

TP7 : Filtre Médian

Important :

- Toutes les ressources à utiliser se trouvent dans le répertoire "**Ressources**" situé sur la racine du disque C.
- Il est demandé au candidat :
 - ✓ De créer, dans le répertoire **Bac2023** situé sur la racine du disque C, un dossier de travail portant son numéro d'inscription (6 chiffres) et dans lequel il doit enregistrer, au fur et à mesure, tous les fichiers solutions de ce sujet.
 - ✓ De copier, dans son dossier de travail, le fichier "**Algo.rar**" situé dans "**C:\Ressources**", puis d'extraire son contenu, en utilisant le mot de passe "**123456**", dans ce même dossier de travail.
 - ✓ D'élaborer une solution modulaire au problème posé.
 - ✓ De vérifier à la fin de l'épreuve que tous les fichiers créés sont dans son dossier de travail.

Une image représentée en niveaux de gris, est formée par des pixels codés chacun sur huit (8) bits et ayant chacun une valeur comprise entre 0 (noir) et 255 (blanc).

Le filtre médian est une technique de filtrage utilisée dans les traitements d'images bitmap.



Image à filtrer



Image filtrée

L'algorithme du filtre médian consiste à remplacer chaque valeur du pixel de l'image à filtrer par la valeur médiane des pixels voisins. Il s'agit d'un voisinage carré autour du pixel considéré.

Un voisinage est donc assimilable à une matrice carrée de taille impaire, exemple : voisinage 3x3, 5x5, 7x7, etc. (voir exemple ci-après)

N.B. : Lorsqu'un pixel est proche du bord de l'image, certains pixels de son voisinage n'existent pas.

On se propose d'écrire un programme permettant :

- ✓ De remplir un fichier « **ImageInit.txt** » par des codes binaires correspondants aux pixels d'une image à filtrer,
- ✓ De remplir une matrice **MF**, par les valeurs, en décimal, des pixels de l'image filtrée par le principe du filtre médian (voisinage 3x3 ou 5x5 ou 7x7) Sachant que :
 1. Le fichier « **ImageInit.txt** » est constitué comme suit :
 - La première ligne, contient la définition de l'image sous la forme **HxL** où **H** est le nombre de pixels en hauteur et **L** le nombre de pixels en largeur (Exemple : "200x180")
 - Chacune des **HxL** lignes suivantes, contient une succession de huit (8) bits représentant le code binaire d'un pixel de l'image.

2. L'image filtrée sera obtenue en procédant comme suit :

- Remplir à partir du fichier « **ImageInit.txt** », une matrice **M** de **H** lignes et **L** colonnes par les conversions en décimal des blocs de huit (8) bits représentant chacun le code d'un pixel.
- Appliquer un filtre médian sur chaque élément **M** [**i**, **j**] de la matrice **M** afin de former une deuxième matrice **MF** et ce de la manière suivante :

- Placer les valeurs de la matrice **M** formant le voisinage choisi de **M** [**i**, **j**], dans un tableau **T** (y compris le pixel lui-même).

N.B. : Les éléments du voisinage qui sont en dehors de la matrice **M**, auront la valeur **zéro** dans le tableau **T**.

- Trier par ordre croissant les éléments du tableau **T**.
- Placer la valeur située au milieu du tableau **T** trié (valeur médiane), dans la case **MF** [**i**, **j**] de la matrice **MF**.

- Transférer la matrice **MF** dans un fichier binaire nommé « **ImageFiltre.dat** » tel que chaque enregistrement, autre que le premier contient les champs suivants :

- **numL** : le numéro de la ligne de la matrice contenant le pixel
- **numC** : le numéro de la colonne de la matrice contenant le pixel
- **val** : la valeur de la case correspondante de la matrice.

N.B. : le premier enregistrement contiendra les dimensions de l'image

Exemple :

M	0	1	2	3	4	5	...
0	7	10	12	7	9	9	...
1	12	8	7	11	9	10	...
2	9	9	10	8	7	12	...
3	10	9	9	6	10	9	...
4	12	12	12	10	7	7	...
...

Pour le pixel **M**[1, 0] :

- Placement des valeurs du voisinage 3x3 y compris le pixel à traiter

T	0	1	2	3	4	5	6	7	8
	0	7	10	0	12	8	0	9	9

- Tri du tableau **T** pour déterminer la valeur médiane

T	0	1	2	3	4	5	6	7	8
	0	0	0	7	8	9	9	10	12

- Placement de la valeur médiane dans **MF**[1,0]

MF	0	1	2	3	4	5	...
0
1	8
2	9
3
4
...

Pour le pixel **M**[2, 4] :

- Placement des valeurs du voisinage 3x3 y compris le pixel à traiter

T	0	1	2	3	4	5	6	7	8
	11	9	10	8	7	12	6	10	9

- Tri du tableau **T** pour déterminer la valeur médiane

T	0	1	2	3	4	5	6	7	8
	6	7	8	9	9	10	10	11	12

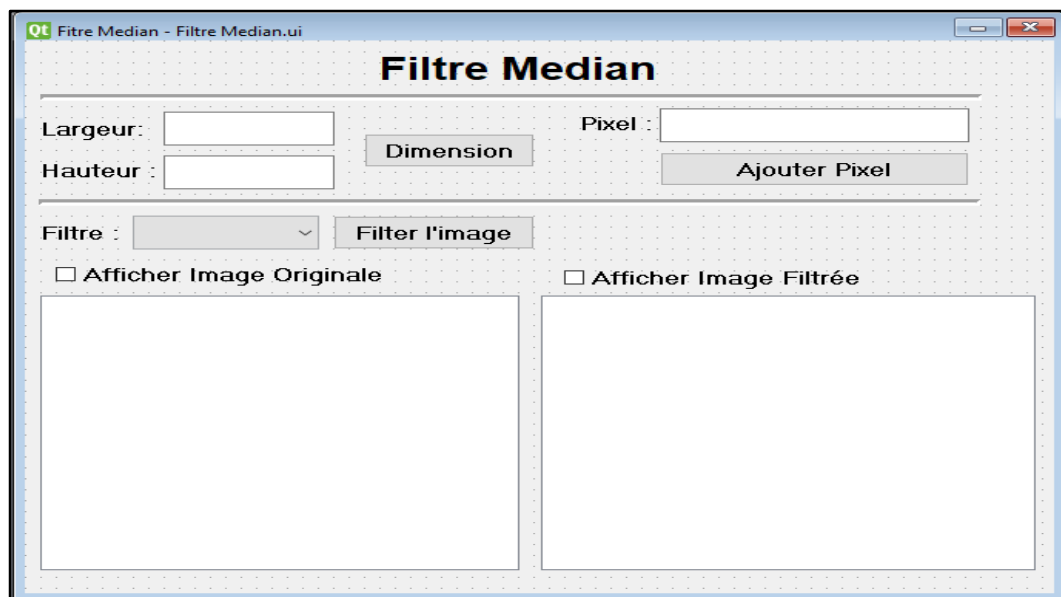
- Placement de la valeur médiane dans **MF**[2,4]

On se propose de concevoir une interface graphique contenant les objets suivants :

- Un label contenant le texte « **Filtre Median** »
- Quatre labels contenant les textes « **Largeur** », « **Hauteur** », « **Filtre** » et « **Pixel** »
- Trois zones de saisie permettant la saisie de « **Largeur** », « **Hauteur** » et « **Pixel** ».
- Une liste de sélection contenant les options «**3x3**», «**5x5**» et «**7x7**» et permettant le choix d'un filtre à appliquer.
- Deux tables widgets pour l'affichage du contenu du fichier " **ImageInit.txt** " et " **Imagefitre.dat**"
- Deux cases à cocher nommé « **Afficher image originale** » et « **Afficher image Filtrée** » permettant d'afficher les codes décimales des pixels des images originale et filtrée.

Travail demandé :

- 1) Compléter l'interface graphique « **Interface_Filtre.ui** » par les éléments présentés précédemment comme illustrée dans la figure suivante :



- 2) Compléter le programme qui existe dans le fichier « **Filtre_Median.py** » par :
 - Le module « **dimension** », qui s'exécute suite à un clic sur le bouton « **Dimension** », et permettant, si la largeur et la hauteur sont entre 5 et 10, de créer le fichier « **ImageInit.txt** » et d'enregistrer dans la première ligne les dimensions comme indiqué ci-dessus.
 - Le module « **Pixel** », qui s'exécute suite à un clic sur le bouton « **Ajouter Pixel** », et permettant, si les contraintes ci-dessus sont respectées, d'ajouter la valeur du pixel dans une ligne du fichier « **ImageInit.txt** ».
 - Le module « **Filtrer** », qui s'exécute suite à un clic sur le bouton « **Filtrer l'image** », et permettant d'appliquer le filtre choisi dans la liste de sélection et d'enregistrer le résultat dans le fichier « **ImageFiltre.dat** » comme expliqué ci-dessus.
 - Les modules « **aff_originale** » et « **aff_filtre** », qui s'exécute suite aux clics sur les cases à cocher et permettant d'afficher les codes décimaux des pixels des images originale et filtrée dans les tables widgets.