

Université Paul Sabatier Toulouse

Compte Rendu

Travaux pratiques de Traitement d'Images

M1 EEA

- ✗ Module KEAX7AK1 : Traitement des images
- ✗ Enseignant : A. Herbulet

- ✗ Réalisé par :
 - KABOU Abdeldjalil
 - IAICHE Achour Mehdi Anis

Travaux pratiques de Traitement d'Images

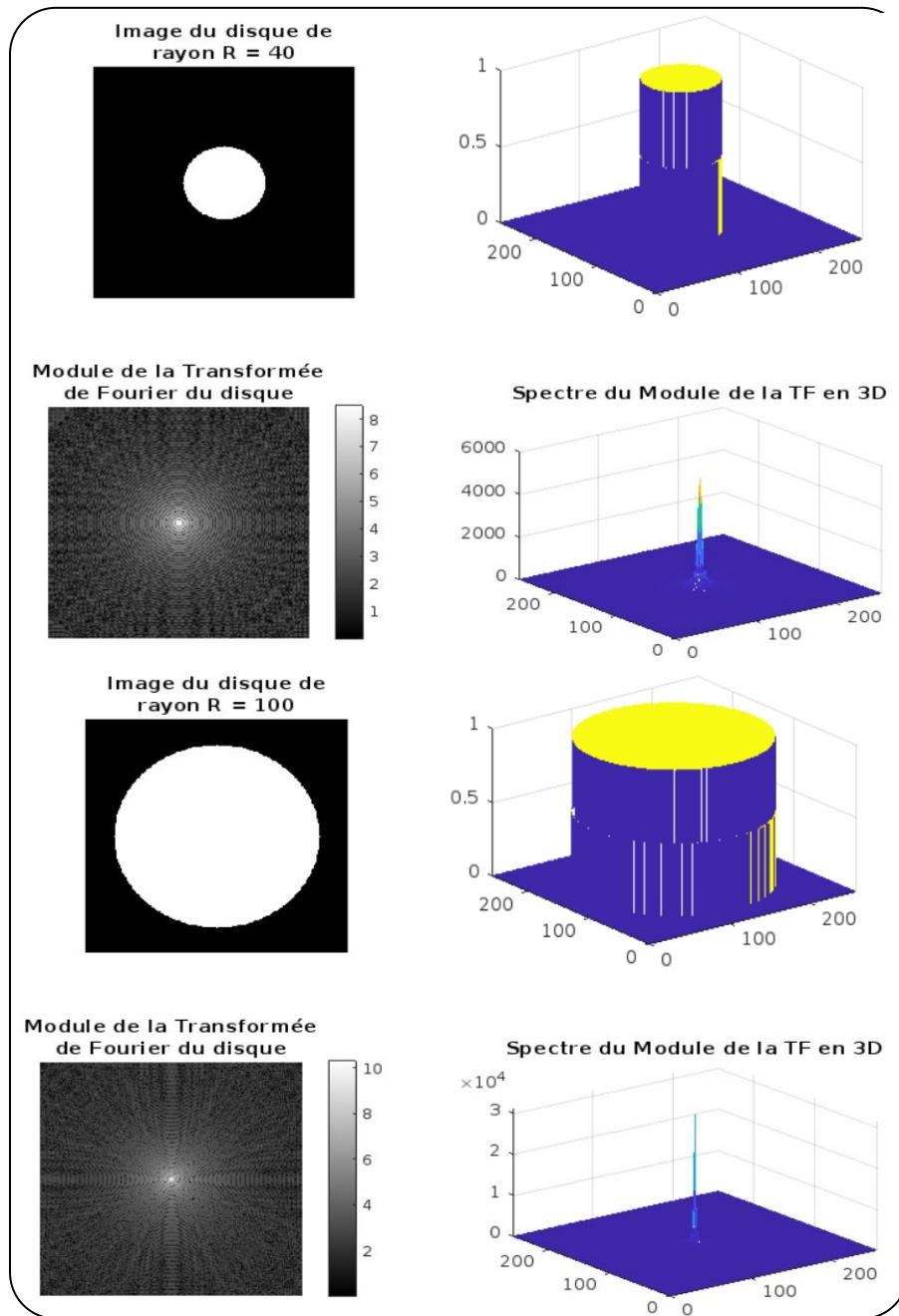
Objectif

Le but est de manipuler la transformée de Fourier (TF) sur des images en niveaux de gris et d'effectuer du filtrage dans le domaine fréquentiel.

1 Transformée de Fourier

1.1 Calcul de TF

- Disque plein (2D)



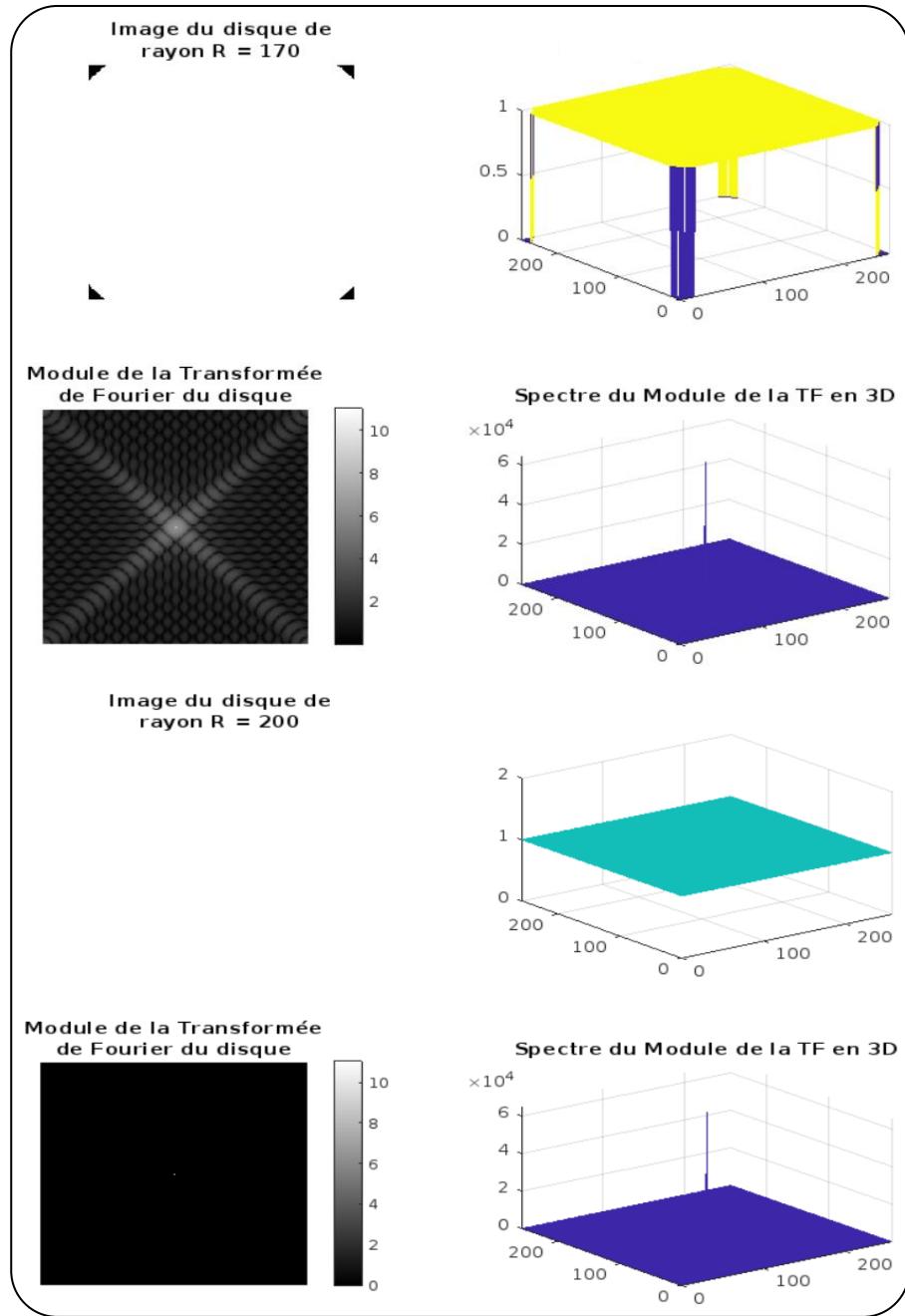
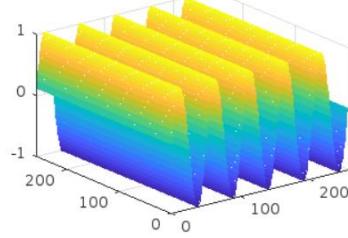
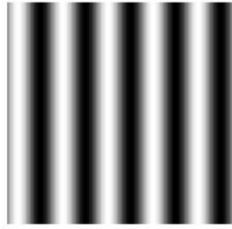


Image disque plein (2D) de différent rayon et sa FFT

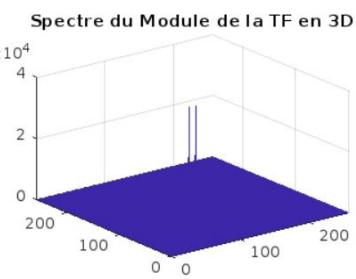
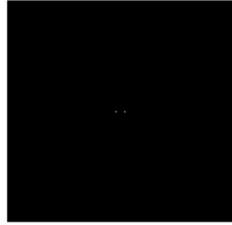
Lorsque le rayon du disque est petit, la TF est large et diffuse (Les anneaux concentriques sont espacés et s'étendent loin du centre), Et lorsque le rayon du disque augmente, la TF devient plus étroite et concentrée autour du centre de la fréquence (le zéro-fréquence). Les anneaux se rapprochent et sont plus fins.

- **Image sinusoïdale verticale / horizontale**

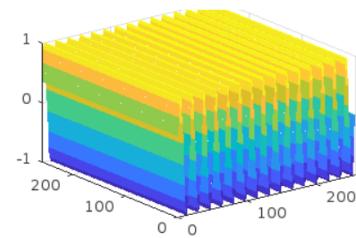
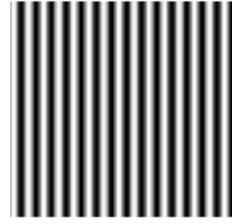
**Image sinusoïdale verticale
de fréquence $f = 5$**



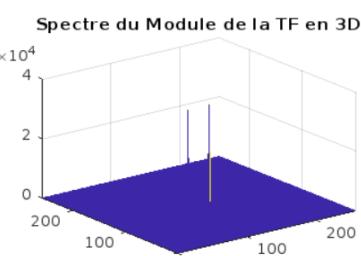
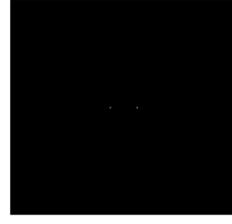
**Module de la TF
de l'Image sinusoïdale verticale**



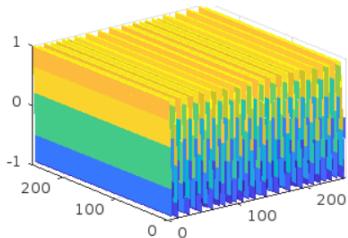
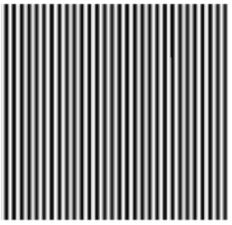
**Image sinusoïdale verticale
de fréquence $f = 15$**



**Module de la TF
de l'Image sinusoïdale verticale**



**Image sinusoïdale verticale
de fréquence $f = 30$**



**Module de la TF
de l'Image sinusoïdale verticale**

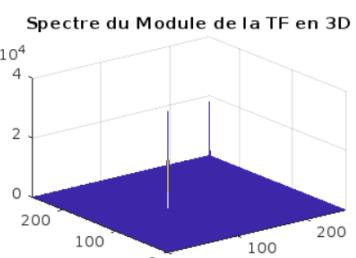
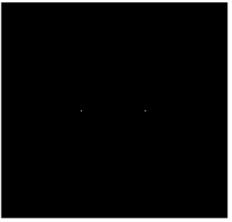
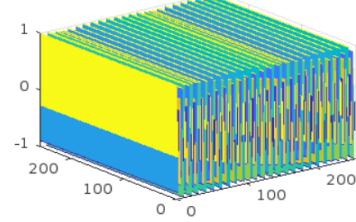
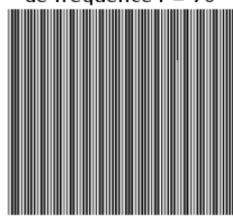
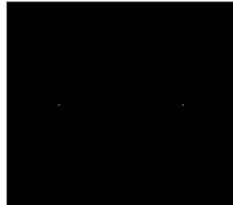


Image sinusoïdale verticale
de fréquence $f = 70$



Module de la TF
de l'Image sinusoïdale verticale



Spectre du Module de la TF en 3D

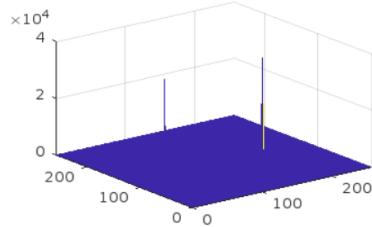
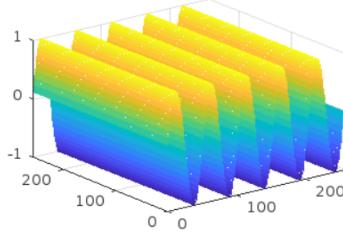
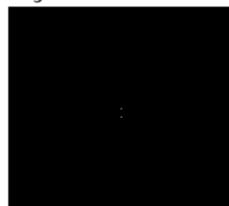


Image sinusoïdale Verticale et sa FFT

Image sinusoïdale horizontale
de fréquence $f = 5$



Module de la TF
de l'Image sinusoïdale horizontale



Spectre du Module de la TF en 3D

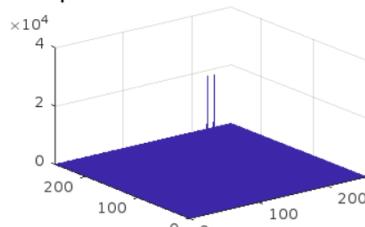
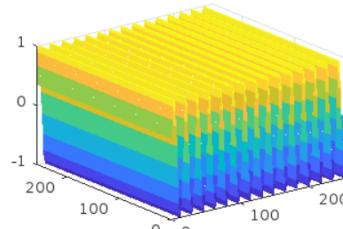
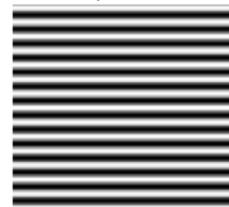
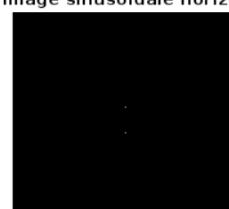


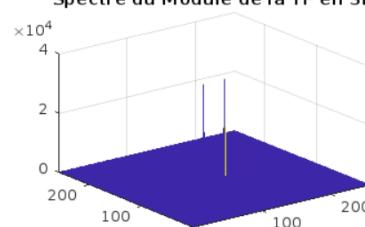
Image sinusoïdale horizontale
de fréquence $f = 15$



Module de la TF
de l'Image sinusoïdale horizontale



Spectre du Module de la TF en 3D



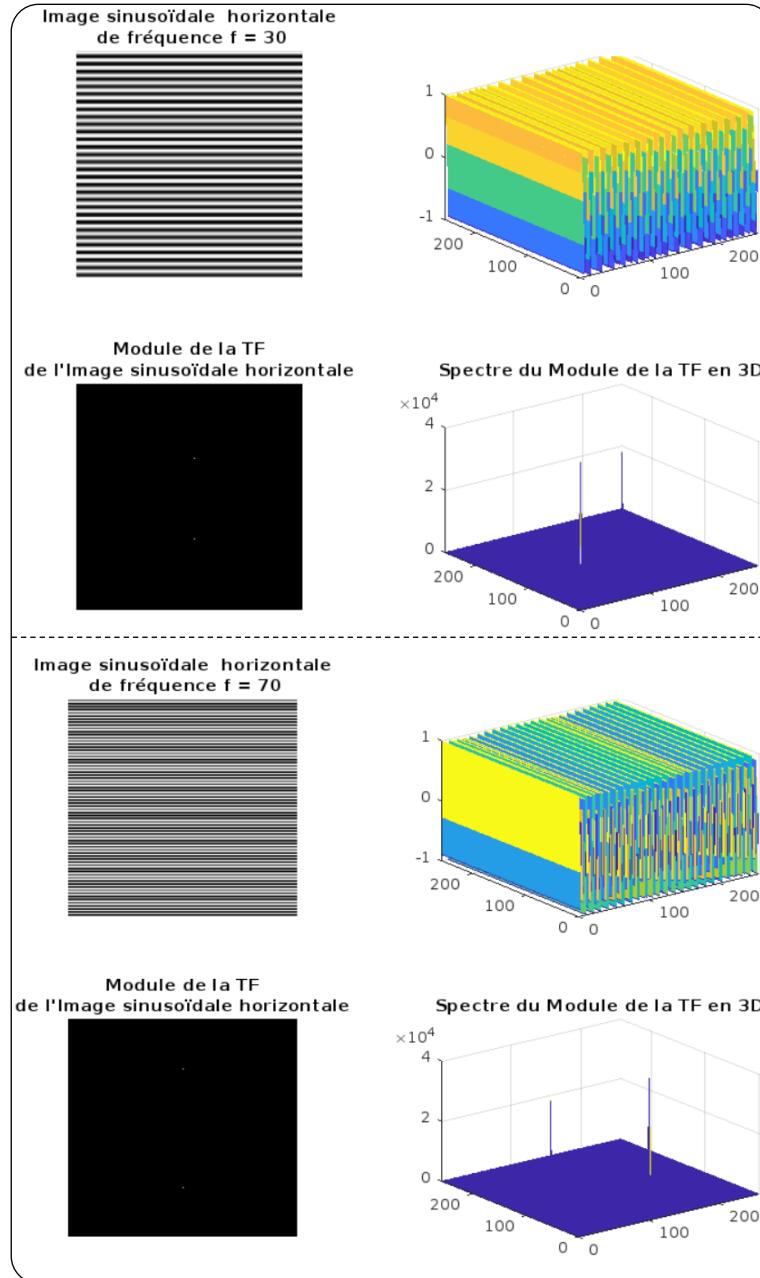


Image sinusoïdale Horizontale et sa FFT

Lorsque la fréquence de la sinusoïde est faible, les pics de la TF se situent plus près du centre de l'image. Et lorsque la fréquence augmente, les pics s'éloignent du centre (plus la fréquence est élevée, plus la distance est grande par rapport au centre).

Direction des pics : Si l'image contient une sinusoïde horizontale, les pics apparaissent sur l'axe vertical de la TF. Si la sinusoïde est verticale, les pics apparaissent sur l'axe horizontal.

- **Image gaussienne centrée 2D**

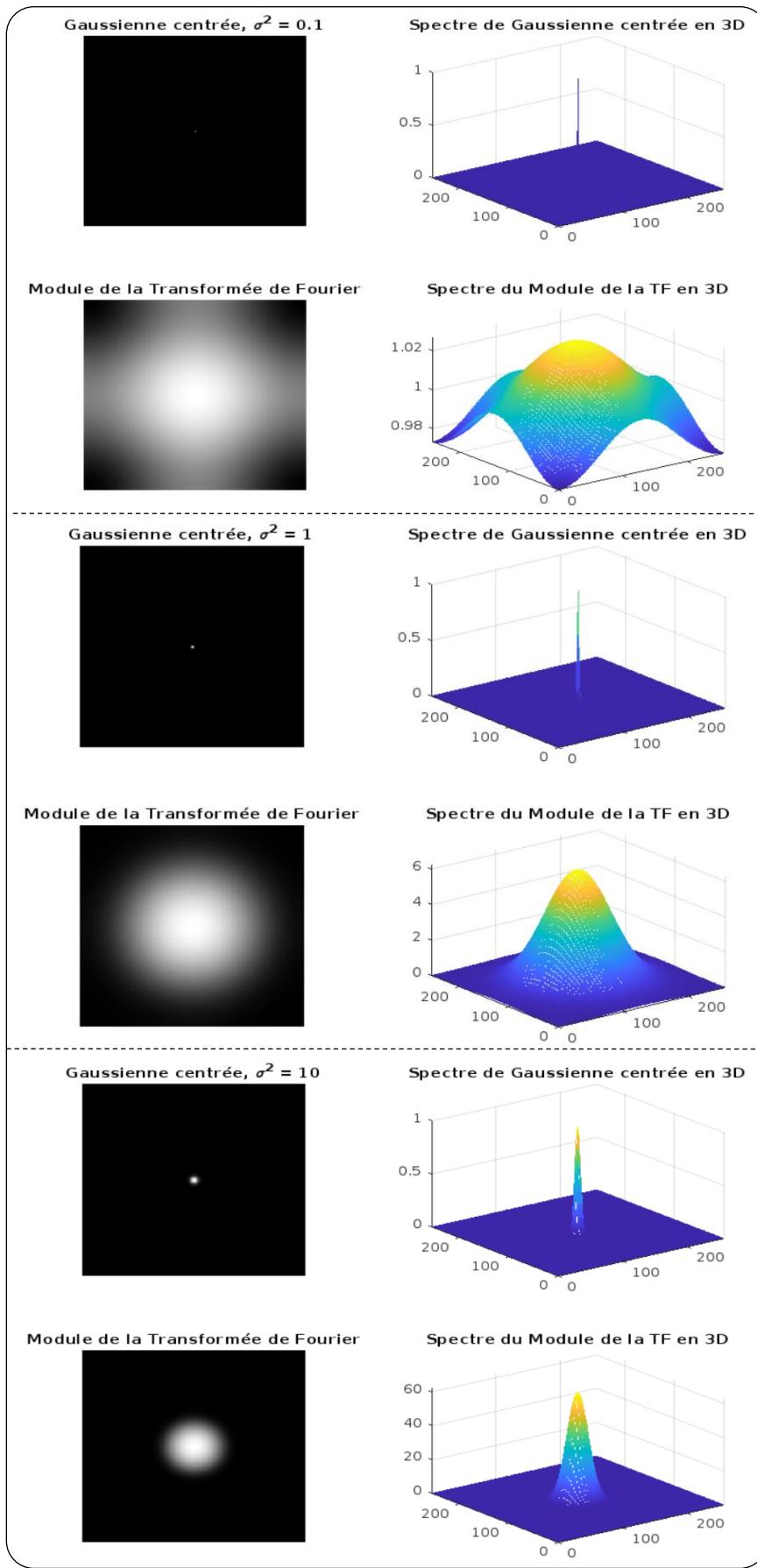
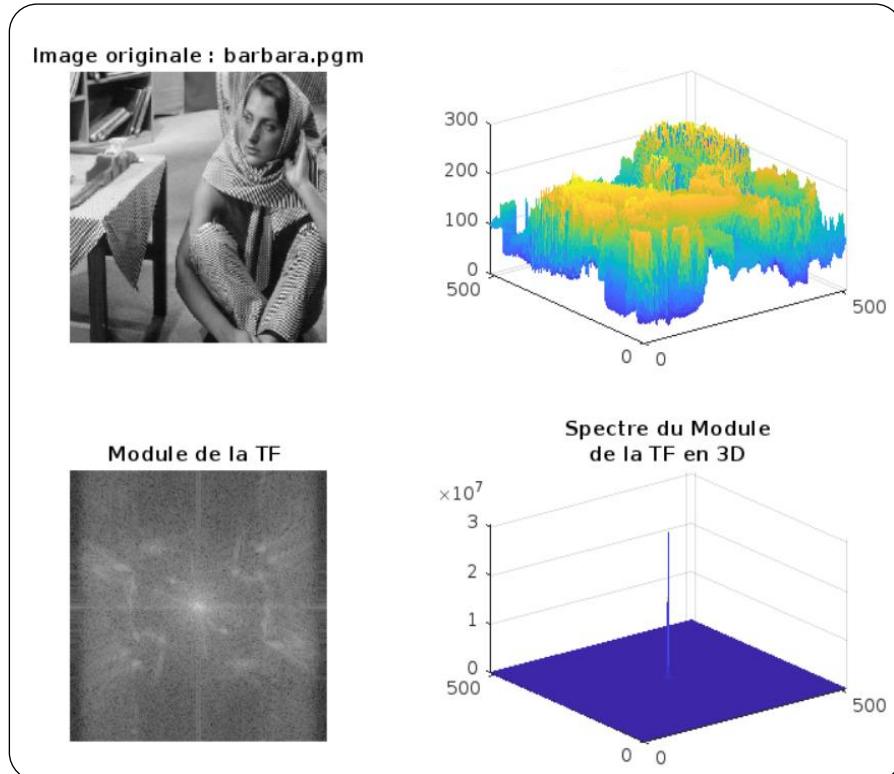


Image gaussienne centrée 2D et sa FFT

Pour une faible variance (σ^2 petite), la TF est large et s'étend sur une grande portion de l'espace fréquentiel (Une gaussienne étroite dans le domaine spatial produit une TF large). Et pour une grande variance (σ^2 grande), la TF est concentrée au centre et décroît rapidement (Une gaussienne large dans le domaine spatial produit une TF étroite).

- **Image « barbara.pgm »**



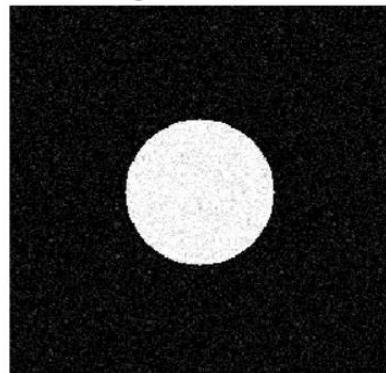
*Image **barbara.pgm** et sa FFT*

Le centre de la transformée de Fourier (TF) représente les basses fréquences, qui correspondent aux structures générales et aux variations globales de luminance de l'image. Les bords de la TF, en revanche, représentent les hautes fréquences, qui sont associées aux détails fins et aux contours de l'image.

1.2 Influence du bruit

- **Bruit blanc gaussien à l'image du disque**

Image du disque avec bruit blanc gaussien, $\sigma^2 = 0.1$



Module de la TF avec bruit blanc gaussien

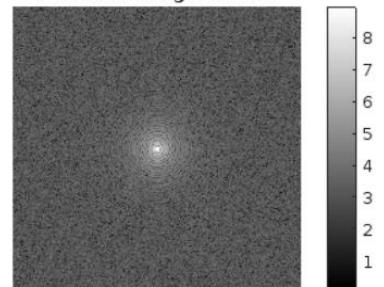
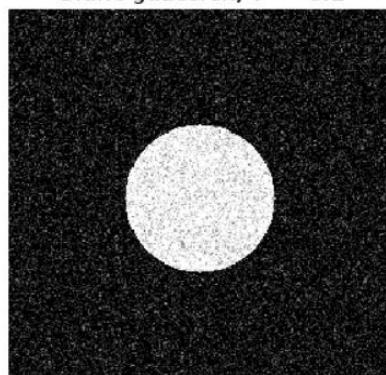


Image du disque avec bruit blanc gaussien, $\sigma^2 = 0.2$



Module de la TF avec bruit blanc gaussien

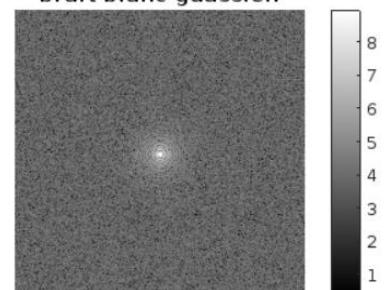
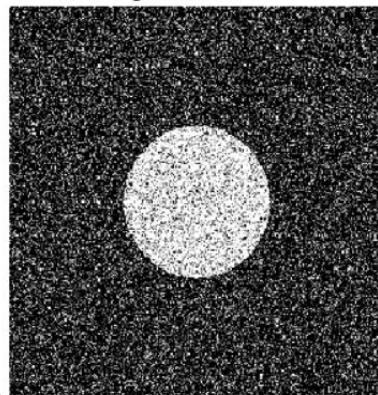


Image du disque avec bruit blanc gaussien, $\sigma^2 = 0.5$



Module de la TF avec bruit blanc gaussien

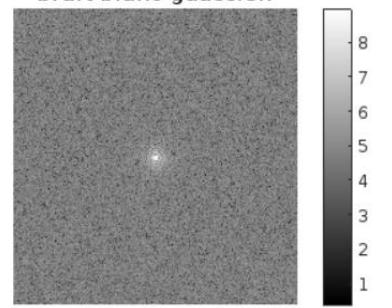


Image du disque avec un bruit blanc et sa FFT

L'ajout du bruit gaussien rend le spectre de Fourier de l'image moins structuré et augmente la présence de composantes de fréquence aléatoires, ce qui diminue la clarté et la netteté des caractéristiques initiales de la TF de l'image du disque.

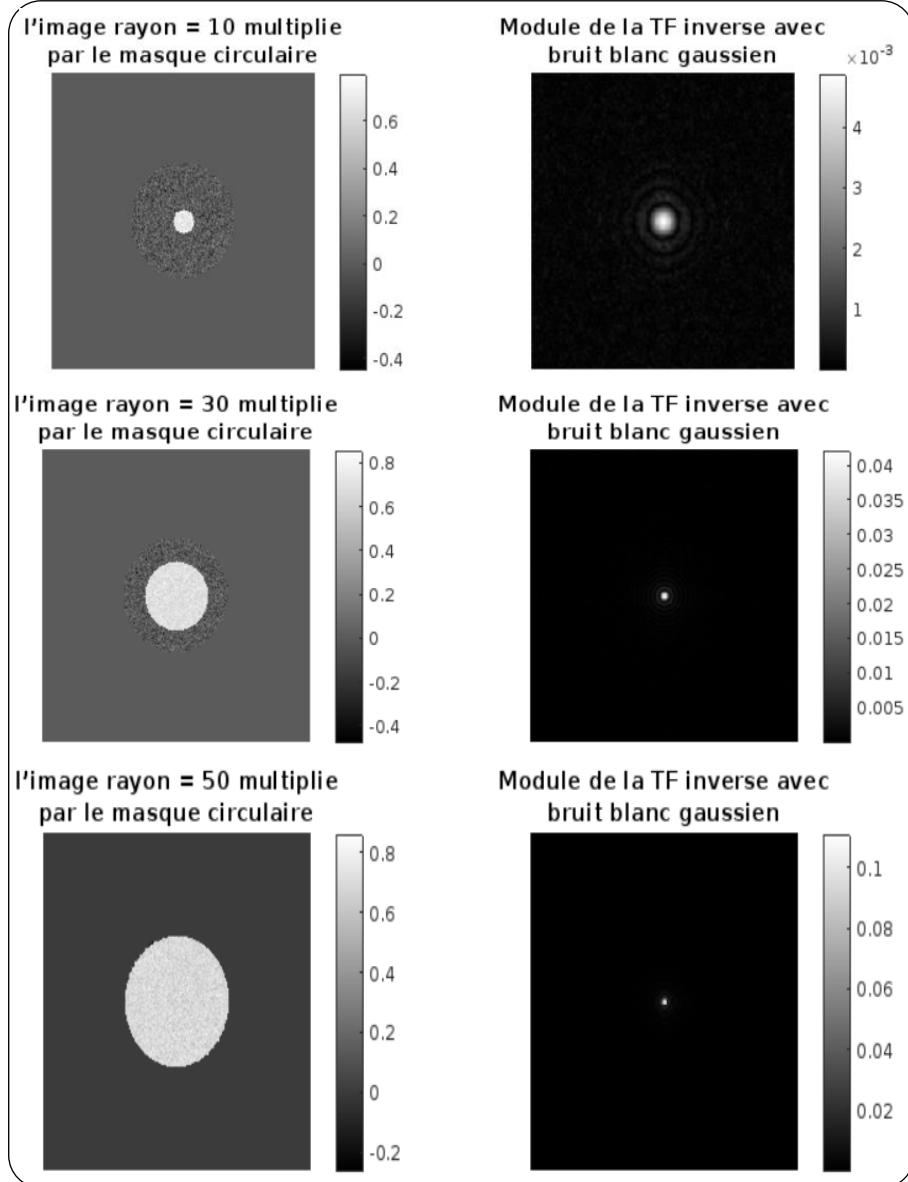


Image Bruitée (en changeant le rayon de l'image) avec un masque et sa IFFT

Ici nous avons fixé le rayon du masque et nous avons varié chaque fois le rayon de l'image du disque bruité, nous avons obtient :

- **Rayon faible** : Le spectre de la TF est large, mais avec une forte suppression des hautes fréquences, ce qui donne une image plus floue.
- **Rayon élevé** : Le spectre de la TF est plus étroit, mais avec moins de suppression des hautes fréquences, donc l'image conserve plus de détails fins.

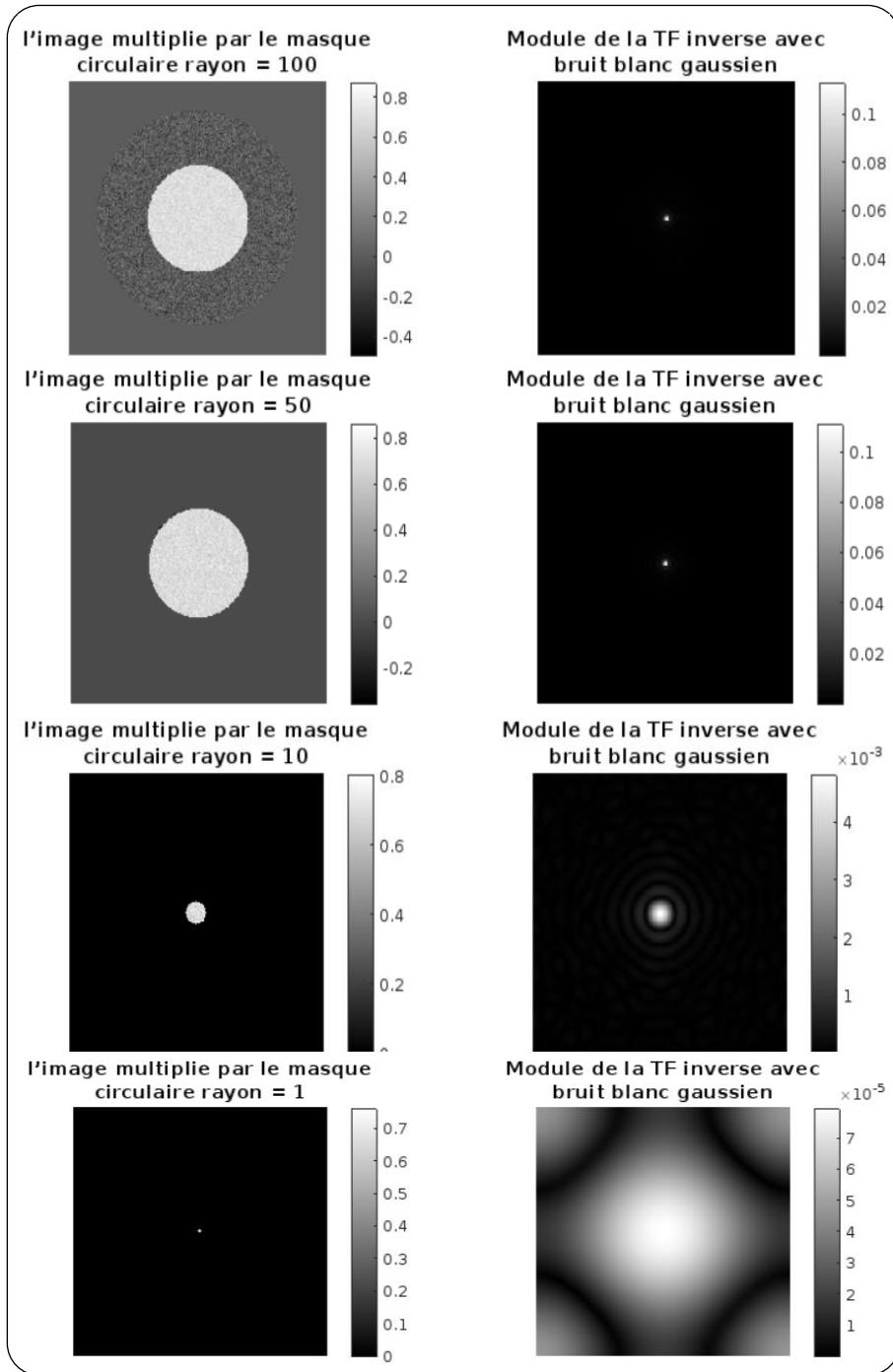


Image Bruitée avec un Masque (en changeant le rayon du masque) et sa IFFT

Ici nous avons fixé le rayon de l'image et nous avons varié chaque fois le rayon du masque, nous avons obtient :

Lorsque le rayon du masque est petit, après le filtrage (la TF inverse), l'image résultante sera très lissée et perdra des détails et Le spectre de la TF sera plus. Et lorsque le rayon du masque est grand, la TF conserve une plus grande partie des fréquences, et l'image résultante après TF inverse sera plus nette, Le spectre est plus concentré autour du centre

- **Type d'opération**

Nous avons réalisé une **opération de filtrage fréquentiel** via un masque dans le domaine de Fourier, suivi d'une reconstruction de l'image via la TF inverse pour évaluer l'effet du filtrage sur la netteté et les détails de l'image.

- **La même chose avec l'Image « barbara.pgm »**

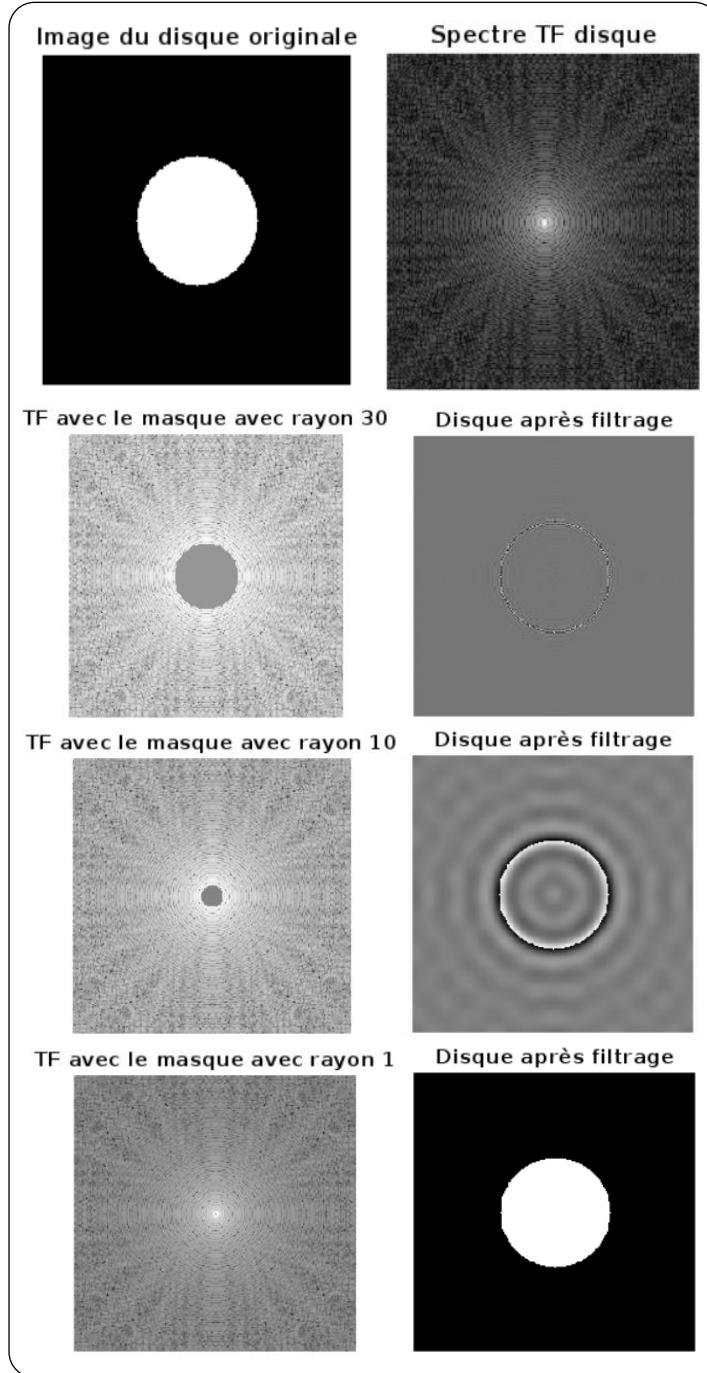


*Image « **barbara.pgm** » (en changeant le rayon du masque) et sa IFFT*

Pour le rayon du masque est petit, seules les basses fréquences sont conservées. Vous verrez une image plus floue, car les détails fins ont été supprimés (La suppression des hautes fréquences entraîne une image lissée. C'est utile pour la réduction du bruit, mais la perte de détails peut affecter la qualité). Et pour le rayon est grand, l'image conservera plus de détails, apparaissant plus nette. Cela montre que des parties plus larges du spectre fréquentiel ont été conservées.

1.3 Filtrage dans le domaine fréquentiel

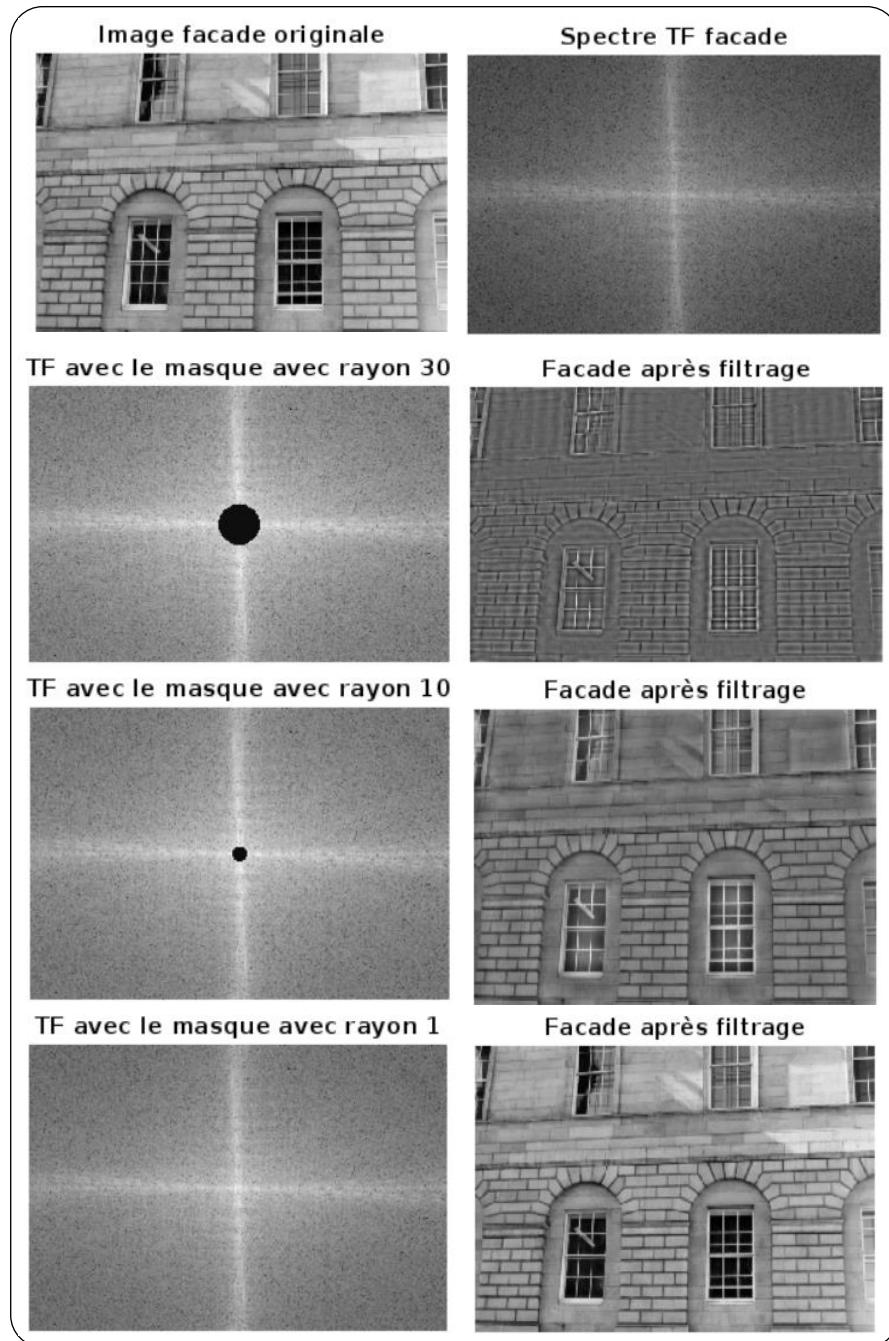
Le filtrage dans le domaine fréquentiel consiste à manipuler le spectre de fréquence de l'image après l'application de la Transformée de Fourier (TF). En multipliant le spectre par un masque spécifique (comme un masque circulaire), on peut atténuer ou éliminer certaines fréquences avant de revenir à l'espace spatial avec la Transformée de Fourier inverse (TFI).



Dans le cas d'une image de disque, lorsque nous appliquons un masque de très petit rayon (qui laisse seulement passer les très basses fréquences) et que nous effectuons la Transformée de Fourier inverse (TF inverse), l'image résultante ressemble encore au disque mais avec des contours flous et lissés.

Cela se produit parce que les détails fins et les hautes fréquences (qui représentent les bords nets et les détails précis) ont été supprimés par le filtrage, laissant seulement la structure de base de l'image.

Pour l'Image « facade.png »



Un masque de rayon 1 dans le domaine fréquentiel agit comme un filtre passe-bas très sévère, supprimant presque toutes les hautes fréquences et ne laissant que les composantes de très basse fréquence. Par conséquent, l'image résultante après la TF inverse affiche la forme globale du disque mais sans détails précis, ce qui explique pourquoi elle apparaît presque comme l'image originale, mais plus lisse et floue.

1.4 Importance de la phase

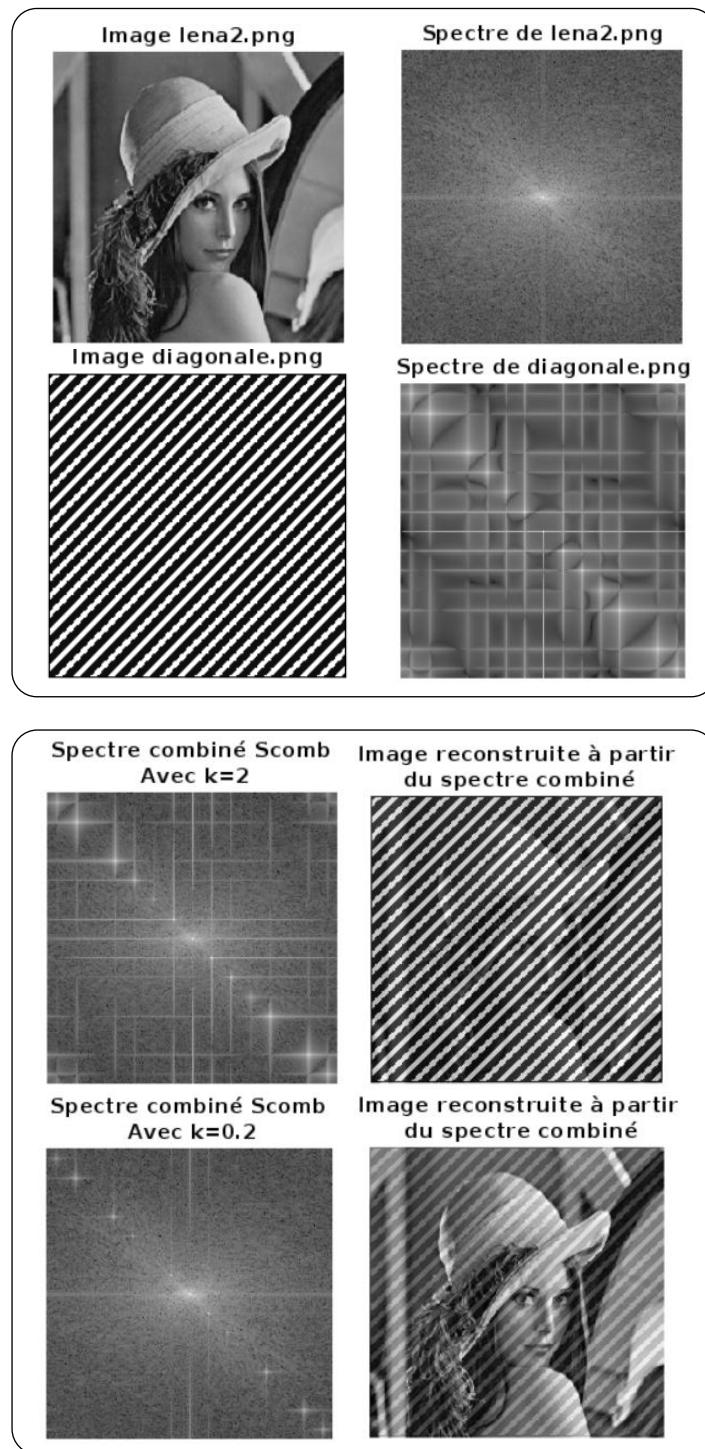


Phase de Barbara + Module de Lena : L'image obtenue ressemble fortement à Barbara, car la phase conserve les détails structurels de cette image.

Phase de Lena + Module de Barbara : De même, l'image ressemble à Lena.

La phase joue un rôle crucial dans la reconstruction de l'image et contient la majorité de l'information visuelle nécessaire pour la perception. Le module affecte principalement l'intensité et le contraste, mais l'apparence globale est dominée par la phase.

2. Correction de défauts



L'image obtenue après la TF inverse montre une version de Lena à de légers motifs diagonaux, tandis que le spectre combiné visualisé met en évidence la fusion des deux spectres, dominée par les composantes de lena2.png avec des ajouts subtils de diagonale.png.

Annexes

1 Transformée de Fourier

Calcul de TF

- *Disque plein (2D)*

```
function Img = CreateCircle(N, r)
    % Crée une image binaire NxN contenant un disque de rayon r au centre
    [x, y] = meshgrid(-N/2:N/2-1, -N/2:N/2-1);
    Img = sqrt(x.^2 + y.^2) <= r;
end
N = 256; % Taille de l'image (NxN)
%R = 40; % Rayon du disque
%R = 100;
%R = 150;
%R = 170;
R = 200;
% Création de l'image du disque avec la fonction CreateCircle
Img = CreateCircle(N, R);
% Affichage de l'image du disque
figure;
subplot(2, 2, 1);
imshow(Img, []);
title(['Image du disque de', newline,'rayon R = ', num2str(R)]);
subplot(2, 2, 2);
mesh(Img);
title(['Spectre de l''Image', newline,'du disque de rayon en 3D']);
% Calcul de la transformée de Fourier
TF = fftshift(fft2(Img)); % Centrer la TF avec fftshift
module_TF = abs(TF); % Calcul du module
% Affichage du module de la transformée de Fourier
subplot(2, 2, 3);
imshow(log(1 + module_TF), []); % Log pour améliorer la visualisation
title(['Module de la Transformée', newline,'de Fourier du disque']);
colorbar;
subplot(2, 2, 4);
mesh(module_TF);
title('Spectre du Module de la TF en 3D');
```

- *Image sinusoïdale verticale / horizontale*

```
function Img = CreateSinus(N, f)
    % Crée une image N x N d'une sinusoïde horizontale de fréquence f
    x = 1:N;
    sinusoidalLine = sin(2 * pi * f * x / N); % Génère une ligne de sinusoïde
    Img = repmat(sinusoidalLine, N, 1); % Réplique cette ligne sur toutes les lignes de l'image
end
function Img = CreateSinus_H(N, f)
    % Crée une image N x N d'une sinusoïde verticale de fréquence f
    y = 1:N;
    sinusoidalColumn = sin(2 * pi * f * y / N).'; % Génère une colonne de sinusoïde
    Img = repmat(sinusoidalColumn, 1, N); % Réplique cette colonne sur toutes les colonnes
```

```

    de l'image
end
% Paramètres
N = 256; % Taille de l'image
% Différentes fréquences pour observer l'évolution
%frequencies = 5;
%frequencies = 15;
%frequencies = 30;
frequencies = 70;
% Créer une image sinusoïdale
Img = CreateSinus(N, frequencies);
Img_H = CreateSinus_H(N, frequencies);
% Calcul de la transformée de Fourier
TF = fftshift(fft2(Img));
module_TF = abs(TF);
TF_H = fftshift(fft2(Img_H));
module_TF_H = abs(TF_H);
% Affichage
figure;
%subplot(2, 2, 1);
%imshow(Img, []);
%title(['Image sinusoïdale verticale', newline, 'de fréquence f = ', num2str(frequencies)]);
%subplot(2, 2, 2);
%mesh(Img);
%title(['Spectre de l''Image', newline, 'sinusoïdale verticale en 3D']);
subplot(2, 2, 1);
imshow(Img_H, []);
title(['Image sinusoïdale horizontale', newline, 'de fréquence f = ', num2str(frequencies)]);
subplot(2, 2, 2);
mesh(Img);
title(['Spectre de l''Image', newline, 'sinusoïdale horizontale en 3D']);
%subplot(2, 2, 3);
%imshow(log(1 + module_TF), []);
%title(['Module de la TF', newline, 'de l''Image sinusoïdale verticale']);
%subplot(2, 2, 4);
%mesh(module_TF);
%title('Spectre du Module de la TF en 3D');
%subplot(3, 3, 3);
%mesh(module_TF);
subplot(2, 2, 3);
imshow(log(1 + module_TF_H), []);
title(['Module de la TF', newline, 'de l''Image sinusoïdale horizontale']);
subplot(2, 2, 4);
mesh(module_TF);
title('Spectre du Module de la TF en 3D');

```

- *Image gaussienne centrée 2D*

```

function Img = CreateGauss(N, v)
    % Crée une image N x N contenant une gaussienne centrée de variance v
    [x, y] = meshgrid(-N/2:N/2-1, -N/2:N/2-1);
    sigma = sqrt(v); % L'écart type est la racine carrée de la variance
    Img = exp(-(x.^2 + y.^2) / (2 * sigma^2));
end

```

```

% Paramètres
N = 256; % Taille de l'image
variances = 0.1; % Différentes valeurs de variance
%variances = 1;
%variances = 10;
% Créer une image gaussienne
Img = CreateGauss(N, variances);
% Calcul de la transformée de Fourier
TF = fftshift(fft2(Img));
module_TF = abs(TF);
% Affichage
figure;
subplot(2, 2, 1);
imshow(Img, []);
title(['Gaussienne centrée, \sigma^2 = ', num2str(variances)]);
subplot(2, 2, 2);
mesh(Img);
title('Spectre de Gaussienne centrée en 3D');
subplot(2, 2, 3);
imshow(log(1 + module_TF), []);
title('Module de la Transformée de Fourier');
subplot(2, 2, 4);
mesh(module_TF);
title('Spectre du Module de la TF en 3D');

```

- *Image "barbara.pgm"*

```

title('Spectre du Module de la TF en 3D');
% Charger l'image "barbara.pgm"
Img = imread('barbara.pgm');
% Calculer la transformée de Fourier
TF = fftshift(fft2(double(Img)));
module_TF = abs(TF);
% Affichage de l'image originale et de son module de la TF
figure;
subplot(2, 2, 1);
imshow(Img, []);
title('Image originale : barbara.pgm');
subplot(2, 2, 2);
mesh(Img);
title(['Spectre de l''Image', newline, 'originale en 3D']);
subplot(2, 2, 3);
imshow(log(1 + module_TF), []);
title('Module de la TF ');
subplot(2, 2, 4);
mesh(module_TF);
title(['Spectre du Module', newline, 'de la TF en 3D']);

```

Influence du bruit

- *TF du disque bruit et le Masque et TF inverse*

```

function Img = CreateCircle(N, r)
% Crée une image binaire NxN contenant un disque de rayon r au centre
1

```

```

[x, y] = meshgrid(-N/2:N/2-1, -N/2:N/2-1);
Img = sqrt(x.^2 + y.^2) <= r;
end
% Génération d'une image de disque
N = 256; % Taille de l'image
r = 50; % Rayon du disque
Img = CreateCircle(N, r);
% Ajouter du bruit blanc gaussien
sigma = 0.10;
%sigma = 0.20;
%sigma = 0.50; % Définir la variance du bruit
bruit = sigma * randn(size(Img));
Img_bruitee = Img + bruit;
% Calcul de la transformée de Fourier
TF = fftshift(fft2(Img_bruitee));
module_TF = abs(TF);
% Créer un masque circulaire de rayon spécifié
[r, c] = size(Img_bruitee); % Taille de l'image
[cols, rows] = meshgrid(1:c, 1:r); % Créer une grille d'indices
centre_x = c / 2; % Centre de l'image
centre_y = r / 2; % Centre de l'image
rayon = 50; % Ajuster le rayon pour tester l'effet
% Calculer le masque circulaire
mask = ( (rows - centre_y).^2 + (cols - centre_x).^2 ) <= rayon^2;
% Appliquer le masque au spectre
TF_mask = TF .* mask;
%TF = fftshift(ifft2(Img_bruitee));
%module_TF = abs(TF);
% Affichage
figure;
subplot(2, 2, 1);
imshow(Img_bruitee, [0 1]); % Ajout de la plage pour imshow
title(['Image du disque avec bruit', newline, 'blanc gaussien, \sigma^2 = ', num2str(sigma)]);
subplot(2, 2, 2);
imshow(log(1 + module_TF), []); % Log pour améliorer la visualisation
title(['Module de la TF inverse avec', newline, 'bruit blanc gaussien']);
colorbar;
subplot(2, 2, 3.5);
imshow(log(1 + TF_mask), []); % Log pour améliorer la visualisation
title('mask avec la TF');
colorbar;

```

- L'image « barbara.pgm »

```

% Lecture de l'image
img = imread('barbara.pgm');
img = double(img);
% Calcul de la Transformée de Fourier
TF_img_shifted = fftshift(fft2(img)); % Centrage de la TF
% Crédit et application du masque circulaire
[m, n] = size(img);
[x, y] = meshgrid(-n/2:n/2-1, -m/2:m/2-1);
%rayon = 100; % À modifier pour observer l'effet du masque

```

```

%rayon = 50;
%rayon = 20;
rayon = 10;
masque = (x.^2 + y.^2 <= rayon^2);
% Filtrage de la TF
TF_filtre = TF_img_shifted .* masque;
% Transformée de Fourier inverse
TF_inverse = ifft2(ifftshift(TF_filtre));
img_filtre = real(TF_inverse); % Extraction de la partie réelle
%img_filtre = abs(TF_inverse);
% Affichage des résultats
figure;
subplot(1, 3, 1), imshow(img, []), title('Image originale');
subplot(1, 3, 2), imshow(log(1 + abs(TF_img_shifted)), []), title('Spectre de la TF originale');
subplot(1, 3, 3), imshow(img_filtre, []), title(['Image après filtrage', newline, ' avec rayon = 10']);

```

Filtrage dans le domaine fréquentiel

L'image du disque

```

function Img = CreateCircle(N, r)
    % Crée une image binaire NxN contenant un disque de rayon r au centre
    [x, y] = meshgrid(-N/2:N/2-1, -N/2:N/2-1);
    Img = sqrt(x.^2 + y.^2) <= r;
end
N = 256; % Taille de l'image (NxN)
R = 50;
Img = CreateCircle(N, R);
% Calcul de la TF et centrage
TF_disque = fftshift(fft2(Img));
% Création du masque circulaire pour le filtrage
[m, n] = size(Img);
[x, y] = meshgrid(-n/2:n/2-1, -m/2:m/2-1);
%rayon = 100; % Ajustez pour observer l'effet
%rayon = 50;
%rayon = 30;
rayon = 1;
masque = (x.^2 + y.^2 >= rayon^2); % Masque pour mettre à zéro la partie centrale
% Application du masque et filtrage
TF_disque_filtre = TF_disque .* masque;
% Transformée de Fourier inverse
TF_disque_inverse = ifft2(ifftshift(TF_disque_filtre));
img_disque_filtre = real(TF_disque_inverse);
% Affichage des résultats
figure;
subplot(2, 2, 1), imshow(Img, []), title('Image du disque originale');
subplot(2, 2, 2), imshow(log(1 + abs(TF_disque)), []), title('Spectre TF disque');
subplot(2, 2, 3), imshow(log(1 + TF_disque_filtre), []), title('TF avec le masque avec rayon 1');
subplot(2, 2, 4), imshow(img_disque_filtre, []), title('Disque après filtrage');

```

L'image facade.png

```

img_facade = imread('facade.png');
img_facade = double(img_facade);
TF_facade = fftshift(fft2(img_facade));
% Création du masque circulaire pour le filtrage
[m, n] = size(img_facade);
[x, y] = meshgrid(-n/2:n/2-1, -m/2:m/2-1);
%rayon = 100; % Ajustez pour observer l'effet
%rayon = 30;
%rayon = 10;
rayon = 1;
masque = (x.^2 + y.^2 >= rayon.^2); % Masque pour mettre à zéro la partie centrale
TF_facade_filtre = TF_facade .* masque;
TF_facade_inverse = ifft2(ifftshift(TF_facade_filtre));
img_facade_filtre = real(TF_facade_inverse);
% Affichage des résultats
figure;
subplot(2, 2, 1), imshow(img_facade, []), title('Image facade originale');
subplot(2, 2, 2), imshow(log(1 + abs(TF_facade)), []), title('Spectre TF facade');
subplot(2, 2, 3), imshow(log(1 + TF_facade_filtre), []), title('TF avec le masque avec rayon 1');
subplot(2, 2, 4), imshow(img_facade_filtre, []), title('Facade après filtrage');

```

Importance de la phase

```

% Lecture des images
img_barbara = imread('barbara.pgm');
img_lena = imread('lena.pgm');
img_barbara = double(img_barbara);
img_lena = double(img_lena);
% Calcul de la TF des images
TF_barbara = fft2(img_barbara);
TF_lena = fft2(img_lena);
% Calcul du module et de la phase
module_barbara = abs(TF_barbara);
phase_barbara = angle(TF_barbara);
module_lena = abs(TF_lena);
phase_lena = angle(TF_lena);
% Reconstruction des images avec échange de phases
TF_barbara_new = module_barbara .* exp(1i * phase_lena);
TF_lena_new = module_lena .* exp(1i * phase_barbara);
% Calcul de la TF inverse
img_barbara_new = real(ifft2(TF_barbara_new));
img_lena_new = real(ifft2(TF_lena_new));
% Affichage des résultats
figure;
subplot(2, 2, 1), imshow(img_barbara, []), title('Image originale Barbara');
subplot(2, 2, 2), imshow(img_barbara_new, []), title('Barbara avec phase de Lena');
subplot(2, 2, 3), imshow(img_lena, []), title('Image originale Lena');
subplot(2, 2, 4), imshow(img_lena_new, []), title('Lena avec phase de Barbara');

```

Correction de défauts

```

% Lecture des images
img_lena = imread('lena2.png');
img_diagonale = imread('diagonale.png');
img_lena = double(img_lena);
img_diagonale = double(img_diagonale);
% Calcul des Transformées de Fourier 2D
TF_lena = fft2(img_lena);
TF_diagonale = fft2(img_diagonale);
% Affichage des spectres (en échelle logarithmique pour une meilleure visualisation)
figure;
subplot(2, 2, 1), imshow(uint8(img_lena)), title('Image lena2.png');
subplot(2, 2, 3), imshow(uint8(img_diagonale)), title('Image diagonale.png');
subplot(2, 2, 2), imshow(log(1 + abs(fftshift(TF_lena)))), [], title('Spectre de lena2.png');
subplot(2, 2, 4), imshow(log(1 + abs(fftshift(TF_diagonale)))), [], title('Spectre de
diagonale.png');
% Paramètre de combinaison
k = 2; % Vous pouvez ajuster cette valeur selon vos besoins
%k = 0.2;
% Combinaison des spectres
Scomb = TF_lena + k * TF_diagonale;
% Affichage du spectre combiné

% Calcul de la TF inverse pour visualiser l'image résultante
img_combined = real(ifft2(Scomb));
% Affichage de l'image reconstruite
figure;
subplot(2, 2, 1),
imshow(log(1 + abs(fftshift(Scomb))), [], title(['Spectre combiné Scomb', newline, 'Avec
k=2']));
subplot(2, 2, 2),
imshow(img_combined, []), title(['Image reconstruite à partir', newline, 'du spectre
combiné']);

```