VI Transformée de Fourier discrète et filtrage

- Revisite du filtrage *linéaire* vu précédemment, mais sous l'angle du filtrage « directement » dans le plan de Fourier, i.e. le plan des fréquences spatiales (par le biais de la Transformée de Fourier (TF) bidimensionnelle).

  => de nouveaux horizons vont s'ouvrir à nous en termes d'amélioration/de restoration d'image…

- Illustrations/applications :
  - filtrage passe-bas (afin de « lisser » une image),
  - filtrage passe-haut (afin de faire ressortir les hautes fréquences => par ex. contours),
  - filtrage passe-bande (pour enlever par ex. des biais périodiques => par ex. tramage).

- D'autres applications abordées à la fin de ce chapitre (restoration/reconstruction d'image => déconvolution).

(1) TF discrète (TFD, ou DFT en anglais) bidimensionnelle

• Soit $f(x, y)$, avec : *x=0,1,2,…,M-1*, et : *y=0,1,2,…,N-1*, une image digitale de taille *MxN*.

La TFD de $f(x, y)$, $\hat{f}(u, v)$, s'écrit :

$$\hat{f}(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \ \exp\{-\imath 2\pi(ux/M + vy/N)\}$$

Remarque : dans le monde continu, on a :

$$\hat{f}(u,v) = \int\int f(x,y) \ \exp\{-\imath 2\pi(ux + vy)\} \ dx \ dy$$

• *x* et *y* sont les coordonnées spatiales des pixels de l'image $f(x, y)$

• *u* et *v* sont les coordonnées fréquentielles des frequels (frequels = « frequency elements » = les pixels dans le plan de Fourier) de $\hat{f}(u, v)$.

• La TFD inverse s'écrit :

$$f(x,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \hat{f}(u,v) \ \exp\{\imath 2\pi(ux/M + vy/N)\}$$

Même remarque :

$$f(x,y) = \int\int \hat{f}(u,v) \ \exp\{\imath 2\pi(ux + vy)\} \ du \ dv$$

- $\hat{f}(0,0)$ = « composante continu » (vient de l'électronique) = intégrale (somme dans le monde discret) de $f(x,y)$

- Même si $f(x,y)$ est réelle, $\hat{f}(u,v)$ est a priori complexe.

- On peut exprimer donc $\hat{f}(u,v)$ également comme :

$$\hat{f}(u,v) = |\hat{f}(u,v)| \exp\{\imath\phi(u,v)\}$$

où $|\hat{f}(u,v)|$ est le module de $\hat{f}(u,v)$ et $\phi(u,v)$ son argument.

$$|\hat{f}(u,v)| = \sqrt{Re^2(\hat{f}(u,v)) + Im^2(\hat{f}(u,v))}$$

$$\phi(u,v) = \arctan\frac{Im(\hat{f}(u,v))}{Re(\hat{f}(u,v))}$$

- Densité spectrale :

$$|\hat{f}(u,v)|^2 = Re^2(\hat{f}(u,v)) + Im^2(\hat{f}(u,v))$$

- Quelques propriétés de la TFD

  - *f* réelle => $Re(\hat{f})$ paire et $Im(\hat{f})$ impaire
  Et : $|\hat{f}(u, v)| = |\hat{f}(-u, -v)|$ module toujours paire
  => Densité spectrale centro-symmétrique

  - *f* paire => $Im(\hat{f})$ = 0

  - La TFD est *circulaire* (i.e. périodique de période *M* en *u* et de période *N* en *v*)

  - La TFD inverse aussi (de période *M* en *x* et *N* en *y)*
  => l'image obtenue par TFD inverse est *périodisée !*

  - Ces deux derniers points : car les images et leurs TFD sont échantillonnées.

  Échantillonnage dans le plan direct (resp. dans le plan de Fourier) <=> périodisation dans le plan de Fourier (resp. dans le plan direct).

- Calculer la TFD sous Matlab/Octave : en fait calcul d'une FFT (*Fast Fourier Transform*) :

  >> $\hat{f}$ = *fft2(f)*

- Mais la FFT produit un décalage de (*M/2*, *N/2*) => *fftshift* sous Matlab/Octave pour réordonner.

- Quand on va devoir placer l'image initiale dans un tableau plus grand pour le filtrage, on utilisera une syntaxe plus complète pour la FFT :

  >> $\hat{f}$ = fft2(f, P, Q)

  où *P* et *Q* plus grands que *M* et *N* (*zero padding*).

- Module de $\hat{f}(u, v)$ obtenu avec la commande *abs* :

  >> mod = abs($\hat{f}$)

- Visualiser |FFT| correctement (*frequel* central au centre du tableau visualisé, pas dans les coins) :

  >> imagesc(abs(fftshift(fft2(f))))

- Pour mieux voir (dynamique souvent limitée par rapport à la dynamique de la TF) :

  >> imagesc((abs(fftshift(fft2(f))))^0.5)

- Opération inverse à *fftshift* : *ifftshift*, ainsi :

  >> $\hat{f}$=ifftshift(fftshift($\hat{f}$))

- Pour calculer la phase de $\hat{f}$: *atan2(Im(f),Re(f))*, qui est un tableau d'angles entre -π et π. Autrement dit :

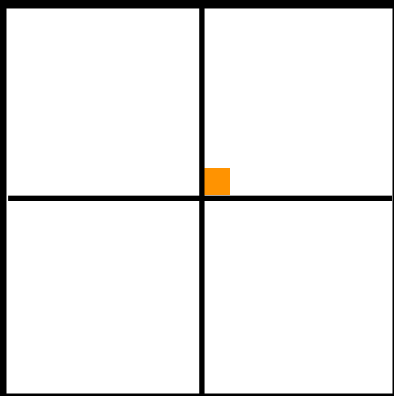
>> phi = atan2(imag($\hat{f}$), real($\hat{f}$))

ou plus directement avec *angle* (ou même *arg*) :

>> phi = angle($\hat{f}$)

(Et on a donc : $\hat{f}$ = *abs($\hat{f}$) x exp(i * angle($\hat{f}$)*))

- Attention, pour la TFD : le centre se situe ici [*M/2+1*, *N/2+1*] si *M* et *N* sont paires :



(si nb impairs (*M* ou *N*) : *floor(M/2)+1*, *floor(N/2)+1*)

- La FFT inverse s'obtient grâce à *ifft2* :

>> f = ifft2($\hat{f}$)

- Attention : *fft2* convertit au passage en classe « double »…

f de type *uint8* —> $\hat{f}$ de type *double* (valeurs réelles en double précision, mais entre 0.0 et 255.0)

=> pour éviter les problèmes : convertir dès le début les images en réels double précision entre 0.0 et 1.0 !

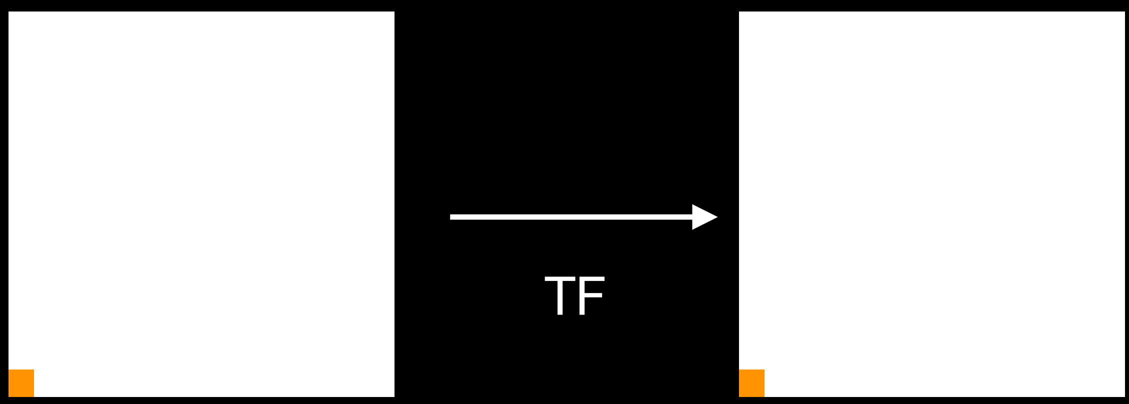
- Unités dans le plan de Fourier :

  - image de longueur $L=N\ \Delta x$, avec N le nb linéaire de pixels et Δx la taille du pixel

  => pas en fréquence = $1/L = 1/(N\ \Delta x)$, en u, et $1/(N\ \Delta y)$ en v (mais normalement $\Delta x=\Delta y$!).

  => vecteur de fréquences spatiales u (ou v) :
  $u=1/(N\Delta x)\ [-N/2\ldots0\ldots N/2-1]$

  => plus grande (haute!) fréquence spatiale disponible = $1/(N\Delta x)\ x\ N/2 = 1/2\ 1/\Delta x$



$\Delta x=L/N$
$L = N\ \Delta x = N\ L/N$

$\Delta u=1/L$
$N\ \Delta u = N/L = 1/\Delta x$

- Exemples sous Matlab :

  —> <u>FFT bidimensionnelle d'une sinusoïde</u> :

  `>> Tx=20;` —> période de 20 pixels (tôle ondulée)
  `>> x=0:99;`
  `>> whos x` —> vecteur de 100 valeurs de 0 à 99
  `>> I = ones(100,1) * (1 + cos(2*pi*x/Tx));`
  `>> imagesc(I), colorbar`

  (`>>improfile` sous Matlab, sinon un `plot` pour vérifier la sinusoïde selon l'axe des x)

  `>> Ichap = fft2(I);`
  `>> whos Ichap`
  `>> figure`
  `>> subplot(1,2,1), imagesc(I), title('Sinusoïde'), colorbar, axis('square')`
  `>> subplot(1,2,2), imagesc(abs(fftshift(Ichap))), title('|FFT(Sinusoïde)|'), colorbar, axis('square')`

  Remarque : ici, la fonction cosinus étant paire, la partie imaginaire de sa TF est nulle, et donc on aurait pu représenter de manière équivalente la partie réelle.

- **<u>Exercice 1</u>** : Reprendre l'exemple ci-dessus. Décaler la sinusoïde de 7.5 pixels. Que se passe-t-il au niveau (du module) de la TF ?