

## Traitement d'Images TP 2 : Le filtrage

### I- Filtrage spatial

Sur l'image 'cameraman.tif' ajoutez un bruit gaussien en utilisant la fonction `imnoise()` et sauvegardez l'image résultat.

#### I. Filtrage spatial linéaire

- 1) Programmer une fonction `filtre_Moy(image)` permettant de réaliser lissage par moyennage dans un voisinage 3x3, appliquez la sur l'image bruitée. généraliser pour un voisinage de  $n \times m$ .
- 2) Programmer une fonction `filtrage(image,H)` permettant de réaliser un filtrage avec le filtre H.
  - a. Tester sur l'image bruitée un filtre passe bas  $H = 1/16 * [1,2,1; 2,4,2; 1,2,1]$ .
  - b. Tester sur l'image bruitée un filtre passe haut  $H = [-1,-1,-1;-1,8,-1;-1,-1,-1]$ .

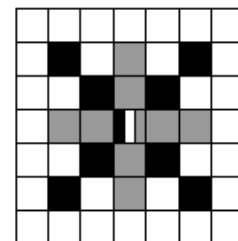
NB : fonctions utiles : `floor`, `mean`, `median` et `std`

#### II. Filtrage spatial non linéaire

- 1) Programmer une fonction `lissage_MinMax(image)` réalisant le lissage selon la méthode Min Max.
- 2) Programmer une fonction `lissage_Median(image,n,m)` permettant d'appliquer un filtrage médian sur une fenêtre de  $n \times m$
- 3) Programmer une fonction permettant d'appliquer un filtrage médian hybride sur une fenêtre de 5x5 (principe ci-dessous), appliquer la sur l'image bruitée et déduire le rôle de ce type de filtre

C'est un lissage où la valeur du pixel est la médiane de ces trois médianes à savoir : croix, diagonale + pixel central

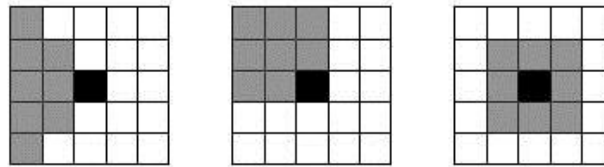
- 4) Le filtre de Nagao



#### Principe

Le principe de ce filtrage est de remplacer chaque pixel de l'image par la valeur moyenne des pixels contenus dans une fenêtre particulière. Il s'agit de choisir la fenêtre la mieux adaptée parmi un certain nombre de fenêtres prédéfinies.

9 fenêtres sont ici définies, chacune contenant 9 pixels, dont le pixel à remplacer. Les fenêtres de filtrage sont déduites de la façon suivante :



- 4 fenêtres sont issues de celle de gauche (par rotations d'angles 0, 90, 180 et 270 degrés) ;
- 4 sont issues de celle du milieu de la même façon ;
- 1 est celle de droite.

La fenêtre qui sera sélectionnée est celle dont la variance est la plus faible. La valeur moyenne de cette fenêtre remplacera celle du pixel central, le fait de choisir la fenêtre de plus petite variance implique que cette moyenne est faite sur une zone relativement homogène. Ainsi, les contours sont conservés : un pixel proche d'un contour sera remplacé par une valeur homogène à la zone à laquelle il appartient, et non pas par une valeur moyenne entre la zone à laquelle il appartient et celle de l'autre côté du contour.

Implémenter cet algorithme de lissage, tester le sur une image et déduire les résultats de ce filtrage ?

## II- Filtrage fréquentiel

Sur l'image 'cameraman.tif' ajouter un bruit gaussien en utilisant la fonction `imnoise()` et sauvegarder l'image résultat.

- 3) Réaliser les filtres passe-bas : idéal, butterworth, et gaussien. Appliquer les sur l'image bruitée.
- 4) Réaliser les filtres passe-haut : idéal, butterworth, et gaussien. Appliquer les sur l'image bruitée.

NB : pour le calcul de TF de l'image utilisez les fonctions `fft2()` et `ifft2()`, et `fftshift()` pour faire correspondre le centre de l'image à la fréquence (0,0).

Donner la fréquence de coupure en argument à votre programme. Cette fréquence vous servira à définir un cercle dans la transformée de Fourier à partir duquel vous mettrez les valeurs (à l'intérieur ou à l'extérieur du cercle) à zéro selon le type de filtre (passe-bas ou passe-haut). Voir les notes de cours pour exemples de ces filtres.