

# **H.E.P.L. : Catégorie Technique**

---

**Bachelier en Informatique et Systèmes : finalité Informatique Industrielle**

## **LABORATOIRE D'INFORMATIQUE INDUSTRIELLE**

---

### **Traitement d'image**

---

3ème année

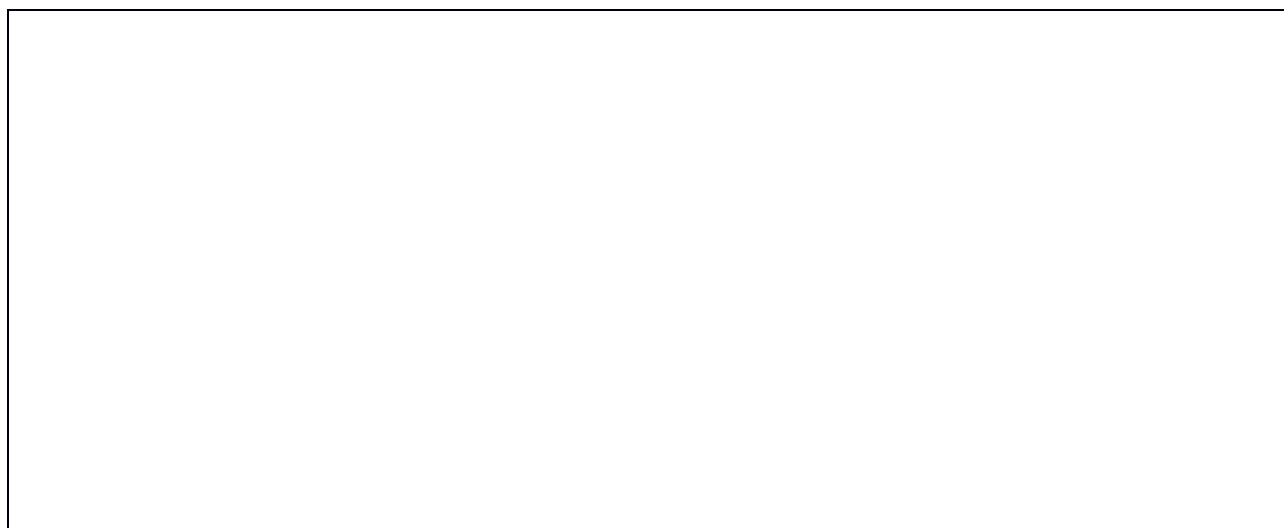
### **Etape 3**

---

2012 - 2013

Van Gysegem Thomas

Groupe : 2322



## Table des matières

H.E.P.L. : Catégorie Technique.....	1
Explication du fonctionnement : .....	3
MainWindows.....	3
Common.....	3
Picture.....	3
GrayScaleStepFour.....	3
Dialog*.....	3
DialogExpansion.....	3
DialogModifTaille.....	3
DialogModifPalette.....	3
DialogThreshold.....	4
DialogMultiThreshold.....	4
DialogHistogramme.....	4
Schéma des processus : .....	5
Modification de la taille.....	5
Modification de la palette.....	6
Récupérer la ROI.....	7
Seuillage.....	8
Seuillage multiple.....	9
Histogramme.....	10
Égalisation d'histogramme.....	11
Expansion.....	12
Extraction.....	13
Filtres.....	14
Aperçu des résultats : .....	14
Affinage des contours.....	15
Commentaires.....	16

## ***Explication du fonctionnement :***

### **MainWindows**

Cette classe instancie la fenêtre de base de l'application, elle définit le comportement des différents boutons, menus et autres boîtes de dialogues . Elle permet d'afficher les images et de lancer les traitements désiré

### **Common**

Fournit plusieurs fonction utilitaire utilisable dans différents contexte et dans d'autres applications.

### **Picture**

Cette classe permet la manipulation d'une image donnée. Elle implémente les traitements demandé dans le cahier des charges (ROI, Taille, Palette et d'autres à venir)

### **GrayScaleStepFour**

Ce fichier fournit un ensemble de fonction utilisée pour implémenter les traitements demandé à l'étape 4 du dossier.

### **Dialog\***

Les classes commençant par Dialog implémentent les boîtes de dialogue et la façon dont elles interagissent avec la fenêtre principale. Pour l'instant, elles sont au nombre de deux : DialogModifTaille et DialogModifPalette.

### **DialogExpansion**

Permet de définir un facteur d'élargissement ou de rétrécissement de l'image ainsi que de sélectionner l'opération voulue (agrandir ou rétrécir).

### **DialogModifTaille**

Se charge de fournir une interface à l'utilisateur permettant de modifier la taille de l'image

### **DialogModifPalette**

Se charge de fournir une interface à l'utilisateur permettant de modifier une couleur de la palette de l'image

### **DialogThreshold**

Permet à l'utilisateur de définir un seuil utilisé pour le seuillage de l'image. Un aperçu du résultat est affiché lorsque l'on modifie la valeur du seuil.

### **DialogMultiThreshold**

Identique à DialogThreshold si ce n'est que cette boîte de dialogue permet de sélectionner 3 seuil simultanément pour réaliser un seuillage multiple.

### **DialogHistogramme**

Affiche deux histogramme, celui de l'image de base et celui de l'image résultat. Un bouton permet d'égaliser l'histogramme de l'image de base.

### **DialogAutre**

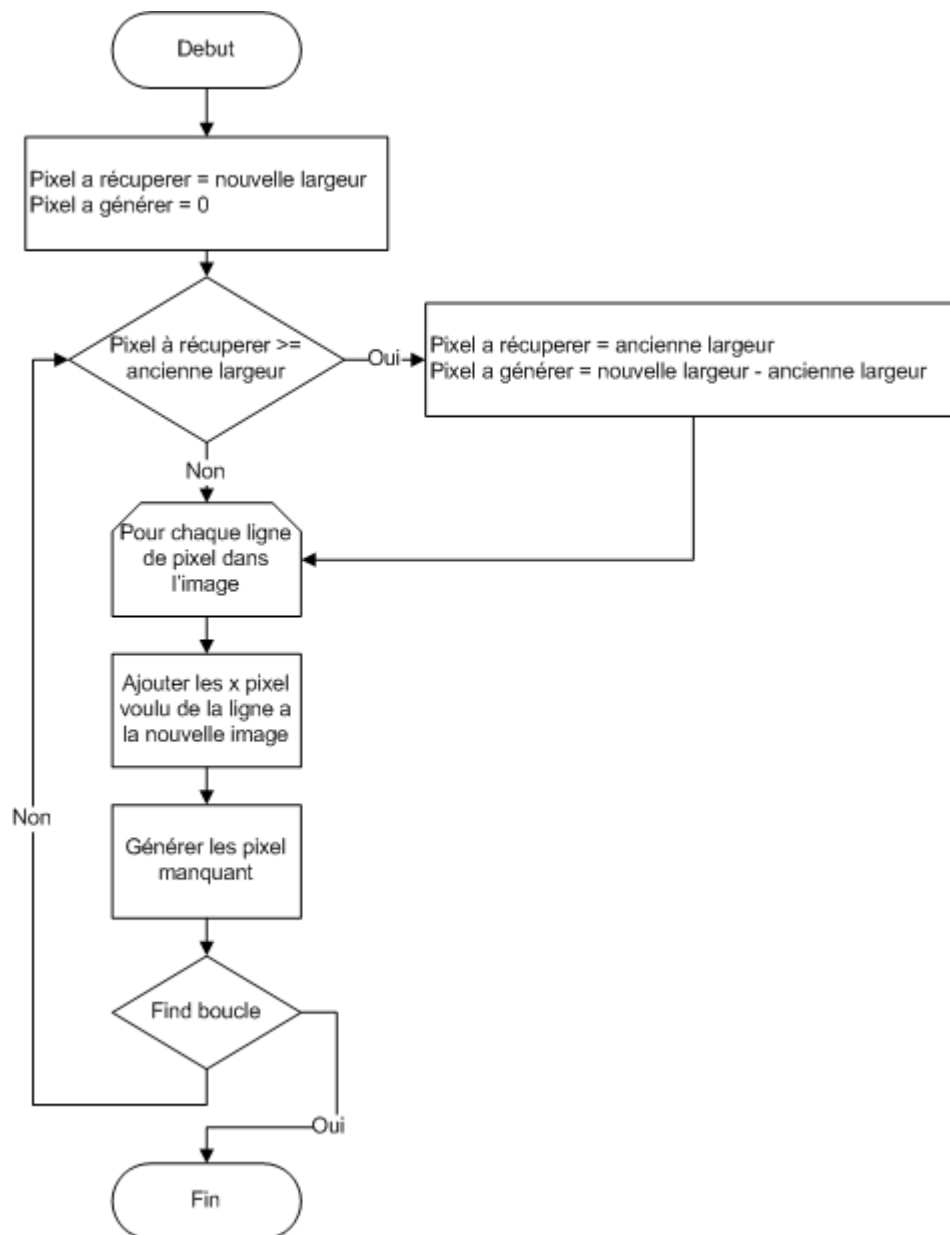
Permet d'appliquer les érosions/dilatations/fermeture et ouverture.

### **DialogFiltre**

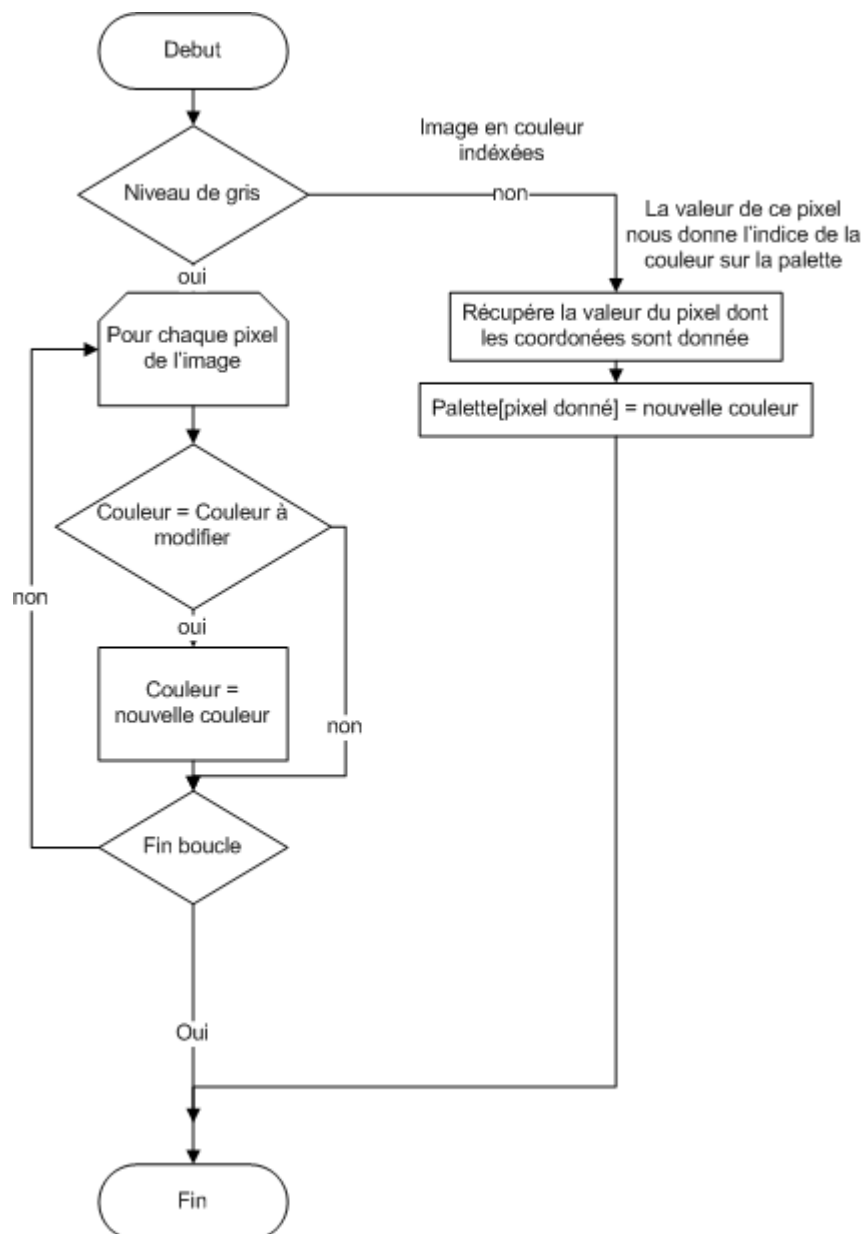
Permet d'appliquer les différents filtres demandé (Sobel, Médian, Moyen, Gaussien, Laplacien, Roberts, Prewitt et Kirch).

## Schéma des processus :

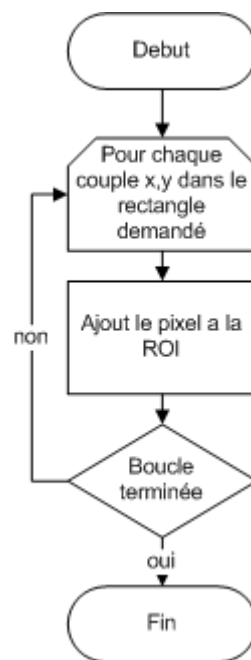
### Modification de la taille



## Modification de la palette

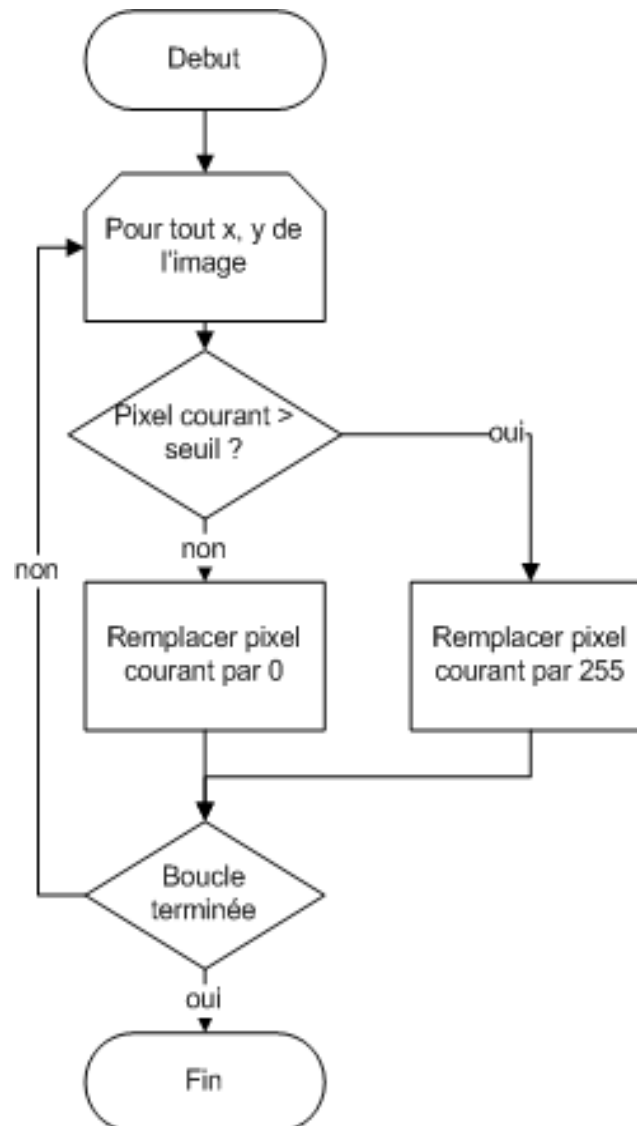


## Récupérer la ROI



## Seuillage

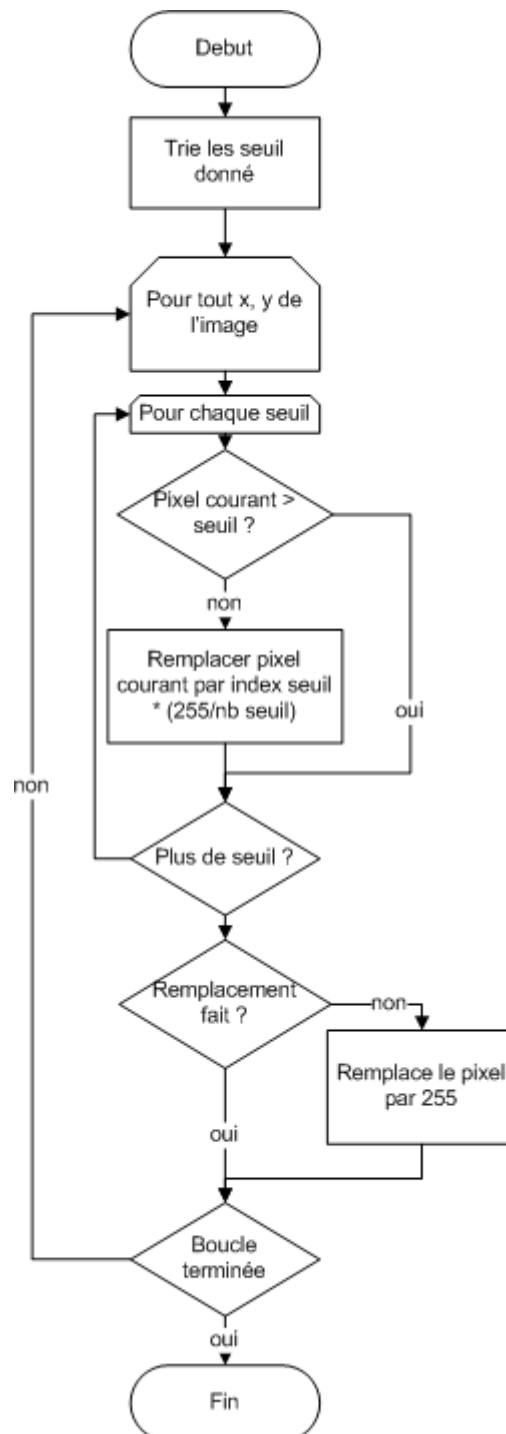
Le seuillage est une opération qui permet de remplacer tout les pixel dont la valeur en niveau de gris ne dépasse pas un seuil donné par 0 et de remplacer les autres par 255.





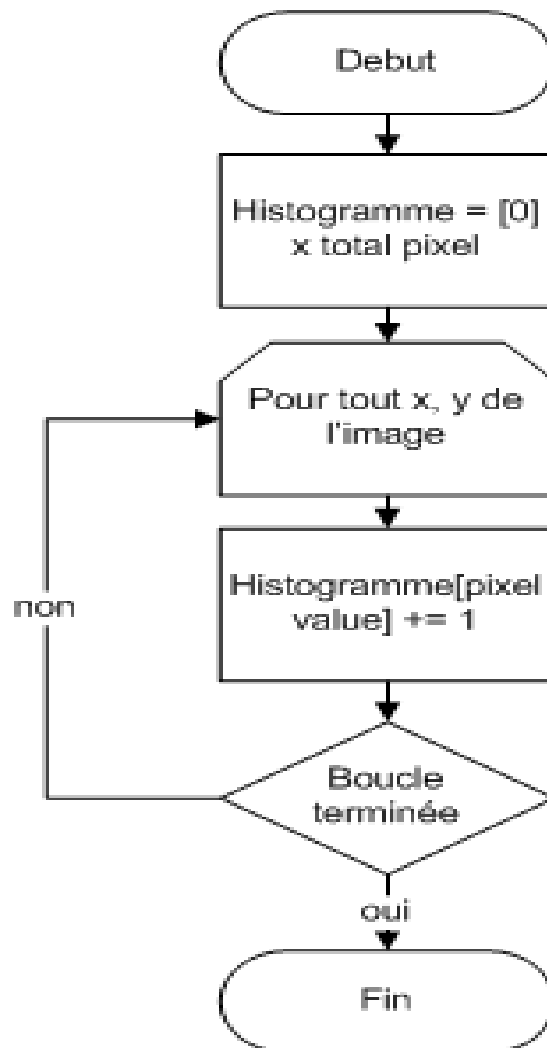
## Seuillage multiple

Cette opération est identique au seuillage sauf que celle-ci permet de spécifier plusieurs seuil et de remplacer les valeurs des pixel par la valeur du plus petit seuil dont la valeur lui est supérieur multiplié par 255 divisé par le nombre de seuil.



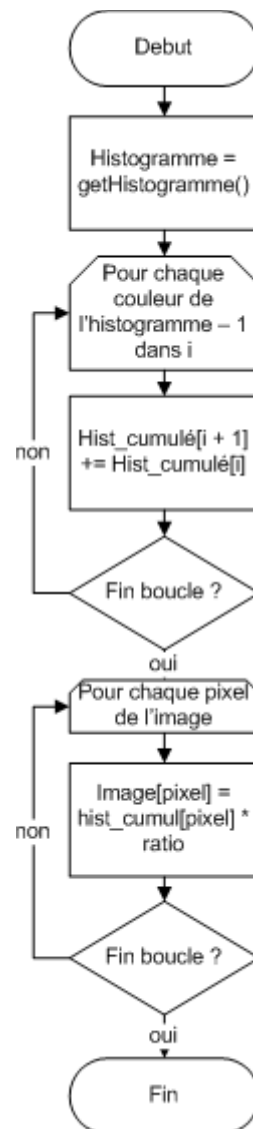
## Histogramme

L'histogramme d'une image est en réalité un tableau dont chaque indice représente une couleur de l'image que l'histogramme représente. Chaque valeur de ce tableau représente le nombre de pixel dans l'image de la couleur correspondant à la case du tableau contenant cette valeur.

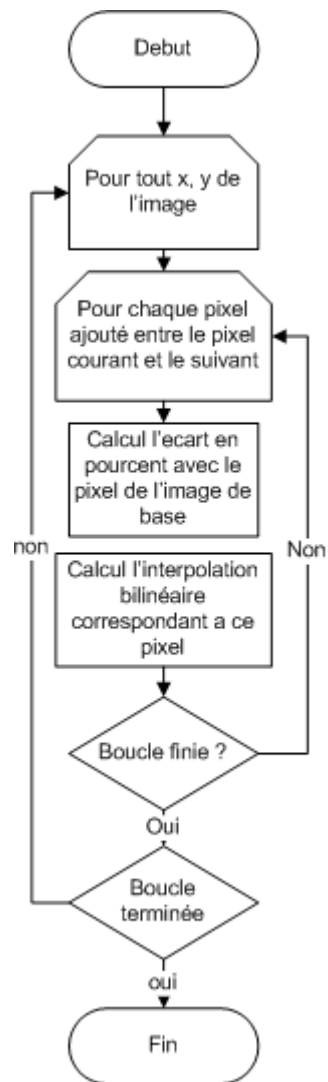


## Égalisation d'histogramme

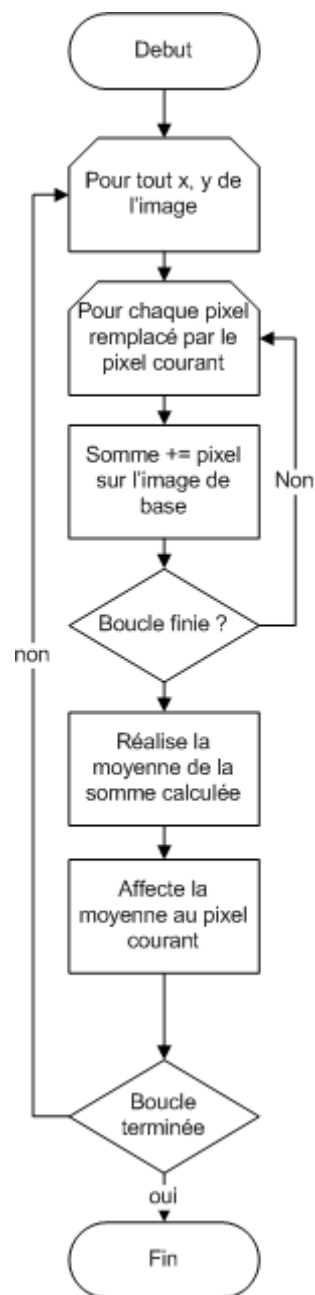
L'égalisation d'un histogramme permet d'ajuster le contraste d'une image et de faire apparaître des détails jusque là imperceptibles (homme sur fond noir, contre jour, ...).



## Expansion



## Extraction



## **Filtres**

**Haute fréquence :** Couleur claire (blanc)

**Basse fréquence :** Couleur foncée (noir)

**Passe-bas :** Atténue les hautes fréquences. Donne un effet de flou sur l'image.

**Passe-haut :** Atténue les basses fréquences. Permet de faire ressortir les contours de l'image.

**Passe-bandes :** Filtre passe-bas combiné avec un filtre passe-haut.

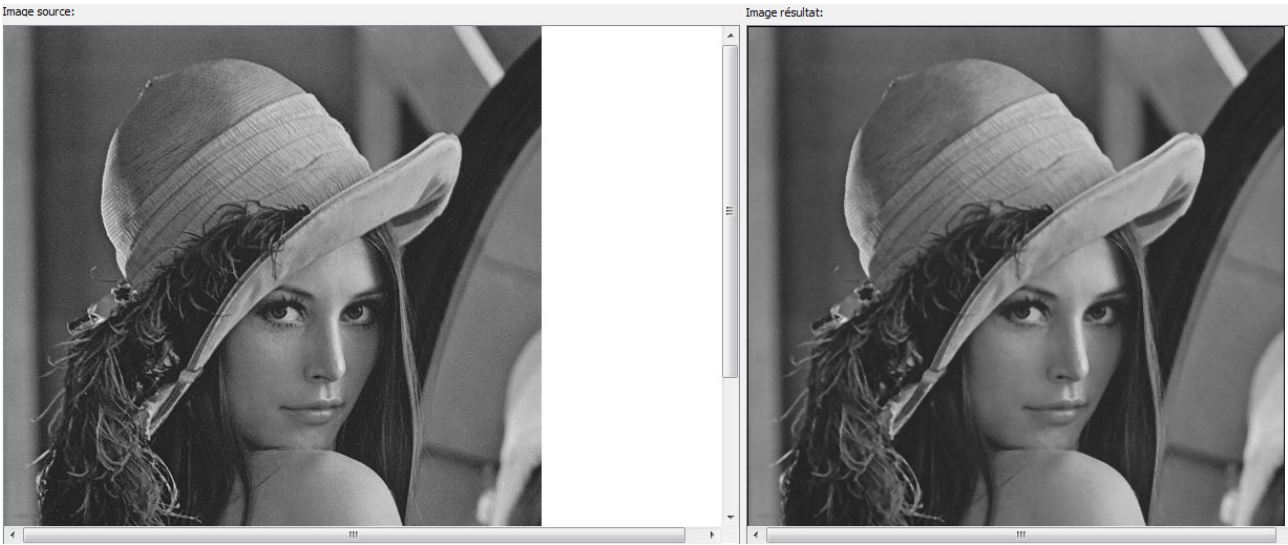
**Filtre moyen :** Filtre passe-bas. Parcourt l'image et remplace chaque pixel par la valeur moyenne des pixels voisins (dans une fenêtre donnée)

**Filtre médian :** Filtre passe-bas. Pour chaque pixel, récupère les valeurs des pixels voisins et remplace le pixel courant par la valeur médiane de la liste des valeurs triée.

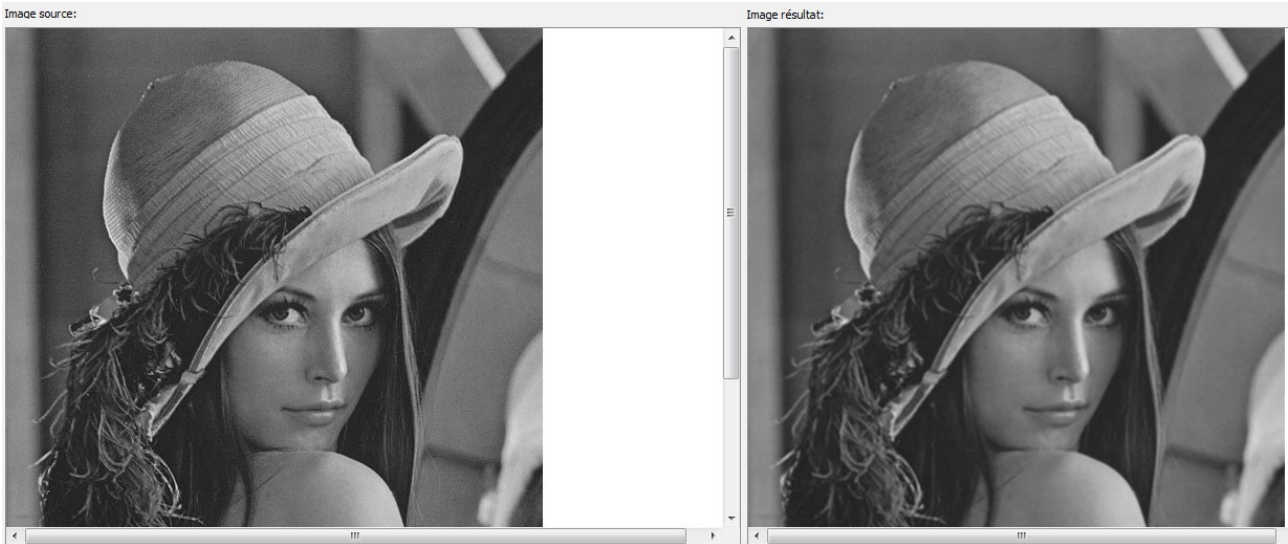
**Filtre Gaussien, Laplacien, Roberts, Prewitt, Kirch, Sobel :** Filtre constitué d'un noyau (appelé noyau de convolution) appliqué sur chaque pixel de l'image par produit de convolution. Hormis le filtre Gaussien, tout ces filtres sont des filtres passe-haut et permettent de faire ressortir les contours de l image.

**Aperçu des résultats :**

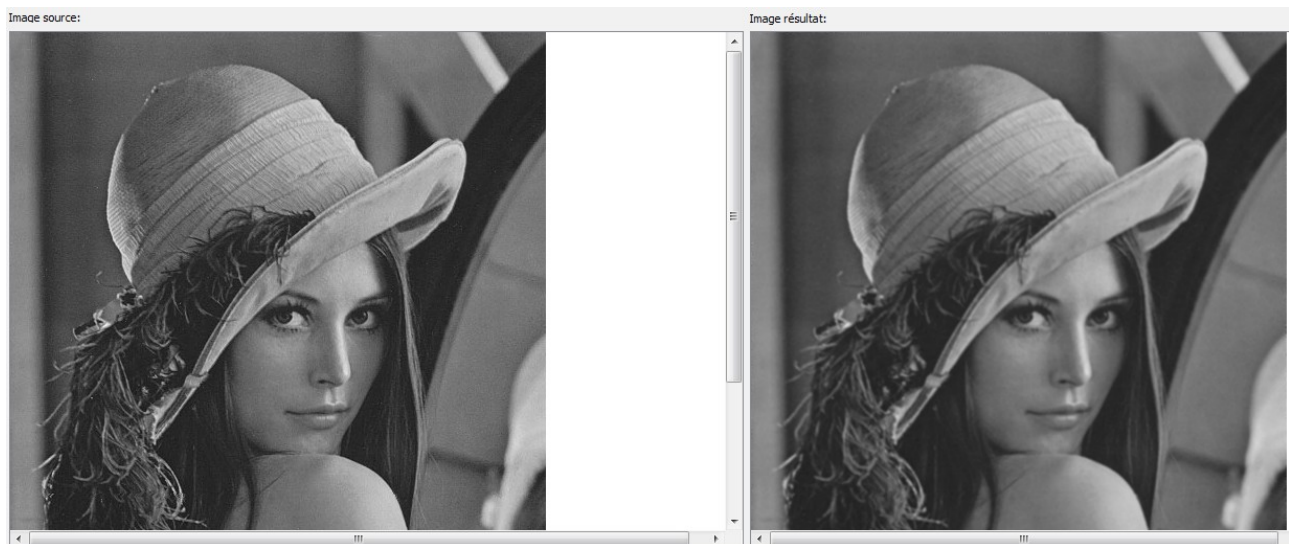
**Filtre médian :**



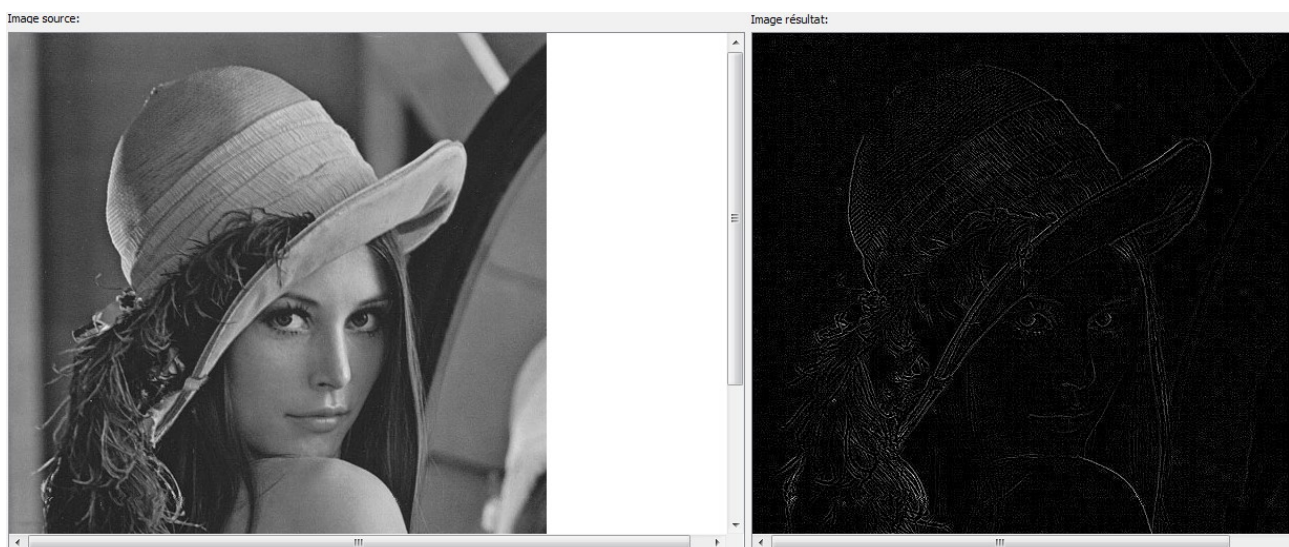
**Filtre moyen :**



### Filtre Gaussien :

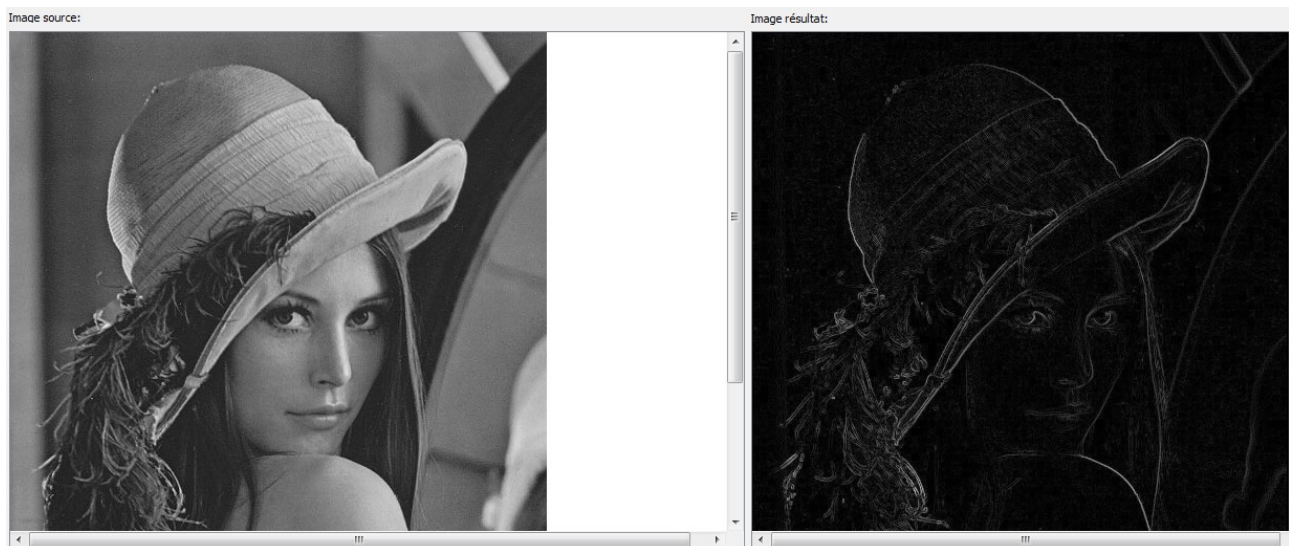


### Filtre Laplacien :

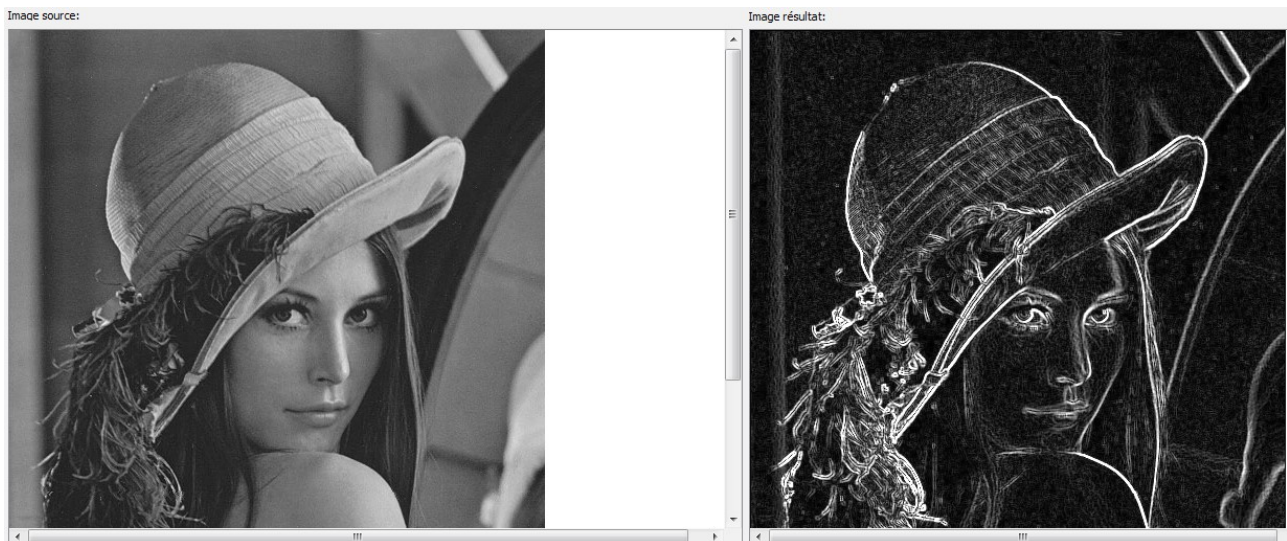




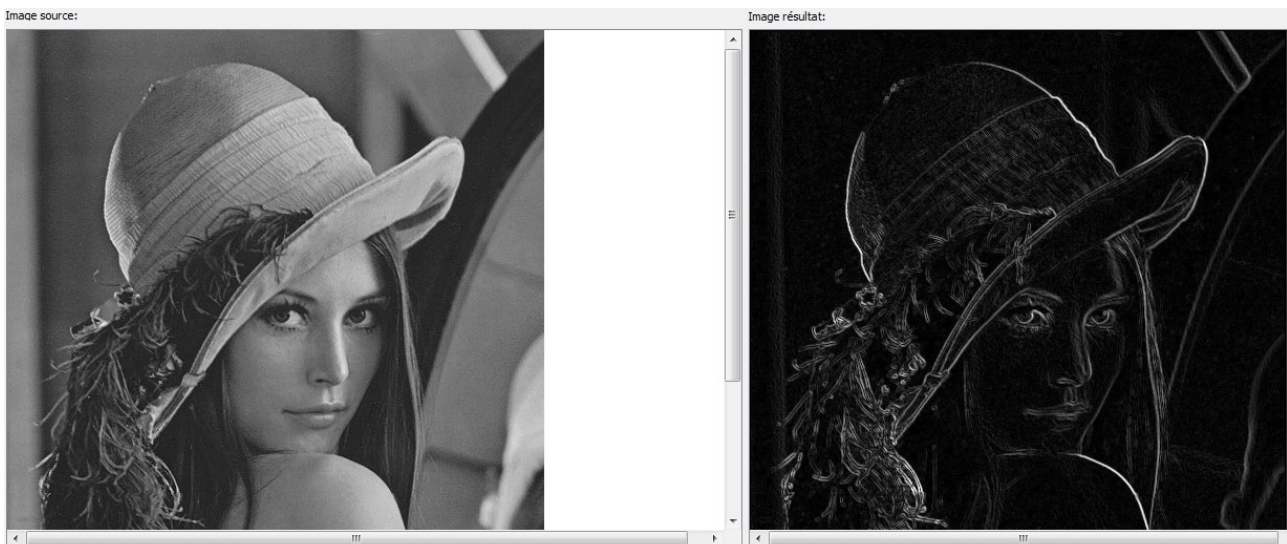
## Filtre de Roberts :



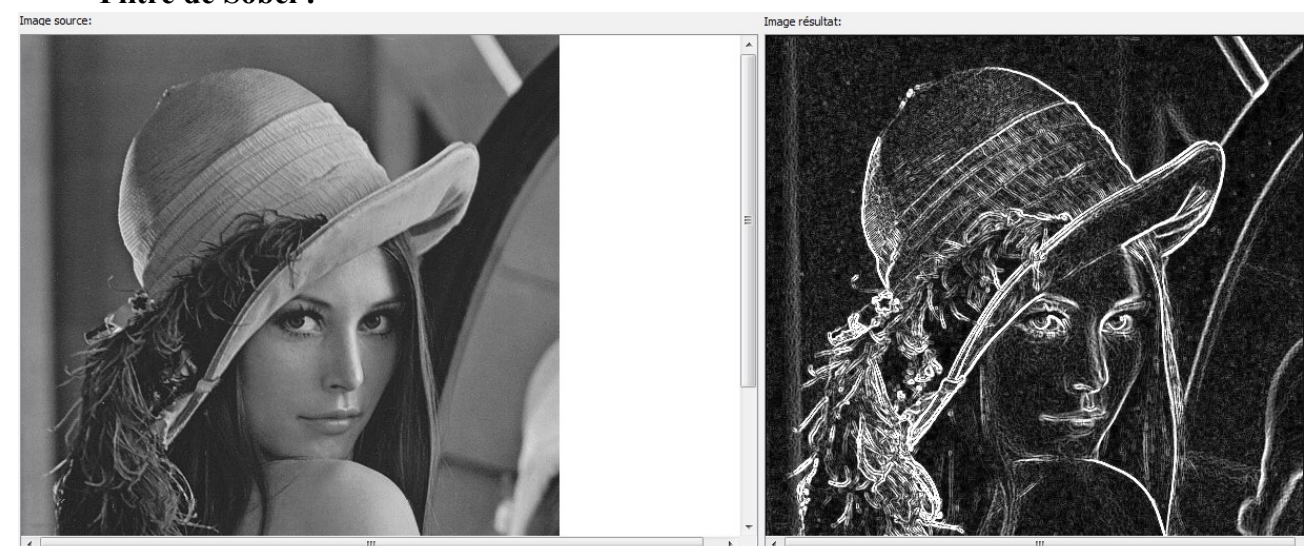
### Filtre de Prewitt :



### Filtre de Kirch :



### Filtre de Sobel :



## Affinage des contours

L'affinage des contours est une opération en traitement d'image qui consiste à parcourir une image binaire (tout les pixel sont soit noir, soit blanc) et à en réduire l'épaisseur des contours afin d'atteindre une taille d'un pixel de large.

L'algorithme utilisé dans mon projet calcule 2 valeurs pour chaque pixel en fonction de son voisinage :

$N(\text{pixel})$  = Somme de tout les pixel blanc autour du pixel donné.

$S(\text{pixel})$  = Nombre de transition de blanc vers noir dans le voisinage du pixel donné.

Voici comment j'ai noté le voisinage du pixel :

p9	p2	p3
p8	p1	p4
p7	p6	p5

Le calcul de S se fait en parcourant les pixel dans l'ordre donné ci-dessus.

La deuxième étape de l'algorithme consiste à marquer tout les pixel vérifiant toutes ces conditions :

- $N(\text{pixel}) \neq 0$
- $N(\text{pixel}) \neq 1$
- $N(\text{pixel}) \neq 7$
- $N(\text{pixel}) \neq 8$
- $S(\text{pixel}) < 2$

Lorsque N vaut 0, le pixel est isolé et ne doit donc pas être supprimé.

Lorsque N vaut 1, le pixel est une extrémité de ligne et ne doit donc pas disparaître.

Lorsque N vaut 8 ou 7, le pixel n'est pas un contour.

Si S est plus grand ou égal à 2, le pixel est une connexion entre deux contours.

On supprime ensuite tout les pixel marqué (blanc -> noir) et on réitère l'algorithme jusqu'à ce que la liste de pixel marqué soit vide après chaque itération.

## **Commentaires**

Ce dernier dossier m'a demandé beaucoup plus de travail de recherche que les pour les précédents dossier. Les différents algorithmes que j'ai pu explorer ne donnaient pas tous des résultats satisfaisant.