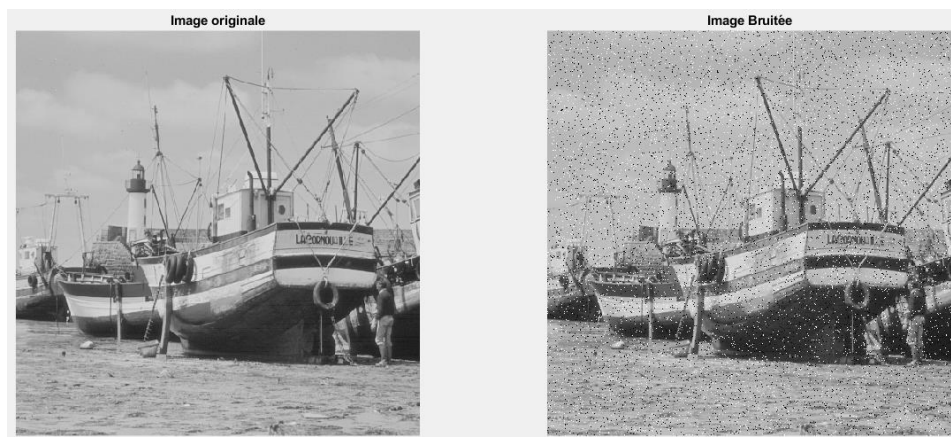


# Compte rendu du TP2

## 1- Filtres non-linéaires vs. Filtres linéaires

### 1.1 Création d'une image bruitée

```
img = imread('BOATS_LUMI.BMP');
figure(1);
subplot(1,2,1); imshow(img);title('Image originale');
noised = imnoise(img, 'salt & pepper');
subplot(1,2,2);imshow(noised);title('Image Bruitée');
```



**Remarque :** on peut créer un bruit dans l'image en changeant les valeurs des pixels de l'image source d'une manière aléatoire par d'autres valeurs afin d'obtenir une image bruitée.

### 1.2 Application d'un filtre linéaire

#### 1- Application du filtre 1 avec un masque de 3\*3 sur l'image bruitée :

```
filter1 = 1/9*ones(3);
fill = imfilter(noised, filter1);
figure; imshow(fill);title('Filtre 1');
```

**Remarque :** la qualité de l'image diminue essentiellement au niveau des contours, elle devient floue et les bruits de l'image ne sont pas trop claires qu'avant l'application du filtre.



#### 2- Masque 5\*5 :

```
filter2 = 1/25*ones(5);
fil2 = imfilter(noised, filter2, 'conv');
figure; imshow(fil2);title('Filtre 2: 5*5');
```



- **Masque 7\*7:**

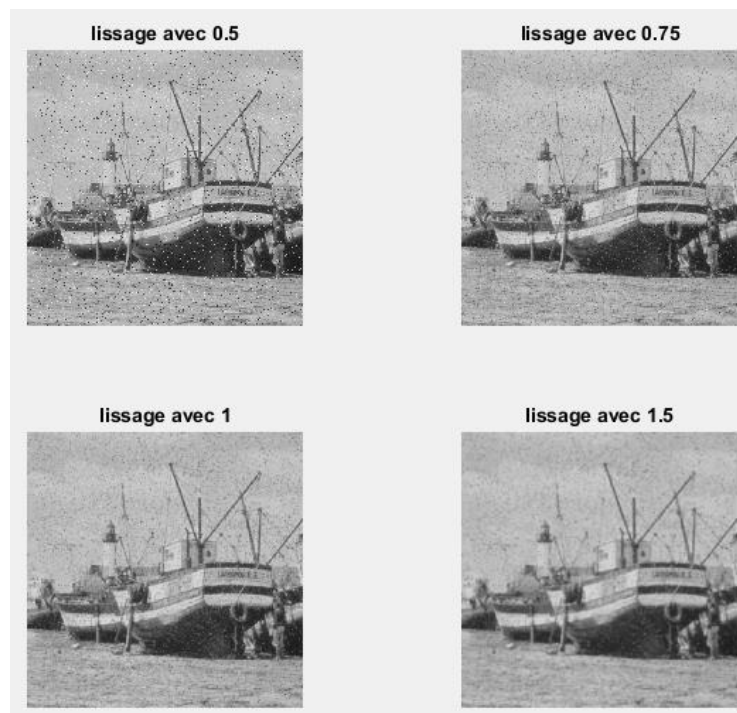
```
filter3 = 1/49*ones(7);
fil3 = imfilter(noised, filter3, 'conv');
figure; imshow(fil3);title('Masque 7*7');
```

**Conclusion :** Lorsque on augmente la taille du masque, on obtient une image de plus en plus de faible qualité et on affect les contours de l'image. Par contre, on remarque qu'il y a une atténuation du bruit.



### 3- Lissage Gaussien avec différentes valeurs de sigma :

```
l = imgaussfilt(noised,0.5);
subplot(2,2,1),imshow(l);title('lissage avec 0.5')
l = imgaussfilt(noised,0.75);
subplot(2,2,2),imshow(l);title('lissage avec 0.75')
l = imgaussfilt(noised,1);
subplot(2,2,3),imshow(l);title('lissage avec 1')
l = imgaussfilt(noised,1.5);
subplot(2,2,4),imshow(l);title('lissage avec 1.5')
```



**Conclusion :** Lorsque la valeur de sigma augmente, on remarque qu'il existe une diminution au niveau de la visibilité des bruits dans l'image, et par suite la qualité de l'image augmente relativement.

### 1.3. Application d'un filtre non-linéaire

La fonction **medfilt2** performe le filtrage médian d'une image dans deux dimensions.

### 1- Application d'un filtre de 3\*3

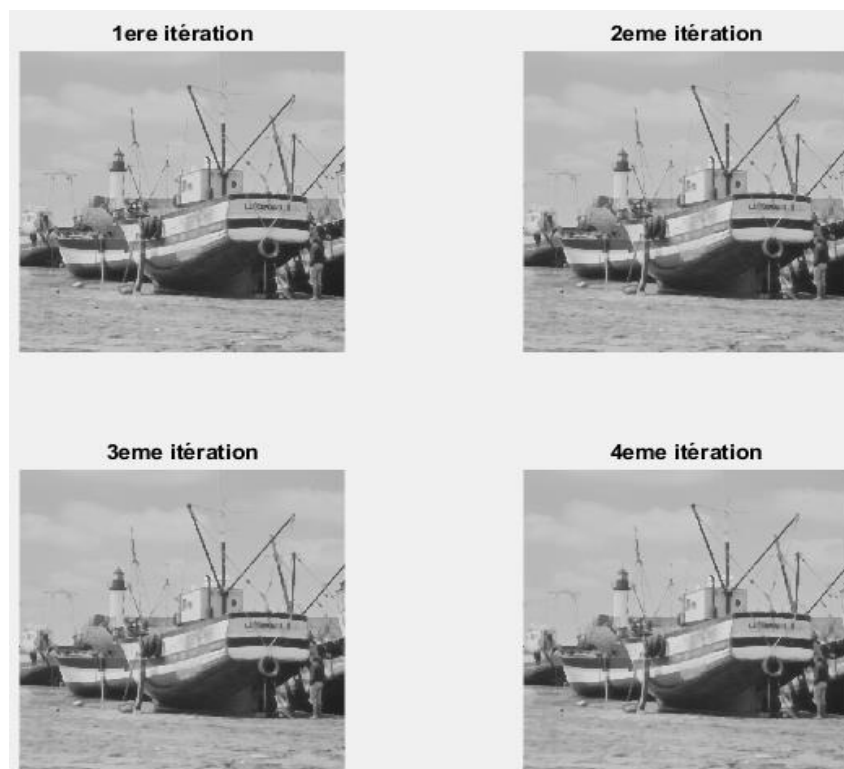
```
filt1 = medfilt2(noised, [3,3]);
figure; imshow(filt1);
```

**Remarque :** On constate que le filtrage médian n'affecte pas les contours de l'image par contre il améliore la qualité et le niveau de visibilité de l'image source contrairement au filtrage moyenner qui affecte les contours de l'image et la visibilité du contenu de l'image.



### 2- Application du filtrage par médian en 4 itérations :

```
figure;
filt = medfilt2(noised, [3,3]);
filt1 = medfilt2(filt, [3,3]);
subplot(2,2,1); imshow(filt1); title('1ere itération')
filt2 = medfilt2(filt1, [3,3]);
subplot(2,2,2); imshow(filt2); title('2eme itération')
filt3 = medfilt2(filt2, [3,3]);
subplot(2,2,3); imshow(filt3); title('3eme itération')
filt4 = medfilt2(filt3, [3,3]);
subplot(2,2,4); imshow(filt4); title('4eme itération')
```



**Conclusion :** Au cours des itérations, le filtrage médian améliore la qualité de l'image.

### 3- Application d'un filtre de 5\*5

```
filt1 = medfilt2(noised, [5,5]);
figure; imshow(filt1);
```



### Application du filtrage par médian en 4 itérations :

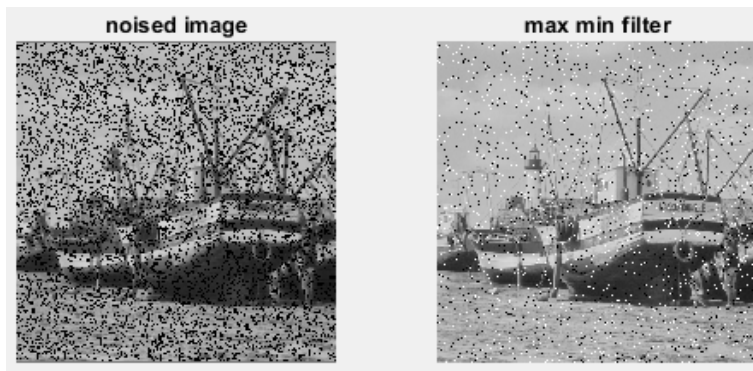
```
figure;
filt = medfilt2(noised, [5,5]);
filt1 = medfilt2(filt, [5,5]);
subplot(2,2,1); imshow(filt1); title('1ere itération')
filt2 = medfilt2(filt1, [5,5]);
subplot(2,2,2); imshow(filt2); title('2eme itération')
filt3 = medfilt2(filt2, [5,5]);
subplot(2,2,3); imshow(filt3); title('3eme itération')
filt4 = medfilt2(filt3, [5,5]);
subplot(2,2,4); imshow(filt4); title('4eme itération')
```



#### 4- Filtrage par le maximum\_minimum :

```
function [img] = filtrage_max_min (image)
    image = double(image);
    [L,C] = size(image);
    img = imnoise(image,"salt & pepper");
    for i=2:L-1
        for j=2:C-1
            filter=img(i-1:i+1, j-1:j+1);
            middle = img(i,j);
            temp = filter(:)';
            temp(5) = [];
            Min = min(temp);
            Max = max(temp);
            if img(i,j)< Min
                img(i,j) = Min;
            elseif img(i,j) > Max
                img(i,j) = Max;
            end
        end
    end
end

img = imread('BOATS_LUMI.BMP');
noised = imnoise(img,'salt & pepper');
filt1 = filtrage_max_min(noised);
figure; imshow(filt1);
```



## 2 Filtrage linéaire dans le domaine fréquentiel et dans le domaine spatial

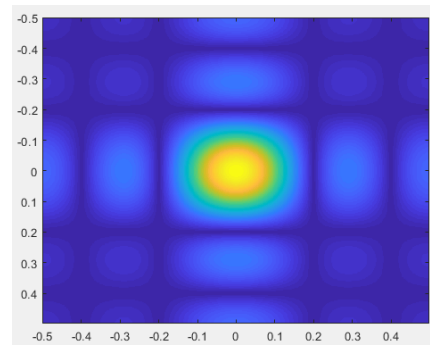
### 1- Filtrage passe-bas :

Un filtre passe-bas est un filtre qui laisse passer les basses fréquences et qui atténue les hautes fréquences, c'est-à-dire les fréquences supérieures à la fréquence de coupure.

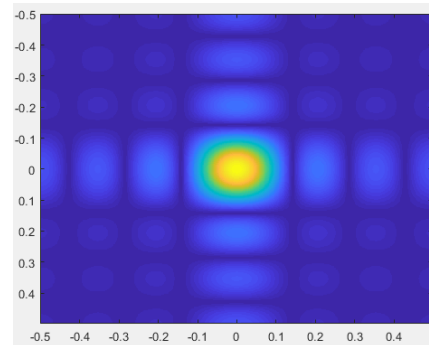
```
support = 5;
filtre = ones(support)/(support*support)
taille = 512;
spectre_filtre = fftshift(fft2(filtre,taille,taille))/(support*support);
vt = (-taille/2:taille/taille:(taille/2-taille/taille))/taille;
figure(1);
imagesc(vt,vt,abs(spectre_filtre));
colormap('default');
```



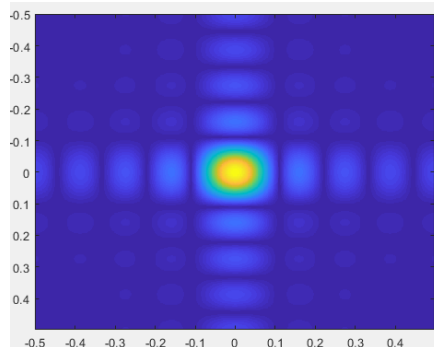
- Support = 5



- Support = 7

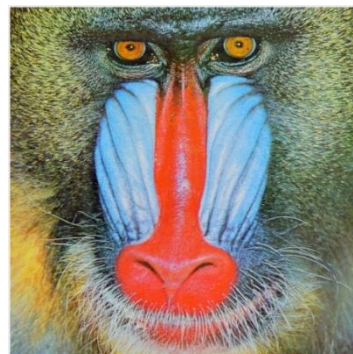


- Support = 9



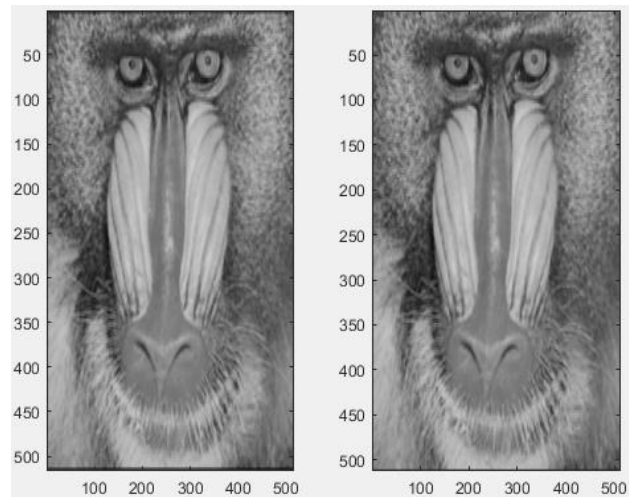
## 2- Filtrage spatial

L'image originale :



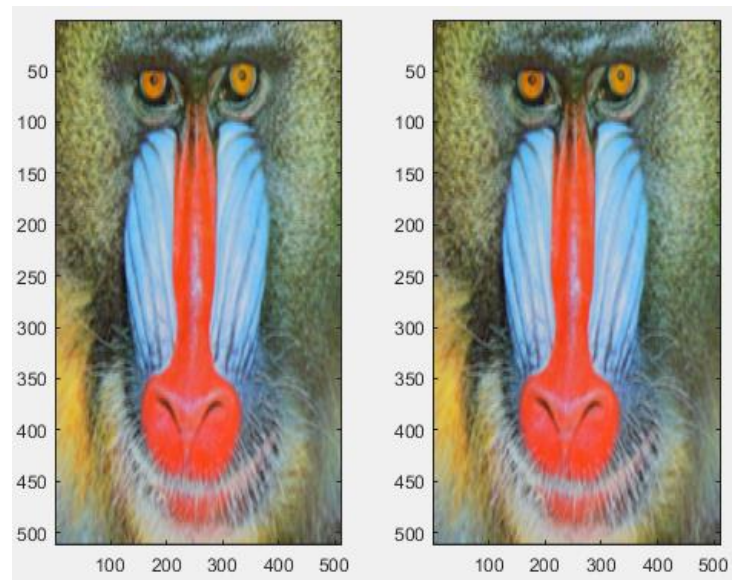
```
imf3 = imfilter(im1,filtre);
imf4 =
imfilter(im1,filtre,'replicate');

figure(4)
subplot(1,2,1)
image(imf3)
subplot(1,2,2)
image(imf4)
colormap(map);
```

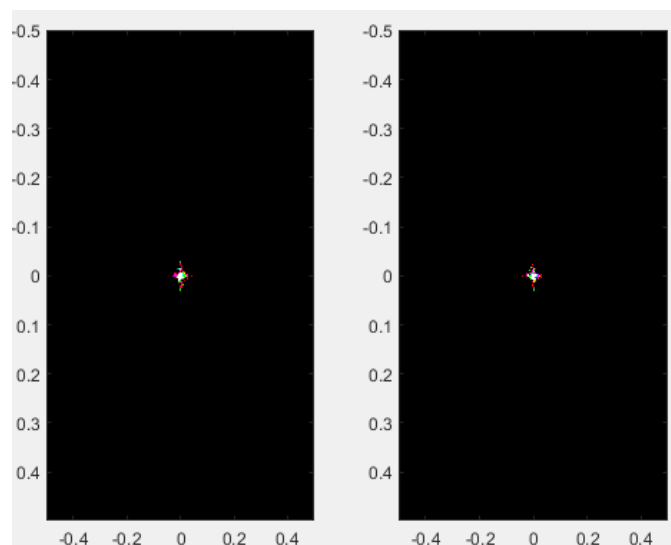


**Conclusion :** On remarque qu'il existe des améliorations au niveau de la qualité des pixels situées dans les bords de l'image.

3-



**Résultat fréquentiel :**



#### 4- Comparaison du filtrage spatial et fréquentiel :

- Pour support = 5 :  

```
temps_filtrage_spatial =
0.0127
```

```
temps_filtrage_freq =
0.0633
```

- Support = 32 :  

```
temps_filtrage_spatial =
0.0914
```

```
temps_filtrage_freq =
0.0635
```

- Support = 49 :  

```
temps_filtrage_spatial =
0.2160
```

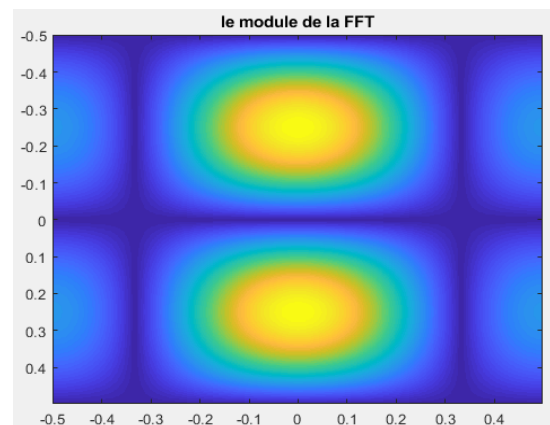
```
temps_filtrage_freq =
0.0766
```

**Conclusion :** En général, le filtrage fréquentiel est plus rapide que le filtrage spatial.

#### 5- Filtrage Passe-haut :

- En utilisant le filtre suivant ci-dessous on obtiendra les résultats suivants :  
**Filtre 1 :**  $[-1 \ -1 \ -1; \ 0 \ 0 \ 0; \ 1 \ 1 \ 1]$ ;

**L'image originale :**





## Filtrage spatial :

- En utilisant le filtre suivant ci-dessous on obtiendra les résultats suivants :

**Filtre 2 :**  $\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$  ;

