

Contrairement au filtrage par convolution (filtrage linéaire), le filtrage non-linéaire fait intervenir les pixels voisins suivant une loi non-linéaire. Le filtre médian (cas particulier du filtrage d'ordre), utilisé dans cet exercice, est un exemple classique de ces filtres. À l'instar du filtrage par convolution, les filtres non-linéaires opèrent sur un voisinage donné.

## 1. Création d'une image bruitée

Chargez l'image *BOATS\_LUMI.BMP*. Mettez à jour la liste des chemins dans le path browser. Le but de l'exercice est de comparer les effets linéaires et non-linéaires de deux filtrages sur une image bruitée.

La fonction *imnoise* de Matlab permet de créer différents types de bruit. À l'aide de cette fonction, créez l'image bruitée de *BOATS\_LUMI* avec un bruit de type « poivre-et-sel » (« *salt-and-pepper* »). Affichez l'image bruitée et expliquez comment peut-on créer ce bruit ?

## 2. Application d'un filtre linéaire

On souhaite réduire le bruit dans l'image. Dans un premier temps, on considère un filtre moyenneur ( $3 \times 3$ ) pour réduire le bruit dans l'image. Son noyau de convolution est :

$$\frac{1}{9} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Appliquez ce filtre (utilisez la fonction *imfilter*), et observez l'image bruitée. Interprétez le résultat obtenu.

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

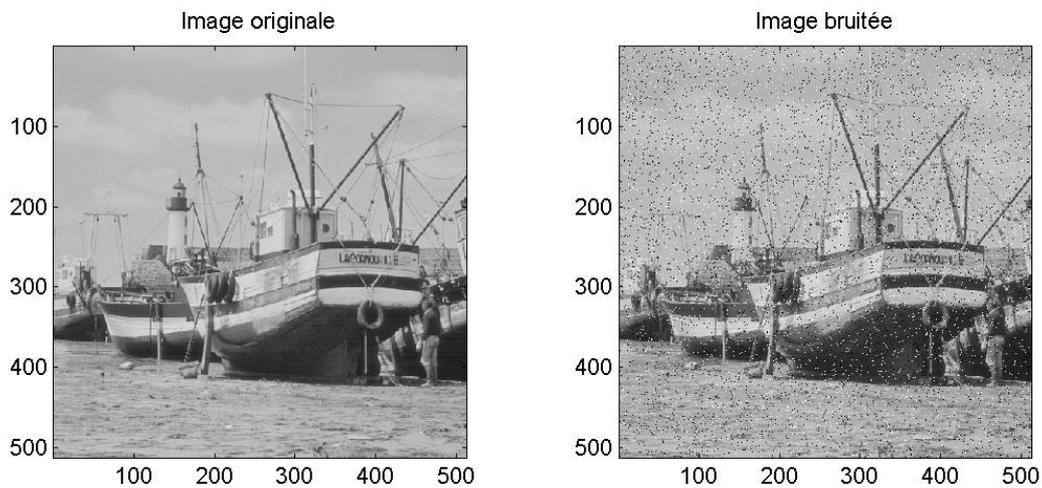
On souhaite à présent réduire le bruit avec un filtre médian ( $3 \times 3$ ). Sous Matlab, le filtrage médian peut être réalisé à l'aide de la fonction *medfilt2*. Expliquez ce qu'effectue cette fonction. Appliquez ce filtre à l'image bruitée et observez les résultats. Expliquez les différences avec le filtre moyenneur précédemment utilisé.

## Correction de l'exercice : Filtrage linéaire vs. Filtrage non-linéaire

1 - Voici les commandes à entrer pour créer et observer l'image bruitée de *BOATS\_LUMI* :

```
I=imread('BOATS_LUMI.BMP') ; % image bateau en niveaux de gris
IB = imnoise(I,'salt & pepper'); % image bruitée
figure(1)
subplot(1,2,1)
subimage(I)
title('Image originale')
subplot(1,2,2)
subimage(IB)
title('Image bruitée')
```

Voici les résultats obtenus :

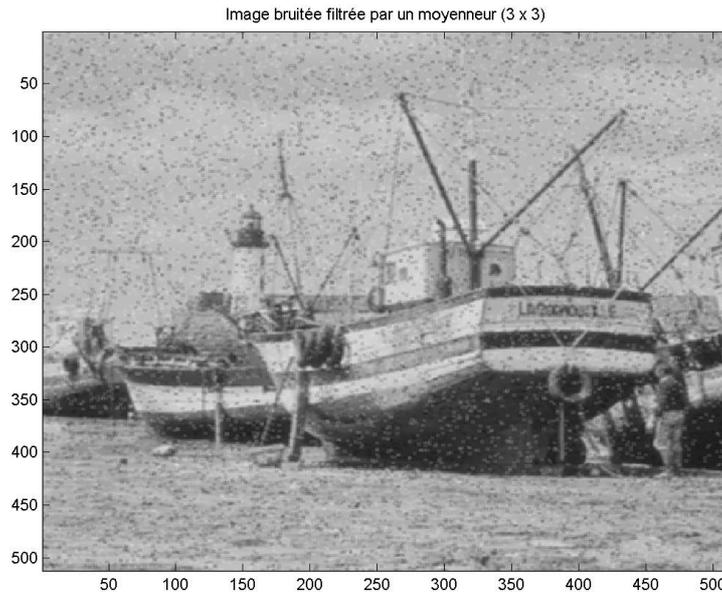


Le bruit « poivre-et-sel » utilisé ici consiste à mettre, aléatoirement, plusieurs pixels aux valeurs 255 ou 0 (valeurs extrêmes de l'intervalle des niveaux de gris). Ce type de bruit impulsionnel peut apparaître par numérisation d'une image ou au cours d'une transmission.

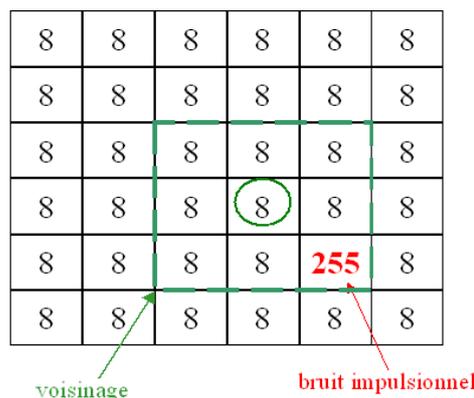
2 - Voici les commandes pour appliquer un filtre moyeneur à l'image bruitée :

```
% Filtre moyeneur
N = ones(3)/9 ; % noyau de convolution du filtre
If1 = imfilter(IB,N) ;
figure(2)
image(If1)
title('Image bruitée filtrée par un moyeneur (3 x 3)')
v=0:1/255:1; colormap([v' v' v']); % LUT pour afficher en niveaux de gris
```

Voici l'image obtenue :



Le bruit « salt-and-pepper » est faiblement réduit. On distingue encore nettement les grains dans l'image. Le filtre moyeneur réalise en effet la moyenne pour chaque pixel sur un voisinage (3 x 3) et le bruit des impulsions (à 0 ou à 255) participe à cette moyenne :

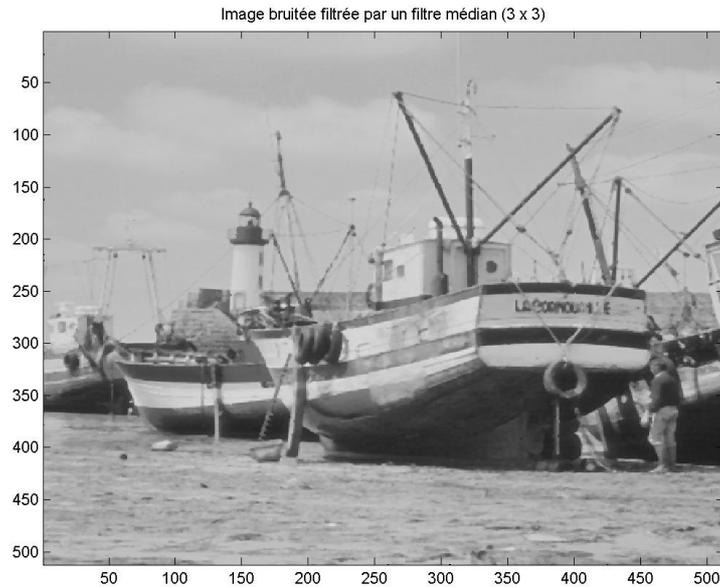


Le pixel cerclé a le niveau de gris 8 ainsi que toute l'image, à l'exception d'un pixel bruité à 255. En sortie du filtrage moyeneur, la valeur de sortie du pixel cerclé (et de tout pixel ayant la valeur 255 dans son voisinage) sera :  $(8 \times 8 + 255) / 9 \approx 35$ . La valeur du pixel après filtrage n'est donc pas représentative du voisinage de ce pixel, le bruit impulsif est trop faiblement réduit. Ce filtre linéaire n'est donc pas adapté.

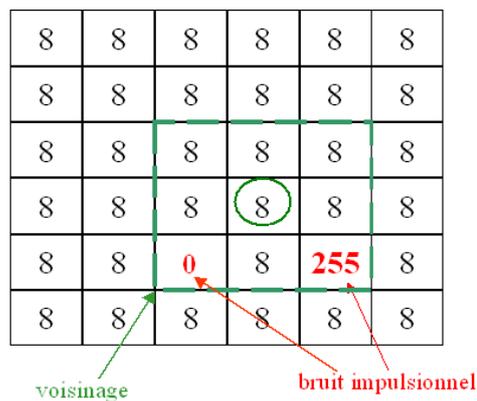
3 - Voici les commandes pour appliquer un filtre médian à l'image bruitée :

```
% Filtre médian
If2 = medfilt2(IB,[3 3]) ; % filtrage médian avec un voisinage 3 x 3
figure(3)
image(If2)
title('Image bruitée filtrée par un filtre médian (3 x 3)')
v=0:1/255:1; colormap([v' v' v']); % LUT pour afficher en niveaux de gris
```

Voici l'image obtenue :



Le bruit impulsionnel est visiblement réduit. Le filtrage médian d'un pixel P, sur un voisinage V(P) de taille (M x N), ordonne les valeurs des pixels de V(P) par ordre croissant, et attribue en sortie la valeur médiane sur ce voisinage au pixel P (opération non-linéaire) :



Reprenons l'exemple précédent : les valeurs des pixels sont ordonnées par ordre croissant : 0, 8, 8, 8, 8, 8, 8, 8, 255. La valeur médiane est donc 8. Pour cette opération non-linéaire, les impulsions 0 et 225 n'ont pas d'influence sur la valeur médiane. Le filtrage médian est donc adapté à la réduction du bruit impulsionnel.