



Mémoire

pour la validation du

Master 1

Option : Mathématique et applications

Présenté Par :

Achraf KATTASS
Kokou Laris EDJINEDJA
Ayoko AYIKA

THÈME

Introduction au traitement d'images

Soutenu le : 23/06/2021

Sous la direction du : Pr. Yves d'Angelo

Table des matières

1	Introduction	4
2	Les images numériques	5
3	Les types d'image	6
3.1	Les différents aspects du traitement numérique d'image	8
3.2	Traitement ponctuel d'une image	8
3.2.1	Histogramme d'une image	8
3.2.2	Recadrage dynamique – Contraste	9
3.2.3	Rehaussement de contraste	11
3.2.4	Egalisation de l'histogramme	12
3.2.5	Binarisation – Seuillage	14
4	Débruitage Par filtre non linéaire et linéaire	16
4.0.1	Notion de bruit dans une image	16
4.0.2	Modélisation du bruit et du flou	17
4.1	Notion de filtre	17
4.2	Filtrage	18
4.2.1	Principe de filtrage linéaire local	18
4.3	Débruitage par filtre linéaire ou de convolution	19
4.4	Filtre gaussien et optimisation	19
4.4.1	Optimisation du filtre gaussien pour trouver la meilleure valeur de σ qui donne une bonne qualité d'une image .	21
4.4.2	SNR	21
4.4.3	Filtre moyenneur	23
4.5	Débruitage par filtre non linéaire	24
4.5.1	Filtre Médian	24
4.6	Comparaison entre filtre non linéaire et filtre linéaire	26
4.7	Détection du contour d'une image	27
4.7.1	Approximation du gradient	27
4.7.2	Calcul du gradient d'une image	27

4.7.3	Contour avec gradient naïf d'une image	28
4.7.4	contour avec gradient de l'image	28
4.7.5	Masque sobel	29
4.7.6	Masque préwitt	30
4.7.7	Contour avec approximation de la dérivée seconde d'une image	30
4.7.8	Laplacien discret -4	31
4.7.9	Laplacien discret -8	31
5	Conclusion et quelques applications du traitement numérique d'image	33
	Bibliographie	34

Chapitre 1

Introduction

Une image est une représentation visuelle, voire mentale d'un objet, d'un être vivant ou d'un concept.

Une image numérique c'est toute image (dessin, photographie...) acquise, créée, traitée et stockée à l'aide des nombres sous forme binaire. Afin de développer des méthodes d'analyse, de traitement de l'information et de prise de décision, des mathématiques avancées dans le traitement du signal doivent être utilisées.

Le traitement numérique d'image est donc une discipline combinant informatique et mathématiques appliquées pour étudier les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information. Le traitement d'image se place généralement après les étapes d'acquisition et de numérisation.

Le traitement d'image commence à être étudié dans les années 1920 pour la transmission d'images par le câble sous-marin allant de New York à Londres. Harry G. Bartholomew et Maynard D. McFarlane effectuent la première numérisation d'image avec compression de données pour envoyer des fax de Londres à New York. Le véritable essor du traitement d'images n'a lieu que dans les années 1960 quand les ordinateurs commencent à être suffisamment puissants pour travailler sur des images. Dès lors, un véritable engouement se fait jour pour le traitement de l'image et surtout pour la compression de l'image par des systèmes experts. A partir des années 1990, il y a une amélioration constante des opérateurs et l'on réussit à appliquer le traitement d'image à de nombreux domaines comme la médecine pour améliorer les diagnostics faits à partir des techniques d'imagerie médicale (IRM).

Dans notre étude, nous nous initierons aux traitements de base d'une image numérique.

Chapitre 2

Les images numériques

Il existe deux types d'images numériques : les images matricielles (ou bitmap) et les images vectorielles. Dans le cadre de notre étude, nous nous intéresseront uniquement aux images matricielles.

Une image matricielle est composée d'une matrice (tableau) de points à plusieurs dimensions, chaque dimension représentant une dimension spatiale, temporelle ou autre (un niveau de résolution par exemple). Nous travaillerons avec les images bidimensionnelles ou 2D. La variable est une variable d'espace représentant les deux coordonnées (x, y) d'un point du plan de l'image.

La représentation informatique (codage) d'une image est sa numérisation dans deux espaces :

1. L'espace spatial où l'image est numérisée suivant l'axe des abscisses et des ordonnées : c'est l'échantillonnage. Les échantillons sont nommées pixels (picture element). L'échantillonnage d'une image en pixels c'est extraire une valeur en chaque position.
2. L'espace des couleurs où toutes les valeurs de luminosité que peut prendre un pixel sont numérisées pour représenter sa couleur et son intensité : c'est la quantification. Autrement dit, la quantification des pixels dans une image numérique est le choix de la valeur qui leur est attribuable parmi les valeurs possibles.

Chapitre 3

Les types d'image

On distingue plusieurs types d'images :

1. **Les images binaires**

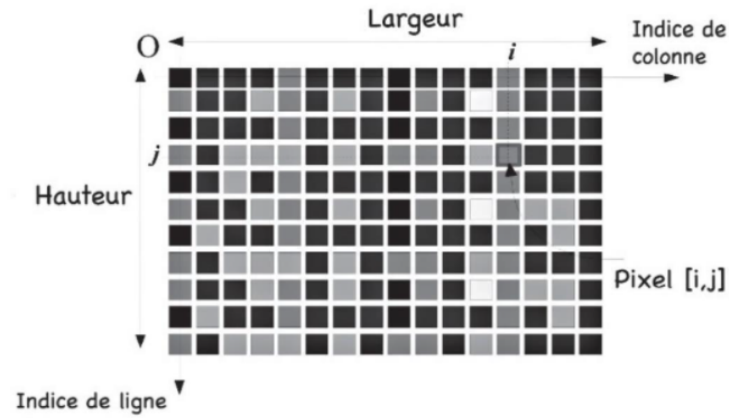
Pour ce type d'image, un pixel peut prendre uniquement les valeurs noir ou blanc. C'est le type d'image que l'on utilise pour numériser du texte quand celui-ci est composé d'une seule couleur.

2. **Les images couleur**

Pour ce type d'image, un pixel est codé par trois valeurs numériques qui dépendent du type de codage utilisé. Le plus utilisé est l'espace de couleur Rouge-Vert-Bleu (RVB).

3. **Les images en niveau de gris**

Ce sont des images qui renferment 256 teintes de gris. La valeur 0 représente le noir (intensité lumineuse nulle) et la valeur 255 est associée au blanc (intensité lumineuse maximale). Chaque entier représentant un niveau de gris est codé sur 8 bits. Il est donc compris entre 0 et $(2^8) - 1$. L'image est donc capable d'afficher 256 nuances de gris différentes.

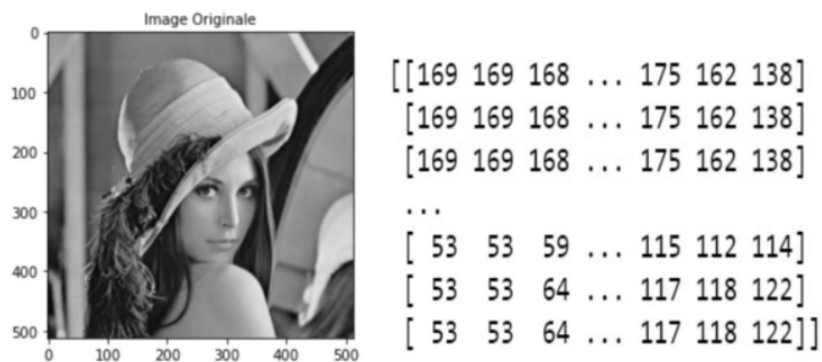


1.png

Une image du point de vue mathématique est une fonction définie sur un sous ensemble de \mathbb{Z}^2 à valeur dans \mathbb{R} et cet sous ensemble n'est que le produit cartésien de la taille de l'image.

Soit I une image numérique. Alors, si on note N_1 le nombre le lignes de I et N_2 le nombre de colonnes, on dira que I est une image $N_1 * N_2$ et le nombre total de pixels de I sera $N = N_1 * N_2$, nous avons aussi $\forall (i, j) \in [0, N_1 - 1,] * [0, N_2 - 1]$ $I[i, j] = n$ où $n \in [0, 255]$, n est donc la valeur du niveau de gris du pixel.

Dans notre étude, nous travaillerons uniquement avec les images en niveau de gris et nous nous servirons de l'image de Lenna pour effectuer nos traitements. Il s'agit d'une image 512*512. La conversion de l'image en matrice nous donne le résultat suivant :



Si on note I notre image numérique, nous aurons par exemple $I[0, 0] = 169 =$ valeur du niveau de gris du premier pixel de l'image.

Notion d'acquisition

Le principe de l'acquisition d'une image numérique consiste à transformer des quantités physiques (ou chimiques) en nombres que l'on stockera dans des tableaux.

On peut distinguer les procédés d'acquisition des images suivants :

1. Le système visuel
2. La photographie argentique
3. La photographie numérique
4. La numérisation des textes et des images
5. L'imagerie médicale
6. La radiographie
7. (...)

3.1 Les différents aspects du traitement numérique d'image

Le traitement d'images nécessite des outils nombreux et variés. Nous commencerons par effectuer des opérations élémentaires sur les images (éclaircissement, contraste ...) et nous étudierons les principales étapes du traitement d'image qui sont : le filtrage, la segmentation et la restauration.

3.2 Traitement ponctuel d'une image

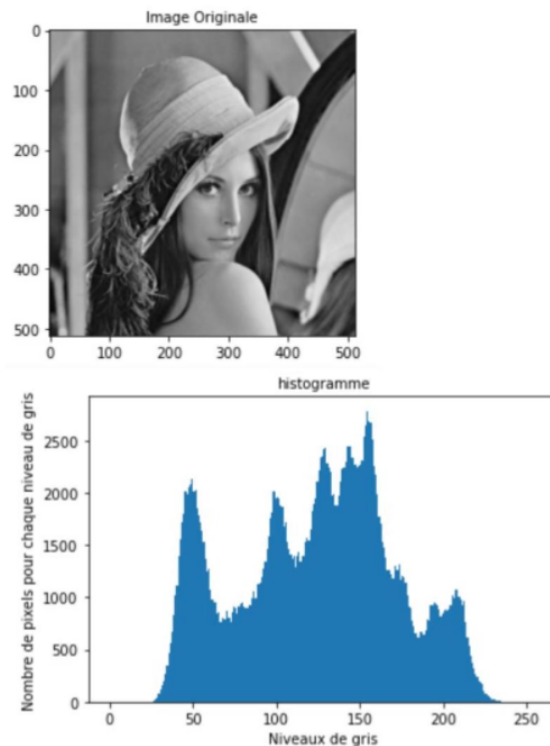
Dans cette partie, nous étudierons le traitement ponctuel d'images ainsi que les principales étapes du traitement d'image.

3.2.1 Histogramme d'une image

L'histogramme d'une image mesure la distribution des niveaux de gris dans l'image. Pour un niveau de gris g , on compte le nombre de pixels ayant la

3.2. TRAITEMENT PONCTUEL ~~CHAPITRE~~ LES TYPES D'IMAGE

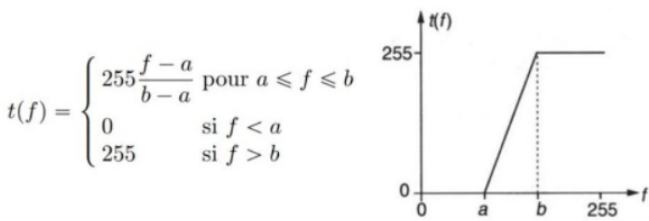
valeur g . Formellement, c'est une fonction discrète définie par : pour tout p dans $0, \dots, 255$, h_p = Nombre de pixels ayant p pour niveau de gris. L'histogramme permet d'obtenir rapidement une information générale sur l'apparence de l'image mais ne constitue pas une caractéristique de l'image. Ainsi, deux images peuvent avoir le même histogramme. Une image « visuellement satisfaisante » aura généralement un histogramme équilibré (proche d'une fonction plate). Dans la suite, nous verrons quelques procédés permettant d'améliorer l'aspect visuel d'une image.



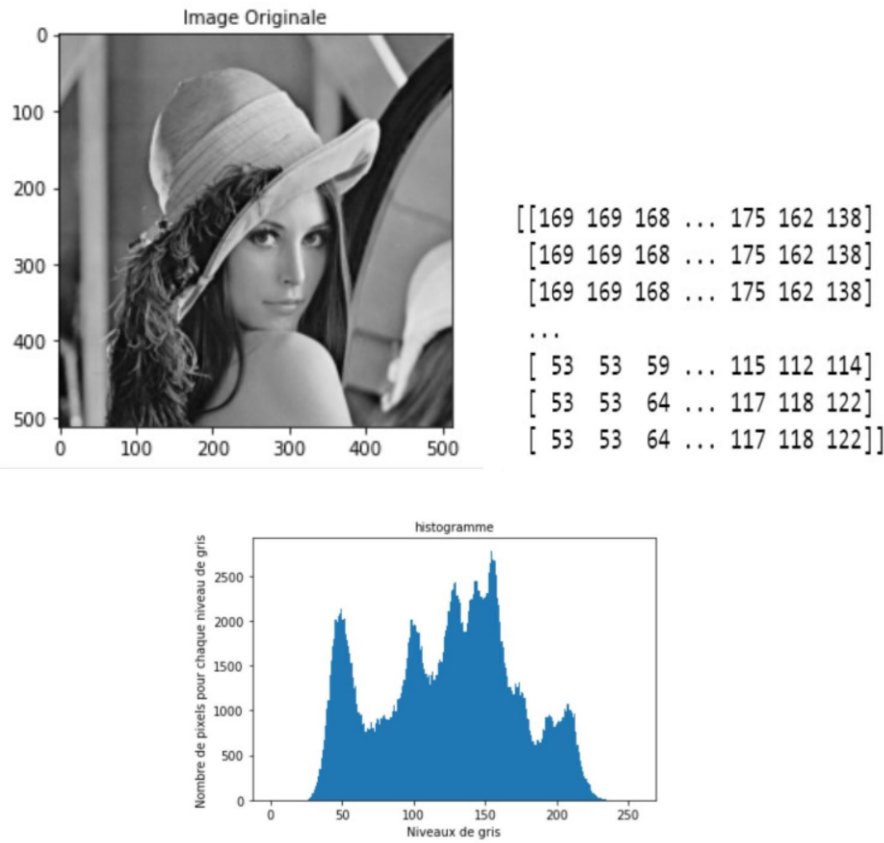
3.2.2 Recadrage dynamique – Contraste

Le contraste est une transformation qui permet de modifier la dynamique des niveaux de gris dans le but d'améliorer l'aspect visuel de l'image. A un niveau de gris f de l'image originale correspond le niveau $t(f)$ dans l'image transformée. Soit une image présentant un histogramme concentré dans l'intervalle $[a, b]$ où a et b représentent les niveaux de gris extrêmes présents dans l'image. Le recadrage dynamique dans ce cas consistera à étendre la dynamique de l'image transformée à l'étendue totale $[0, 255]$ de la façon suivante :

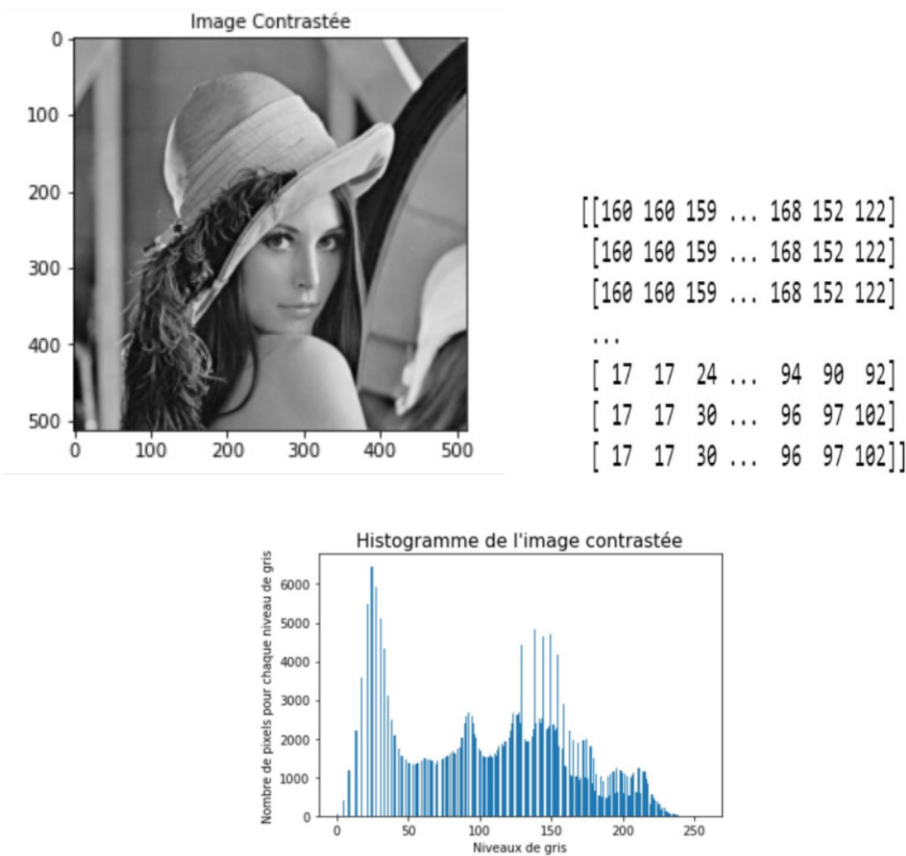
3.2. TRAITEMENT PONCTUEL CHAQUE PIXEL SEULEMENT LES TYPES D'IMAGE



Nous obtenons le résultat suivant :



3.2. TRAITEMENT PONCTUEL CHAQUE PIXEL SEULEMENT UN FOIS



3.2.3 Rehaussement de contraste

Le rehaussement des contrastes permet d'accentuer le contraste dans un intervalle précis de niveau suivant la fonction :

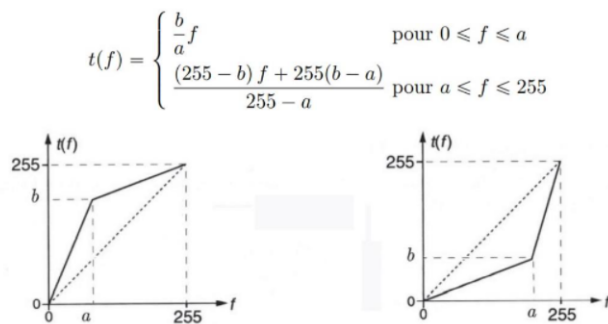


Figure 4-1- Fonction de rehaussement de contraste :dilatation de la dynamique des zones sombres (à gauche) et des zones claires (à droite)

3.2. TRAITEMENT PONCTUEL ~~CHANTRE~~ ~~TRIA~~GELES TYPES D'IMAGE

Nous obtenons les résultats suivants :

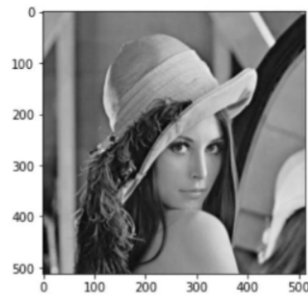


Figure 4-2- Image originale

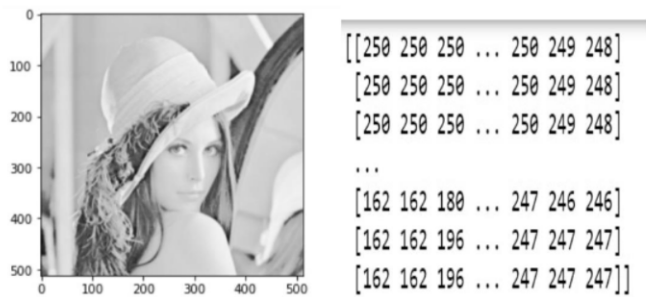


Figure 4-3- Image rehaussée (dilatation de la dynamique des zones claires)

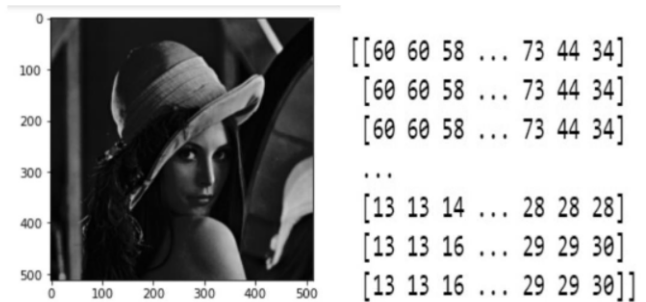


Figure 4-4- Image rehaussée (dilatation de la dynamique des zones sombres)

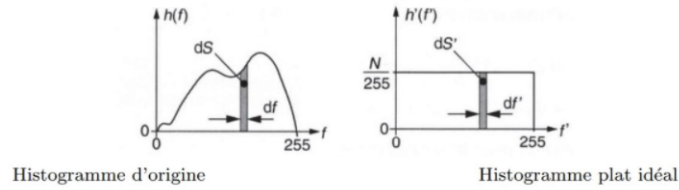
3.2.4 Egalisation de l'histogramme

Comme vu précédemment, une image « visuellement satisfaisante » a généralement un histogramme proche d'une fonction plate. Mais, force est de constater que l'histogramme d'une image est rarement plat. L'égalisation d'histogramme

3.2. TRAITEMENT PONCTUEL D'IMAGES LES TYPES D'IMAGE

consiste à corriger une image qui manque de contraste. Cette transformation permet donc d'utiliser toute l'étendue des niveaux de gris et d'avoir pratiquement autant de pixels de chaque niveau.

Considérons l'histogramme continu $f \mapsto h(f)$. En notant $f' = t(f)$, l'histogramme égalisé $f' \mapsto h'(f')$ doit s'approcher de la forme idéale décrite ci-dessous :



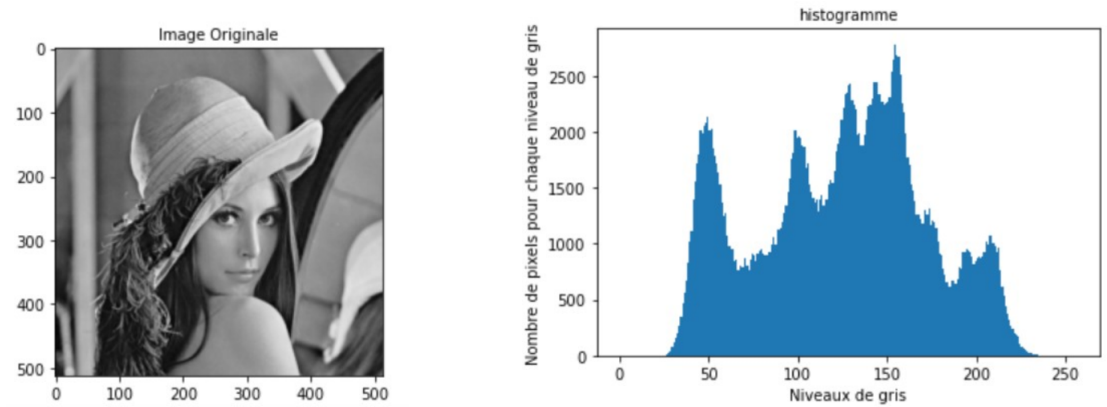
Deux surfaces élémentaires en correspondance dans ces histogrammes initiaux et égalisés, présentent le même nombre de points, ce qui permet d'écrire :

$$f' = t(f) = \frac{255}{N} \int_0^f h(s) ds$$

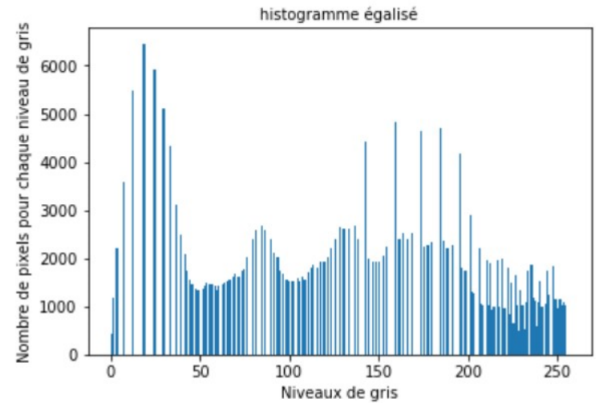
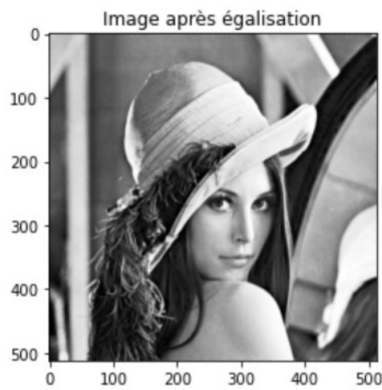
N représente le nombre total de pixels de l'image et h/N est l'histogramme normalisé. La transformation d'égalisation discrète est donné par :

$$f' = t(f) = \frac{255}{N} \sum_{i=0}^f h_i$$

Nous obtenons donc les résultat suivant :



3.2. TRAITEMENT PONCTUEL CHAQUE TYPE D'IMAGE



3.2.5 Binarisation – Seuillage

Le but de la binarisation d'une image est d'affecter un niveau uniforme aux pixels pertinents et d'éliminer les autres. Le seuillage consiste à affecter le niveau 255 aux pixels dont la valeur est supérieure à un seuil S et le niveau 0 aux autres. La valeur du seuil peut être déterminé manuellement ou automatiquement à partir de l'histogramme. Pour la détermination manuelle, nous fixerons la valeur du seuil à un minimum local de l'histogramme. Pour la détermination automatique, nous utiliserons la méthode d'Otsu .

Valeur du seuil avec la méthode d'Otsu : 124

On a le résultat suivant :

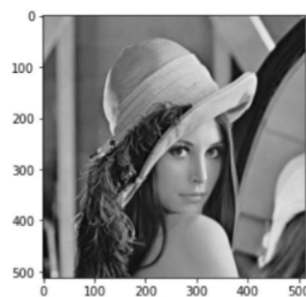


Figure 4-5- Image originale

3.2. TRAITEMENT PONCTUEL D'ANIMATIONS TYPES D'IMAGE

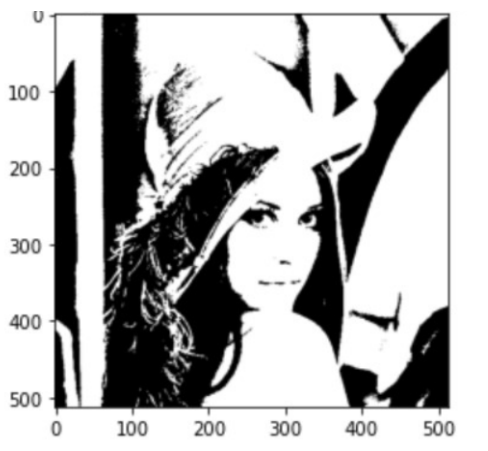


Figure 4-6- Image après seuillage par la méthode d'Otsu (S=124)

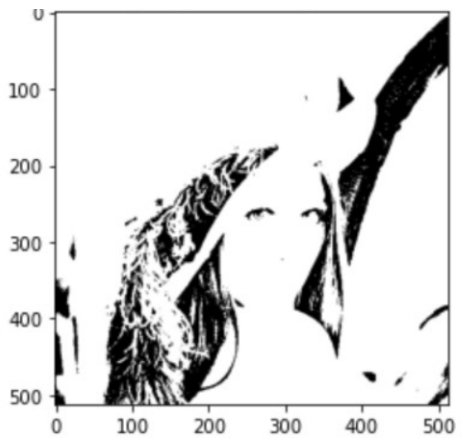
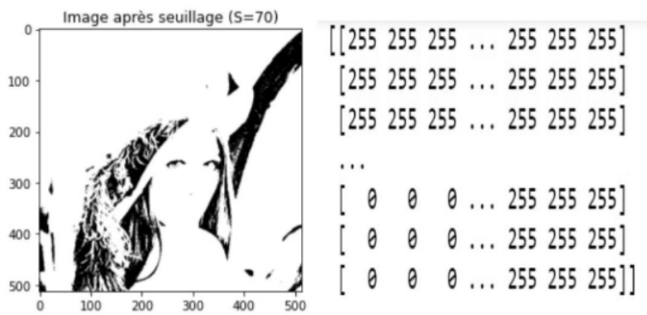


Figure 4-7- Image après seuillage pour S = 70



Chapitre 4

Débruitage Par filtre non linéaire et linéaire

4.0.1 Notion de bruit dans une image

Dans toute image numérique on peut observer des anomalies. Ces anomalies sont souvent les tâches de couleurs quelconques qui détruisent la qualité de l'image et la rendent illisible.

Cette anomalie est due à la défaillance du capteur de l'appareil utilisé pour la prise de l'image.

Cette défaillance se traduit par la perturbation que subit le capteur à compter la valeur de couleur en recevant et subissant des fluctuations électrostatique à cause de leurs charges et décharges.

On a deux types de bruit d'image :

1. Bruit de chrominance qui est la composante colorée des pixels bruités. Il est visible sous forme de tâches de couleurs aléatoires.
2. Bruit de luminance qui est la composante lumineuse des pixels bruités. Il est visible sous la forme de tâches plus foncées ou plus, claires donnant un aspect granuleux à l'image.

On distingue plusieurs sortes de bruit tels que bruit poivre et sel, bruit thermique et bruit de grenaille

1. Bruit poivre et sel

Le bruit poivre et sel ou bruit impulsionnelle est un bruit qui assigne à un certains nombres de pixels de l'image, une valeur 0 ou 255 aléatoirement.

Pour le représenter il suffit de choisir un pixel aléatoirement et de remplacer la valeur de niveau de gris par 0 pour le bruit poivre et 255 pour le bruit sel.

2. Bruit thermique

Le bruit thermique est dû à l'agitation naturelle des électrons qui s'augmente avec la température de capteur.

flou

Le flou est un caractère de l'image qui la rend peu nette.

4.0.2 Modélisation du bruit et du flou

Notons par U_b une image bruitée et U une image originale. Dans le cas des bruits additifs le bruit b est mathématiquement défini comme le reste entre U_b et U tel que $U_b - U = b$, et dans le cas multiplicatif par un rapport entre U_b et U diminué de 1 tel que $U_b = U * (1 + b)$.

Si b est gaussien alors on parle de bruit gaussien.

Par ailleurs, il existe d'autres types de bruits tels que :

- Le bruit de Bernoulli (à valeurs dans 0 et 1).
- Le bruit uniforme entre 0 et 255.

Pour modéliser le bruit gaussien il suffit pour chaque valeur de pixel de l'image originale prise aléatoirement On ajoute une valeur aléatoire gaussienne, de moyenne 0 et de variance σ^2 qui détermine la variance entre l'image originale et l'image bruitée. La méthode est pareille pour la modélisation des bruits uniformes et de Bernoulli qui sont additifs. Cette processus est valable pour les bruits multiplicatifs sauf qu'au lieu de faire une addition on fait une multiplication.

4.1 Notion de filtre

Pour définir un filtre, définissons deux espaces vectoriels X (entrée) et Y (sortie) munis d'une topologie (Par exemple espace vectoriel normé) et un opérateur A linéaire qui à un signal $e \in X$ dit signal d'entrée (input) associe un signal $s \in Y$ appelé signal de sortie (output) $A : e \mapsto s$.

Par définition un filtre linéaire est formé des espaces d'entrée et de sortie et

d'un opérateur linéaire continu qui vérifie les deux propriétés suivantes :

1. Invariance temporelle
si $T_a : x \mapsto x \pm a$ est l'opérateur de translation, alors $T_a A = A T_a$,
c'est à dire que les deux opérateurs sont commutatifs.
2. Causalité (ou réalisabilité) :
Si pour tout la donné on a la propriété suivante :
 $\forall t < t_a \quad x_1(t) = x_2(t) \Rightarrow t < t_a$, alors $Ax_1(t) = Ax_2(t)$.

Il existe plusieurs types de filtre.

Filtres passe bas

Les filtres passe bas éliminent fortement les hautes fréquences et ne laisse passer que les basses fréquences.

Les effets des filtres passe bas :

- Augmentation du flou des images.
- Les détails disparaissent ou sont moins visibles.
- Les contrastes dans l'image sont diminués.

Filtres passe haut

Les filtres passe haut éliminent fortement l'énergie des basses fréquences d'un spectre. Ils sont utilisés dans la détection du contour pour la netteté et augmente le contraste.

Filtres passe haut

Le filtre coupe bande qui est le complément du précédent.

4.2 Filtrage

4.2.1 Principe de filtrage linéaire local

Dans le traitement de signal, le filtrage est défini comme une convolution. Par contre, dans le domaine de traitement d'image cette opération est définie comme une corrélation, en pratique cette différence est traduite par une rotation de 180 degrés de la matrice représentante du filtre. Pour pouvoir faire la convolution dans le cadre de traitement d'image il suffit de faire subir au filtre une symétrie de 180 degrés autour de l'origine.

CHAPITRE 4. DÉBRUITAGE PAR FILTRE NON LINÉAIRE ET 4.3. DÉBRUITAGE PAR FILTRE LINÉAIRE OU DE CONVOLUTION

La convolution d'une image avec un filtre de convolution ou masque est une technique pour supprimer les petites perturbations subies par une image tout en gardant sa qualité (contour dynamique, texture, etc...).

Dans cette partie de notre rapport nous allons présenter les techniques linéaires de base pour effectuer un filtrage permettant de supprimer les effets d'un bruit additif.

Soit f une image à filtrer et K un noyau de convolution ou PSF (Point spread function). Alors dans le cas (2D) la convolution dans un espace continue se traduit par :

$$(f * K)(x, y) = f * K|_{x,y} = \int \int_{Voisinage} f(x - u, y - v) K(u, v) du dv$$

Par contre si c'est une convolution dans un espace discret cette opération se traduit par :

$$(f * K)(x, y) = f * K|_{x,y} = \sum_{i,j \in Voisinage} f(x - i, y - j) K(i, j)$$

Pour faire notre convolution, on choisit un noyau dont le support est petit de manière à minimiser le coût lors de l'implémentation.

Ce noyau nous permettra d'effectuer les opérations de convolutions au voisinage de chaque pixel.

4.3 Débruitage par filtre linéaire ou de convolution

4.4 Filtre gaussien et optimisation

Convolution avec un noyau gaussien

Considérons un masque de taille $3 * 3$ représenté par une fonction gaussienne

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$G(-1, -1)$	$G(0, -1)$	$G(1, -1)$
$G(-1, 0)$	$G(0, 0)$	$G(1, 0)$
$G(-1, 1)$	$G(0, 1)$	$G(1, 1)$

Alors la convolution entre f et notre noyau gaussien revient à faire l'opération suivant pour chaque pixel $f(x, y)$.

$$g_{x,y} = (f * G)(x, y) \sum_{i=-1}^1 \sum_{j=-1}^1 f_{x-i, y-j} G_{i,j}$$

avec g la transformée de f par convolution avec G .

$$\begin{aligned} g_{x,y} &= (f * G)(x, y) \sum_{i=-1}^1 \sum_{j=-1}^1 f_{x-i, y-j} G_{i,j} \\ &= G(0, 0)f_{x,y} + G(0, 1)f_{x,y-1} + G(0, -1)f_{x,y+1} + G(1, 0)f_{x-1,y} + G(1, 1)f_{x-1,y-1} \\ &\quad + G(1, -1)f_{x-1,y+1} + G(-1, 0)f_{x+1,y} + G(-1, 1)f_{x+1,y-1} + G(-1, -1)f_{x+1,y+1} \end{aligned}$$

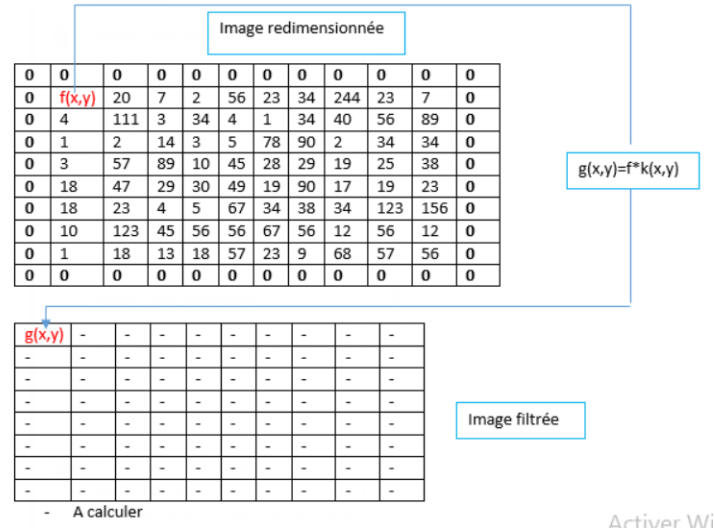
. Afin de conserver la moyenne de l'image f (la luminosité) la somme des éléments du filtre est normalisée à 1.

$$G(0, 0) + G(1, 0) + G(-1, 0) + G(-1, -1) + \dots = 1$$

On remarque que si nous appliquons notre masque sur tout les pixels, nous aurons un problème de bord, puisque $f(-1, 0)$ est n'est pas définie. Pour remédier à ce problème nous allons redimensionner l'image en mettant des valeurs nulles aux bords de l'image. Cette technique prends le nom de " prolongement dans les zéros ".

Le prolongement dans les zéros est une alternative qui suppose que l'image est définie sur un support infini et prend des valeurs nulles en dehors du support initial.

Parmi tant d'autres solutions théoriques optimales pour résoudre les problèmes du bord on peut citer aussi la solution alternative du **résultat tronqué**. son principe est de ne retenir que les pixels pour lesquels les calculs peuvent être faits, ce qui induit la perte de deux lignes et de deux colonnes



ce type de filtre utilisé est un filtre passe bas.

4.4.1 Optimisation du filtre gaussien pour trouver la meilleure valeur de σ qui donne une bonne qualité d'une image

Dans cette partie nous allons optimiser notre méthode de débruitage en utilisant le rapport signal sur bruit SNR.

4.4.2 SNR

Par définition, le SNR d'un signal donné x entaché d'un bruit b sur un intervalle de temps finis I est la quantité.

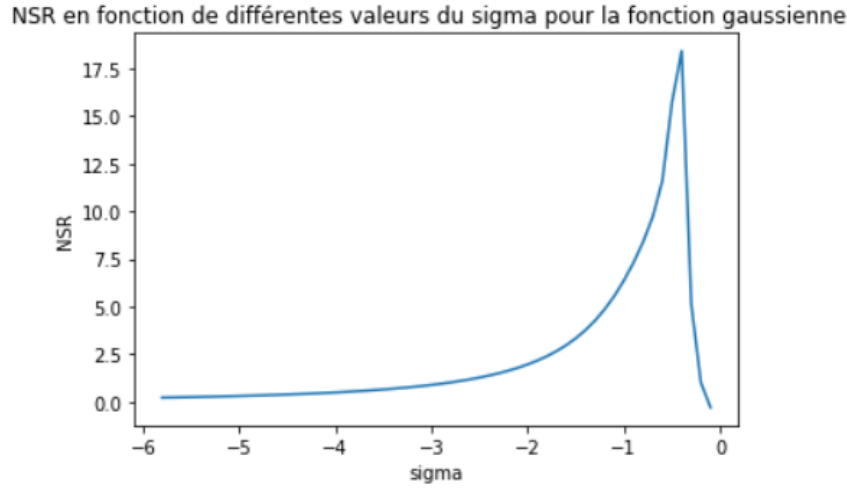
$$SNR = 20 * \log_{10} \left(\frac{\|x\|}{\|b\|} \right)$$

On désigne par $\|\cdot\|$ la norme euclidienne du signal discret.

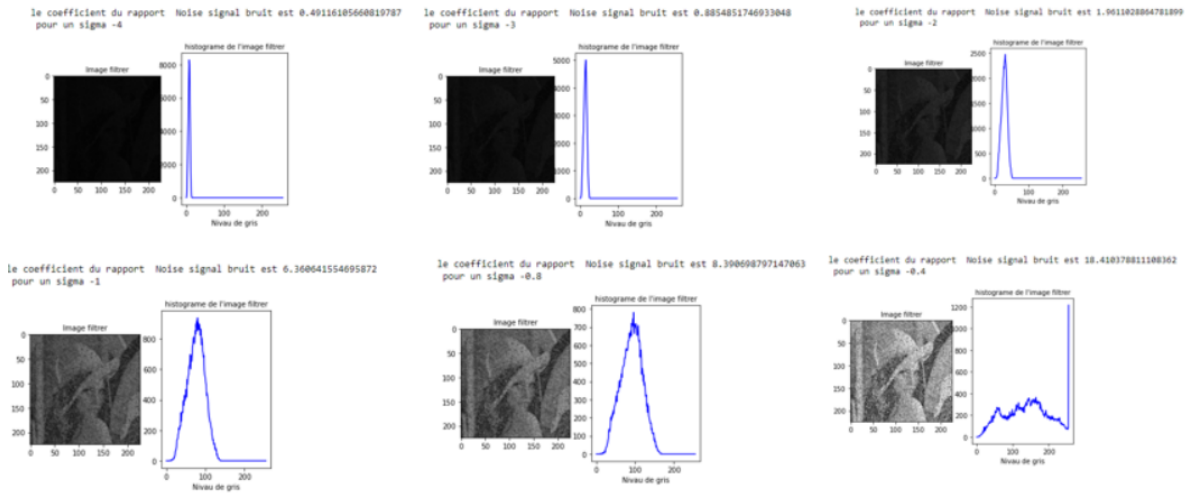
Comme dans notre cas on travaille sur une image, considérons U une image original (image restaurée) et U_b l'image bruitée alors le bruit additif est $U_b - U$. Notre SNR s'écrit $SNR = 20 * \log_{10} \left(\frac{\|U_b\|}{\|U_b - U\|} \right)$, on constate que plus le rapport $\left(\frac{\|U_b\|}{\|U_b - U\|} \right) \Rightarrow \infty$ plus notre image bruitée devient plus nette. Dans le cadre de la comparaison entre le filtre non linéaire et le filtre linéaire nous allons prendre dans le cas du débruitage la catégorie de filtre qui a un SNR plus élevé.

CHAPITRE 4. DÉBRUITAGE PAR FILTRE NON LINÉAIRE ET 4.4. FILTRE GAUSSIEN ET OPTIMISATION LINÉAIRE

faisons varier le sigma et calculons le rapport SNR, on obtient la courbe ci-dessous :

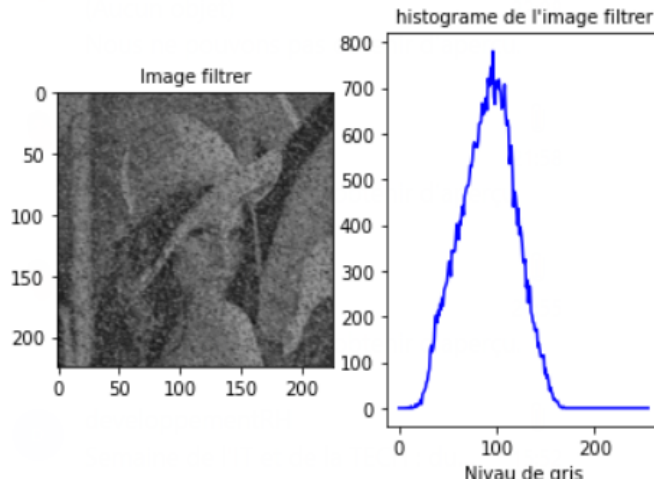


Pour les différentes valeurs de sigma on obtient les images débruitées suivantes :



On constate que quand le SNR tends vers zéro, l'image devient toute noire. Par contre, quand on prend le sigma qui donne le maximum de SNR, l'image n'est pas totalement débruitée. Pour le choix de la qualité de l'image, nous allons choisir $\sigma = -0.8$.

le coefficient du rapport Noise signal bruit est 8.390698797147063
pour un sigma -0.8



4.4.3 Filtre moyenneur

Le filtre moyenneur est aussi un filtre linéaire car il est caractérisé par un noyau.

Soit h le noyau du filtre moyenneur alors pour pouvoir faire la convolution avec f on choisit,

$$h = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Comme dans le cas du filtre gaussien chaque valeur de niveau de gris composant notre image sera remplacée par la convolution avec le noyau h en considérant un voisinage de taille 3×3 pris autour de la valeur de niveau de gris.

Afin de conserver la moyenne (la luminosité) de l'image f , on va plutôt normaliser le noyau h avant de faire la convolution. Typiquement, cela revient à remplacer chaque valeur de niveau de gris composant notre image par la valeur moyenne des valeurs pris au voisinage du pixel $f(x, y)$.

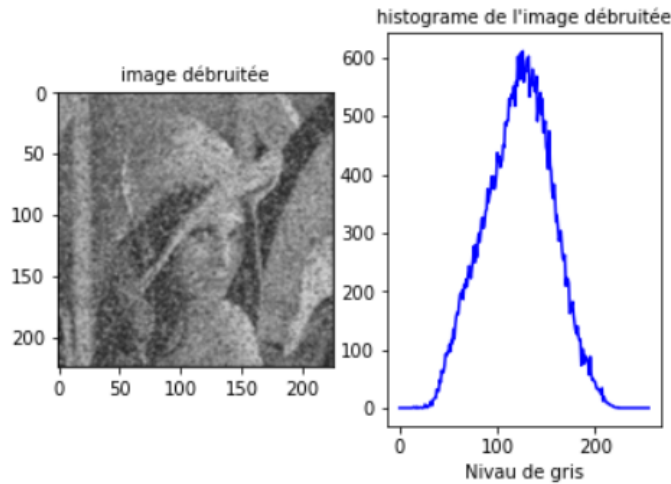
Le noyau utilisé maintenant est le noyau que l'on note

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Résultat après application du filtre moyennneur :

Le coefficient du rapport Noise signal bruit est 8,31571078543007.

le coefficient du rapport Noise signal bruit est 9.073119013445472



4.5 Débruitage par filtre non linéaire

4.5.1 Filtre Médian

Considérons une image bruitée dont la matrice qui lui est associée est f . Le filtre médian est un filtre numérique non linéaire qui a pour but de réduire les bruits d'une image.

L'idée générale dans cette section est de remplacer chaque entrée $f(x, y)$ par la valeur médiane en considérant les valeurs au voisinage 3×3 du pixel $f(x, y)$. Par exemple si nous découpons une portion de valeur de niveau de gris de taille 3×3 de notre image.

CHAPITRE 4. DÉBRUITAGE PAR FILTRE NON LINÉAIRE ET 4.5. DÉBRUITAGE PAR FILTRE NON LINÉAIRE LINÉAIRE

10	20	7	2	56	23	34	244	23	7
4	111	3	34	4	1	34	40	56	89
1	2	14	3	5	78	90	2	34	34
3	57	89	10	45	28	29	19	25	38
18	47	29	30	49	19	90	17	19	23
18	23	4	5	67	34	38	34	123	156
10	123	45	56	56	67	56	12	56	12
1	18	13	18	57	23	9	68	57	56

10	20	7
14	111	3
1	2	14

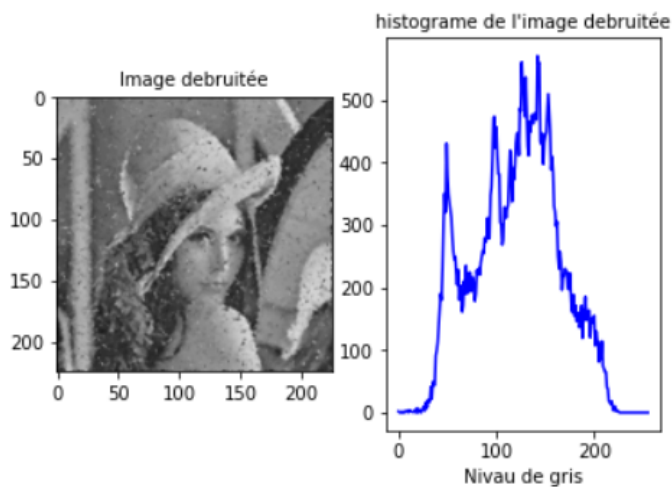
Notre entrée $f(x,y)=111$ serait remplacer par la valeur médiane des valeurs prises au voisinage du pixel $f(x,y)$ rangés dans l'ordre croissant

1	2	3	7	10	14	14	20	111
---	---	---	---	----	----	----	----	-----

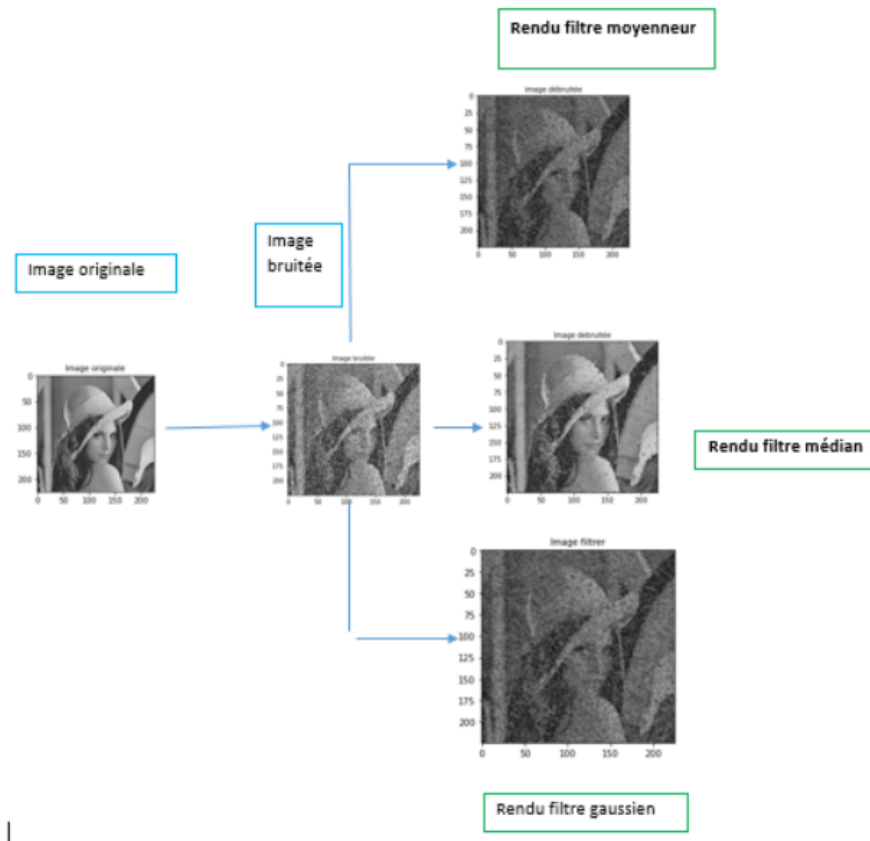
Donc $f(x,y)$ sera remplacé par 10

Comme au niveau du filtre gaussien en considérant un voisinage de 3×3 cela nous conduit à un problème de bord. Pour pallier à ce problème nous allons simplement surdimensionner la matrice de l'image en rajoutant des valeurs nulles pour pouvoir faire le calcul et après couper la vraie image.

le coefficient du rapport Noise signal bruit est 8.604991835680927



4.6 Comparaison entre filtre non linéaire et filtre linéaire



— Dans le cas Filtre Moyenneur et filtre médian

Du point de vue qualité de l'image, le filtre médian est plus adapté que le filtre linéaire (Filtre Moyenneur) pour réduire le bruit.

Dans ce cas il est préférable d'utiliser le filtre non linéaire (Filtre Médian)

— Dans le cas du Filtre Médian et du filtre avec noyau gaussien optimiser

On constate que compte tenu de la qualité de l'image le filtre médian est plus optimisé.

En conclusion pour plus supprimer les bruits il est préférable d'utiliser les filtres non linéaires (filtre médian).

4.7 Détection du contour d'une image

Le contour est la frontière que forme un objet avec l'arrière-plan dans une image donc détecter le contour revient à détecter un objet ou une partie d'un objet dans une image.

La détection de contour nous permet de garder une information importante de l'image.

La détection de contour est une technique qui permet de réduire les informations dans une image et de transformer l'image à un ensemble de courbe.

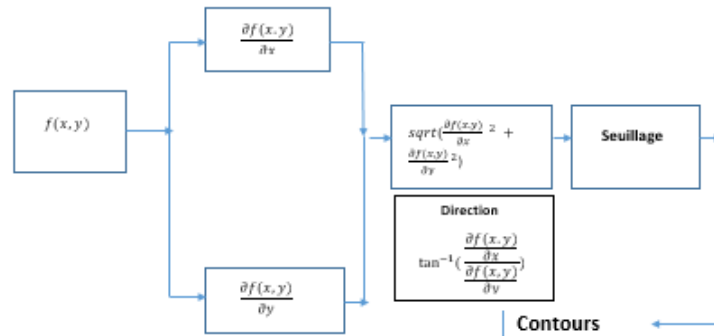
4.7.1 Approximation du gradient

Le gradient d'une fonction d'une image f en un point pixel $M(x, y)$ s'écrit :

$$\nabla f(x, y) = \left(\frac{\partial f(x, y)}{\partial x}, \frac{\partial f(x, y)}{\partial y} \right)$$

où $\frac{\partial f(x, y)}{\partial x}$ est la dérivée de f dans la direction Ox et $\frac{\partial f(x, y)}{\partial y}$ est la dérivée de f dans la direction Oy .

Pour faire l'approximation du gradient afin de déterminer le contour d'une image il suffit dans un premier temps de calculer d'abord le gradient dans les deux directions orthogonales et ensuite calculer les modules du gradient.



4.7.2 Calcul du gradient d'une image

Comme on travaille avec une image, on va travailler avec le gradient discret. Dans ce cas le gradient discret est le taux de variation calculable même si l'image présente une discontinuité.

On se donne une fonction f continue sur $[0, L]$ et dérivable sur $]0, L[$.

On peut approcher la dérivée de la fonction en des points X_1, \dots, X_{N-1} d'une

subdivision de $[0, L]$ de la forme $xi = ih, i = 0, \dots, N$ ou $h = \frac{L}{N}$ est le pas de grille n .

Pour tout $i = 1, \dots, N - 1$ $f'(x_i) = \frac{f(xi+h)+f(xi)}{h} = \frac{N}{L}(f(xi + 1) - f(xi))(*)$.

Par convention nous allons prendre $h = \frac{N}{L} = 1$.

Dans ce cas nous allons calculer les dérivées directionnelles avec la forme établie en (*).

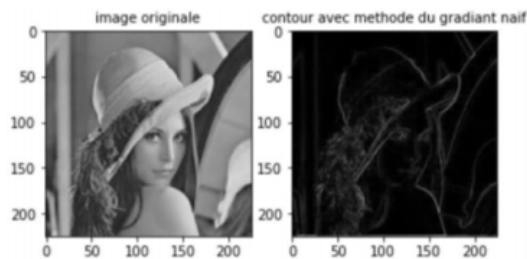
On a donc $f_x(x, y) = f(x + 1, y) - f(x, y)$ suivant (Ox) et $f_y(x, y) = f(x, y + 1) - f(x, y)$.

Si nous considérons l'approximation f_x de $\frac{\partial f(x,y)}{\partial x}$ on voit que le calcul peut se faire par calcul avec le noyau $K = \begin{bmatrix} 0 & -1 & 1 \end{bmatrix}$.

4.7.3 Contour avec gradient naïf d'une image

Comme dans les autres sections, dans ce cas le noyau de convolution qu'on utilise est $\begin{bmatrix} 0 & -1 & 1 \end{bmatrix}$ pour estimer la dérivée dans la direction (ox) et sa transposer pour estimer la dérivée suivant (oy) . chaque pixel est évalué après comme norme du gradient de f

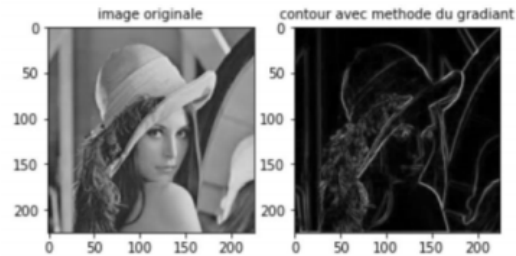
Résultat après application du filtre du gradient naïf



4.7.4 contour avec gradient de l'image

Dans le sens de produire des frontières épaisses nous allons utiliser le noyau $\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$ pour estimer la dérivée dans la direction (ox) et sa transposé pour estimer la dérivée suivant (oy) .

Résultat après application du filtre du gradient



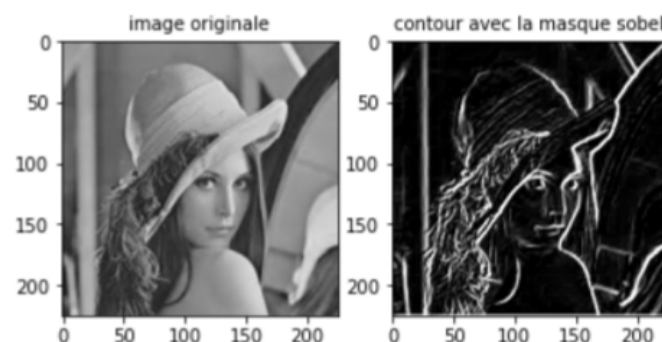
4.7.5 Masque sobel

Comme dans les autres sections, le masque qu'on utilisera pour évaluer les dérivées sont :

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Le produit tensoriel est fait pour atténuer les perturbations de l'image car l'opération de gradient est sensible au bruit.
Résultat après application du masque sobel.



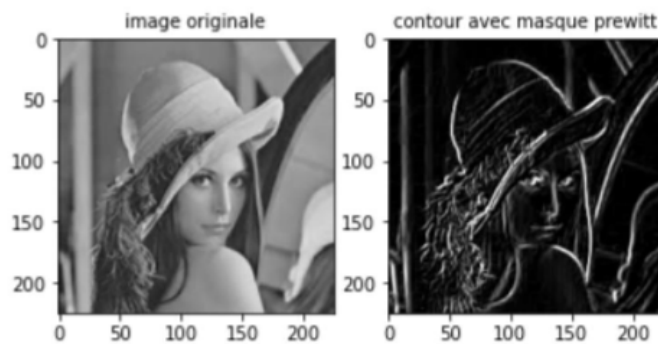
4.7.6 Masque préwitt

Comme le masque de sobel on va faire le produit de convolution avec le noyau :

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & -1 & -1 \end{bmatrix}$$

Résultat après application du masque préwitt.



4.7.7 Contour avec approximation de la dérivée seconde d'une image

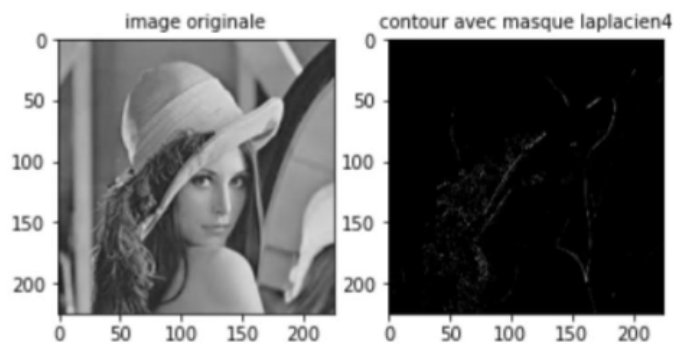
Pour approcher f_{xx} de $\frac{\partial^2 f}{\partial^2 x}$ on va utilisé le noyau $[-1 \ 2 \ 1]$ et son transposé pour f_{yy} de $\frac{\partial^2 f}{\partial^2 y}$.

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

4.7.8 Laplacien discret -4

Pour avoir le contour ici on doit faire la convolution avec le noyau ci-dessus :

0	1	0
1	-4	1
0	1	0

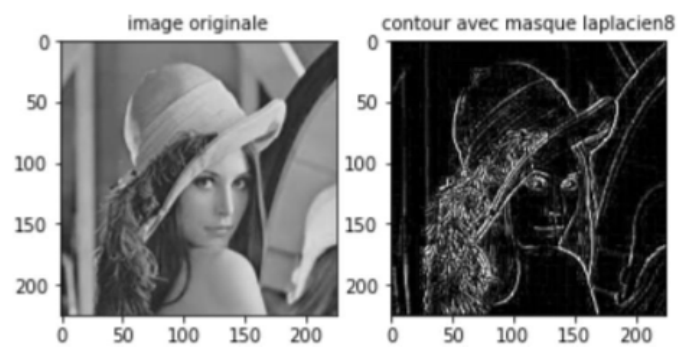


4.7.9 Laplacien discret -8

De même Pour avoir le contour ici on doit faire la convolution avec le noyau ci-dessus :

1	1	1
1	-8	1
1	1	1

CHAPITRE 4. DÉBRUITAGE PAR FILTRE NON LINÉAIRE ET
4.7. DÉTECTION DU CONTOUR D'UNE IMAGE LINÉAIRE



Chapitre 5

Conclusion et quelques applications du traitement numérique d'image

Ce projet nous a permis de nous initier aux traitements de base d'une image numérique. Nous avons appris à : représenter numériquement une image, réaliser des traitements ponctuels sur des images, filtrer des images. Nous avons également appris à utiliser la bibliothèque de traitement d'images OpenCV sur Python et à implémenter un algorithme de traitement d'image. Le traitement d'image est utilisé aujourd'hui pour détecter la présence d'un objet ou son absence ou encore pour calculer les caractéristiques d'un ou de plusieurs éléments d'une image. Dans tous les cas, l'idée est d'extraire des informations en partant d'une image initiale. Pour cela, il faut utiliser un ensemble de techniques en les combinant et en les enchaînant dans le but de créer des applications complexes et évoluées. Ces techniques sont à la base des systèmes de vision industrielle qui est l'application de la vision par ordinateur aux domaines industriels de production et de recherche. Le traitement d'image est également utilisé dans la reconnaissance d'objets qui est une branche de la vision artificielle. Elle consiste à identifier des formes pré-décrites dans une image numérique, et par extension dans un flux vidéo numérique.

En définitive, le traitement d'image intervient dans plusieurs domaines permettant ainsi de les faire progresser. On peut citer : la reconnaissance de l'écriture, la recherche d'image par le contenu , la reconnaissance faciale.

Bibliographie

- [1] *Mathématiques et applications 76, Introduction au traitement mathématique des images-méthodes déterministes – Maitine Bergounioux*
- [2] *Traitement-d’images-Wikipédia-(wikipedia.org)*
[http ://www.tsi.enst.fr/pages/enseignement/ressources/mti/egal-histo/rapport.htm](http://www.tsi.enst.fr/pages/enseignement/ressources/mti/egal-histo/rapport.htm)
- [3] *Seuillage-d’image-Wikipédia-(wikipedia.org)**[http ://www.tsi.enst.fr/pages/enseignement/ressources/mti/egal-histo/rapport.htm](http://www.tsi.enst.fr/pages/enseignement/ressources/mti/egal-histo/rapport.htm)*
- [4] *[http ://www.tsi.enst.fr/pages/enseignement/ressources/mti/egal-histo/rapport.htm](http://www.tsi.enst.fr/pages/enseignement/ressources/mti/egal-histo/rapport.htm)*
- [5] *[https ://fr.wikipedia.org/wiki/Loi-uniforme-continue](https://fr.wikipedia.org/wiki/Loi-uniforme-continue)*
- [6] *[http ://aymeric.histace.free.fr/documents/Segmentation.pdf](http://aymeric.histace.free.fr/documents/Segmentation.pdf)*
- [7] *Méthode d’Otsu — Wikipédia (wikipedia.org)*
- [8] *Initiez-vous aux traitements de base des images numériques - OpenClassrooms*
- [9] *filtres introduction et moyennneur - YouTube*
- [10] *Traitement d’image : suite chapitre 4 : Filtrage et restauration d’image - YouTube*
- [11] *[Deepmath] 12.2. Neurone et couche de convolution - YouTube*
- [12] *Master Informatique Biomédicale - Traitement d’images Cours 3 - Erosion et dilatation - YouTube*