



# FILTRE D'IMAGE

Par : ANDRIAMAMY Rova Isaia  
ANDRIAMANANA Ranto Ny Aina  
RANDRIANANTENAINA Jean Olivier

# PLAN

- INTRODUCTION
- TRANSFORMEE DE FOURIER
- TRANSFORMATION EN ONDELETTES
- FONCTION MATLAB
- COMPARAISON
- CONCLUSION

# INTRODUCTION

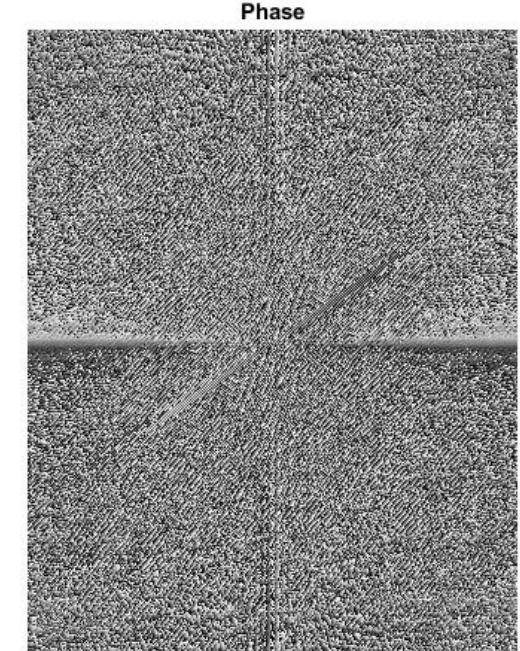
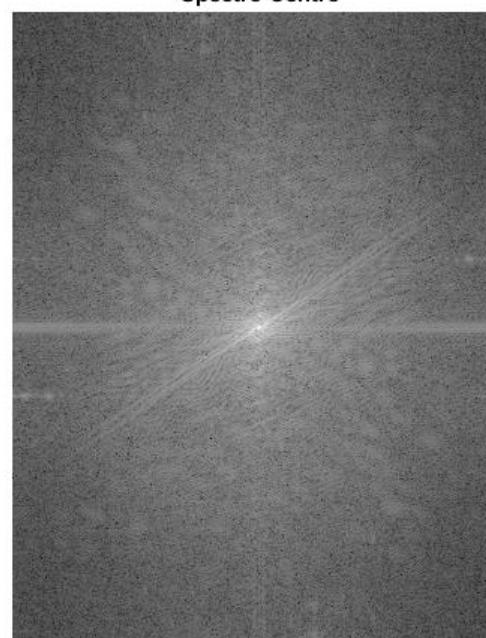
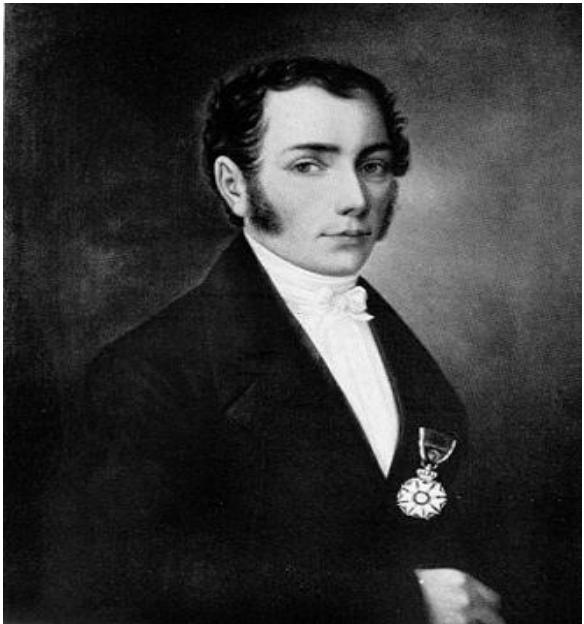
- Traitement d'image : améliorer ou extraire des informations visuelles.
- Filtrage d'image : atténuer le bruit, lisser, ou détecter des contours.
- Techniques clés : Transformée de Fourier et Transformée en Ondelettes.

# Transformée de Fourier

- Principe
- Application au filtrage
- Avantages et inconvénient

# Principe

- Transformée de Fourier décompose une image en sinusoïdes de différentes fréquences.
- Elle est utilisée pour convertir une image du **domaine spatial** (où chaque pixel représente une valeur de couleur ou d'intensité lumineuse) au **domaine fréquentiel**



# Application au filtrage

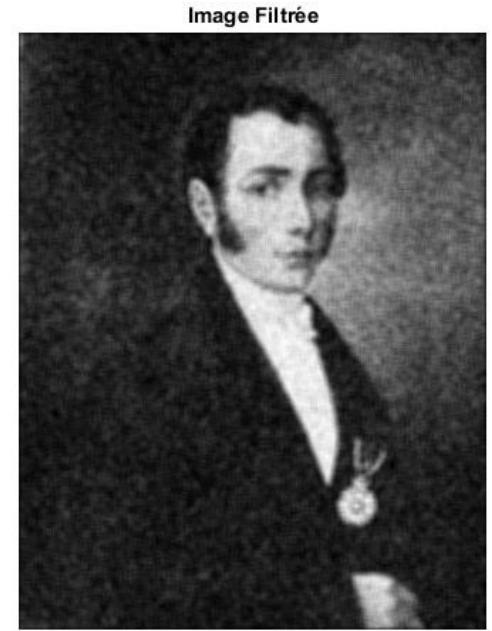
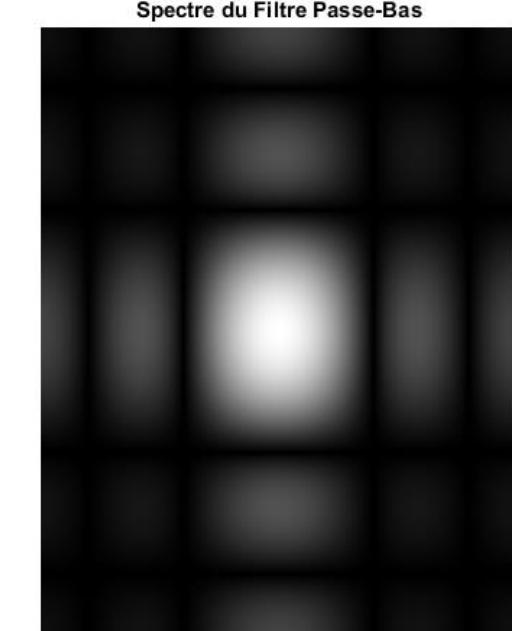
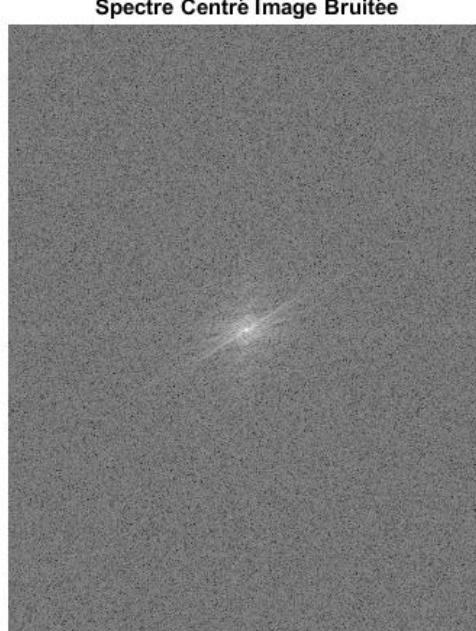
Utilisée pour filtrer l'image.

Filtrage fréquentiel, suppression des composantes indésirables (bruit haute ou basse fréquence).

$$a_f = a * h$$



$$a_f = TF^{-1}[A(f) \cdot H]$$



# Avantages et Inconvénients

- Avantages : Simple à implémenter, efficace pour les images périodiques.
- Inconvénients : Moins efficace pour capturer les détails locaux, signaux non stationnaires.

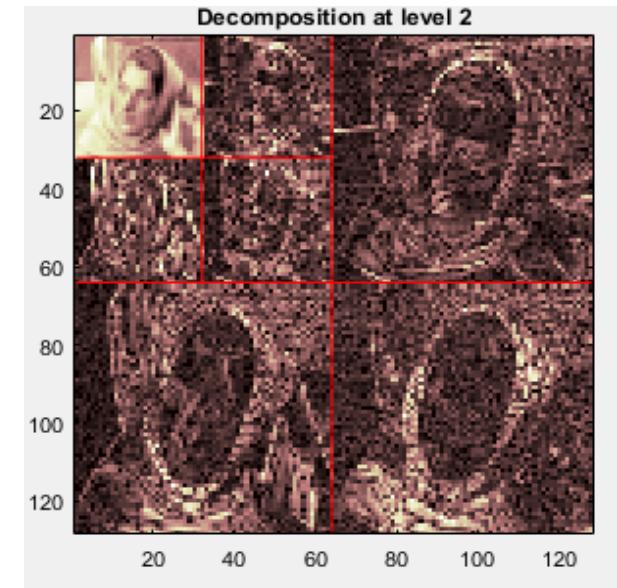
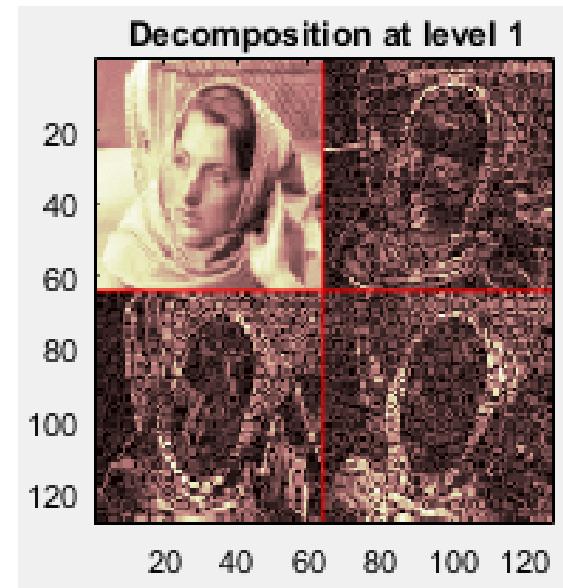
# Transformation en ondelettes

- Principe
- Application au filtrage
- Avantages et inconvénient

# Principe

La **transformation en ondelettes** est utilisée pour analyser une image à différentes résolutions, permettant une localisation à la fois en fréquence et en espace.

- Représentation multi-résolution des signaux.
- Utilisation d'ondelettes de forme localisée pour capter des détails temporels.
- Décomposition du signal en approximation et détails.

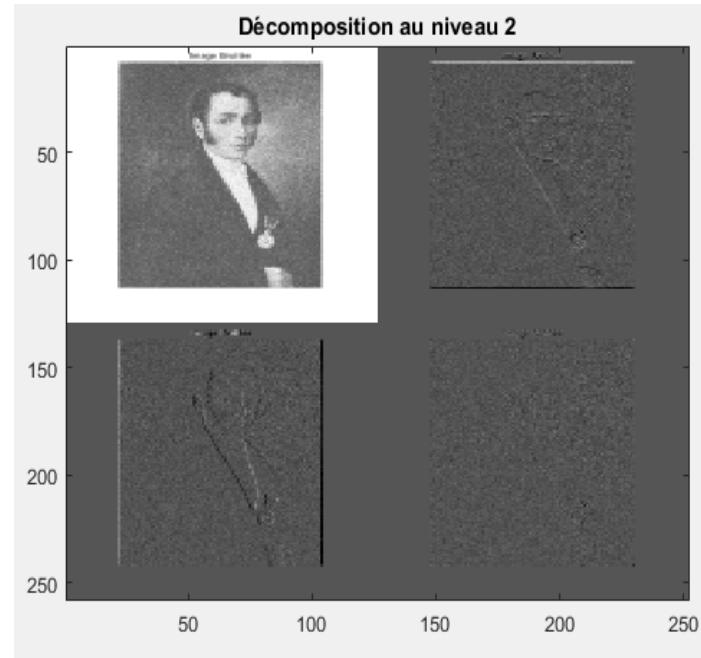


# Application au filtrage

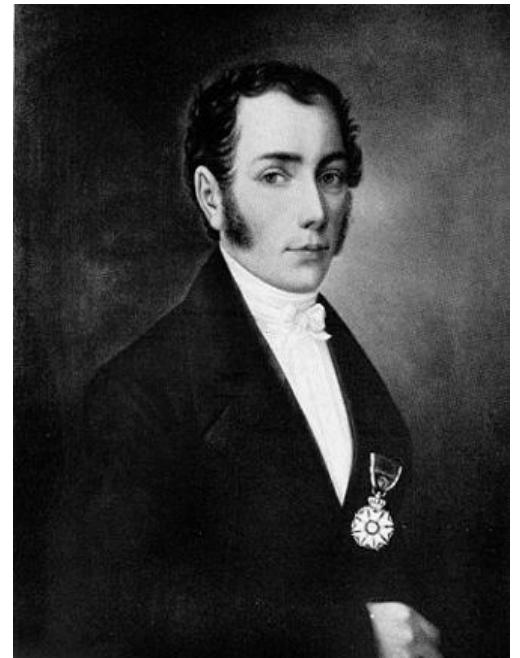
- Décomposition en plusieurs échelles (étaillants des hautes fréquences aux basses).
- Suppression du bruit à différentes résolutions.
- Reconstruction



Décomposition  
→



Seuillage  
→  
Reconstruction



# Avantages et Inconvénients

Avantages : Capacité à capturer les détails locaux et les structures multi-échelles.

Inconvénients : Plus complexe à implémenter et à interpréter que la TF.

# Fonction matlab

- ddencmp
- wdencmp

# Fonction ddencmp

$$[thr, sorh, kp] = ddencmp(in1, in2, x)$$

## Paramètres

*in1*: Type d'opération ‘den’ ou ‘cmp’

*in2*: Type de décomposition ‘wv’ ou ‘wp’

*x*: Donnée d'entrée

## Sorties

*thr*: Seuil calculé

*sorh*: Type de seuillage ‘s’ ou ‘h’

*kp* : Conserver les coefficients d'approximation (1) ou non (0)

# Fonction wdencmp

$$xd = \text{wdencmp}(\text{type}, x, \text{wname}, N, \text{thr}, \text{sorh}, kp)$$

## Paramètres

Type: Type de seuil à appliquer 'gbl' ou 'lvd'

wname: Nom de l'ondelette utilisée

N: Niveau de décomposition

## Noms des ondelettes

'haar'

'db1' à 'db20' (Daubechies)

'sym2' à 'sym45' (Symlets)

'coif1' à 'coif5' (Coiflets)

'bior1.1' à 'bior6.8' (Biorthogonal)

'rbio1.1' à 'rbio6.8' (Reverse Biorthogonal)

'meyr' (Meyer)

'dmey' (Discrete Meyer)

# **Comparaison sur Matlab**

- Code
- Résultats

# Code Matlab

```
% Charger l'image
a = imread('moon.jpg');

% Ajouter du bruit gaussien
noiseLevel = 0.01;
noise = noiseLevel * randn(size(a));
img_noisy = im2double(a) + noise;
img_noisy = min(max(img_noisy, 0), 1);
%Normalisé

% Afficher l'image bruitée
figure;imshow(img_noisy, []);
title('Image bruitée');

% Créer un filtre gaussien
H = fspecial('gaussian', [9 9], 2);

% Spectre de l'image bruitée
ma = abs(fft2(img_noisy));

% Spectre du filtre gaussien
[nblij, nbcol] = size(img_noisy);
mH = abs(fft2(H, nblij, nbcol));

% Reconstruction
maf = ma .* mH;
phase = angle(fft2(img_noisy));
z = maf .* exp(li * phase);
z2 = ifft2(z);
img_filtered = real(z2);
```

```
% Affichage de l'image filtrée
figure;
imshow(img_filtered, []);
title('Image filtrée avec un filtre gaussien');

% Filtrage par ondelettes
[thr, sorh, kp] = ddencmp('den', 'wv', img_noisy);
imgden_wavelet = wdencmp('gbl', img_noisy, 'sym4', 2, thr, sorh,
kp);

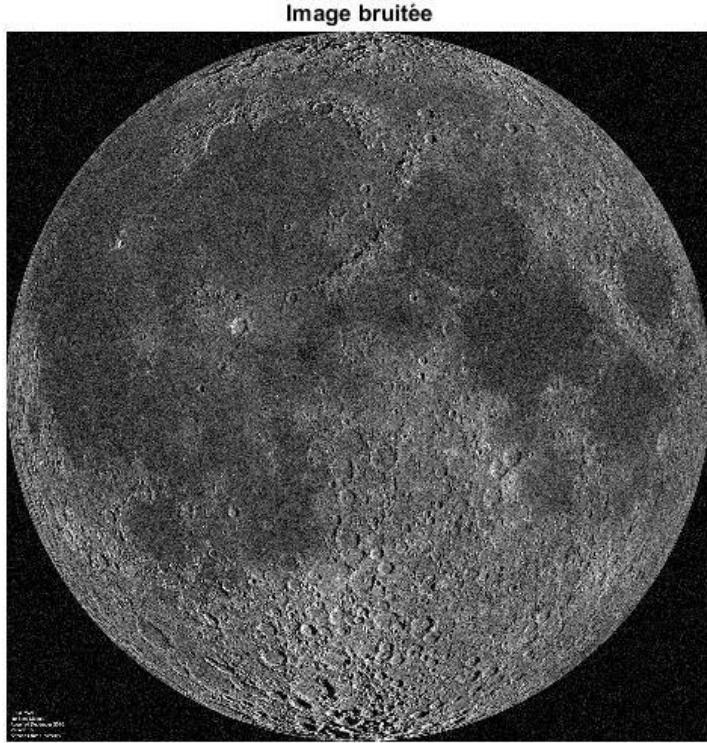
% Affichage de l'image filtrée par ondelettes
figure;
imshow(imgden_wavelet, []);
title('Image filtrée par ondelettes');

% Calculer le PSNR entre l'image originale et l'image filtrée (TF)
mse_filtered = mean((im2double(a(:)) - img_filtered(:)).^2);
psnr_filtered = 10 * log10(1 / mse_filtered); % PSNR pour l'image
filtrée par TF

% Calculer le PSNR entre l'image originale et l'image filtrée par
ondelettes
mse_wavelet = mean((im2double(a(:)) - imgden_wavelet(:)).^2);
psnr_wavelet = 10 * log10(1 / mse_wavelet); % PSNR pour l'image
filtrée par ondelettes

% Afficher le PSNR
disp(['PSNR de l''image filtrée par TF : ' num2str(psnr_filtered,
'%.2f') ' dB']);
disp(['PSNR de l''image filtrée par ondelettes : '
num2str(psnr_wavelet, '=%.2f') ' dB']);
```

# Résultats



*PSNR: 21.61 dB*

*PSNR: 26.94 dB*

# Conclusion

- TF : Efficace pour le filtrage global, mais ne capture pas bien les détails locaux.
- TO : Analyse plus fine et localisée, adaptée aux signaux non stationnaires.
- Le choix dépend du contexte : TF pour les images régulières, TO pour les images complexes et bruitées.