

Introduction au traitement d'images

Enseignement intégré

IT220 | Informatique 2A | 2023-2024

Chapitre 5 : Transformée de Fourier

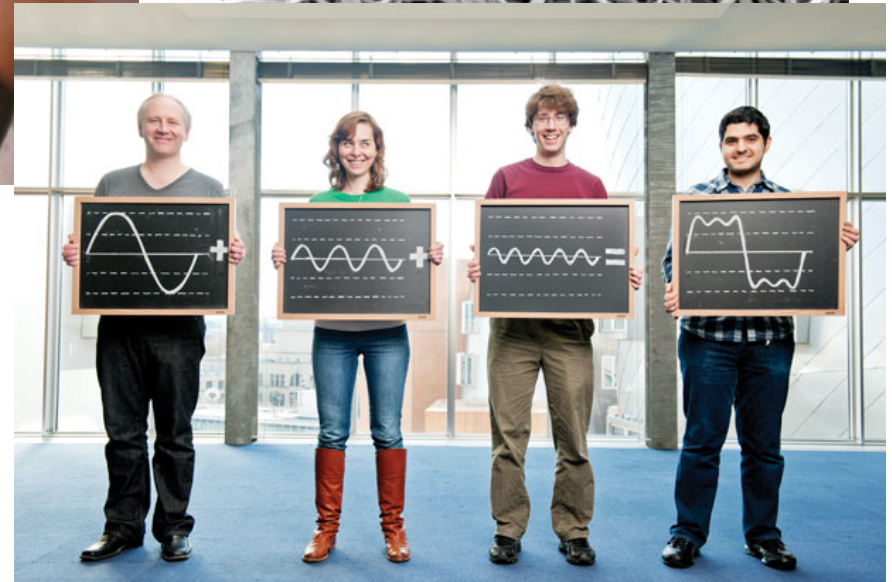
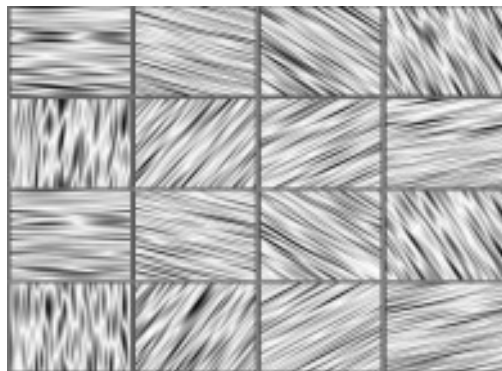
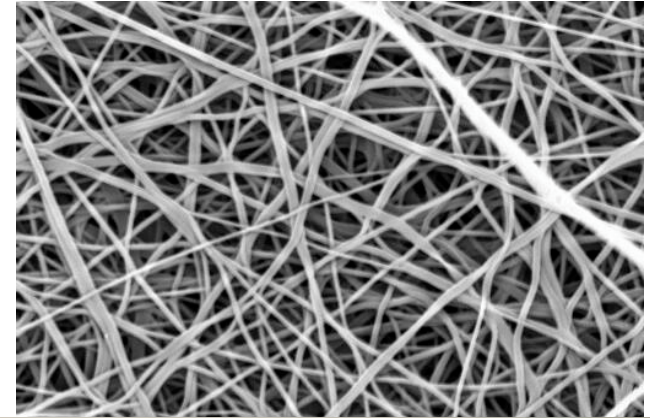
Rémi Giraud

remi.giraud@enseirb-matmeca.fr

<https://remi-giraud.enseirb-matmeca.fr/>

- **Introduction**
- **Formation / Acquisition**
- **Image couleur**
 - Format/Affichage/Synthèse
 - Espaces couleur caractéristiques (YCbCr)
 - Applications : compression, esquisse
- **Traitements**
 - Filtrage linéaire / non linéaire
 - Applications : débruitage, anonymisation
 - Détection de contours
 - Applications : réhaussement de contraste
- **Transformée de Fourier**
 - Application : recouvrement fréquentiel
- **Compression d'images**
 - Application : algorithme JPEG
- **Transformation spatiales**

Une richesse fréquentielle omniprésente

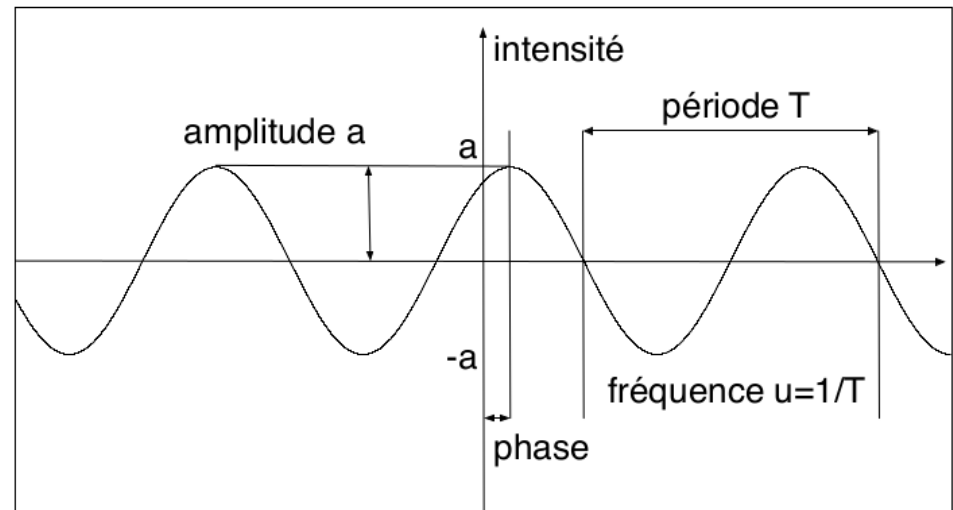


Signal simple 1D

- Fonction sinusoïdale :

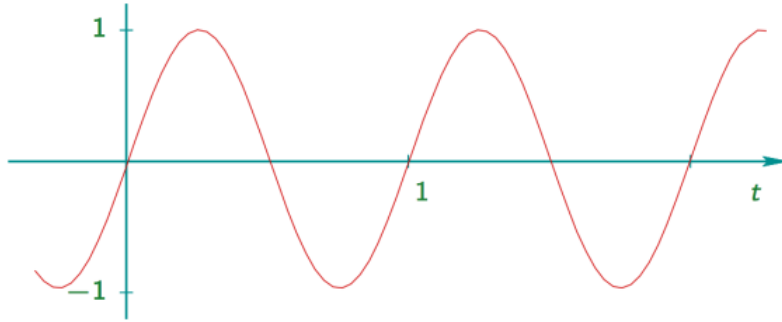
Une fonction sinusoïdale $f(x) = a.\sin(2\pi ux + \phi)$ est caractérisée par

- u : fréquence
- a : amplitude
- ϕ : phase ($\phi = 2\pi ux_0$)

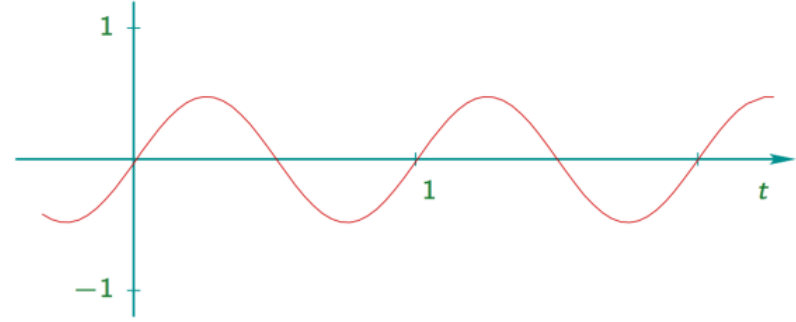


en 1D : $f(x) \rightarrow F(u)$, $F(u)$ décrit la **contribution** de chaque fréquence u dans la reconstruction de f : (amplitude, phase) $\rightarrow F(u) = (a_u, \phi_u)$

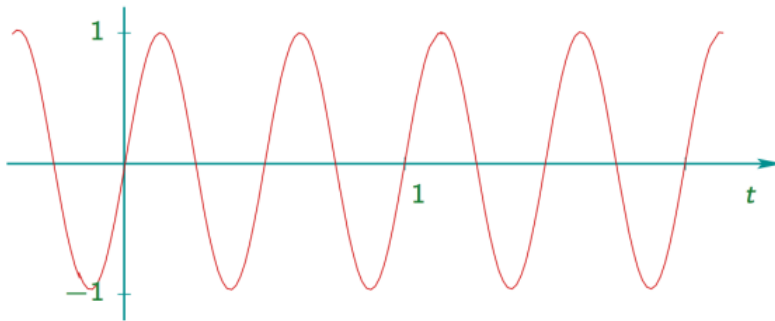
Signal simple 1D



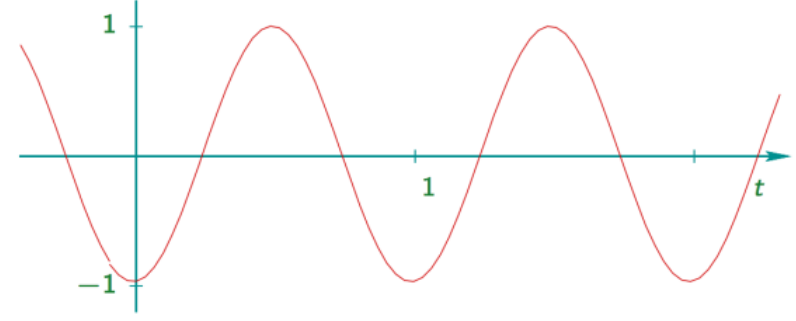
$$F(u = 1) \rightarrow (a = 1, \phi = 0)$$



$$F(u = 1) \rightarrow (a = 0.5, \phi = 0)$$



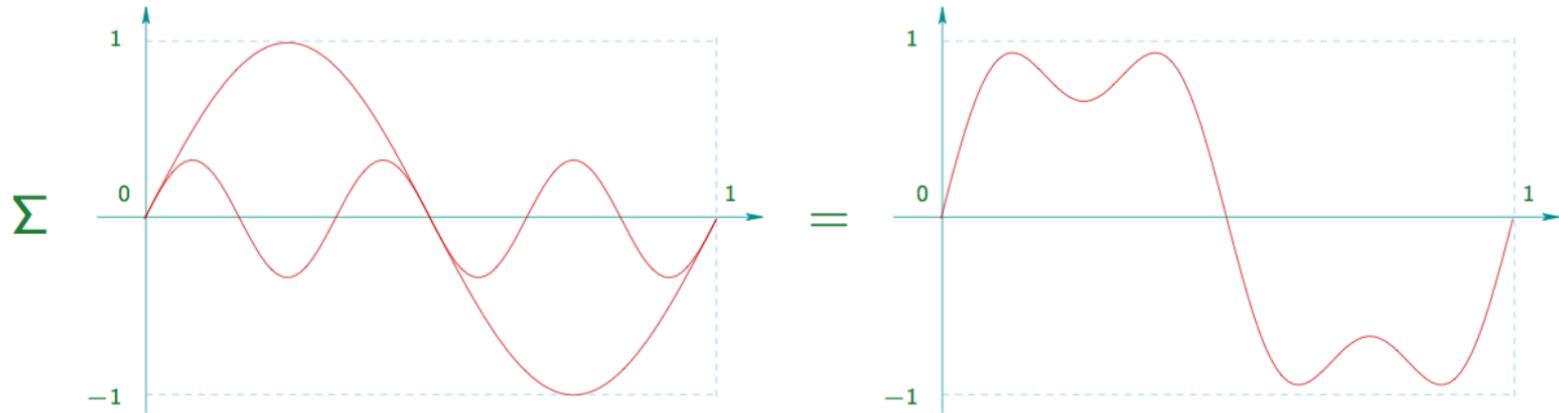
$$F(u = 2) \rightarrow (a = 1, \phi = 0)$$



$$F(u = 1) \rightarrow (a = 1, \phi = -0.25)$$

Signal complexe 1D

- Somme de signaux simples



$$[F(u = 1) \rightarrow (a = 1, \phi = 0)] + [F(u = 3) \rightarrow (a = 1/3, \phi = 0)]$$

- Transformée de Fourier :
 - Écriture sous la forme d'une somme infinie de fonctions sinusoïdales

$$f(x) = \sum_{n=-\infty}^{n=+\infty} c_n e^{i2\pi n u_0 x}$$

Transformée de Fourier 2D

- Transformée directe

$$F(u, v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) e^{-j2\pi(ux+vy)} dx dy = TF(f(x, y))$$

variables
fréquentielles

variables
spatiales

- Transformée inverse

$$f(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} F(u, v) e^{j2\pi(ux+vy)} du dv = TF^{-1}(F(u, v))$$

Transformée de Fourier 2D

- Transformée continue d'une séquence discrète (TFCD)

$$F_n(u, v) = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} f(mT_e^x, nT_e^y) e^{-j2\pi(u m T_e^x + v n T_e^y)}$$

transformée
 (f_e^x, f_e^y) -périodique

périodes
d'échantillonnage

fenêtrage et
échantillonnage des
fréquences

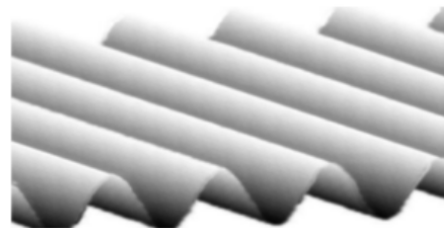
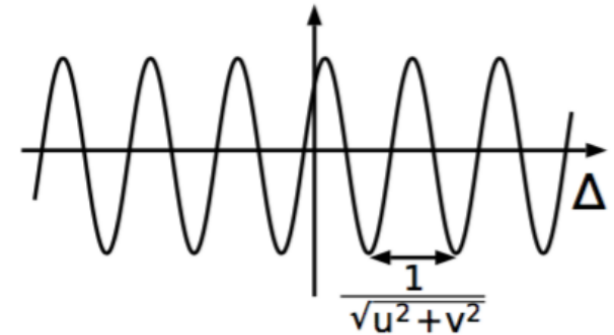
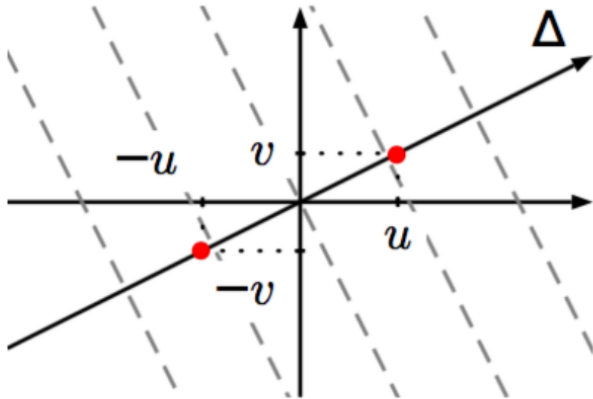
- Transformée discrète d'une séquence discrète (TFD)

$$F(k, l) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) e^{-j2\pi\left(k \frac{m}{M} + l \frac{n}{N}\right)}$$

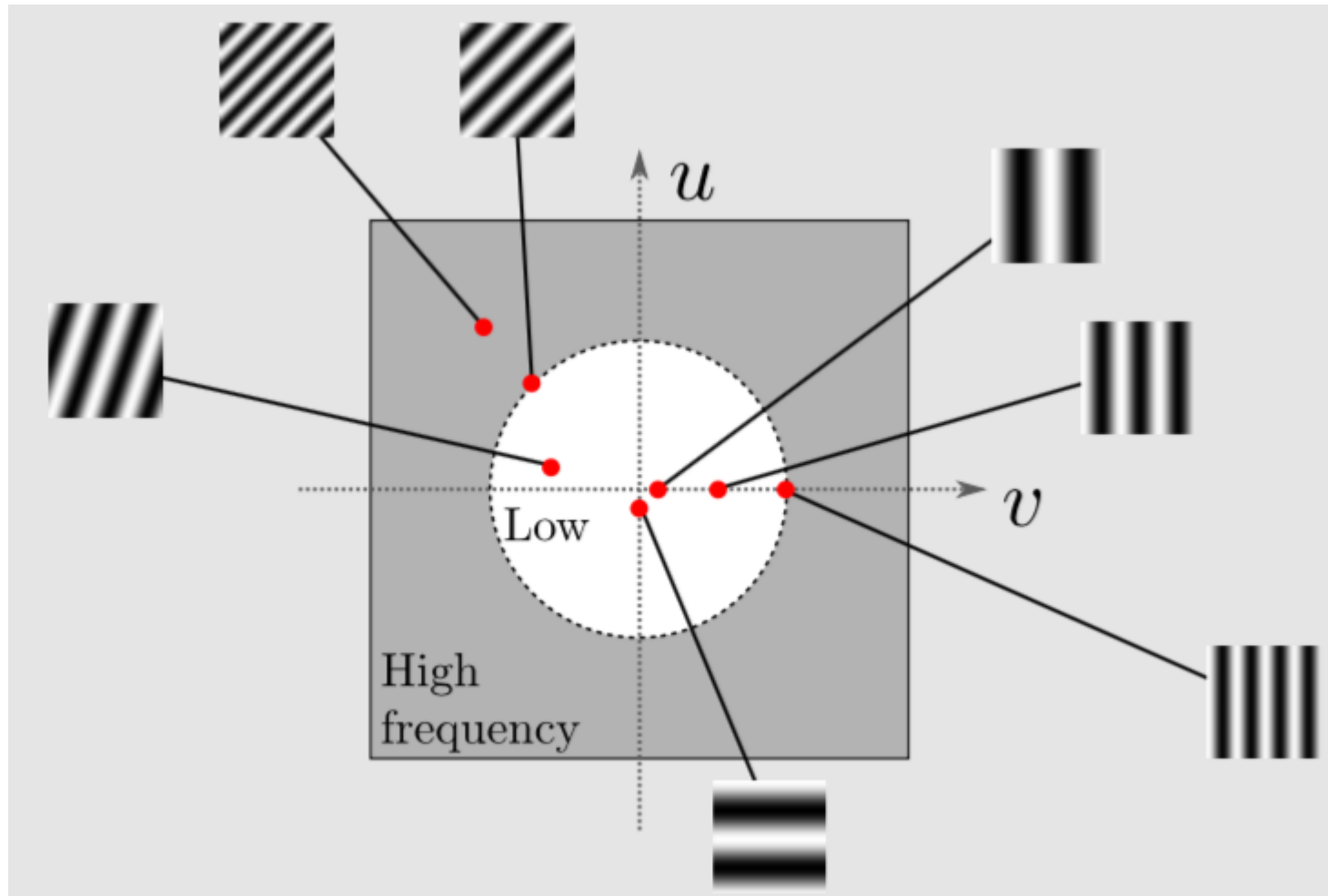
variables discrètes $\begin{cases} k \in [0, M - 1] \\ l \in [0, N - 1] \end{cases}$

Fréquences spatiales

Chaque $(u, v) \rightarrow (a_{(u,v)}, \phi_{(u,v)})$ dans le domaine de Fourier correspond à une fonction sinusoïdale de fréquence $\sqrt{u^2 + v^2}$ le long de Δ et orientée par le vecteur (u, v)

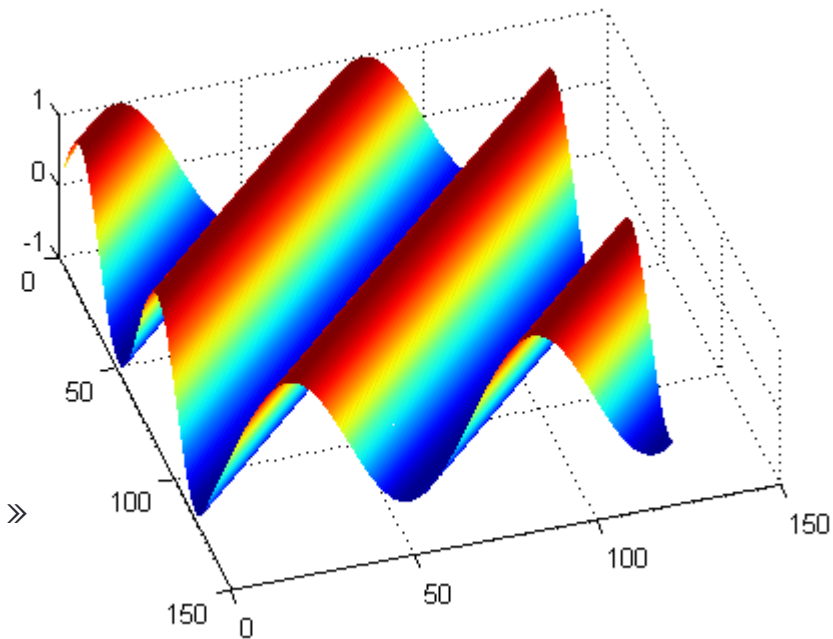
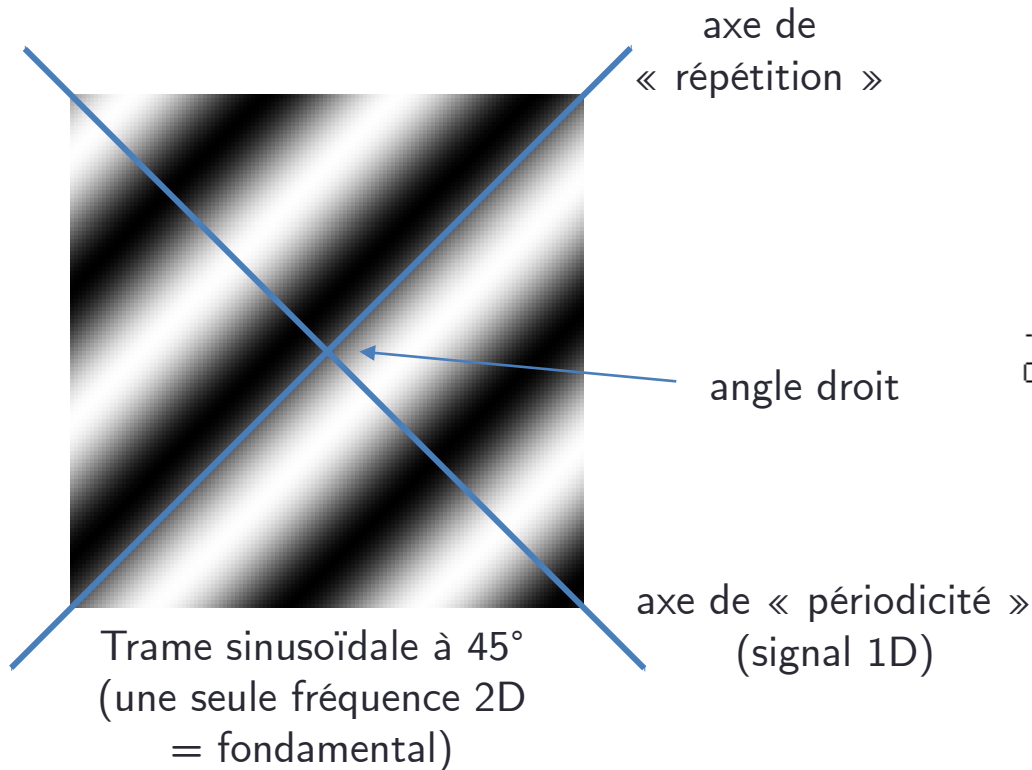


Basses / Hautes fréquences



Signal harmonique 2D

- Sinusoïde parfaite = répétition d'un motif selon un certain axe
- Texture = image périodique



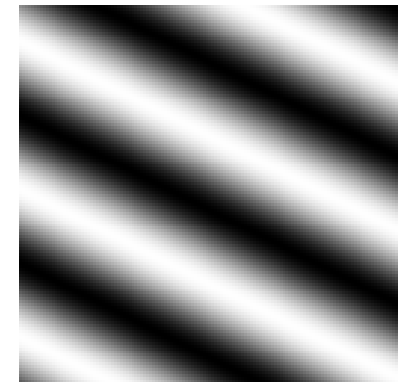
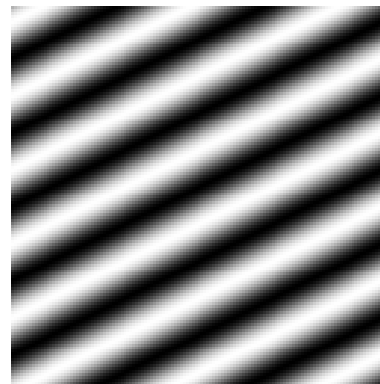
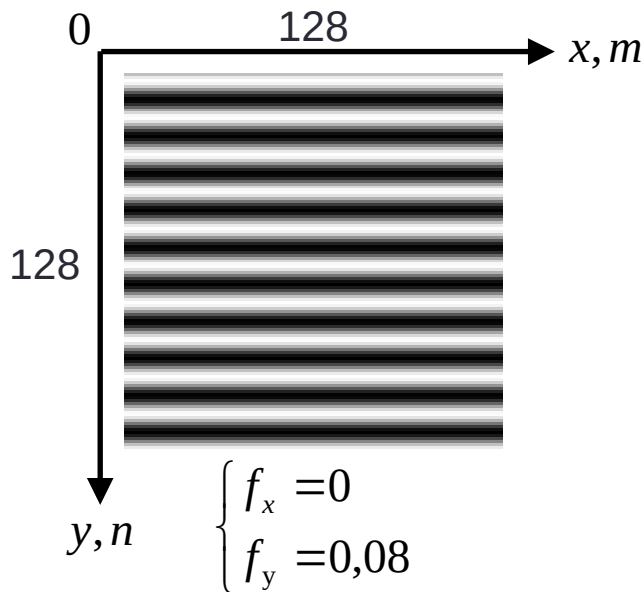
Représentation 3D

Signal harmonique 2D

$$f(x, y) = \sin 2\pi (f_x x + f_y y)$$

fréquence horizontale fréquence verticale

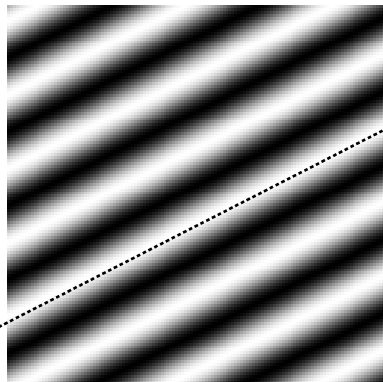
$$\begin{cases} x = mT_e^x, & m \in [0, M - 1] \\ y = nT_e^y, & n \in [0, N - 1] \end{cases}$$



Signal harmonique 2D

$$f(x, y) = \sin 2\pi(f_x x + f_y y)$$

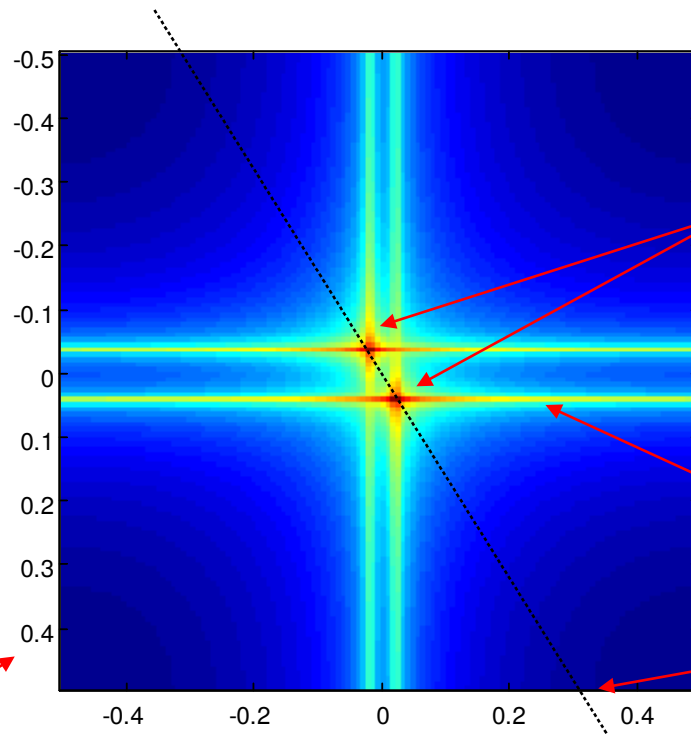
$$\begin{cases} f_x = 0,02 \\ f_y = 0,04 \end{cases}$$



orientation

$|TF(\cdot)|$

Signal 2D harmonique à une seule fréquence (fondamental)



fréquences réduites

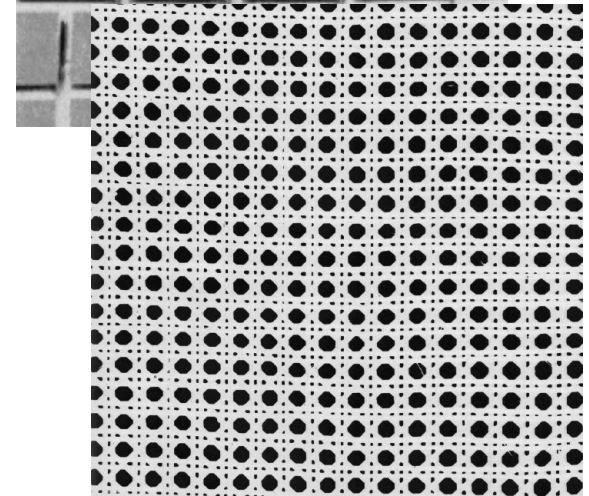
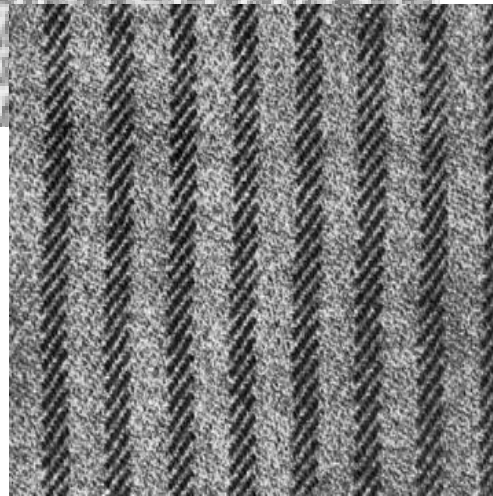
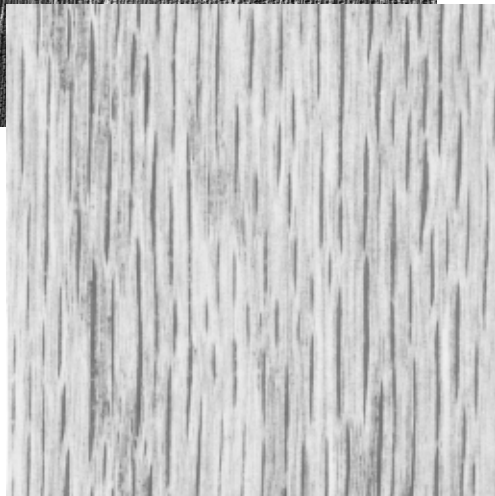
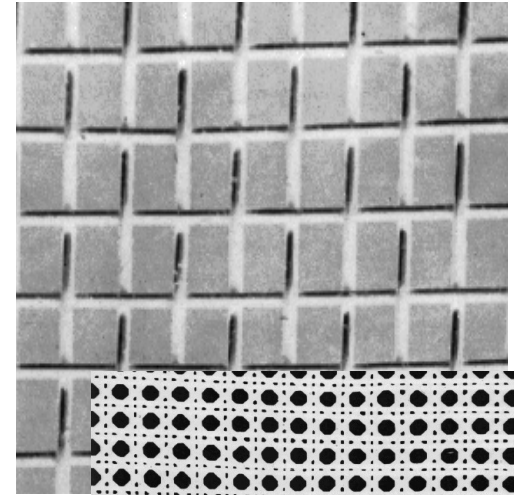
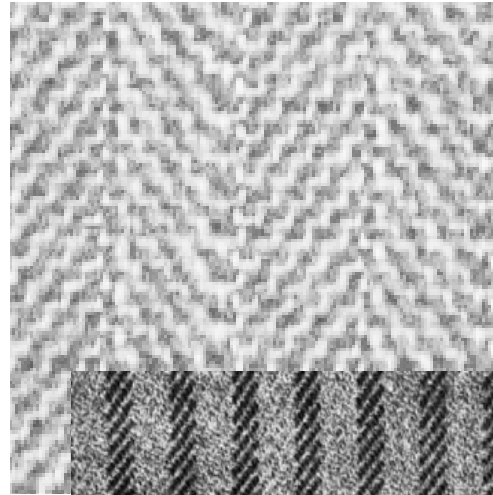
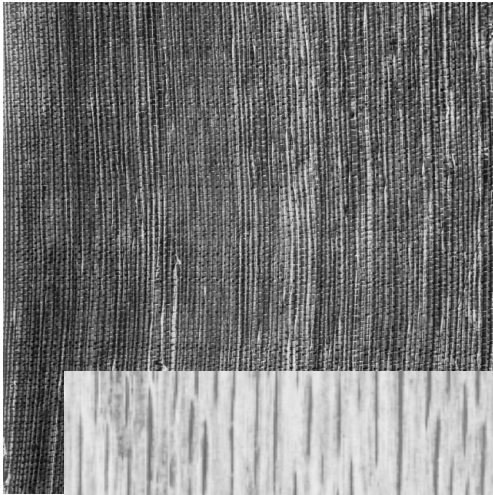
Amplitude du spectre (échelle logarithmique)

paire de Dirac de coordonnées $(-f_x, -f_y), (f_x, f_y)$

convolution par un sinus cardinal 2D (fenêtrage)

axe fréquentiel orthogonal à l'orientation de la trame

Textures pseudo-périodiques

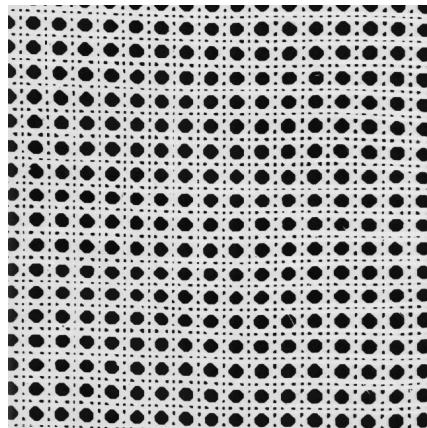


« 1 période »

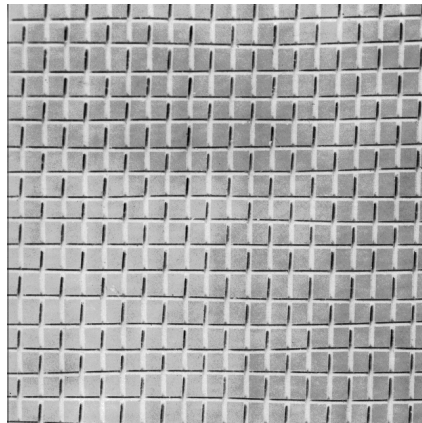
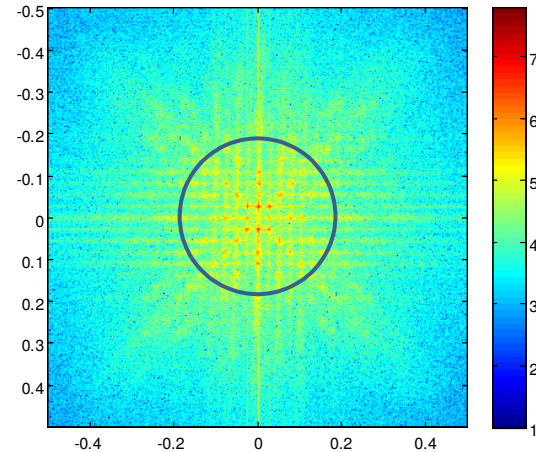
« 2 périodes »

Textures pseudo-périodiques

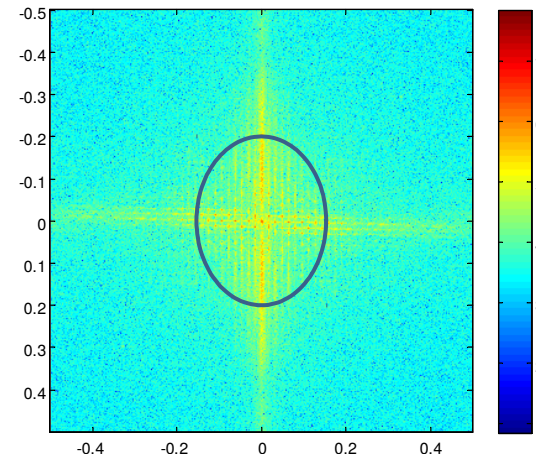
- « Paires de Dirac » diffuses correspondant à des fréquences fondamentales et des harmoniques



$|TF(\cdot)|$



$|TF(\cdot)|$

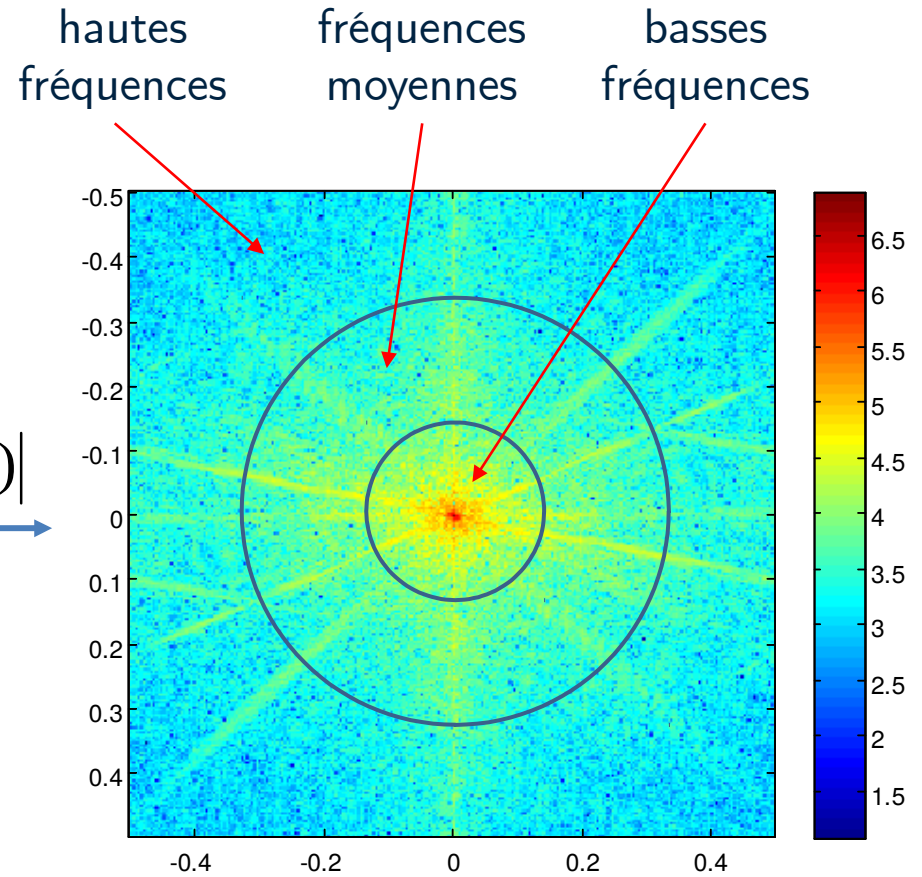


Sur une image naturelle



Image naturelle

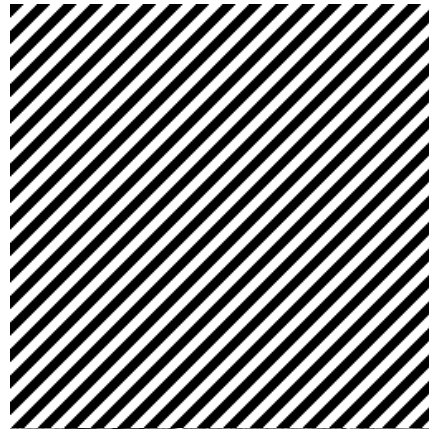
$|TF(\cdot)|$



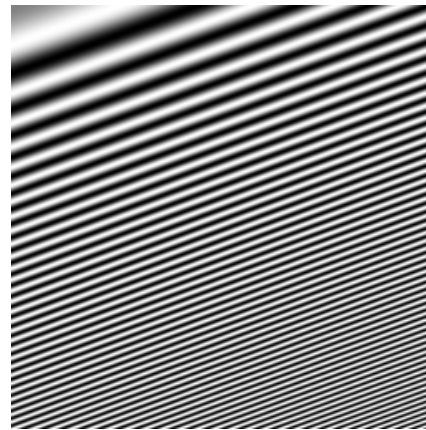
Quiz : Retrouver à quelles images correspondent ces transformées



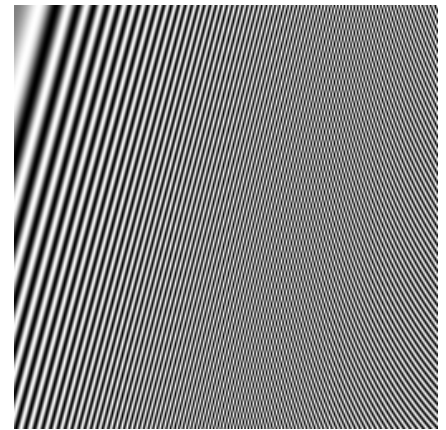
(1)



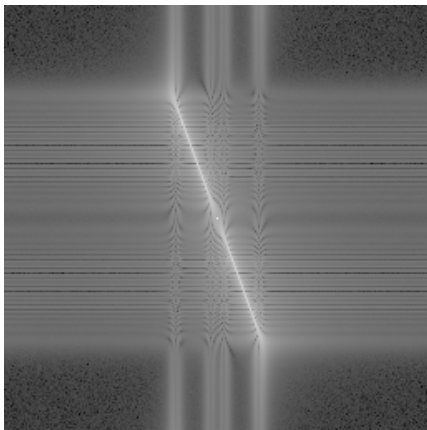
(2)



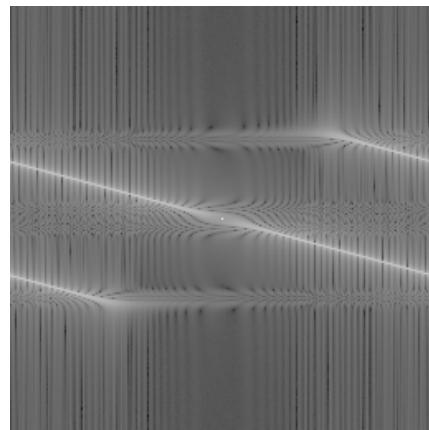
(3)



(4)



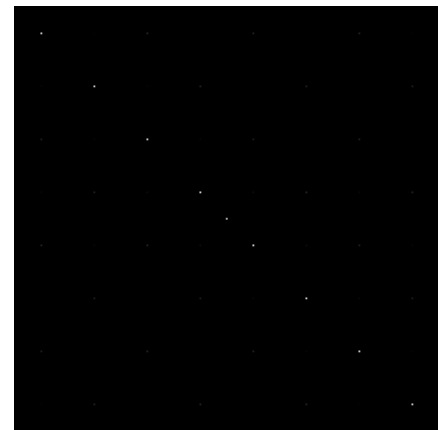
(a)



(b)



(c)



(d)

Quiz : Retrouver à quelles images correspondent ces transformées



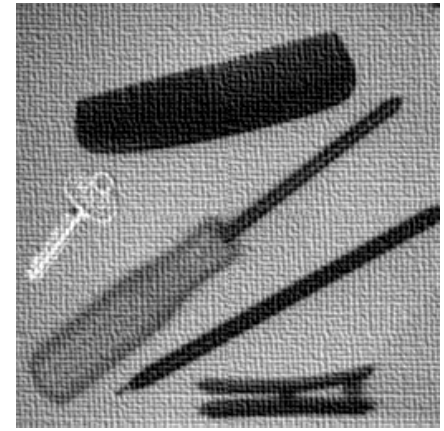
(1)



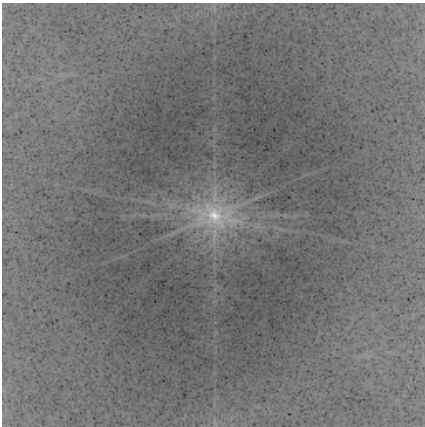
(2)



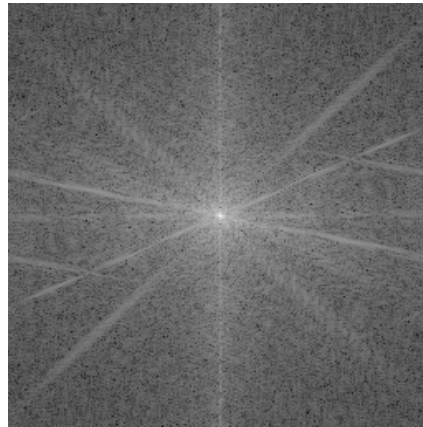
(3)



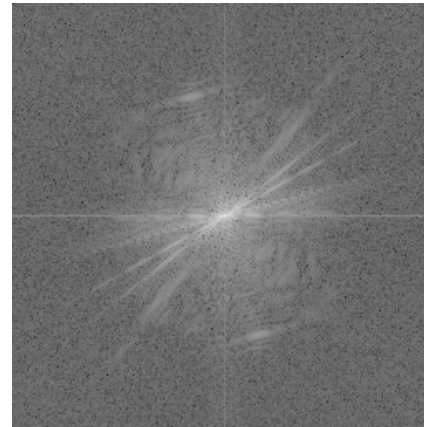
(4)



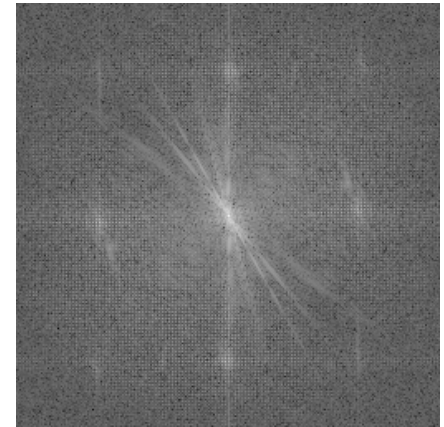
(a)



(b)



(c)



(d)

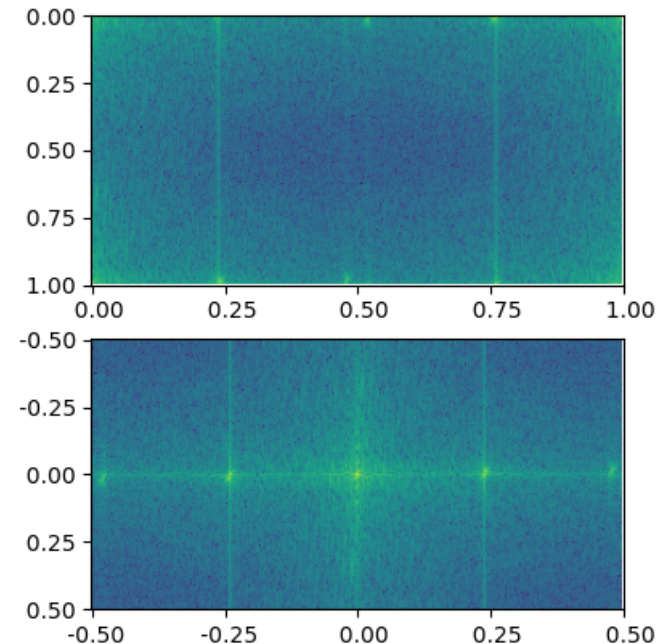
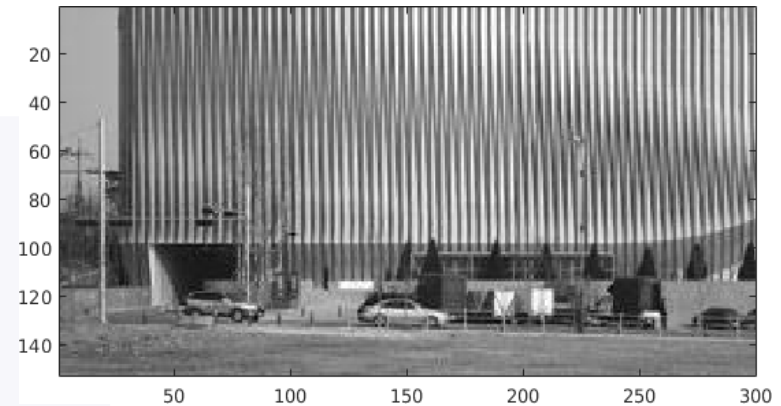
Transformée de Fourier 2D

Affichage

```
A = plt.imread('../img/building.jpg')
plt.figure(1)
plt.imshow(A, cmap='gray')
[h,w] = A.shape

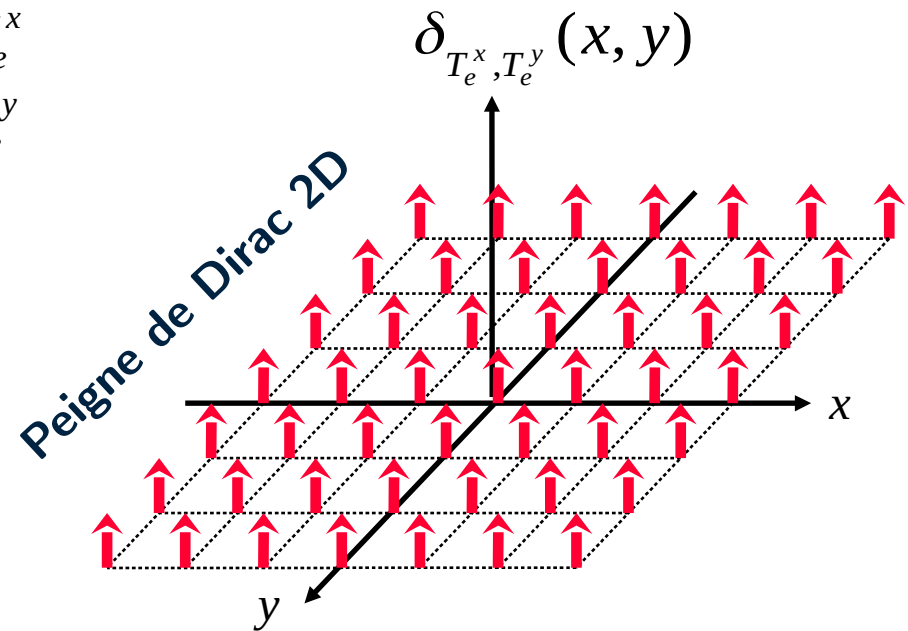
A_fft2 = np.fft.fft2(A)
plt.figure(2)
fig,(ax1,ax2) = plt.subplots(2,1)
plt.subplot(2,1,1)
plt.imshow(np.log10(abs(A_fft2)))
ticksx = np.linspace(0,w,5)
ticksy = np.linspace(0,h,5)
ticklabelsx = ["{:6.2f}".format(i) for i in ticksx/w]
ticklabelsy = ["{:6.2f}".format(i) for i in ticksy/h]
ax1.set_xticks(ticksx)
ax1.set_xticklabels(ticklabelsx)
ax1.set_yticks(ticksy)
ax1.set_yticklabels(ticklabelsy)

plt.subplot(2,1,2)
plt.imshow(np.fft.fftshift(np.log10(abs(A_fft2))))
ticklabelsx = ["{:6.2f}".format(i) for i in ticksx/w-0.5]
ticklabelsy = ["{:6.2f}".format(i) for i in ticksy/h-0.5]
ax2.set_xticks(ticksx)
ax2.set_xticklabels(ticklabelsx)
ax2.set_yticks(ticksy)
ax2.set_yticklabels(ticklabelsy)
```



Domaine discret : Échantillonnage par un peigne de Dirac

$$f_n(x, y) = \begin{cases} f_a(x, y) & \text{pour } \begin{cases} x = mT_e^x \\ y = nT_e^y \end{cases} \\ 0 & \text{sinon} \end{cases}$$



$$\delta_{T_e^x, T_e^y}(x, y)$$

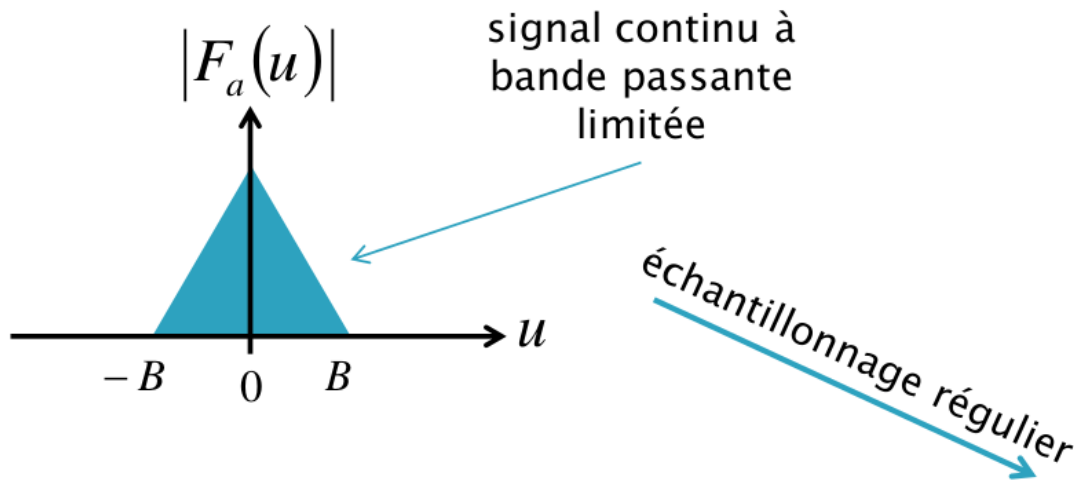
$$f_a(x, y) \rightarrow \text{⊗} \rightarrow f_n(x, y)$$

$$f_n(x, y) = f_a(x, y) \delta_{T_e^x, T_e^y}(x, y)$$

$$\delta_{T_e^x, T_e^y}(x, y) = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} \delta(x - mT_e^x, y - nT_e^y)$$

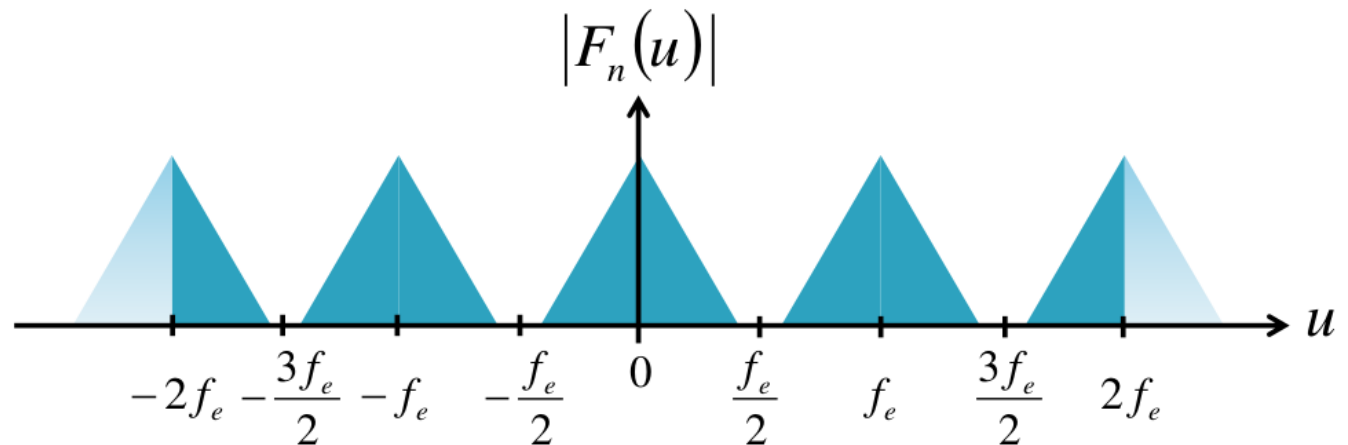
→ Séquence discrète $f_n(x, y)$ notée $f(mT_e^x, nT_e^y)$ ou $f(m, n)$

Périodisation du spectre dans l'espace fréquentiel

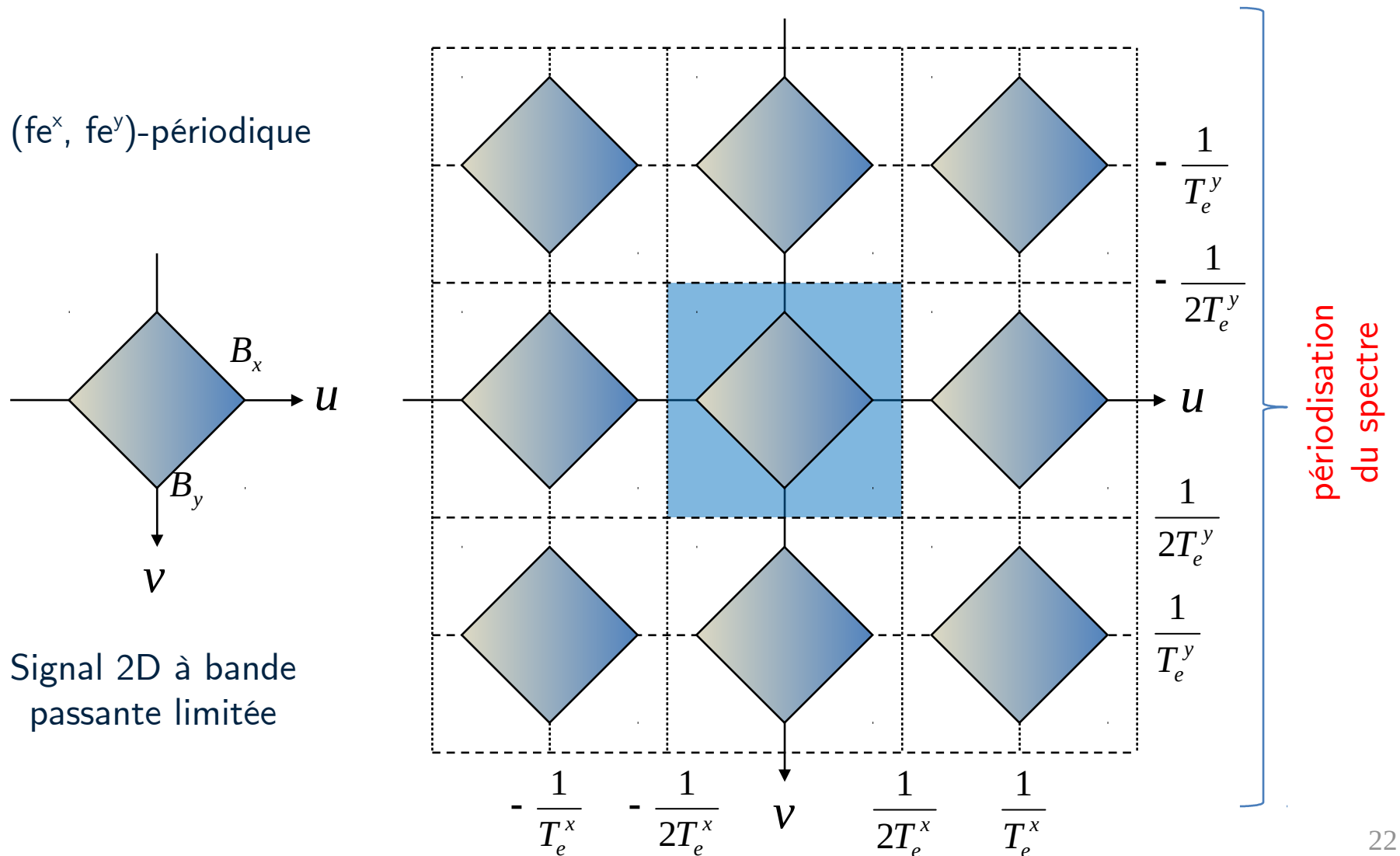


$$\begin{aligned} F_n(u) &= TF(f_n(u)) \\ &= TF(f_a(t)\delta_{T_e}(t)) \\ &= TF(f_a(t)) * TF(\delta_{T_e}(t)) \\ &= F_a(u) * \delta_{1/T_e}(u) \end{aligned}$$

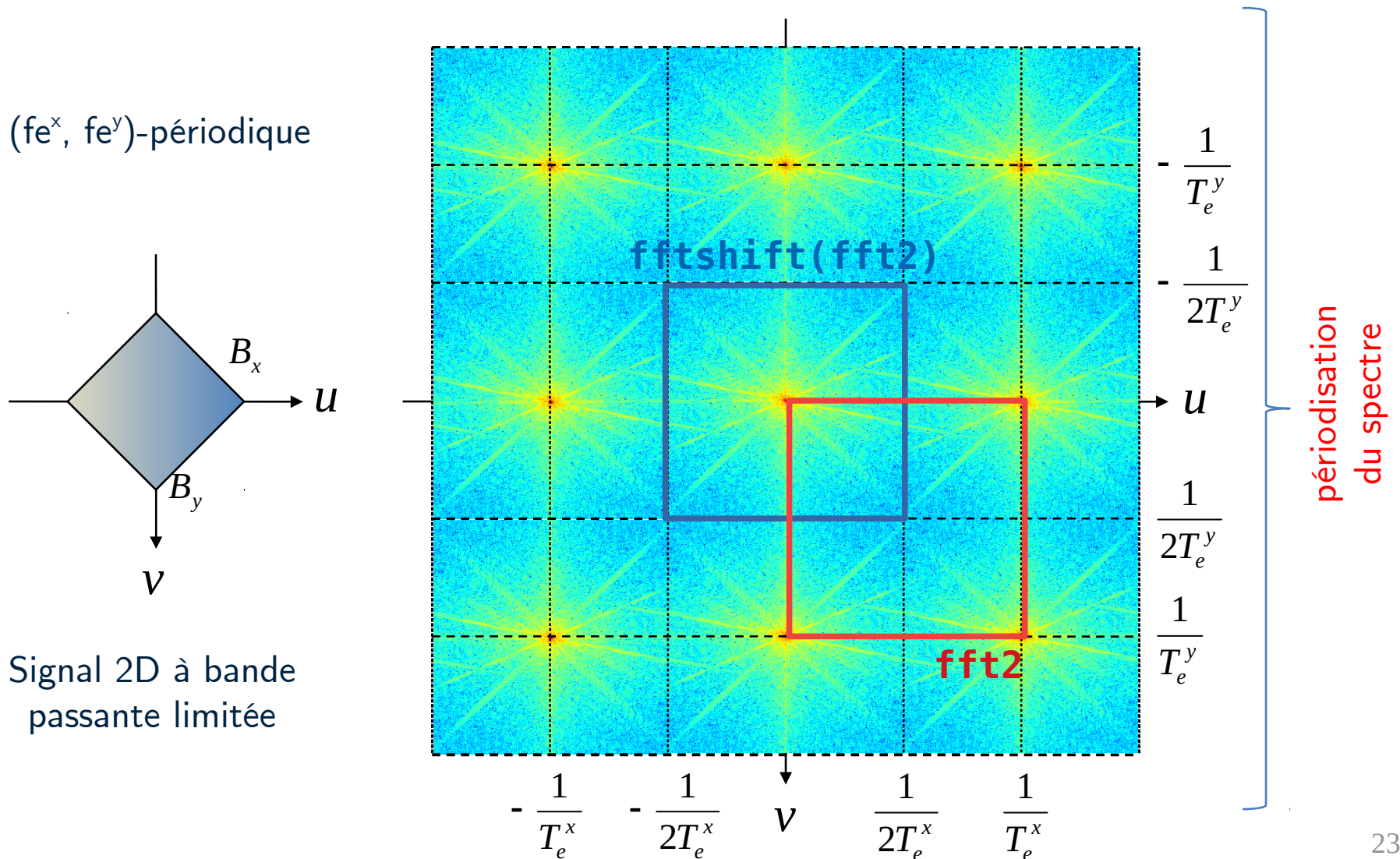
Périodisation
du spectre



Périodisation du spectre dans l'espace fréquentiel

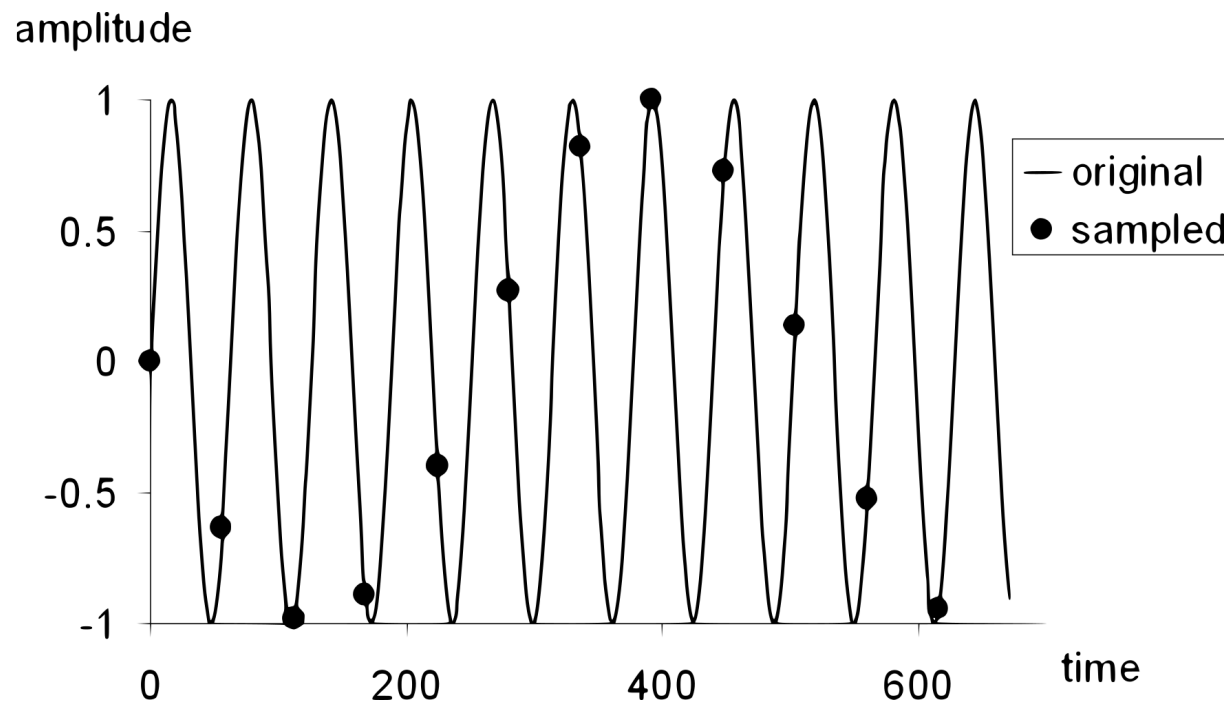


Périodisation du spectre dans l'espace fréquentiel



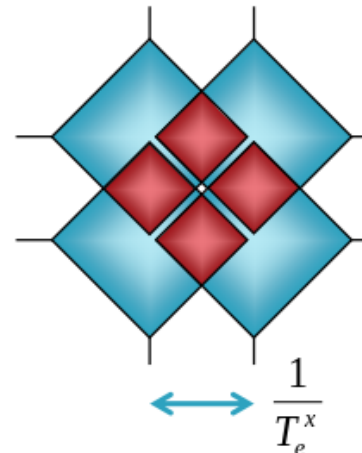
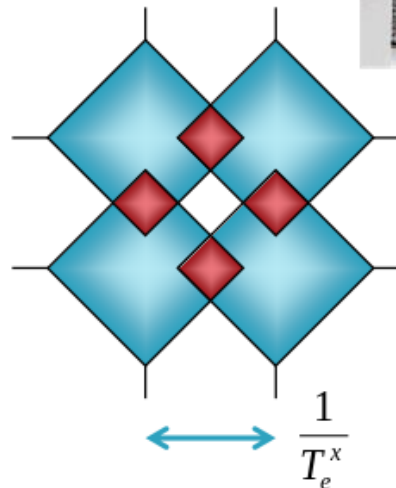
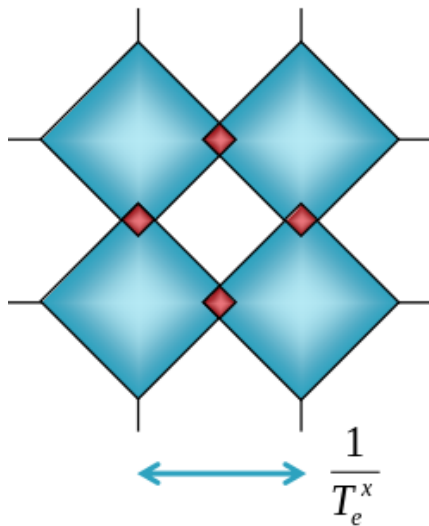
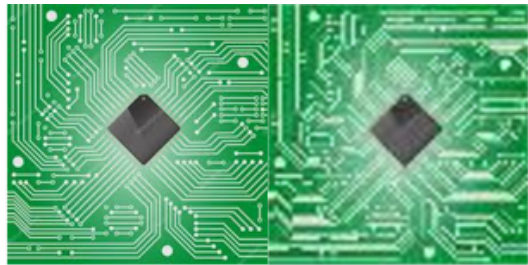
Phénomène d'Aliasing (ou repliement spectral)

- Choisir une période / fréquence d'échantillonnage adaptée pour capturer les variations du signal
- Sinon risque de dénaturer le signal échantillonné



Phénomène d'Aliasing (ou repliement spectral)

- Sur des images naturelles



Filtrage anti-aliasing

- Observer ce problème de repliement de spectre liés à l'échantillonnage spatial en sous-échantillonnant *barbara.png* ou *bricks.png* d'un facteur 4.
- Observer également la transformée de Fourier des images.
- Lisser l'image initiale par un filtre passe-bas d'anti-repliement, type Gaussienne, avant le sous-échantillonnage. On comparera ce résultat également à celui obtenu par `skimage.transform.resize`.



Image initiale (h x w)



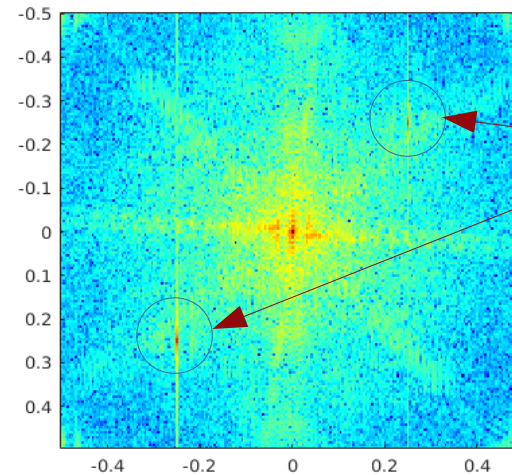
Image sous-échantillonnée
(h/4 x w/4)



Image filtrée et sous-échantillonnée
(h/4 x w/4)

Filtrage fréquentiel

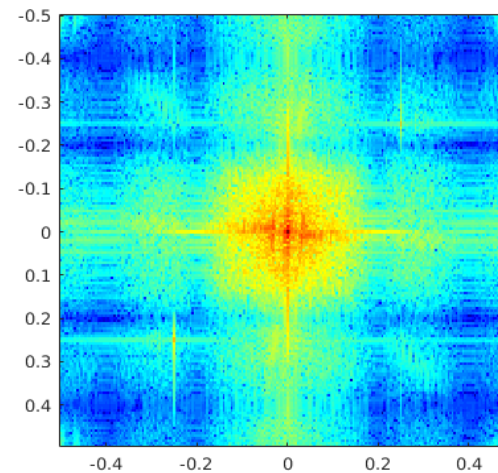
- Bruit fréquentiel. Inefficacité des filtres moyeneurs classiques



présence
du bruit

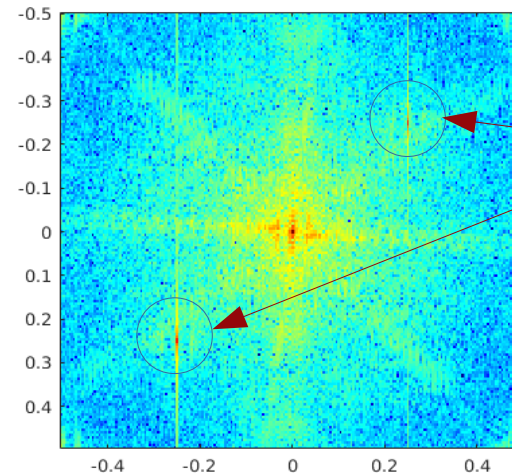


Filtre moyeneur carré 5x5

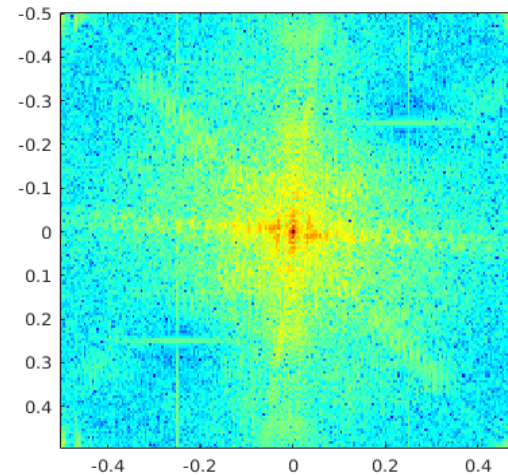


Filtrage fréquentiel

- Filtrage par un filtre coupe bande \rightarrow implémentation en TP



présence
du bruit



Filtre coupe bande spécifique