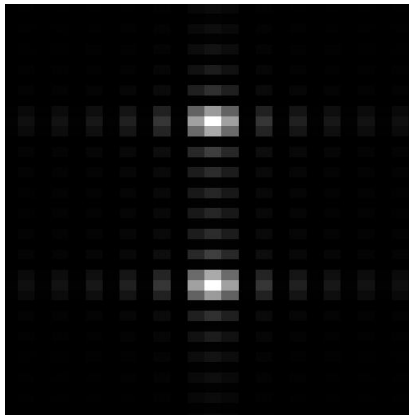


FIGURE 10 – Son spectre (TFD bi-dimensionnelle)

## Echantillonnage et transformée de Fourier

On observe le spectre de l'image *maison.tif*



FIGURE 11 – Une maison

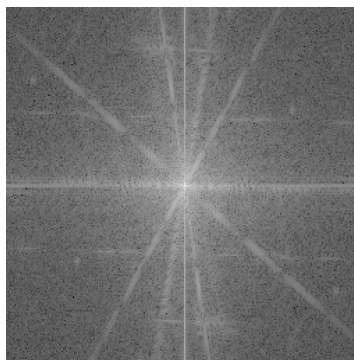


FIGURE 12 – Son spectre (TFD bi-dimensionnelle)

*Les traits horizontaux représentent les variations de couleurs verticales, réciproquement pour les traits verticales, et les traits obliques représentent les variation de couleur obliques (symétrique)*

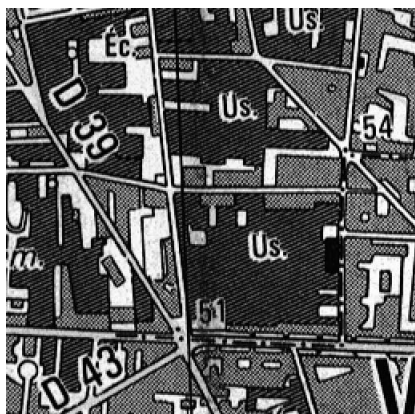


FIGURE 13 – Une carte

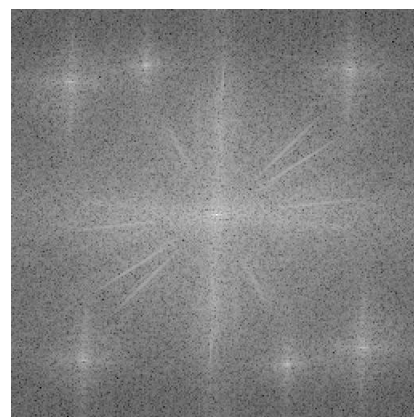


FIGURE 14 – Son spectre (TFD bi-dimensionnelle)



FIGURE 15 – Montreuil



FIGURE 16 – Son spectre (TFD bi-dimensionnelle)

*La principale différence avec la maison est le fait que ces photos ont la forme de plusieurs contours (des carrés) dont le spectre nous donne une série de points (constellations avec des traits verticaux et horizontaux)*  
A présent on sous-échantillonne l'image de la carte :

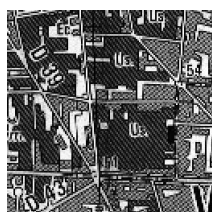


FIGURE 17 – La carte sous-échantillonnée

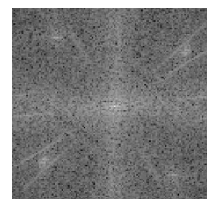


FIGURE 18 – Son spectre (TFD bi-dimensionnelle) (repliement spectral)

# Importance de la phase de la TFD selon le type d'image



FIGURE 19 – Image géométrique



FIGURE 20 – Image géométrique

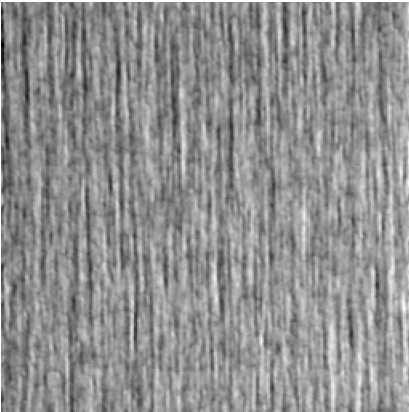


FIGURE 21 – Textile

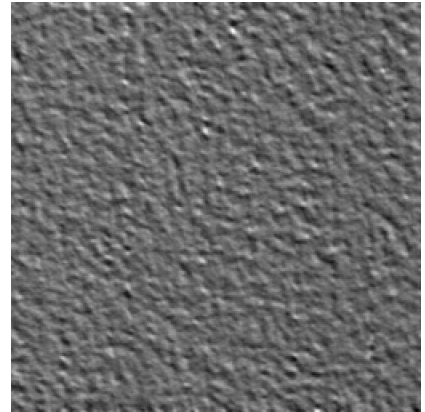


FIGURE 22 – Textile

- On en déduit que le module et **la phase** joue un rôle plus important pour les images géométrique comparée aux textures.
- On a réalisé ci-dessous la synthèse de texture, elles ont la même allure que l'image de base mais ne sont pas des copies exact, en effet on impose les phases d'un bruit blanc gaussien.

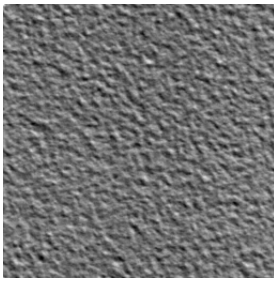


FIGURE 23 – Synthèse de texture V1

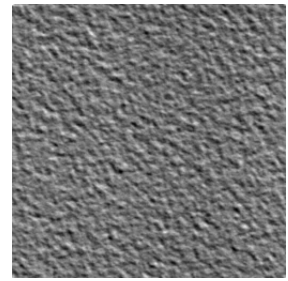


FIGURE 24 – Synthèse de texture V2

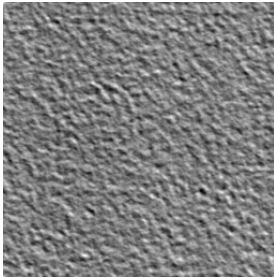


FIGURE 25 – Synthèse de texture V3

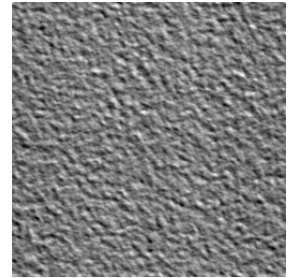


FIGURE 26 – Synthèse de texture V4

## Filtrage passe-haut et image

- L'objectif maintenant est de mettre en évidence les contours d'une image (c'est à dire la trace des bords d'un objet), une méthode pour les mettre en évidence est l'utilisation d'un filtre passe-haut :

$$fx=[0 \ 0 \ 0;-1 \ 1 \ 0; \ 0 \ 0 \ 0]; \ fy=fx'; \ gx=filter2(fx,a); \ gy=filter2(fy,a);$$

On peut éventuellement passer par une régularisation préalable de l'image.

- Le résultat :



FIGURE 27 – Synthèse de texture V3

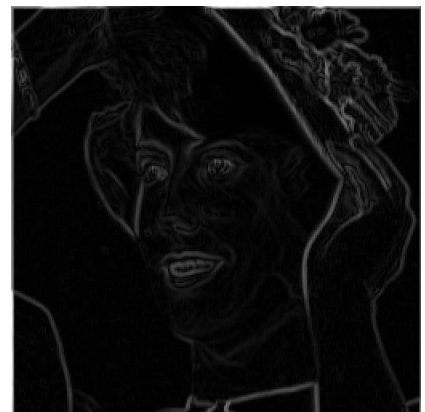


FIGURE 28 – Synthèse de texture V4



- **Unsharp Masking** : L'objectif est d'appliquer un filtre qui permet de donner plus de netteté à une image. Une modélisation possible du flou est l'évolution de l'image comme le ferait une distribution de température suivant l'équation de la chaleur, c'est-à-dire qu'à chaque pas de temps l'image, pour être floutée, se voit augmentée une proportion de son laplacien. Le filtre Unsharp correspond au renversement de cette évolution en retranchant à l'image une portion de son laplacien.

```
a = double(imread('flou.jpg'));  
laplace = ones(3,3); laplace(1:2:3,1:2:3) = 0; laplace(2,2) = -4;  
dirac = zeros(3,3); dirac(2,2) = 1; alpha = 1; b = filter2(dirac - alpha * laplace, a);  
affiche(a,'orig',[0255]); affiche(b,'net',[0255]);
```

- Résultat :



FIGURE 29 – Image floue



FIGURE 30 – Après passage par le filtre

- On peut augmenter le taux de mélange pour obtenir des résultats plus nets.

# Super-résolution

La super-résolution est une technique qui permet de reconstruire une image de haute résolution à partir d'images basses résolution.

- En analysant la fonction `appareilphoto.m` on remarque qu'elle sous-échantillonne notre image pour nous fournir une image de taille réduite. La valeur *sous* nous donne le facteur de sous-échantillonnage de l'appareil photographique. et *dx*, *dy* le début de la partie prise dans l'image retournée
- Résultat :



FIGURE 31 – Image initiale



FIGURE 32 – Après passage par la fonction ( $dx=dy=10$ )



FIGURE 33 – Image initiale



FIGURE 34 – Après passage par la fonction ( $dx=dy=100$ )