REPUBLIQUE DU CAMEROUN
Paix-Travail-Patrie

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

UNIVERSITE DE YAOUNDE I

ECOLE NATIONALE SUPERIEURE POLYTECHNIQUE

DEPARTEMENT DE GENIE INFORMATIQUE

REPUBLIC OF CAMEROON
Peace-Work-Fatherland

MINISTRY OF HIGHLY EDUCATION

UNIVERSITY OF YAOUNDE I

NATIONAL ADVANCED SCHOOL OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCES

Implémentation des méthodes de traitement d'image

Réalisé par

- LEKANE TATSAGOUM Arthur-Carnis
- MAFO DE HEDZO Leonie Ariane
- TATSAWOUM FOMETIO Sidoin
- NIAYAKO SOFFACK Kapoel Césaire

Devoir Proposé par :

- M. Hippolyte TAPAMO

TABLE DES MATIERES

LISTE DES FIGURES

INTRODUCTION

Avant de parler de traitement d'images, il serait plus intelligent de comprendre la notion d'image. La définition d'une image est très divergente selon les domaines. On peut cependant dire globalement qu'il s'agit d'une représentation visuelle d'une chose ou d'une personne. Dans le contexte présent, on pourrait classifier une image en fonction de la représentation (pixel ou vecteur) ou de son mode d'obtention (naturel ou artificiel). Tout au long de notre travail, nous avons essentiellement traité des images matricielles de type PGM.

Les intérêts du traitement d'image sont variés. En allant du contrôle de présence à la reconnaissance de l'écriture, ses domaines d'application s'étendent à la photographie, le cinéma numérique, l'astronomie, la médecine, la sécurité, la microscopie, et à la micro-tomographie.

Dans le présent rapport, nous allons commencer par présenter les fonctionnalités de base de notre application. Par la suite, nous verrons l'implémentation des traitements ponctuels et liés à la géométrie de l'image. Puis, nous aborderons les traitements locaux et globaux pour finir avec les opérateurs morphologiques.

I- PRESENTATION GLOBALE DE NOTRE INTERFACE

Au lancement, notre application a l'aspect de la figure 1.

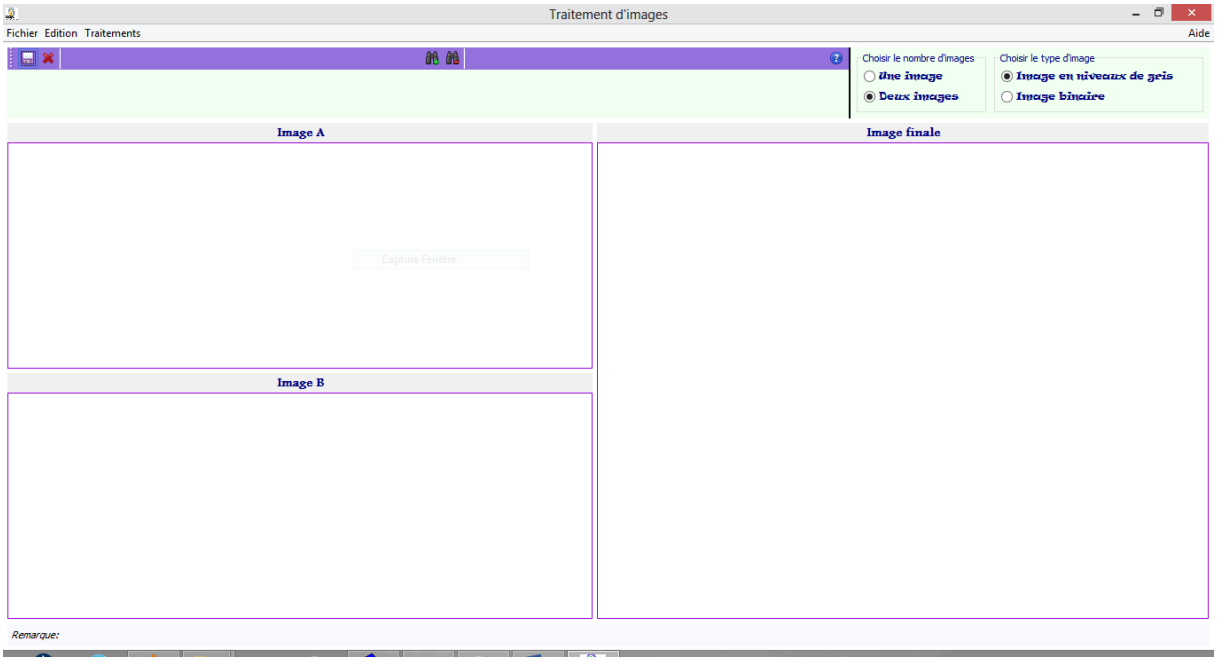

Figure 1: Aspect principal de l'application

Elle pourrait aussi avoir la forme de la figure 2.

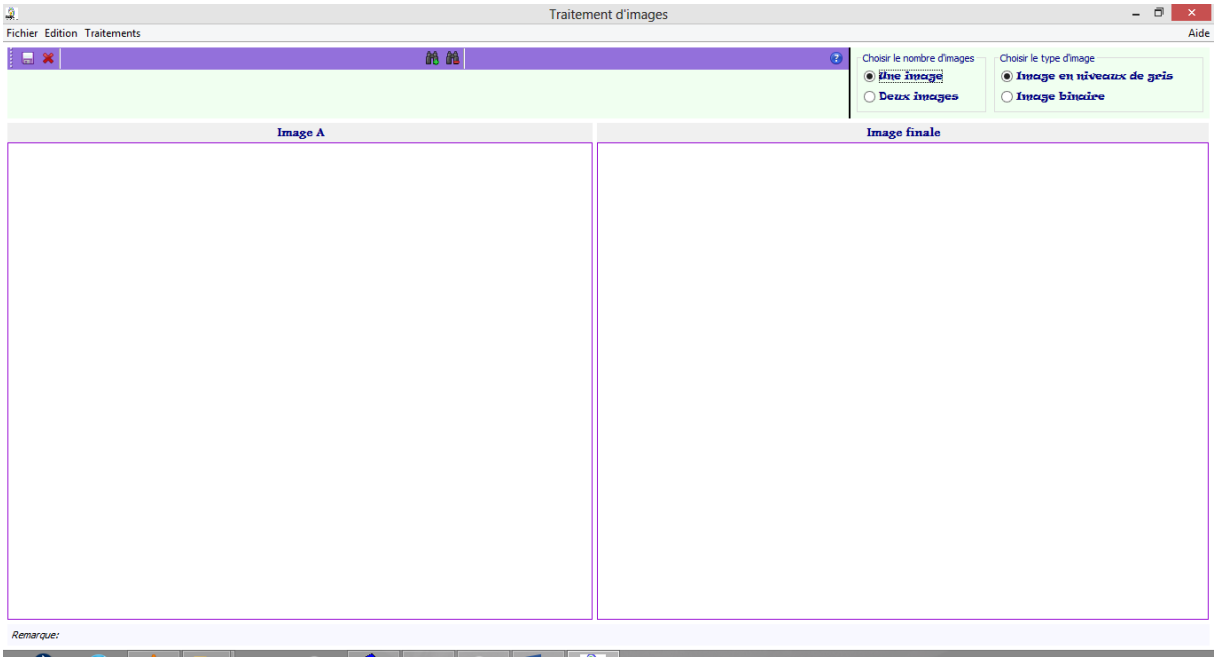

Figure 2: Aspect secondaire de l'application

Dans la zone centrale, nous apercevons facilement trois zones au plus: une pour la première image, une pour la seconde et une autre (à droite) pour l'image résultante du traitement. On pourrait comme le montre la figure 2, n'avoir que deux zones. Dans ce cas, l'image B n'existe pas.

Les premières configurations dans l'interface se font dans le panel en haut à droite (figure 3). On a le choix d'une part entre une et deux images et d'autres part entre une image en niveaux de gris ou une image binaire. Le choix de l'un ou de l'autre dans les deux cas entrainera une modification conséquente de l'interface et la suppression ou l'ajout automatique des sous menus. Par exemple, le choix d'une image binaire entrainera une apparition parmi les sous menus du menu *traitements* des opérations morphologiques.

Figure 3: Configuration initiale de l'interface

Avant de commencer à travailler, nous devons aussi charger une image pgm (figure 4). Ceci ce fait soit en allant dans *fichier > charger > ImageA* soit en cliquant tout simplement sur la zone d'affichage de l'image qu'on souhaite charger. Il est également possible d'enregistrer une image au format pgm (figure 5) en allant dans *fichier > enregistrer* ou en cliquant sur la zone d'affichage de l'image résultat.

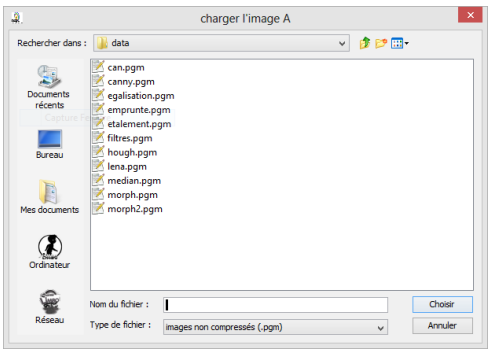

Figure 4: chargement d'une image

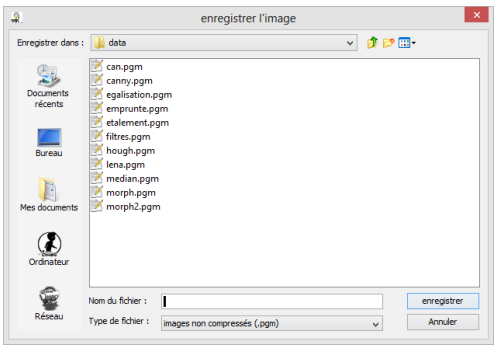

Figure 5: enregistrement d'une image

Lorsque l'image est chargée, on peut la voir apparaitre dans l'interface (figure 6) et tous les traitements fournissent des images qui sont affichés dans le panel de gauche. (exemple pour la binarisation à la figure 6).

Nous pouvons également afficher l'histogramme d'une image en allant dans *traitements > histogramme* (figure 7).

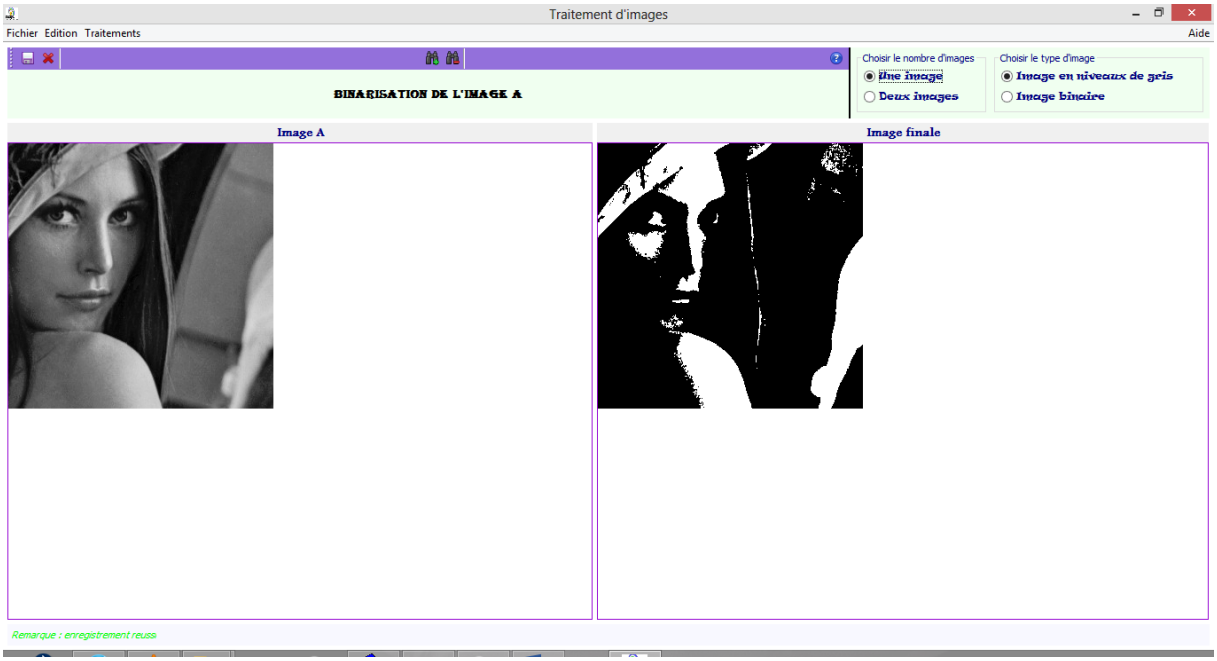

Figure 6: binarisation d'une image

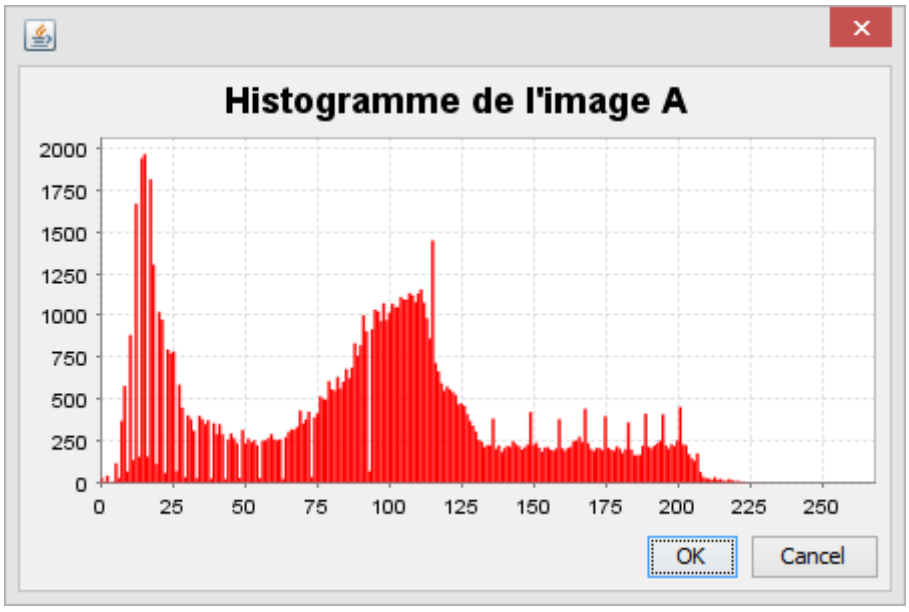

Figure 7: histogramme d'une image

II- TRAITEMENTS PONCTUELS ET GEOMETRIE DE L'IMAGE

Les transformations ponctuelles sont celles dans lesquelles un pixel n'a pas besoin des autres pour subir une transformation.

II.1- L'inversion

Cette opération est fort simple, elle consiste tout simplement à prendre le complémentaire de chaque pixel x (à savoir 255 – x). L'option de binarisation est disponible à *traitement > inversion* lorsqu'on a fait le choix d'une image en niveau de gris (la figure 3 montre comment faire ce choix). La figure 8 nous montre un exemple d'inversion.

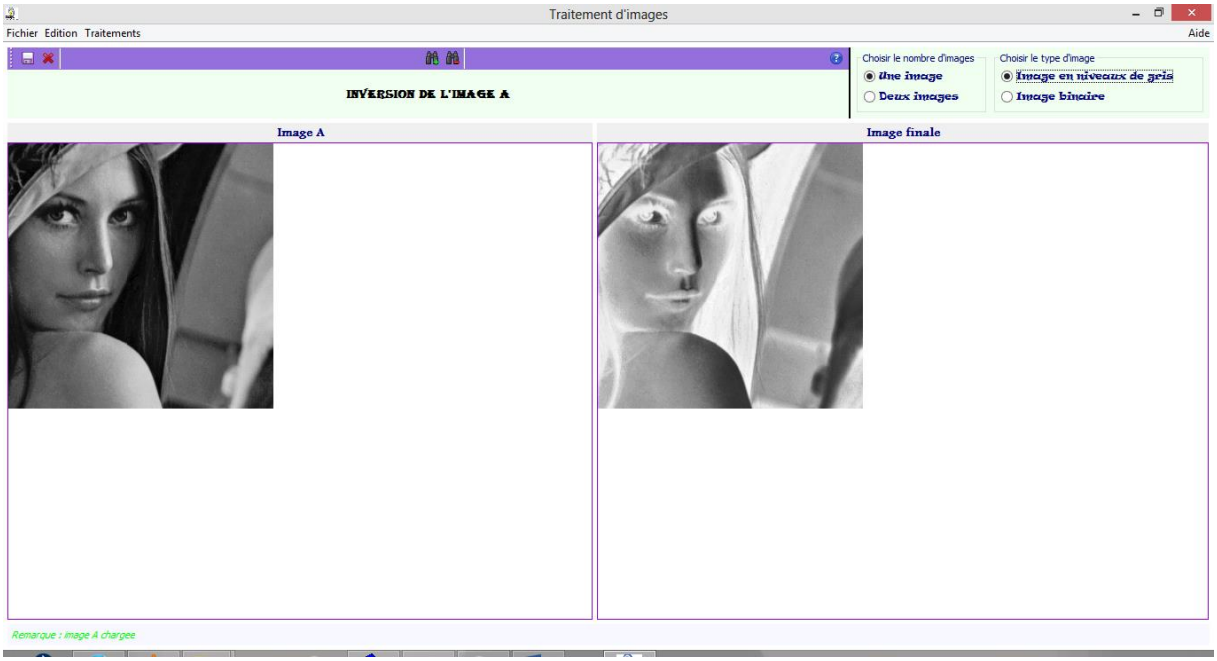

Figure 8: inversion d'une image

II.2 - Rotation

La rotation consiste tout simplement à changer la position de chaque pixel de sorte que l'image finale subisse une rotation d'un angle donné. Pour mettre en place cet algorithme, nous avons appliqué une rotation à chaque pixel selon la formule suivante (en considérant x' et y' comme étant les nouvelles positions du pixel) : $\begin{cases} x' = xcos\theta - ysin\theta \\ y' = xsin\theta + ycos\theta \end{cases}$

Bien que l'idée première fut de partir des pixels de l'image de base avant d'appliquer la formule et faire le recadrage, nous avons plutôt opté pour le chemin inverse c'est-à-dire prendre chaque pixel de l'image finale et deviner sa couleur en cherchant le pixel dont il provient. Ceci se fait en utilisant la réciproque de la formule proposée plus haut et l'avantage est d'éviter les trous dans l'image. La figure 9 montre une rotation d'un angle 45° (angle entré par l'utilisateur).

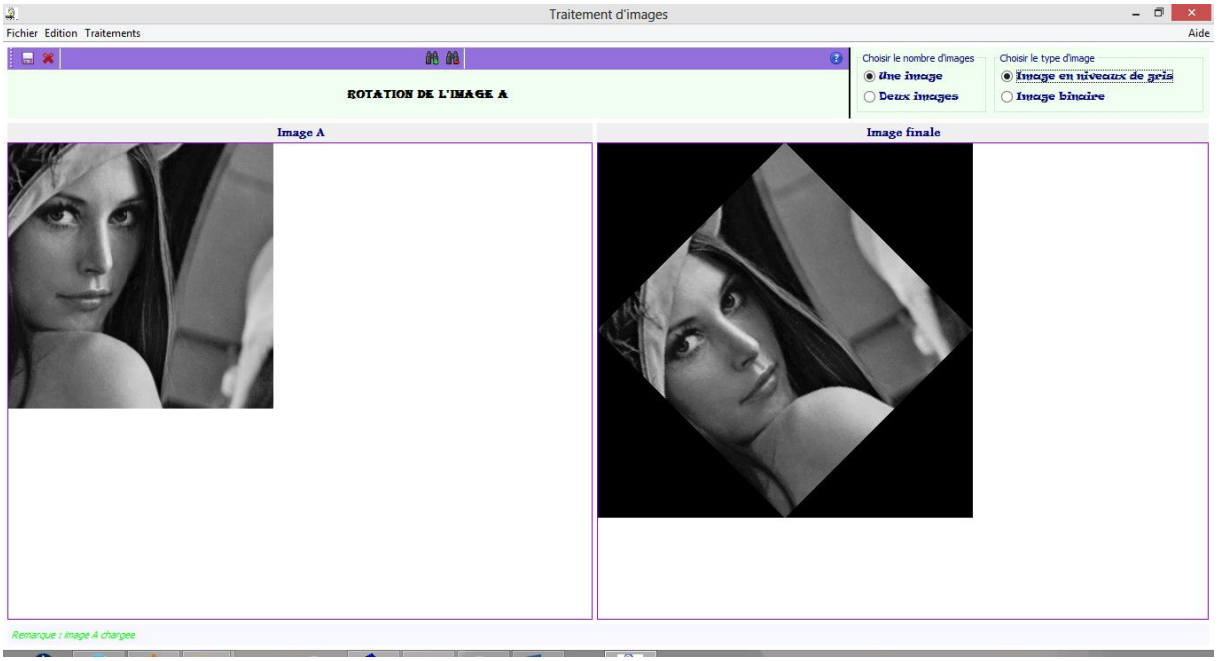

Figure 9: rotation de 45° d'une image

II.3 – Flip horizontal et vertical

Le travail ici consistait à réaliser les fonctions de réflexion de l'image par rapport à un miroir. La figure 10 montre un exemple de flip horizontal (*traitement > flip > horizontal*).

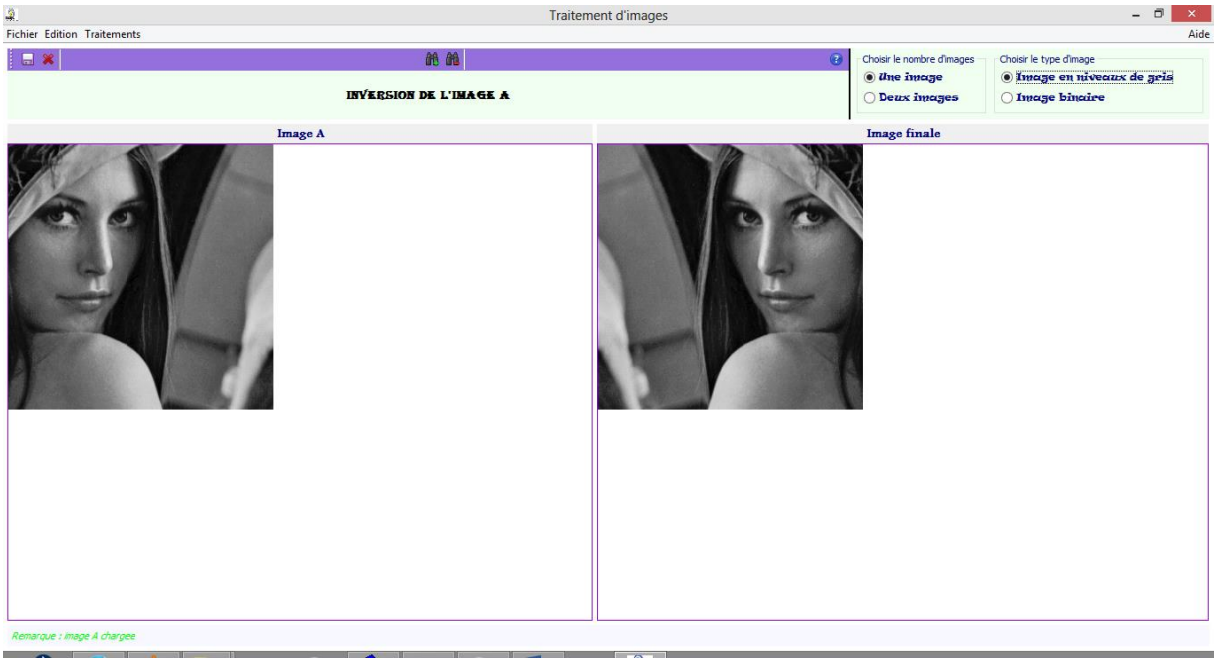

Figure 10: flip horizontal

II.4 – Etalement et égalisation d'histogramme

Les opérations d'étalement et d'égalisation d'histogramme ont pour but d'améliorer le contraste d'une image. L'étalement d'histogramme (figure 11) est utile dans les images où la valeur des pixels s'étend d'une certaine valeur *min* à une valeur *max* (figure 12). Pour réaliser l'étalement d'histogramme, nous réalisons la bijection de cet intervalle [min, max] vers l'intervalle [0, 255] (figure 13).

Ceci étant si le pixel min d'une image est 0 et son pixel max 255, alors l'étalement d'histogramme n'a strictement aucun effet. Dans ce cas, l'égalisation d'histogramme s'avère utile. Elle vise à aplatir au maximum l'histogramme de l'image. Les figures 14, 15 et 16 en sont des illustrations.

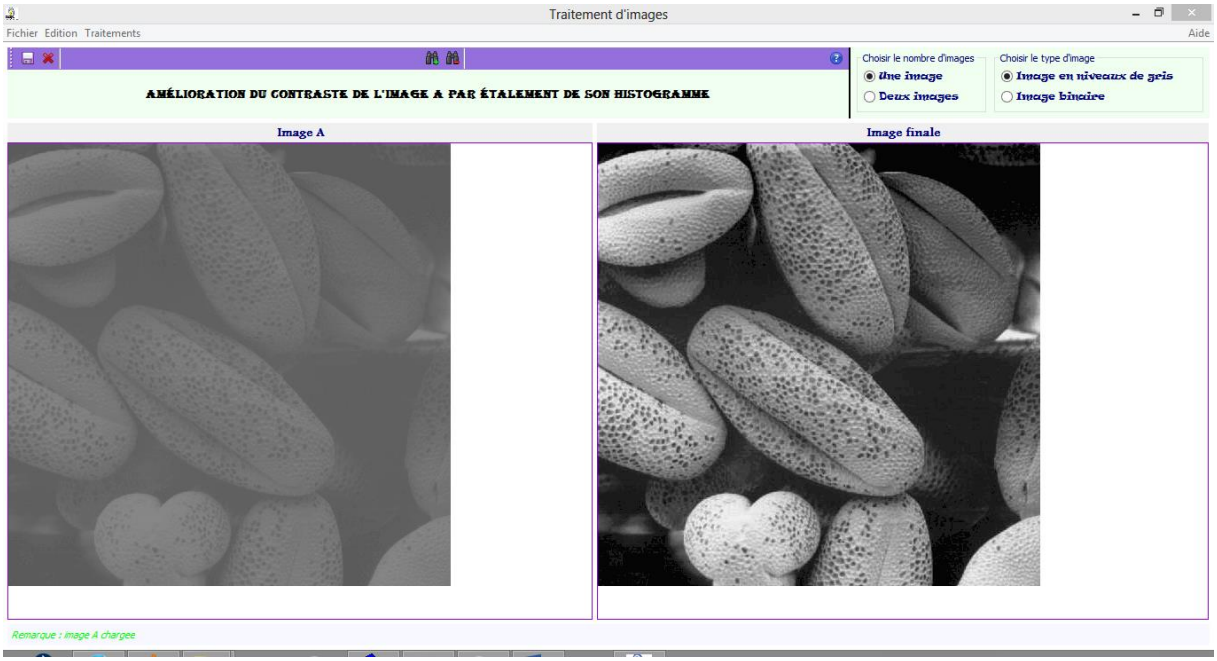

Figure 11: étalement d'histogramme

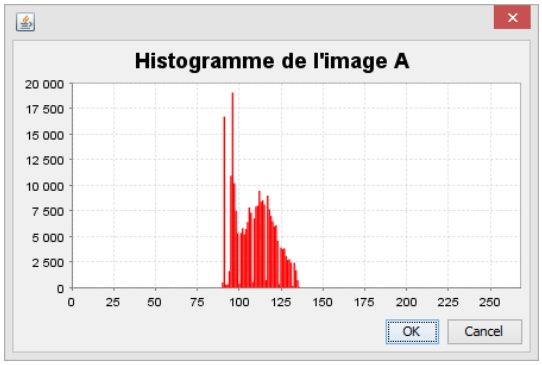

Figure 12: exemple histogramme d'une image à étaler

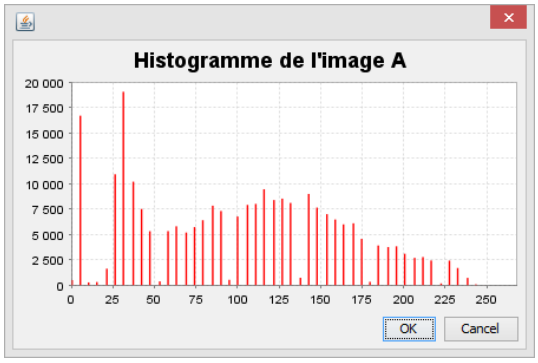

Figure 13: histogramme de l'image étalée

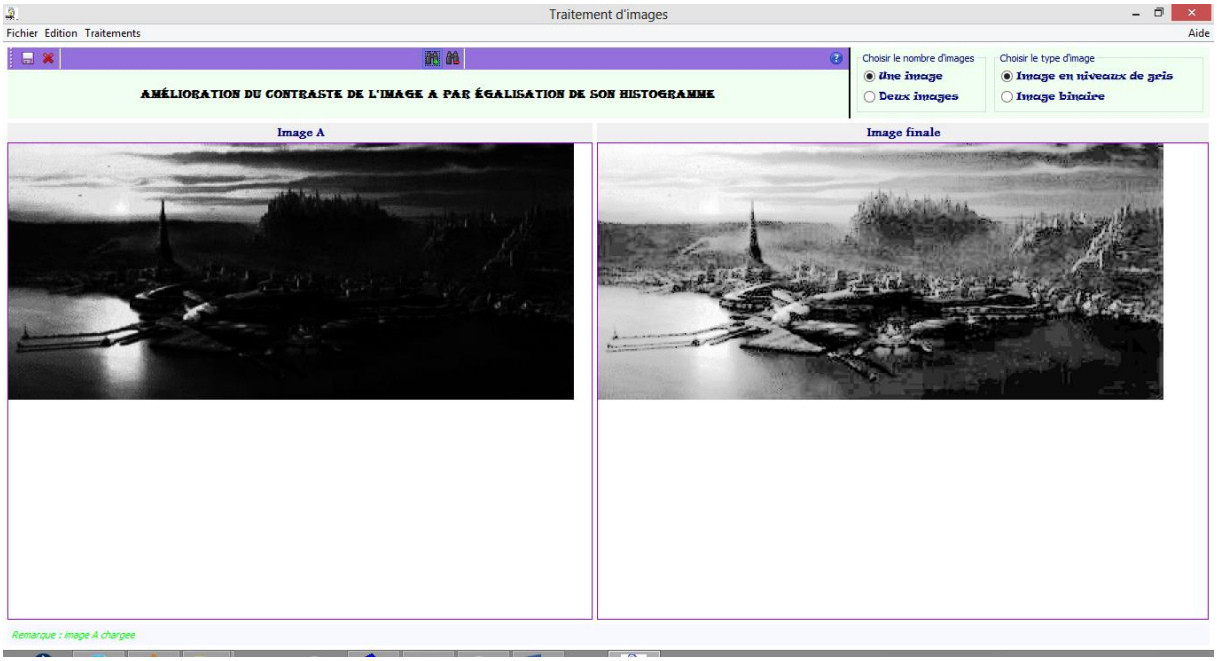

Figure 14: égalisation d'histogramme d'une image

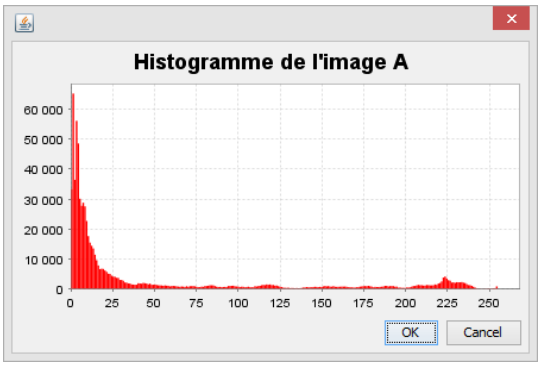

Figure 15: histogramme d'une image bonne à être égalisée

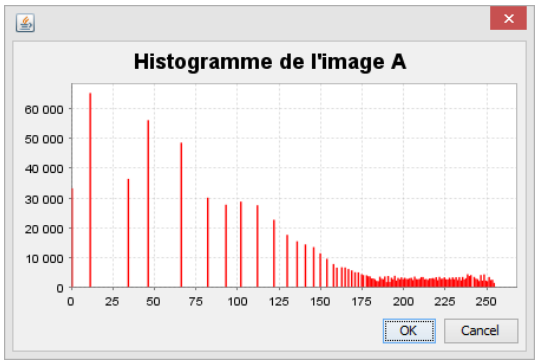

Figure 16: Histogramme de l'image de la figure 15 égalisée

II.5 – Addition et soustraction d'images

Pour pouvoir faire des additions et des soustractions d'image, il faudrait au préalable activer le mode en deux images en utilisant le panneau de la figure 3. Afin que les deux images aient la même taille, nous avons complété les pixels de l'une ou l'autre image à 0. La figure 17 montre l'addition de deux images; La soustraction se faire par le même processus (*traitement > opération > addition*).

L'un des avantages de l'addition est qu'on peut l'utiliser pour améliorer le contraste d'une image, ceci est additionnant l'image avec elle-même. En effet, soit d la dynamique d'une image, $d = max - min$, si on multiplie tous les pixels de l'image par 2 (addition de l'image par elle-même) alors la dynamique devient $d' = (max + max) - (min + min) = (max - min) + (\max - min) = d * 2$

On voit bien avec la formule ci-dessus que la dynamique de l'image se multiplie par 2 quand on additionne une image par elle-même. On peut donc en conclure que cette opération a pour conséquence d'améliorer le contraste de l'image.

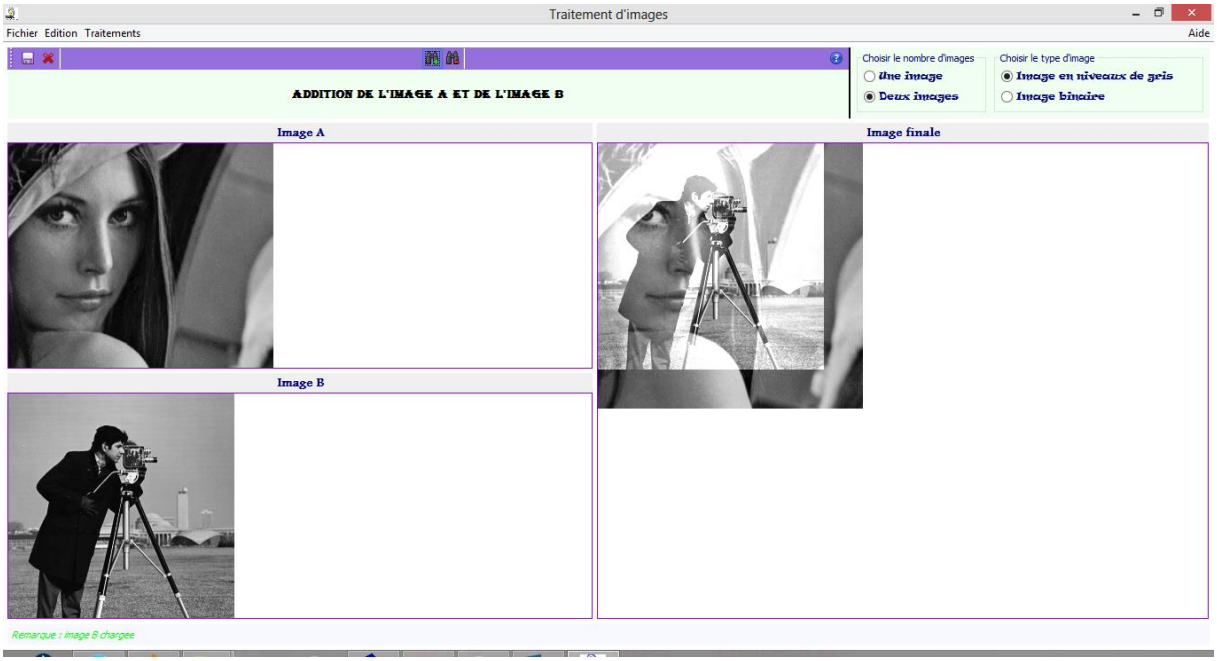

Figure 17: addition de deux images

La soustraction quant à elle permet entre autres de détecter les mouvements sur une image. Lorsqu'on charge deux images pris à des instants différents, la soustraction de ces deux images va uniquement montrer les zones de différence.

III- LES TRAITEMENTS LOCAUX ET GLOBAUX

Les traitements locaux et globaux sont celles dans lesquelles la transformation d'un pixel implique l'utilisation des voisins.

III.1 – convolution d'une image

L'avantage de la convolution est qu'elle est utilisée pour coder la plus part des filtres. Nous pouvons néanmoins l'utiliser de façon indépendante, il suffit d'aller à *traitement > convolution*. Alors une fenêtre apparait (figure 18) et demande de renseigner le masque de convolution. Lorsque ceci est fait, l'image A est convoluée suivant le masque entré.

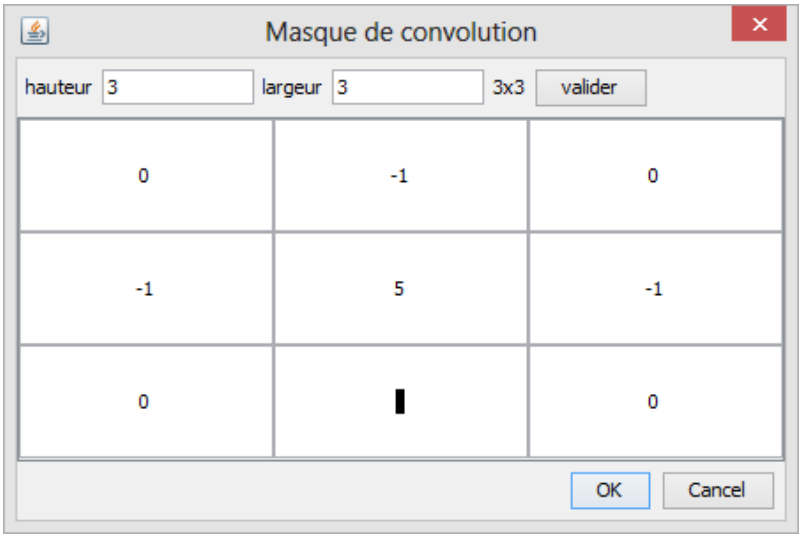

Figure 18: masque de convolution

Il est à noter ici que pour gérer les bords, nous avons tout simplement conservé les anciennes valeurs de pixel à ces endroits.

III.2 – Les filtres

III.2.a – Les filtres de sobel et prewitt

Il s'agit là des filtres de gradient, permettant de faire la détection des contours. Leur mise en œuvre s'est faite simplement en appliquant la convolution avec un certain masque.

Le filtre de sobel (figure 19) consiste tout simplement à faire la convolution de l'image avec les matrices Mx et My et d'obtenir les images Gx et Gy tels que:

$$Mx = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix};\ My = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

L'image finale résultante est obtenue en- faisant $G = \sqrt{Gx^2 + Gy^2}$

Le filtre de sobel s'accompagne souvent d'un seuillage (figure 20). Lorsque l'on choisit *traitements > filtres > sobel*, une boite de dialogue apparait et demande le seuil. Lorsque la valeur est 0, on suppose qu'il n'y a pas de seuil. Autrement, le seuil représente la valeur entrée.

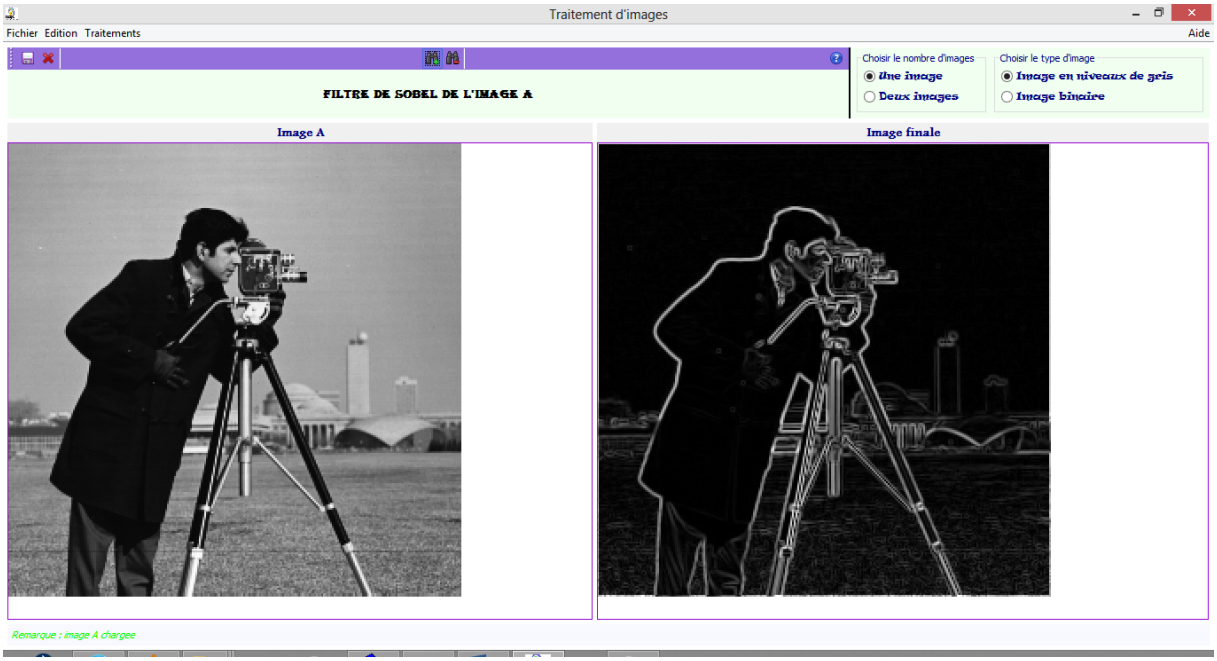

Figure 19: filtre de sobel sans seuillage

Figure 20: filtre de sobel avec pour seuillage 70

Le filtre de prewitt se manipule quasiment de la même façon. La seule différence se situe au niveau des masques de convolution.

$$Mx = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}; \; My = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

III.2.b – Le filtre de canny

Le filtre de canny (figure 21) est aussi un filtre permettant la détection des contours. Cependant, sa mise en œuvre est plus difficile et se fait suivant les étapes suivantes :

- **Le lissage** (en utilisant par exemple un filtre gaussien - on verra ça par la suite -)
- **Le gradient** (pour la détection classique des contours - nous avons utilisé prewitt -)
- **La suppression des non-maxima :** consiste pour chaque profil d'intensité coupant les contours (direction du profil trouvé à partir de la direction du contour - $\arctan(\frac{Gy}{Gx})$ -) à mettre à zéro tous les pixels qui ne sont pas des maxima
- **Le double seuillage** : Ici, on se donne deux seuils SInf et SSup. Lorsqu'un pixel est inférieur à SInf, alors on le met à 0, s'il est supérieur à SSup, alors, on le met à 255. S'il est compris entre SInf et SSup, on le conserve
- **La suppression des hystérésis :** Parmi les pixels compris entre SInf et SSup, ceux qui sont liés à un contour sont des contours et ceux qui ne le sont pas sont supprimés (remis à 0)

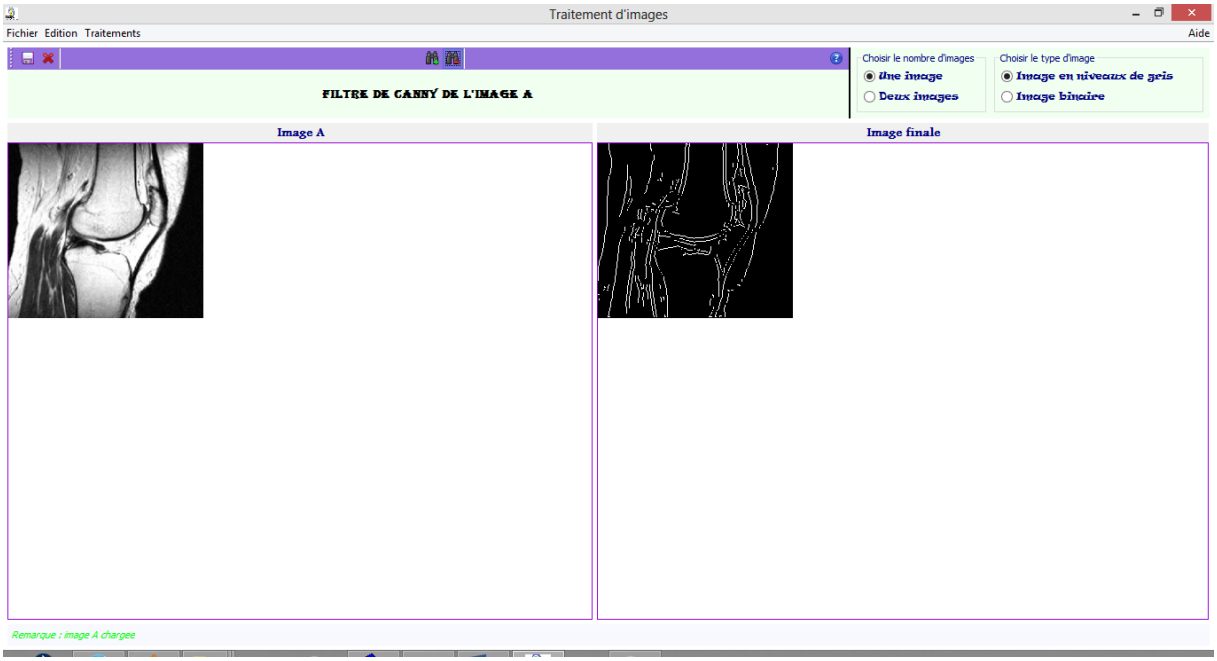

Figure 21: filtre de canny

III.2.c – Le filtre moyenneur et le filtre gaussien

A la différence des premiers filtres vus dans cette partie, ces filtres permettent plutôt le lissage des images. Le principe est d'utiliser un masque qui va rendre l'image un peu floue.

Pour le filtre moyenneur, c'est simple car on utilise un masque dont les valeurs sont égales et dont leur somme donne 1.

Le filtre gaussien (figure 23) quant à lui est plus doux que le filtre moyenneur car son masque suit l'allure de la fonction de gauss. Lorsque l'on clique sur *traitements > filtres > gaussien*, un boite de dialogue s'ouvre (figure 22) et demande de renseigner la taille du masque et le facteur σ qui déterminerait l'allure de la courbe. L'option "afficher le masque" nous permet de voir la matrice du masque en fonction de sa taille et de σ.

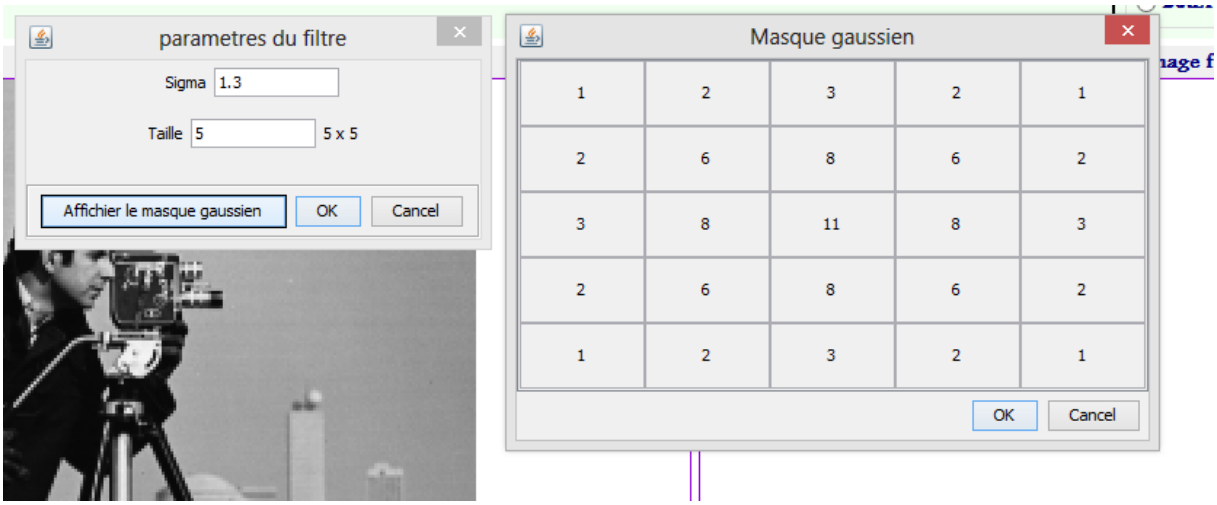

Figure 22: paramètres du filtre gaussien

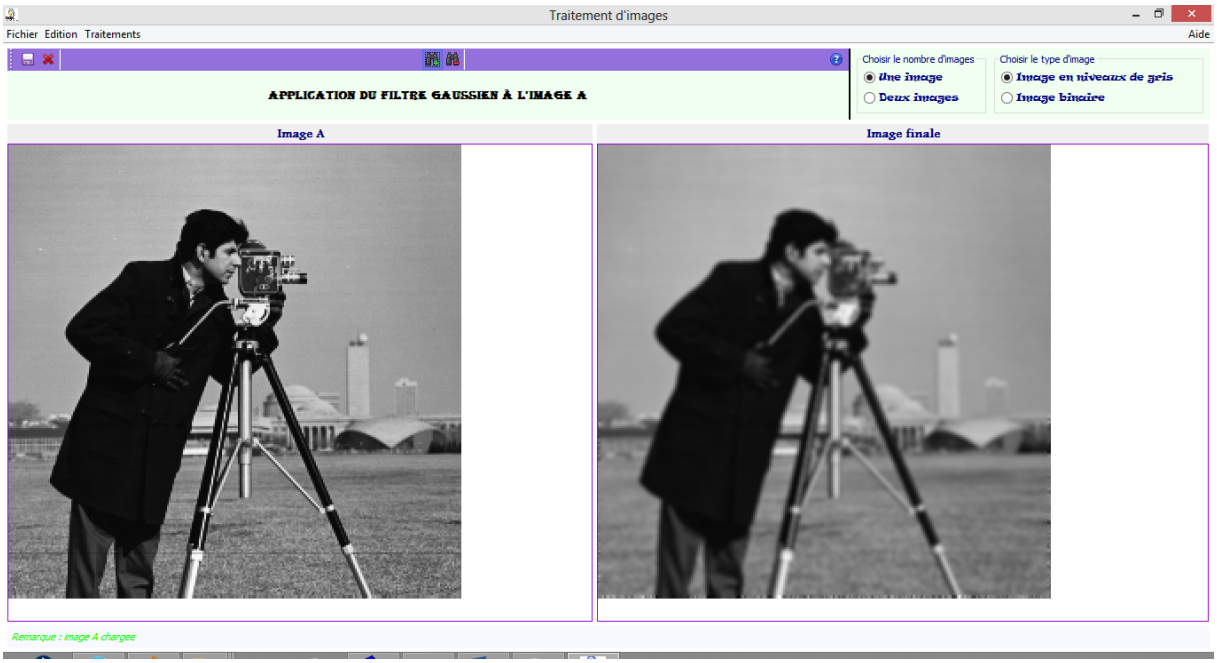

Figure 23: filtre gaussien

III.2.d – Le filtre médian

Ce filtre permet aussi de faire le lissage (figure 24) et il est adapté pour supprimer les bruits de type poivre. Son principe est qu'il trie tous les pixels autour de lui et est remplacé par celui du milieu.

Figure 24: filtre médian

III.3 – La transformée de hough

A l'origine, la transformée de hough a été inventée pour détecter les droites sur une image. Mais elle a évolué et permet désormais de détecter des formes complexes sur l'image. Nous ne nous intéressons ici qu'à sa fonction de base c'est-à-dire à celle permettant la détection des droites.

L'algorithme de hough est basé sur un changement de repère vers un repère paramétrique (figure 25). Ceci implique qu'une droite du repère de base devient un point (de coordonnées ρ et θ) dans le repère paramétrique.

Pour chaque point du repère de base, on trace une infinité de droites tout autour et on place les points équivalents sur le repère paramétrique: l'ensemble forme une courbe paramétrique. Au final, pour un point du repère initial, on a une courbe paramétrique dans le repère paramétrique.

Pour trois points, on aurait quelque chose comme la figure 26. Là, on se rend compte que les trois courbes se coupent en un point unique lorsqu'on regarde l'intervalle de θ entre 0 et 2π. Ce point correspond en fait à la droite que ces 3 points partagent dans le repère de base.

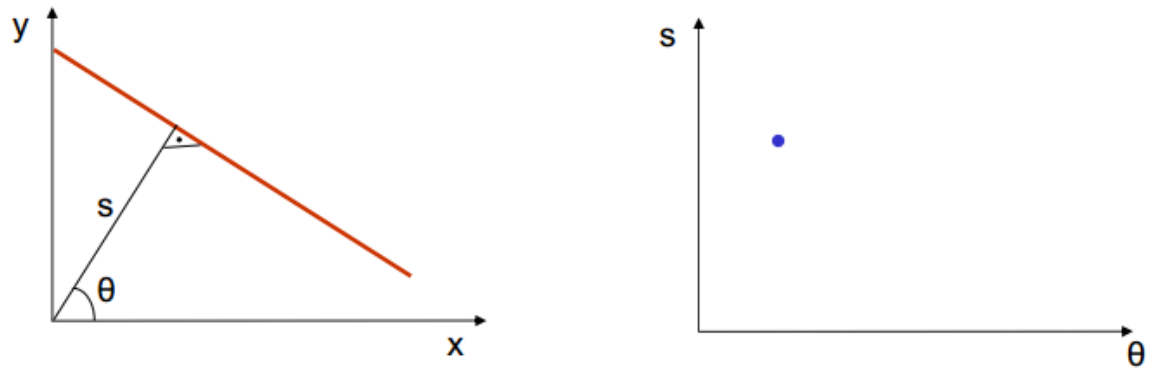

Figure 25 : transformation de repère

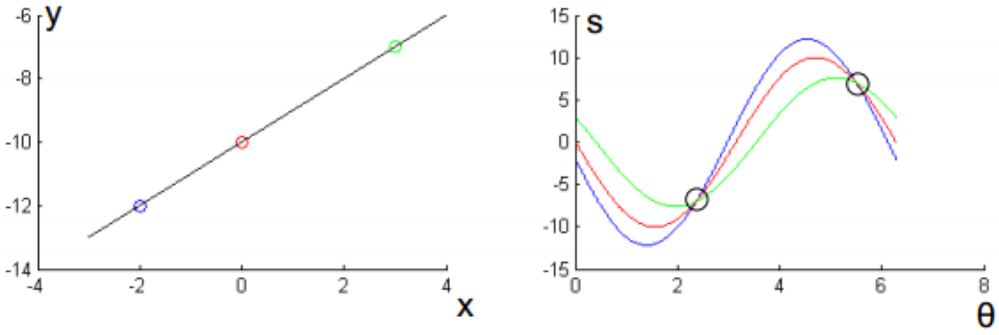

Figure 26: transformation de hough de 3 points

Par conjecture, lorsqu'on a des milliers de points sur l'image, on se retrouve avec une courbe vraiment touffue (figure 27). Cette courbe s'appelle la transformée de hough et pour l'exploiter, on recherche tout simplement les points ou le plus grand nombre de courbes se touchent, ces points correspondraient aux droites sur l'image.

La mise en œuvre de la transformée de hough nécessite un prétraitement. Il s'agit en effet de faire une détection de contours avec seuillage sur l'image. Ainsi, les courbes paramétriques vont être tracées uniquement pour les pixels appartenant aux contours. Une difficulté réside aussi dans la façon de représenter l'intersection. Pour régler cela, on se dit tout simplement que lorsqu'une courbe coupe une autre, la valeur au point d'intersection est la somme des valeurs des pixels sur les deux courbes.

Pour vérifier que notre tranformée de hough fonctionne, nous avons pris une image dans laquelle la présence d'une droite est flagrante (figure 28).

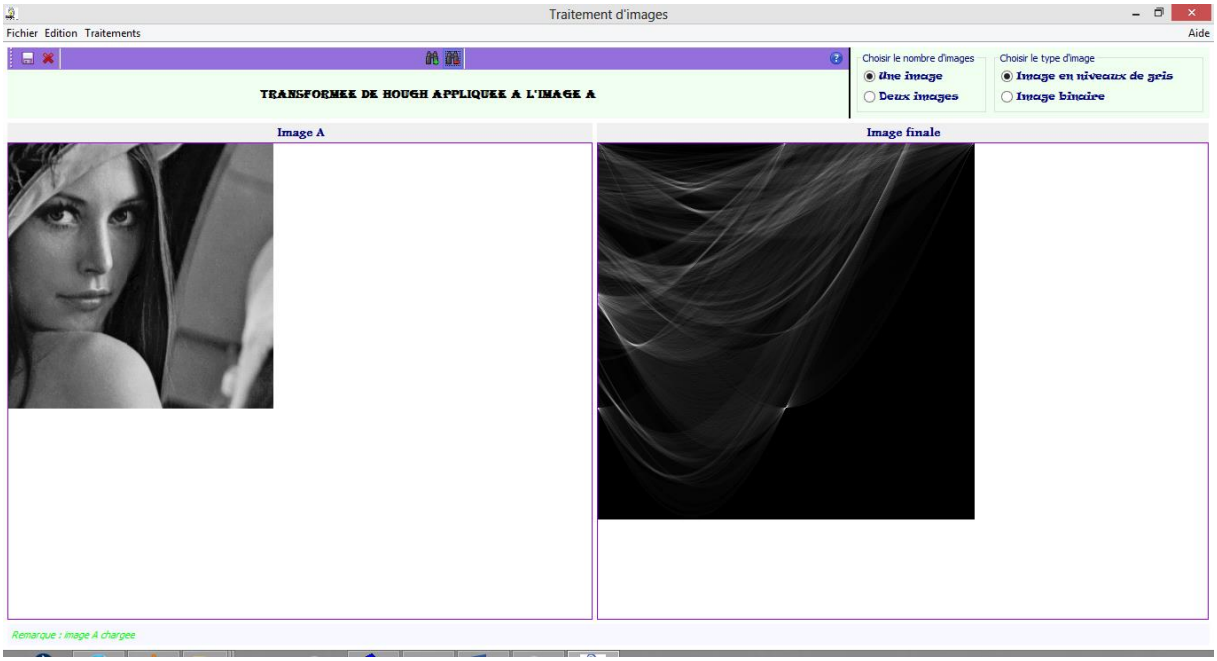

Figure 27: transformée de hough d'une image

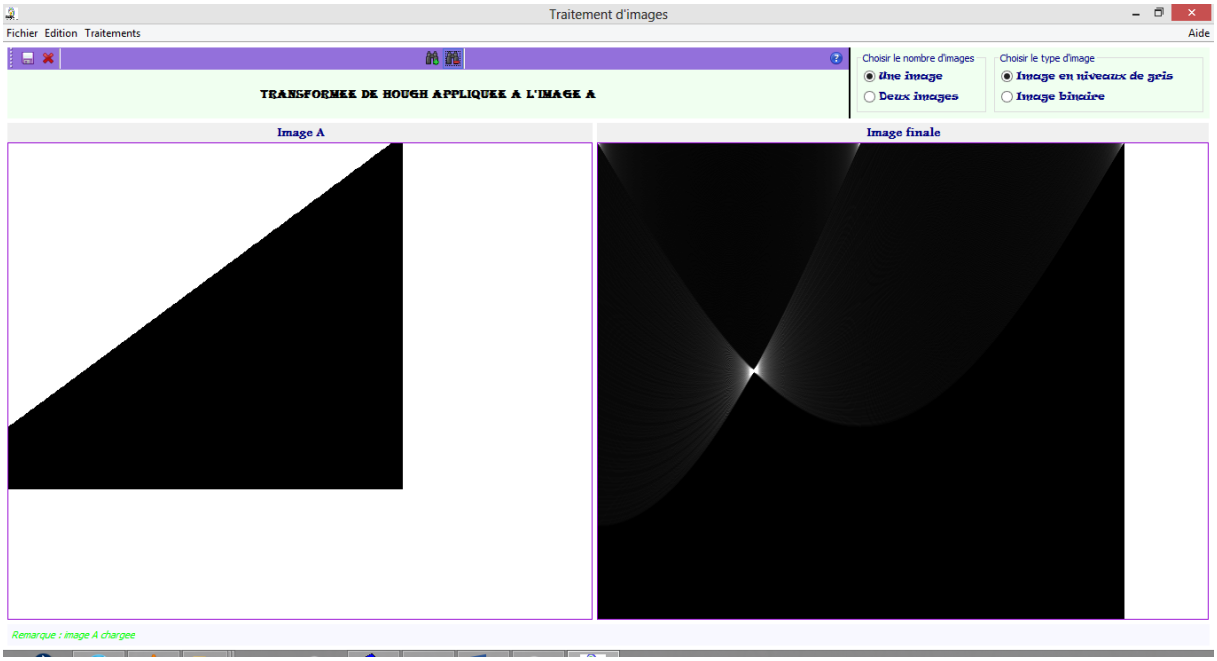

Figure 28: transformée de hough d'une droite

III.4 – Transformée de fourrier, filtre passe haut et passe bas

La transformée de fourrier que nous avons implémenté et les FFT (Fast Fourrier Transform) (figure 29). L'utilité de la transformée de fourrier se voit uniquement lorsqu'on veut faire les filtres passe haut et passe bas.

Vu que les basse fréquences sont au centre et les hautes fréquences sont au bord, le filtre passe haut consiste à supprimer les pixels du centre dans la transformée de fourrier (figure 30), et ensuite de faire

une transformée inverse pour voir le résultat (figure 31). Le filtre passe bas se fait dans les conditions inversées.

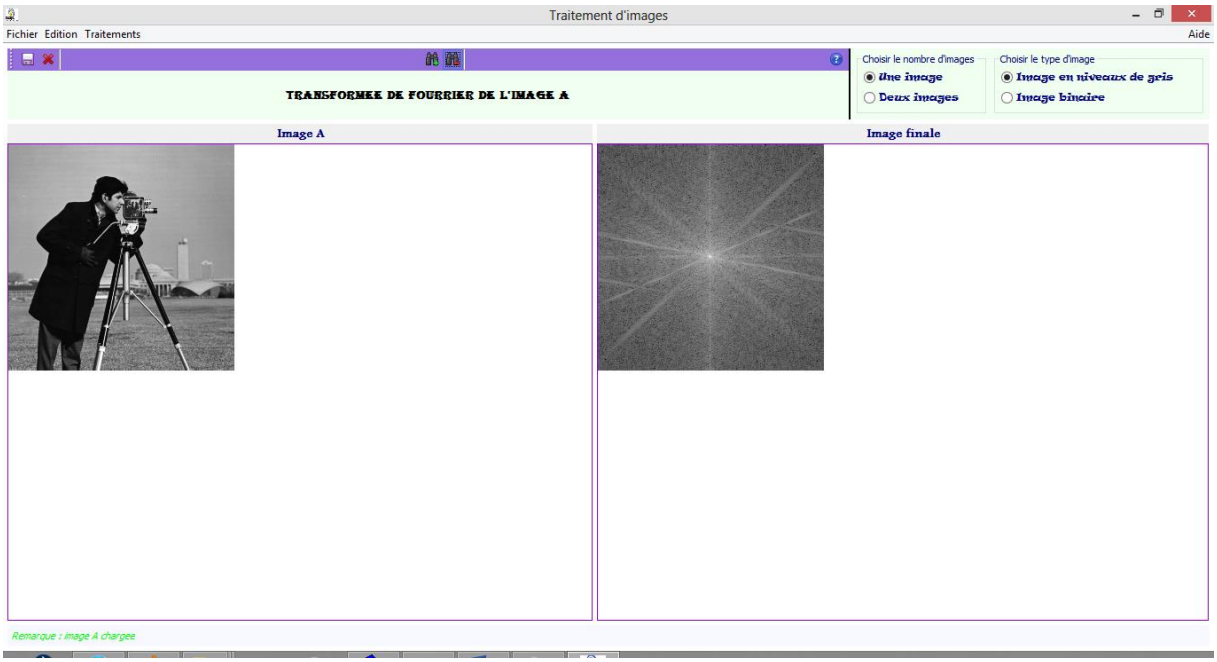

Figure 29: transformée de fourrier d'une image quelconque

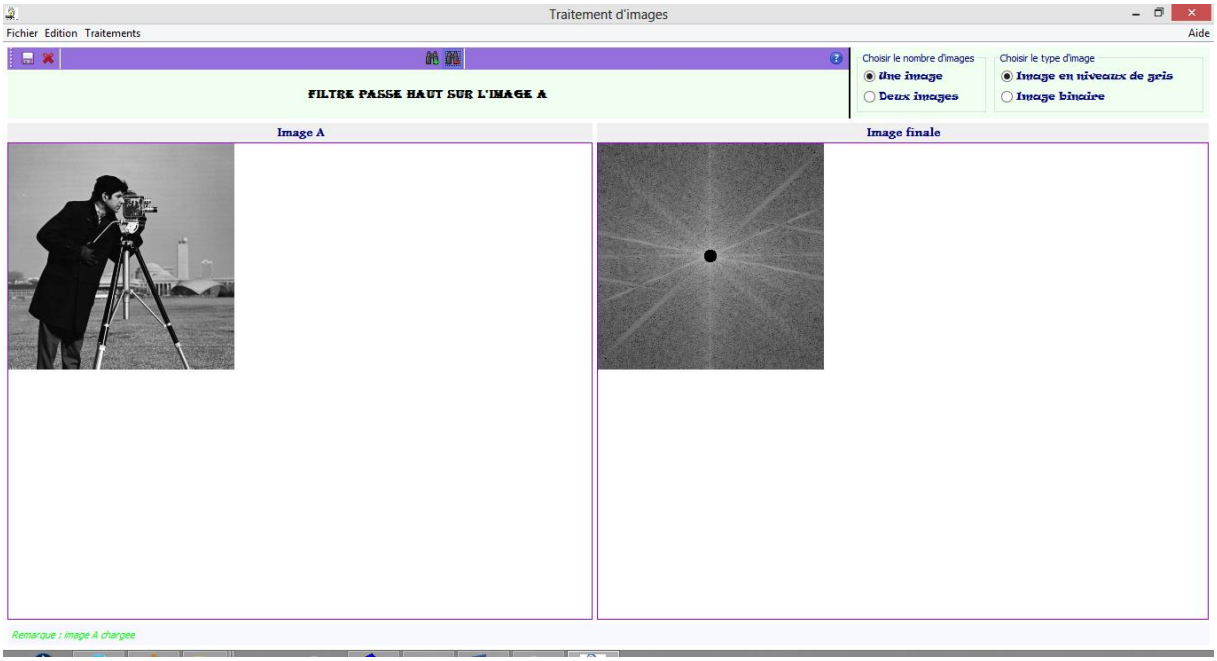

Figure 30: filtre passe haut

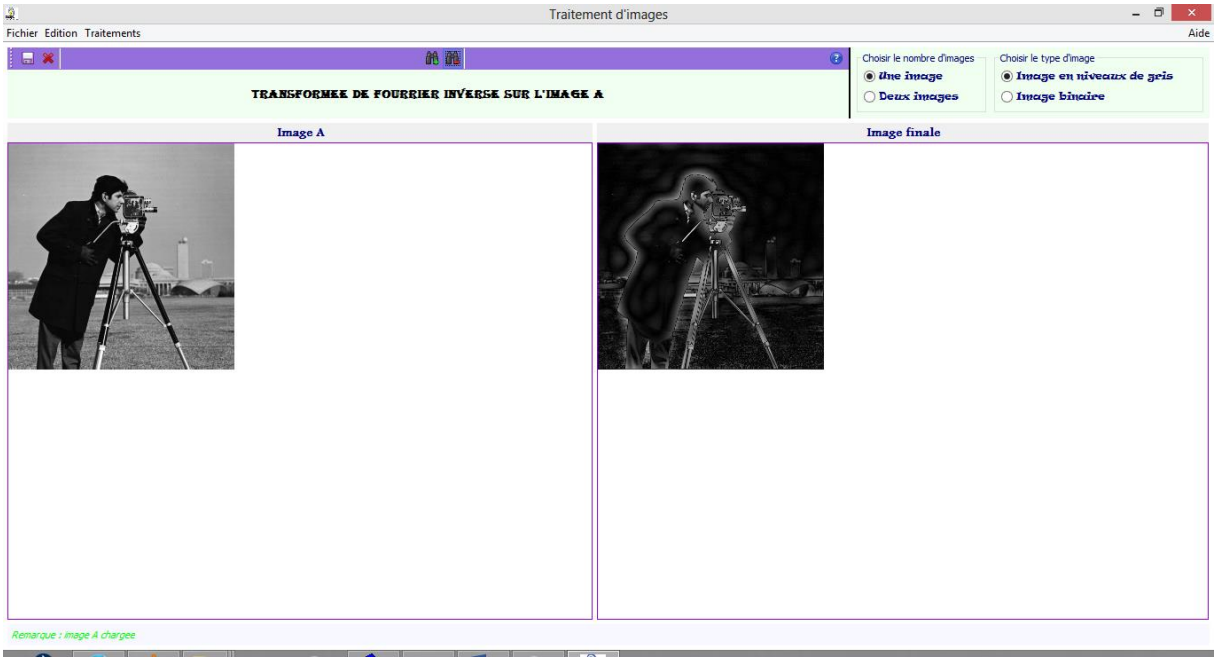

Figure 31: résultante de l'image après le filtrage passe haut

IV- LES IMAGES BINAIRES

Nous finissons avec les images binaires; c'est-à-dire dont la résolution tonale est réduite à deux valeurs (0 et 1). Il existe des fichiers (PBM) qui sont fait pour être manipulés en tant qu'image binaire. Toutefois, vu que nous n'utilisons que les fichiers PGM, on a supposé qu'une image dans laquelle il n'y a que des 0 et des 255 est une image binaire bien que plus gourmande en ressources.

Ainsi les traitements morphologiques sont activés en configurant le panneau de la figure 3.

Une image binaire c'est un objet et un fond. Etant donné la présence de deux valeurs de pixel seulement, le fond prend une couleur et l'objet prend la couleur opposé. Le problème dans les opérations morphologiques est qu'il faut toujours savoir quelle est la couleur de l'objet.

Nous avons donc opté pour une heuristique selon laquelle les pixels majoritaires sur le bord appartiennent au fond. Ainsi, on détermine à chaque fois la couleur de l'objet (blanc ou noir).

IV.1 – L'érosion

L'érosion consiste à dégrader l'image (figure 32). Ici, on utilise un élément structurant (en étoile, en carré, …) pour parcourir l'objet dans l'image. Lorsqu'on se trouve sur un pixel donné de l'objet, si une partir de l'élément structurant touche le fond, alors, ce pixel devient le fond.

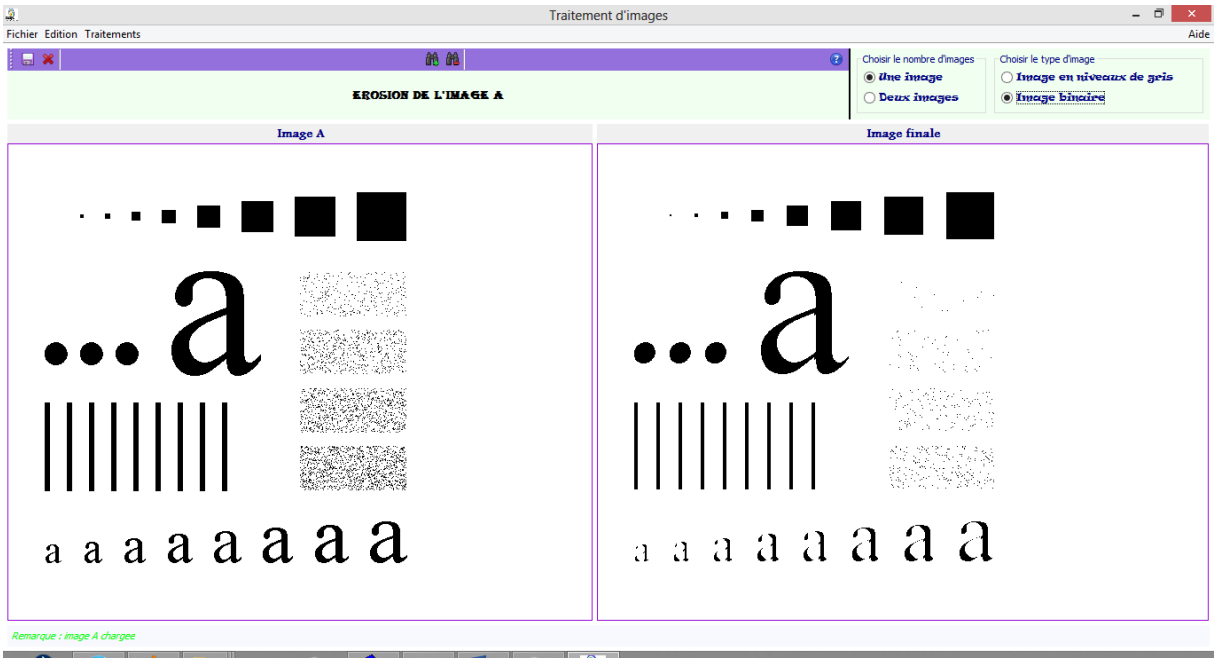

Figure 32: érosion d'une image

IV.2 – Dilatation

Son fonctionnement est similaire à l'érosion, sauf que lorsqu'un pixel de l'élément structurant touche le fond, celui-ci est intégré dans l'objet. La figure 33 en illustre le fonctionnement

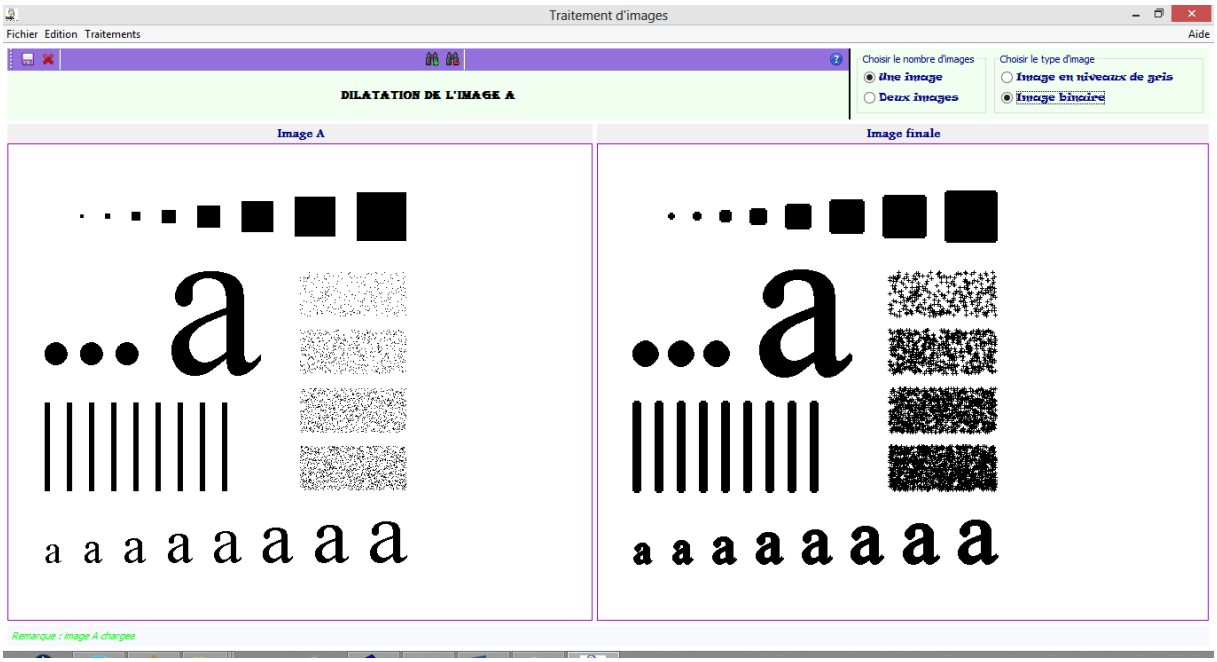

Figure 33: dilatation d'une image

IV.3 – ouverture, fermeture et gradient morphologique

Ces derniers opérateurs ne sont que la combinaison des précédents. L'ouverture c'est l'érosion puis la dilatation, tandis que la fermeture c'est la dilatation puis l'érosion.

Le gradient morphologique quant à lui est la différence entre la dilatation et l'érosion. Voir figure 34.

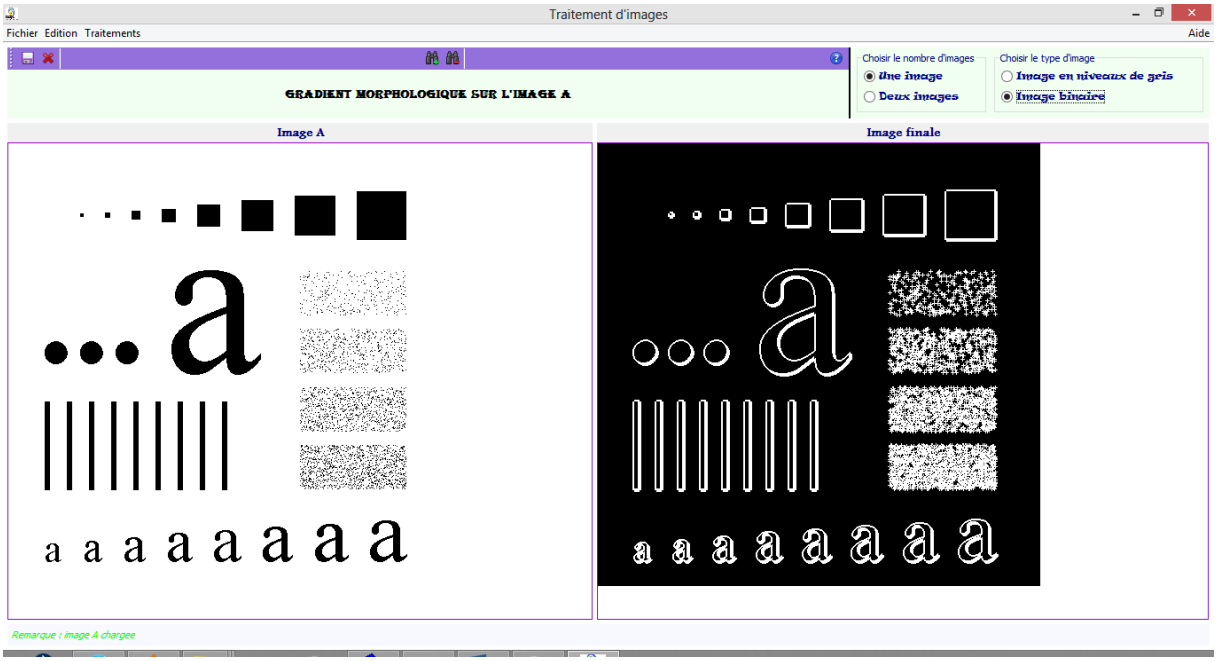

Figure 34: gradient morphologique

CONCLUSION

Parvenus au terme de notre travail, nul doute que ce travail nous a été on ne peut plus édifiante. Dans ce rapport, nous avons tout d'abord présenté les fonctionnalités principales de notre application. Puis nous avons montré comment effectuer quelques traitements ponctuels. Ensuite, nous avons vu comment appliquer les traitements locaux et globaux, et enfin nous avons abordé les images binaires.

Ce travail nous a permis de découvrir les méandres des traitements d'images, et au sortir, on réalise que ce domaine est à la fois fastidieux et passionnant. Nous avons compris l'utilité et le fonctionnement de la plus part des algorithmes de traitement d'image.

ANNEXES

Nous avons fourni dans ce projet :

- Les fichiers "*setup image processing.exe*" qui est le rendu du logiciel développé pouvant être déployé dans toutes les machines Windows disposant d'une assez bonne puissance de calcul.
- Les fichiers sources du logiciel developpé.

REFERENCES

[1] L. Diane, Cours de traitement d'images, Nice, 2004.

[2] TAUVY Alexandre, CARAYON Nicolas et SOISSONS Sébastien, «Egalisation d'histogramme».

[3] G. VINCENT, «Détection efficace des contours,» 2004.

[4] Basile Graf et Christophe Espic, «Transformée de hough».