



## AHP.4

([mai.nguyen-verger@cyu.fr](mailto:mai.nguyen-verger@cyu.fr))

### Réduction/Suppression de bruit (Débruitage)

#### Rappel sur le filtrage :

##### Filtre médian :

Ce filtre affecte à un pixel la valeur médiane des intensités dans son voisinage. Autrement dit, le filtre médian retient une valeur d'intensité telle que dans le voisinage, il y ait d'autant d'intensités supérieures que d'intensités inférieures à cette valeur.

Exemple: un voisinage de 3x3 pixels dans une image :

$$I_{\text{originale}} = \begin{pmatrix} 7 & 5 & 3 \\ 4 & 2^* & 7 \\ 6 & 5 & 1 \end{pmatrix}$$

Le filtre médian range par ordre croissant les intensités du voisinage:

1 2 3 4 **5** 5 6 7 7

Il affecte au pixel central (qui valait initialement 2\*) l'intensité qui se trouve au milieu du

rangement ci-dessus, c'est à dire **5**, donc le résultat est:  $I_{\text{mediane}} = \begin{pmatrix} 7 & 5 & 3 \\ 4 & 5^* & 7 \\ 6 & 5 & 1 \end{pmatrix}$

##### Filtre moyenneur :

Ce filtre remplace le niveau de gris du pixel central par la moyenne des niveaux de gris des pixels autour du pixel central. Pour cela on peut utiliser un masque  $h$  rectangulaire du type (8-connexe) et convoluer l'image originale avec le masque :

$$h = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad \text{le résultat est : } I_{\text{moyenne}} = \begin{pmatrix} 7 & 5 & 3 \\ 4 & 4^* & 7 \\ 6 & 5 & 1 \end{pmatrix}$$

La taille du masque est un paramètre variable. Plus la taille du masque est grande plus l'effet du filtrage est fort. Exemple : la taille du masque rectangulaire : 3x3, 5x5, 9x9.

Afin de conserver la dynamique des niveaux de gris, il faut que la somme des éléments/coefficients du masque soit égale à 1.

Pour que le filtrage soit isotopique (même effet dans toutes les directions), le voisinage considéré devra avoir une forme circulaire (filtre circulaire, filtre conique).

L'inconvénient du filtre moyenneur est l'introduction de flou, les contours sont donc dégradés.

## Programmation : Suppression de bruit (débruitage) par filtres médian et moyenneurs

**Exo.1 :** Débruitage par filtre médian de l'image suivante déjà traitée dans la séance d'Histogramme :

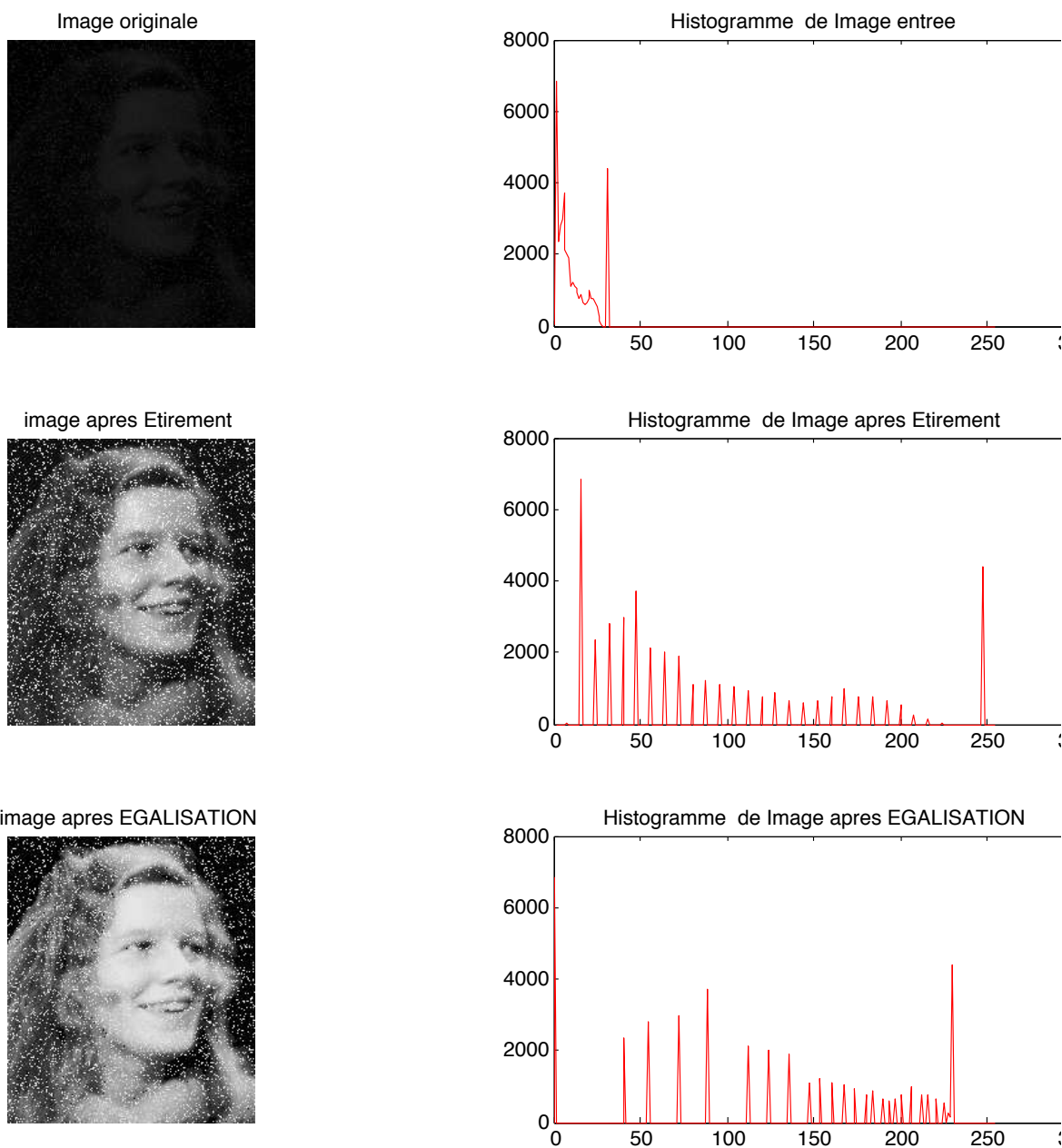


Fig.1 Images originale et après l'étirement et l'égalisation de l'histogramme



Fig. 2 Image débruitée

**Exo. 2.**

- 1) Lire le fichier d'image (l'image de test pour cette séance est 'lena.gif', ou 'circuit.tif' ou 'cameraman.tif').
- 2) Ajouter à l'image originale le bruit de type gaussien de moyenne nulle et de variance 0.04 (4% de la valeur max), et de type 'salt & pepper'. (fonction *imnoise(I, 'gaussian/salt & pepper', mean, var)* ). Afficher les deux images ainsi bruitées.
- 3) Appliquer sur les deux images bruitées un filtre moyennneur rectangulaire de taille 3x3, afficher et commenter les résultats obtenus.
- 4) Appliquer sur les deux images bruitées le «filtre médian », afficher et commenter les résultats obtenus.
- 5) Comparer 3) et 4).

**Exemple :**

Image+bruit gaussien



Image+bruit gaussien+ Filtre Moyennneur



Image+bruit impulsionnel



Image+bruit impulsionnel+ Filtre Median



- 6) Commenter sur l'atténuation du bruit et la qualité (ou la détérioration des détails) en fonction de la largeur du filtre moyennneur, puis le choix de la largeur adaptée ou optimale.

Exemple :

**Image originale**



**Rayon = 5**



**Image originale**



**Rayon = 10**



## Rappel sur Matlab:

1) Les fonctions utiles: ***abs()*** pour calculer la valeur absolue, ***size()***, ***sum()***, ***sum(sum())***, ***zeros()***,  
opération matricielle terme par terme: ***.\**** (point étoile pour la multiplication terme par terme), ***./*** (point barre oblique), ***imnoise()*** pour ajouter des bruits, etc..

**2) Attention à l'effet de bord : on peut ajouter les zéros autour de l'image avant d'effectuer les opérations.**

3) Les niveaux de gris d'images sont entre [0,255] ou entre [0,1] avec noir (0) et blanc (255 ou 1).

4) Les opérations dans les images sont réalisées souvent avec des données de type « double » (exemple : log). S'assurer que les opérations s'effectuent sur une image I de ce type (en utilisant ***double(I)*** par exemple).

5) Changement de type des valeurs d'une image :

***uint8()***, ***uint16()*** : entiers non signés 8, 16 bits

***double()*** : double précision

***logical()*** : binaire