

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne Démocratique et Populaire  
وزارة التعليم العالي والبحث العلمي  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



N° Ref :.....

**Centre Universitaire de Mila**

**Institut des sciences et de la technologie**

**Département de Mathématiques et Informatique**

**Utilisation des outils Matlab pour la segmentation  
d'image :**

**Application sur des pièces d'immatriculation**

**Mémoire préparé En vue de l'obtention du diplôme de License  
Académique en Informatique**

**Préparé par : khenfi Moussab  
Daikh Aissa**

**Encadré par : Yassaadi Sabrina**

**Filière : Informatique**

**Spécialité : Informatique**

**Année universitaire : 2012/2013**

## Table des matières

### Chapitre 01 : prétraitement

Remerciements.....	2
Résumé.....	3
Table des matières .....	4
Table des figures .....	8
Introduction générale .....	10
1. Introduction .....	12
1.1 Définition.....	12
1. 2 Définition.....	12
1. 3 Définition .....	12
2. Le processus de traitement des images .....	12
2.1 Traitement (ou analyse) :.....	12
2.2 Traitement bas-niveau.....	13
2.3 Traitement haut-niveau :.....	13
2.4 Analyse (ou décision ou interprétation) :.....	13
3.1 Synthèse d'images (infographie).....	13
3.2. Définition du prétraitement .....	14
3.3. Méthodes du prétraitement :.....	15
3.3.1 Restauration d'images: .....	15
3.3.2 Amélioration d'images: .....	16
3.3.2.1 La modification d'histogramme ou l'égalisation d'histogramme .....	16
3.3.2.2 Le filtrage .....	16
3.3.3 Seuillage d'image: .....	17
3.3.4 Lissage: .....	17
3.3.5 Nettoyage de l'image:.....	17

3.3.6 Effet miroir et inversion :.....	17
3.3.7 Réalignement automatique de l'image :.....	17
3.4. Techniques du prétraitement :.....	17
3.4.1. Amélioration du contraste:.....	17
3.4.2. Amélioration de la netteté:.....	18
3.4.3. Réduction de bruit par moyenne spatiale (filtres passe-bas) ou temporelle :.....	19
3.4.4. Compensation de non-uniformité d'éclairage et de réponse du capteur:.....	19
3.4.5. Compensation des variations de contraste:.....	19
3.4.6. Compensation de non-linéarité de réponse du capteur:.....	20
3.4.7. Réalignement (correction de position et direction):.....	20
3.4.8. Compensation des distorsions optiques, perspective, des pixels non carrés:.....	20
3.4.9. Ré-échantillonnage géométrique:.....	20
3.4.10.-Comparaison à une référence:.....	20
3.5. Les fonctions du prétraitement sur Matlab :.....	21
4. Conclusion.....	25

## **Chapitre 02 : technique de segmentation**

1. Introduction .....	27
2. Définition de la segmentation :.....	27
2.3 Buts de la segmentation .....	27
2.4 Qu'est-ce que la segmentation .....	27
2.5 Segmentation régions/contours .....	28
3. Définition Seuillage :.....	28
3.1 Choix des seuils :.....	29
3.1.1Seuillage d'histogramme simple :.....	29
3.1.2Seuillage d'histogramme multiple :.....	30
3.2 Comment définir le seuil .....	31
4. Binarisation :.....	32

4.1 Définition.....	32
4.2 Les techniques de binarisation .....	33
4.2.1 Méthode de la moyenne de tous les pixels .....	33
4.2.2 Méthode La valeur médiane entre le ton maximum et le ton minimum .....	33
5. Conclusion .....	33

### **Chapitre 03 : filtrage**

1. Introduction .....	36
2. Définition de filtrage .....	36
2.1 Qu'est-ce qu'un filtre .....	36
2.2 Principe de filtrage .....	37
2.3 Pourquoi on doit filtrer une image.....	37
3. La convolution d'une image .....	37
3.1 Définition.....	37
3.2 Masque de convolution .....	37
3.3 Types de convolution .....	38
4. Les différents types des filtres .....	38
4.1. Les filtres passe-bas .....	38
4.2-Les filtres passe-haut .....	39
5-Types de filtrage .....	40
5.1-Filtrage adaptatif .....	40
5.2-Filtrage des couleurs. ....	41
5.3-Le filtre moyenne. ....	41
5.4 Le filtre Gaussien .....	42
5.5 Le filtre médian.....	43
5.6 Le filtre de Canny : (ou détecteur de Canny) .....	44
6. Utilisation des filtres sous Matlab.....	44

6.1 Filtre Gaussien.....	46
6.2 Filtre moyen... ..	48
6.3 Filtres de sobel .....	49
6.4 Filtres de canny.....	50
7. Conclusions .....	51

### **Chapitre 04: détection de contours**

1 Introduction .....	53
2. Qu'est-ce qu'un contour.....	53
2.1 Définition de la détection de contours.....	53
3. Les méthodes de la détection de contours .....	53
3.1 Dérivée première .....	54
3.1.1 Filtre de Prewitt .....	54
3.1.2 Filtre de Sobel .....	54
3.1.3 Filtre de Canny .....	54
3.2 Dérivée secondes .....	55
4. implémenter le contour sous Matlab .....	55
4.1 filtre de Canny.....	55
4.2 Détecteur de Prewitt .....	56
5. Conclusion.....	57
Résumé.....	58
Référence Bibliographiques et Webographiques.....	59

## Liste des figures

### Chapitre 01 : prétraitement

Figure 1.1 : image pixellisée .....	12
Figure 1.2 : exemple d'une image améliorée .....	13
Figure 1.3 : image d'un prétraitement d'image.....	15
Figure 1.4 : image restaurée .....	16
Figure 1.5 : contraste amélioré d'une image.....	18
Figure 1.6: netteté amélioration d'une image 1 .....	18
Figure 1.7 : netteté améliorée d'une image 2 .....	19

### Chapitre 02 : technique de segmentation

Figure 2.1 : image avant la segmentation et après la segmentation.....	27
Figure 2.2 : image segmentée.....	28
Figure 2.3 : histogramme d'un seuil simple.....	29
Figure 2.4 : Application du seuillage.....	30
Figure 2.5 : histogramme d'un seuil multiple.....	31
Figure 2.6: image d'une pièce d'immatriculation binarisée par méthode 1.....	33
Figure 2.7 : image d'une pièce d'immatriculation binarisée par méthode 2.....	33

### Chapitre 03 : Filtrage

Figure 3.1 : Filtrage par le filtre passe-bas.....	39
Figure 3.2 : Filtrage par le filtre passe-haut.....	40
Figure 3.3 : Filtrage par le filtre adaptatif.....	40
Figure 3.4 : Filtrage par le filtre moyen.....	42
Figure 3.5 : présentation de la forme gaussienne.....	42
Figure 3.6: Filtrage par le filtre gaussien. ....	43

Figure 3.7 : Filtrage par un filtre médian.....43

Figure 3.8 : Filtrage de Sobel et Canny.....44

Figure 3.9 : Application du bruit 'salt & pepper'.....45

Figure 3.10 : image filtrée par le filtrage gaussien (SIGMA=1).....47

Figure 3.11 : image filtrée par le filtrage gaussien (SIGMA=0.75).....47

Figure 3.12 : image filtrée par le filtrage gaussien (SIGMA=0.5).....47

Figure 3.13 : image filtrée par le filtrage moyenne.....48

Figure 3.13 : image filtrée par le filtre de Sobel.....49

Figure 3.13 : image filtrée par le filtre de Canny .....50

**Chapitre 04 : détection de contour**

Figure 4.1 : exemple d'une détection de contour.....53

Figure 4.2 : exemple sur le détecteur de Canny .....56

Figure 4.3 : exemple sur le détecteur de Prewitt .....56

## *Remerciement*

*Nous remerciant ALLAH qui nous aide et nous donne la patience et le courage de continuer et de bien donner.*

*On n'oublie pas nos parents pour leur contribution, leur soutien et leur patience.*

*Je tiens à remercier sincèrement Madam Passaadi Sabrina qui, en tant que Directeur de mémoire, s'est toujours montré à l'écoute et très disponible tout au long de la réalisation de ce mémoire, ainsi pour l'inspiration, l'aide et le temps qu'elle a bien voulu me consacrer et sans qui ce mémoire n'aurait jamais vu le jour.*

*Daikh Aissa, Je ne sais comment te remercier de m'avoir soutenu comme tu l'as fait pendant cette période, qui a été si douloureuse pour moi. Et je sais qu'essayer de redonner un souffle de vie à un pauvre homme, qui se retrouve seul, après tant d'années de vie commune, cela n'est ni facile ni toujours très agréable. Tant d'amis se sont éloignés de moi, alors que je croyais pouvoir compter sur eux ! Aussi, ce que tu as fait n'a pas de prix et je ne l'oublierai jamais. Si j'ai réussi à refaire surface, c'est à ta gaieté, à ton écoute, à ta disponibilité de tous les instants que je le dois. Un seul mot : merci. Que dieu de garde avec sa grande miséricorde dans son vaste paradis*

*Je souhaite adresser mes remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.*

*Enfin, j'adresse mes plus sincères remerciements à tous nos proches et amis, qui nous ont toujours soutenue et encouragée au cours de la réalisation de ce mémoire.*

*Merci à tous et à toutes.*

*Moussaab*



## **Introduction Générale**

Le traitement numérique des images trouve des applications dans un nombre croissant des domaines, aussi divers que le tri automatique du courrier, le contrôle industriel, les télécommunications (compression) et la reconnaissance de cibles militaires. La croissance exponentielle des capacités des processeurs est pour une grande part dans le développement du traitement d'images, ce domaine étant, par nature, très gourmand en puissance de calcul.

Le traitement d'images, par sa diversité même. Les informations que l'on souhaite extraire d'une image, les applications, les types de données, sont si variés qu'il est difficile, lorsque l'on doit traiter une nouvelle application, de savoir à quel modèle se raccrocher et quelle méthodologie suivre.

### **Objectif:**

Notre mémoire consiste à connaître la segmentation et comment appliquer ses méthodes des images numériques, et plus précisément sur les pièces d'immatriculation.

### **Chapitres mémoire :**

Le mémoire est constitué d'une introduction générale, de quatre chapitres.

**Chapitre 01 : le prétraitement :** ce chapitre a pour objectif essentiel la présentation de la première étape du processus global de traitement d'image : le prétraitement. En plus des méthodes du prétraitement et les techniques.

**Chapitre 02: Technique de segmentation :** dans le deuxième chapitre, nous définissons quelques techniques de segmentation et nous exposons les techniques et les méthodes qu'utilise Matlab pour la segmentation.

**Chapitre 03: Filtrage :** Ce chapitre nous représentons c'est quoi le filtrage et quelle sont les types de filtrage, comment les appliquer sur Matlab.

**Chapitre 04 : détection de contour :** ce chapitre nous allons parler des méthodes de la détection de Contour dont l'objectif est de segmenter les différentes parties d'une pièce d'immatriculation.

**CHAPITRE 01 :**  
**LE**  
**PRÉTRAITEMENT**

## 1. Introduction :

Dans ce chapitre on va présenter le prétraitement de l'image. Plus précisément, ce chapitre vise à répondre aux questions suivantes :

Qu'est ce qu'un prétraitement ?

Qu'elles sont les méthodes du prétraitement?

Qu'elles sont les techniques du prétraitement?

Qu'elles sont les Algorithmes du prétraitement sur Matlab?

Commençons d'abord par connaitre quelques définitions :

1.1 Définition: image numérique c'est une matrice de  $X \times Y$  pixels (Picture élément) correspondant à l'échantillonnage et la quantification d'un signal acquis avec une caméra.

-Chaque pixel est associé à un niveau de gris  $n$  ou des niveaux de composante couleur codé sur  $N$  bits et qui représentent respectivement le niveau de luminosité ou de couleur de la zone Correspondante dans la scène observée. Chaque pixel est localisé par ses coordonnées  $x$  et  $y$  dans l'image. [Van2 09]

1.2 Définition : l'image numérique est un ensemble structuré d'information qui après affichage sur écran ont une signification pour l'œil humain. [Dib 13]

1.3 Définition: une image est une matrice (tableau) de point et chaque point est un « pixel », Chaque pixel contient une couleur. (Voire figure 1.1)

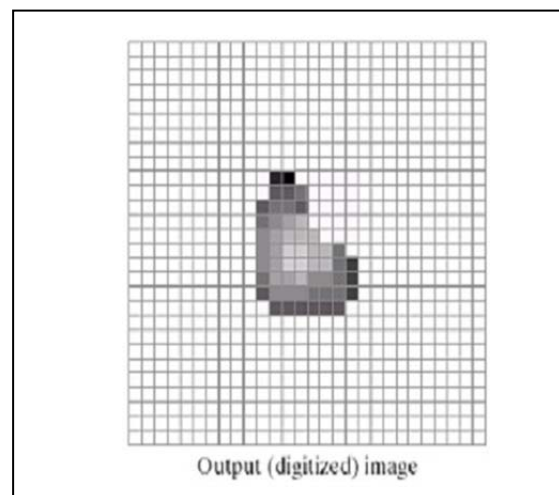


Figure 1.1 : image pixellisée

## 2. Le processus de traitement des images :

### 2.1 Traitement (ou analyse) :

C'est extraire l'information utile. Ou un traitement désigne l'action de traiter une donnée: son, image ou document.

### 2.2 Traitement de bas-niveau :

L'image est décrite de manière numérique (structurelle), il n'y a pas de lien avec la réalité qu'elle représente.

### 2.3 Traitement de haut-niveau :

L'image est décrite de manière symbolique, un lien existe avec la scène observée.

### 2.4 Analyse (ou décision ou interprétation) :

Interpréter les informations et décider d'une action à engager : l'image est alors décrite de manière sémantique. [Van2 09]

## 3. Les différents types de traitements :

### 3.1 Synthèse d'images (infographie) :

- est un ensemble complexe de techniques et méthodes de traitements et de représentation des informations graphiques. [Dib 13]
- Des techniques de génération d'images par ordinateur.
  - a) Amélioration : Modification de l'image dans le but de la rendre plus agréable à l'œil. C'est-à-dire augmenter la qualité de la perception visuelle d'une image. (voir figure 1.1). [Dib 13]

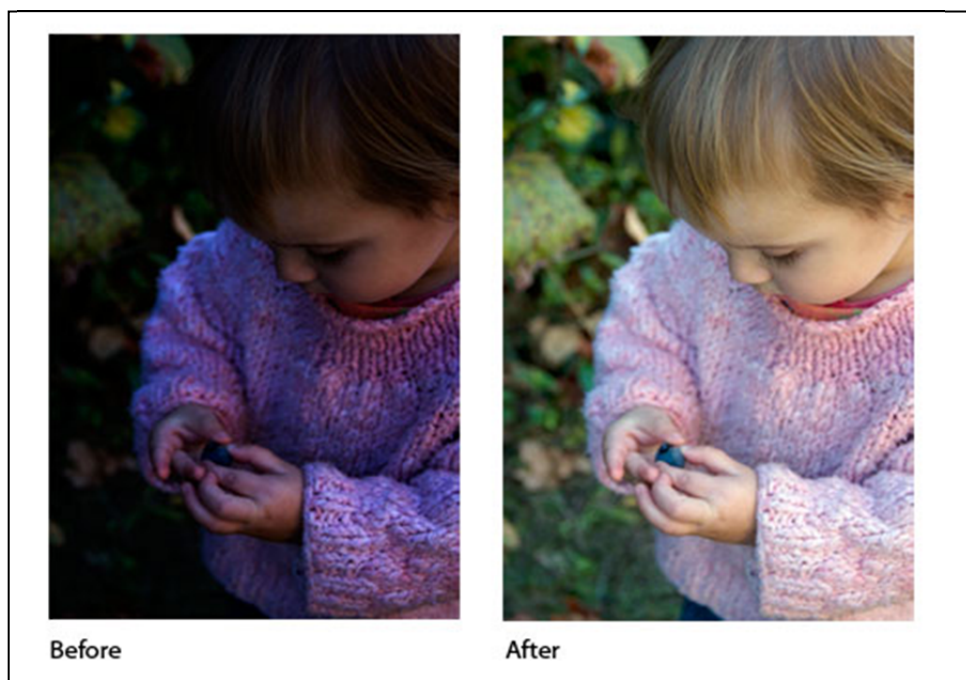


Figure 1.2 : exemple d'une image améliorée

- b) Restauration :** Correction des défauts dus à une source de dégradation. Ou La *restauration* C'est-à-dire remettre en place un état précédent qui a été altéré ou le résultat de cette action. [1]
  
- c) Compression :** De façon générale, la compression est un processus qui permet de diminuer la taille des fichiers. Dans l'imagerie, c'est la réduction du volume de l'image. [1]
  
- d) Quantification :** C'est le Codage de l'image.
  
- e) Segmentation :** La segmentation vise à diviser l'image en morceaux. Pour le but de séparer les entités d'une image afin d'appliquer un traitement spécifique. [1]
  
- f) Classification :** Classification ou système de classification veut dire un système organisé et hiérarchisé de catégorisation d'objets. [1]  
  
Dans le domaine de l'imagerie, c'est l'affectation des données de l'image à une classe donnée.
  
- g) Reconnaissances des formes :** De façon générale, c'est l'ensemble des techniques et méthodes visant à identifier des motifs informatiques à partir de données brutes [1]. Dans le traitement d'image, c'est identification du contenu de l'image.
  
- h) Indexation et recherche d'images :** Transformer l'information des pixels en un autre type d'information pour faciliter l'utilisation l'image (en termes informatiques, taille compacte, vitesse, tout en conservant une sémantique proche à l'utilisateur). [1]
  
- i) Visualisation et vision 3D :** Localisation en 3dimensions des objets dans l'image [Van2 09]

### 3.2 Définition du prétraitement :

« Opération ou ensemble d'opérations consistant à rendre des données brutes aptes à une analyse thématique. Ces opérations consistent, essentiellement, à effectuer des formatages, le délignage, des corrections géométriques, radiométriques, des échantillonnages ».[2]

Donc, le prétraitement consiste d'améliorer l'image soit la clarté, le contour, ou éliminer les informations indésirables sur les images.

Les informations indésirables sont :

-le signal de précharge. : Appelé également offset ou bias en anglais, il s'agit d'un signal constant généré par l'électronique de la caméra.

-Le signal thermique ou le bruit d'image : c'est la présence d'informations parasites qui s'ajoutent de façon aléatoire aux détails de la scène photographiée numériquement. Il est plus particulièrement visible dans les zones peu éclairées.

-Les défauts optiques : sont dus à l'objectif de l'appareil (fixe sur les compacts, interchangeables sur les réflexes). [Mic 08]

Voilà un exemple sur le prétraitement d'image (voire figure1.2).

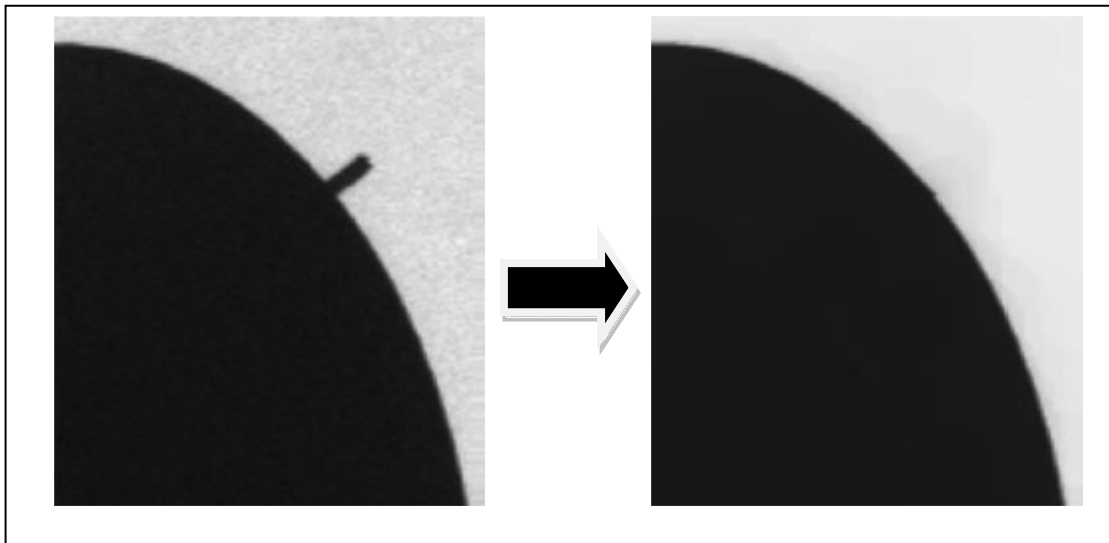


Figure1.3 : image d'un prétraitement d'image

### 3.3 Méthodes du prétraitement :

#### 3.3.1 Restauration d'images :

La restauration d'image est le résultat d'une ancienne image remise à neuf par un infographiste à l'aide d'un ordinateur et d'un logiciel d'image. (Voir figure 1.4) [Van1 09]

-une image déchirée, grafignée, décolorée, peut, avec ce procédé, être rendue à son état d'origine.

-Une image en noir et blanc par exemple peut, avec ce procédé, être colorisée.

Le but est d'obtenir de la restauration d'image c'est que l'image soit la plus proche possible de l'image idéale qui aurait été obtenue si le système d'acquisition était parfait. [Van1 09]



Figure 1.4 : image restaurée

### 3.3.2 Amélioration d'images :

L'amélioration a pour but de satisfaire l'œil de l'observateur humain. L'œil humain est essentiellement sensible aux forts contrastes.

C'est pourquoi les techniques d'amélioration tentent d'augmenter ceux-ci dans le but d'accroître la séparabilité des régions composant une scène. [Van1 09]

Différentes approches :

- La modification d'histogramme ou l'égalisation d'histogramme.
- Le filtrage (fréquentiel).

#### 3.3.2.1 La modification d'histogramme ou l'égalisation d'histogramme :

Est une méthode d'ajustement du contraste d'une image numérique qui utilise l'histogramme.

Elle consiste à appliquer une transformation sur chaque pixel de l'image, et donc d'obtenir une nouvelle image à partir d'une opération indépendante sur chacun des pixels. Cette transformation est construite à partir de l'histogramme cumulé de l'image de départ.

#### 3.3.2.1 Le filtrage :

Un filtre linéaire est, en traitement du signal, un système qui applique un opérateur linéaire à un signal d'entrée. Les filtres linéaires sont rencontrés le plus souvent en électronique.

Le filtrage sera discuté dans le chapitre 3.

### 3.3.3 Seuillage d'image :

Est la méthode la plus simple de segmentation d'image. À partir d'une image en niveau de gris, le seuillage d'image peut être utilisé pour créer une image comportant uniquement deux valeurs, noir ou blanc (monochrome) [7] (Le seuillage sera discuté dans le chapitre 2.)

### 3.3.4 Lissage :

On appelle "lissage" (parfois *débruitage* ou *filtre anti-bruit*) l'opération de filtrage visant à éliminer le bruit dans une image.

L'opération de lissage spécifique consistant à atténuer l'effet d'escalier produit par les pixels en bordure d'une forme géométrique est appelée *anti-crénelage* (en anglais *anti-aliasing*). [8]

### 3.3.5 Nettoyage de l'image :

Lors de la numérisation de documents de qualité médiocre à moyenne, il se peut que vous obteniez des images comportant beaucoup de « bruit », c'est-à-dire beaucoup de points et de traces. Ces traces, lorsqu'elles apparaissent près de lettres ou de nombres, peuvent compromettre la qualité de l'OCR.

Cette fonctionnalité supprime ce bruit. La taille des traces à supprimer peut être définie par l'utilisateur. Celle-ci peut s'appliquer à une image, ainsi qu'à un bloc (ou à une zone isolé dans l'image). [5]

### 3.3.6 Effet miroir et inversion :

Une option disponible permet de réfléchir l'image préparée sur son axe vertical. Il est également possible d'inverser les couleurs de l'image préparée. [5]

### 3.3.7 Réalignement automatique de l'image :

Il s'agit d'une fonction d'imagerie de documents indispensable qui est appliquée aux documents numérisés nécessitant un réajustement au niveau de l'alignement des images. Cette fonction ne requiert pas de bords ou de lignes frontales. (Flexi Capture Engine) propose plusieurs méthodes de réalignement d'images : par paires de carrés noirs, par lignes ou par lignes de texte. [5]

## 3.4. Techniques du prétraitement :

### 3.4.1 Amélioration du contraste:

Lorsque la quantité de lumière incidente est faible ou que les surfaces observées présentent des teintes voisines (en couleur comme en noir et blanc), le contraste peut être insuffisant. L'étirement d'histogramme et son égalisation peuvent améliorer grandement la lisibilité de l'image. (Voir figure 4)



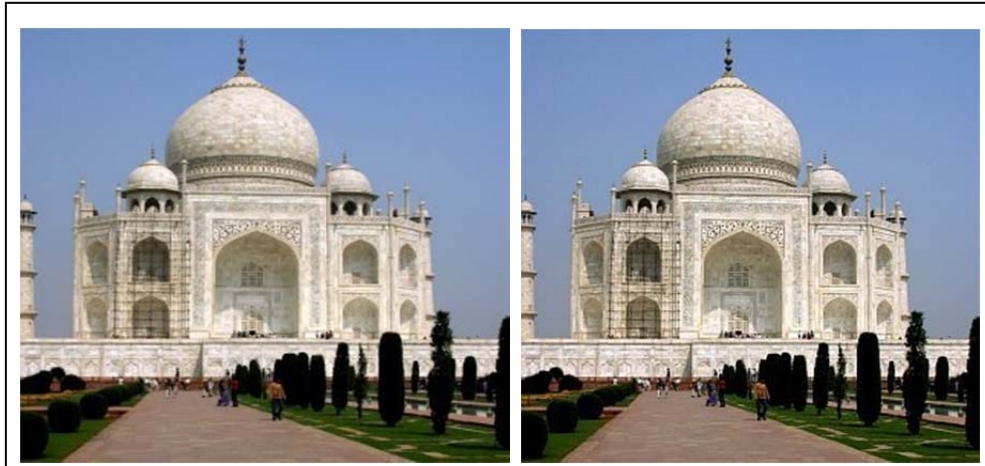


Figure 1.5 : contraste amélioré d'une image

### 3.4.2 Amélioration de la netteté:

Le flou peut être réduit très simplement par utilisation de filtres passe-haut, qui combinent d'une certaine façon l'image donnée et ses dérivées (rehaussement des bords), de façon à accentuer les variations au détriment des zones uniformes.

(Voir figure 5 et 6)

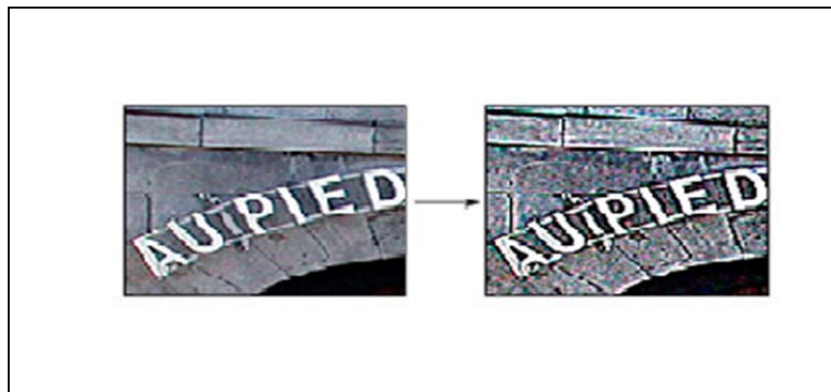


Figure 1.6: netteté et amélioration d'une image 1.

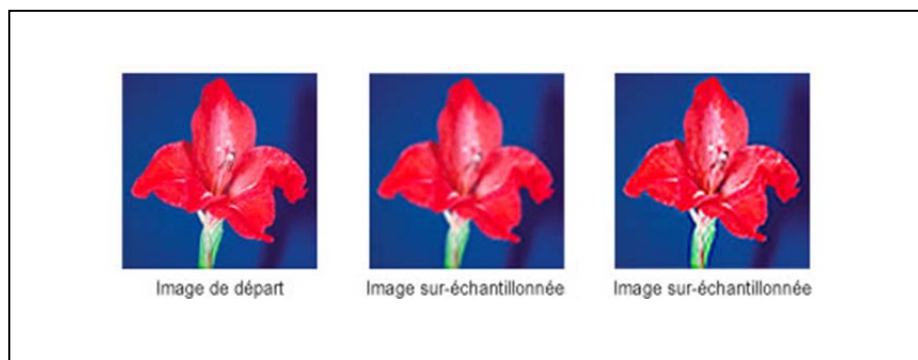


Figure 1.7 : netteté améliorée d'une image 2.

### 3.4.3 Réduction de bruit par moyenne spatiale (filtres passe-bas) ou temporelle :

Les pixels d'une image sont normalement fortement corrélés spatialement (d'un pixel à ses voisins) et temporellement (d'un instant à l'autre). Cette corrélation peut être exploitée pour éliminer les pixels « déviants ». [3]

### 3.4.4 Compensation de non-uniformité d'éclairage et de réponse du capteur:

Les non-uniformités d'éclairage (plus de lumière à certains endroits qu'à d'autres) sont en général lentes, c'est-à-dire qu'elles concernent essentiellement les fréquences spatiales basses. Un filtre passe-haut (voire chapitre 3) peut les atténuer.

Une autre approche consiste à calibrer l'éclairage en conservant une image d'une mire uniforme qui sert de référence pour l'intensité des pixels de même intensité en chaque point. Le même procédé permet de tenir compte de la non-uniformité de la sensibilité du capteur (Une seule mire permet de corriger l'intensité. Deux mires, l'une claire et l'autre sombre, permettent de corriger à la fois l'intensité et le contraste). [3]

### 3.4.5 Compensation des variations de contraste:

En raison du vieillissement des sources d'éclairage, des conditions de lumière ambiante ou d'interventions sur les réglages du système, la quantité totale de lumière incidente peut varier. Cela peut influencer le résultat des traitements. Une façon de compenser de telles variations est la normalisation: on mesure la luminosité moyenne dans les conditions d'apprentissage, puis lors des inspections suivantes, on mesure à nouveau la luminosité moyenne de l'image et on applique un coefficient correcteur approprié. Cette technique n'est efficace que si le contenu de l'image ne change pas d'une image à l'autre. [3]

### 3.4.6 Compensation de non-linéarité de réponse du capteur:

Une réponse linéaire peut être décrite par deux points de mesure (une relation linéaire en  $x$ , de la forme  $ax + b$  comporte deux paramètres inconnus).

Dans le cas général, par contre, la relation entre l'intensité lumineuse reçue et la tension mesurée peut être non-linéaire. Si nécessaire, la relation en question peut être étalonnée au moyen d'instruments de précision et être tabulée pour permettre la correction. [3]

### 3.4.7 Réalignement (correction de position et direction):

Pour diverses raisons, en particulier à cause du jeu mécanique de la fixation des pièces, ou un déplacement involontaire de la caméra ou des pièces elles-mêmes, ... les objets observés n'occupent pas toujours exactement la même position dans le champ de vue.

Si on est capable, par des méthodes exposées de déterminer la position effective des pièces, on peut ramener celles-ci en position nominale par réalignment (translation et rotation, éventuellement remise à l'échelle). Cela permet de ramener l'objet inspecté dans une position telle que les régions d'intérêt utiles peuvent rester fixes. [3]

### 3.4.8 Compensation des distorsions optiques, perspective, des pixels non carrés:

La chaîne de formation de l'image peut introduire différents types de déformations qui donnent des objets observés une image faussée.

Si on est capable de mesurer les déformations correspondantes, notamment par l'emploi de mires de formes connues -grilles, matrices de points, damiers- on est également capable de compenser ces déformations et de restituer la géométrie d'origine de la scène. [3]

### 3.4.9 Ré-échantillonnage géométrique:

Parfois, le contenu brut de l'image n'est pas exploitable et on s'intéresse à des sous-parties d'images de forme particulière: les profils correspondent aux valeurs des pixels prises le long d'un segment de droite ou d'une courbe (arc de cercle ou autre). Ceux-ci sont particulièrement utiles pour analyser des formes longilignes. [3]

### 3.4.10 Comparaison à une référence:

Dans de nombreux cas, les imperfections introduites par la chaîne d'acquisition peuvent être acceptées si l'on travaille par comparaison. Si les imperfections correspondent à des erreurs systématiques, c'est-à-dire se présentent toujours avec la même intensité et le même signe (distorsion optique permanente, non-uniformité stable dans le temps, désalignement constant, ...), il suffit de conserver une image de référence d'un objet sans défaut et de comparer cette image aux images à inspecter pour observer les défauts de la pièce sans être gêné par les défauts de la prise de vue. Souvent, une simple soustraction d'image, ou la comparaison à une tolérance de variation près suffit. [3]

## 3.5. Les fonctions du prétraitement sur Matlab :

Matlab supporte 4 formats d'images :

- les images binaires,
- les images d'intensités (à niveaux de gris),
- les images couleur RGB,
- les images couleur indexées.

-Il est possible de changer de format en utilisant les fonctions suivantes :

- **ind2gray** : indexé → intensité
- **ind2rgb** : indexé → RGB,
- **rgb2ind** : RGB → indexé,

- **rgb2gray** : RGB —→ intensité
- **im2bw** : intensité, indexé, RGB —→ binaire

-Il est possible de changer le type des variables en utilisant les fonctions suivantes :

- **im2double** : codage d'images en type réel,
- **im2uint8** : codage d'images en type entier non signé sur 8 bits,
- **im2uint16** : codage d'images en type entier non signé sur 16 bits,
- **double** conversion de données en type réel,
- **uint8** conversion de données en type non signé sur 8 bits,
- **uint16** conversion de données en type non signé sur 16 bits.

Certaines fonctions ou certains outils de Matlab permettent des manipulations interactives sur une image contenue dans une figure ou non :

- **imageinfo** : retourne les informations de l'image dans la figure ou d'un fichier image,
- **zoom** : zoom sur une zone de l'image de la figure,
- **imcrop** : sélectionne une zone de l'image,
- **improfile** : affiche le profil d'une ligne sélectionnée,
- **impixel** : retourne les valeurs des pixels sélectionnés,
- **impixelinfo** : affiche la position et les valeurs d'un pixel pointé avec la souris,
- **impixelregion** : affiche les valeurs des pixels dans une région sélectionnée avec la souris,
- **imdistanline** : affiche la distance entre deux pixels sélectionnés,
- **imdisplayrange** : affiche l'intervalle des valeurs des pixels de l'image,
- **imcontrast** : réajuste une image,
- **imtool** : outil qui utilise les outils précédents.

- D'autres fonctions permettent des opérations géométriques sur l'image :

- **imresize** : ré-échantillonnage de l'image (homothétie),

• **imrotate** : rotation de l'image. [Van2 09]

- Afficher une image :

Pour lire une image sur le disque, connaître ses caractéristiques, et l'afficher à l'écran :

```
f= imread('image.jpg')
whos f
imshow(f)
```

La commande : **imshow (f, [])** permet d'ajuster automatiquement le contraste. pour lire les informations sur une image sans la charge, on peut se servir de la commande :

```
imfinfo nom_image
```

Pour accéder à des informations particulières, par exemple, la taille :

```
donnees=imfinfo('nom_image');
taille=K.FileSize
```

- Sauver une image :

On utilise la commande :

```
imwrite(f,'nom_de_l'image')
```

On peut aussi spécifier le format et d'autres caractéristiques. Par exemple, en format TIFF (on n'est pas obligé de mettre l'extension dans le nom du fichier)

```
imwrite(f,'image','tif')
```

ou bien une qualité de JPEG :

```
imwrite(f,'image.jpg','quality',q) avec q compris entre 0 et 100.
```

- Améliorer une image :

La fonction **imhist** permet le calcul et l'affichage de l'histogramme d'une image. En exploitant cet histogramme plusieurs opérations sont possibles en utilisant les fonctions suivantes :

• **imadjust** : recadrage de la dynamique selon une correction gamma

- **histeq** : égalisation et spécification d'histogrammes,
- **adapthisteq** : égalisation adaptative d'histogrammes,
- **im2bw** : binarisation d'une image. [Van2 09]

A l'aide de la fonction *adjust*, on peut corriger la distribution des niveaux d'une image en jouant sur trois paramètres

- les niveaux d'entrée ;
- les niveaux de sortie ;
- le coefficient gamma.

ce qui permet de modifier le contraste. Par exemple, sans jouer sur les niveaux, on peut imposer un très fort contraste en posant

```
g=imadjust(f,[ ],[ ],2)
```

Pour trouver les seuils à choisir dans les niveaux d'entrée, on peut utiliser des histogrammes, qui permettent de savoir comment sont repartis les niveaux de gris. [4]

```
p=imhist(f)/numel(f);  
hscale=1:10:256  
bar(hscale,p(hscale))
```

Avec la commande *imhist* on crée un histogramme, que l'on normalise par *numel* (nombre d'éléments de f). Pour tracer l'histogramme avec la commande *bar*, on définit une échelle allant de 10 en 10 (de 1 à 256 niveaux de gris).

```
p=imhist(f)/numel(f);  
hscale=1:10:256  
bar(hscale,p(hscale))  
print -djpeg histogramme.jpg
```

On note que la plupart des points sont concentrés avec des niveaux de gris < 100 (0 = noir absolu ; 256 = blanc absolu). Ces points constituent en fait l'information peu utile (fond de l'image), il faut donc prendre comme seuil inférieur 100 ou 150 pour accentuer l'information utile et supprimer l'information inutile (fond) [Van2 09]

```
i1=imadjust(f,[150 250]/256,[ ],2);  
i2=imadjust(f,[100 250]/256,[ ],2);  
imwrite(i1,'transformation150.jpg')  
imwrite(i2,'transformation100.jpg')
```

Dans certains, on peut chercher à faire tous les niveaux, ce qui veut dire que l'on cherche à rendre plus visibles les niveaux de gris peu présents et, réciproquement, à réduire ceux trop présents. Il s'agit donc d'égaliser l'histogramme, ce qui se fait avec la commande *histeq*. [4]

```
j=histeq(f,256);  
figure, imshow(j).
```

La fonction **medfilt2** (ou plus généralement **ordfilt2**) permet de réaliser le filtrage par un filtre médian.

#### 4. Conclusion :

Pour éliminer les bruits parasites et les artéfacts nuisibles il faut faire le prétraitement, on peut dire que le prétraitement est une étape très importants dans le traitement d'image car si on voulait faire les autres opérations de traitement d'image sans le prétraitement les résultats sont un peu mauvais, est pour cela il faut faire le prétraitement avant d'appliquer des autres opérations.

**CHAPITRE 02 :**  
**TECHNIQUES**  
**DE**  
**SEGMENTATION**



## 1. Introduction :

Dans ce chapitre on va présenter la segmentation d'image, exactement sur les méthodes seuillage. On va aussi exposer la binarisation et étudier comment le faire sur Matlab. Ce chapitre vise à répondre aux questions suivantes :

C'est quoi la segmentation d'image ? Et c'est quoi la binarisation d'une image ?

Qu'est-ce qu'un seuil ?

Quelles sont les Algorithmes de binarisation sur Matlab?

Quelle est la différence entre la binarisation et la segmentation ?

## 2. Définition de la segmentation :

La segmentation vise à diviser l'image en morceaux. Dans le but de séparer les entités d'une image afin d'appliquer un traitement spécifique. [Bou 09]

La segmentation est le découpage d'une image en Différentes régions et/ou contours. (Voire figure 2.1)

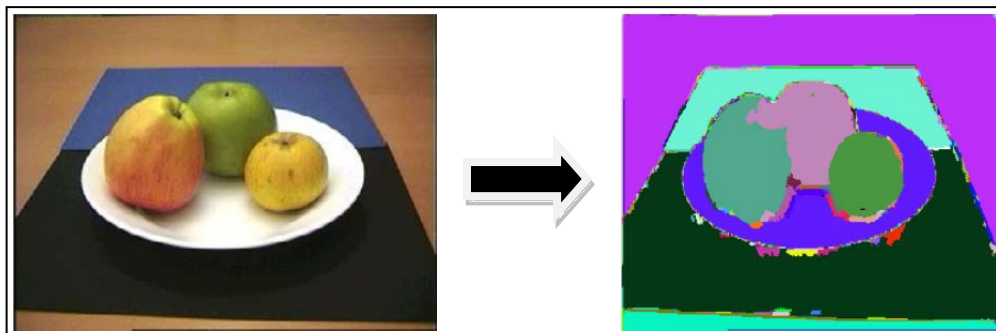


Figure2.1 : image avant la segmentation et après la segmentation

### 2.3 Buts de la segmentation :

- ✓ Extraire (séparer) les entités d'une image
- ✓ Pour y appliquer un traitement spécifique
- ✓ Pour interpréter le contenu de l'image

Dans la pratique : construire une image de masques et chaque masque est une composante connexe. [Jou 2011]

Les masques (des objets) permettront d'interpréter le contenu de l'image On traite chaque zone de l'image (masques) séparément. [Bou 09]

### 2.4 Qu'est-ce que la segmentation ?

La segmentation est normalement basée sur:

Les discontinuités : contours c'est les changements abrupts, frontières entre régions...

Les zones homogènes : régions de même couleurs, textures, intensités, ...

## 2.5 Segmentation régions/contours :

- Approches régions : Rechercher les zones dans l'image sur un critère d'homogénéité.
- Approches contours : Rechercher les discontinuités entre régions.
- Approches duales (régions et contours).[Bou 09]

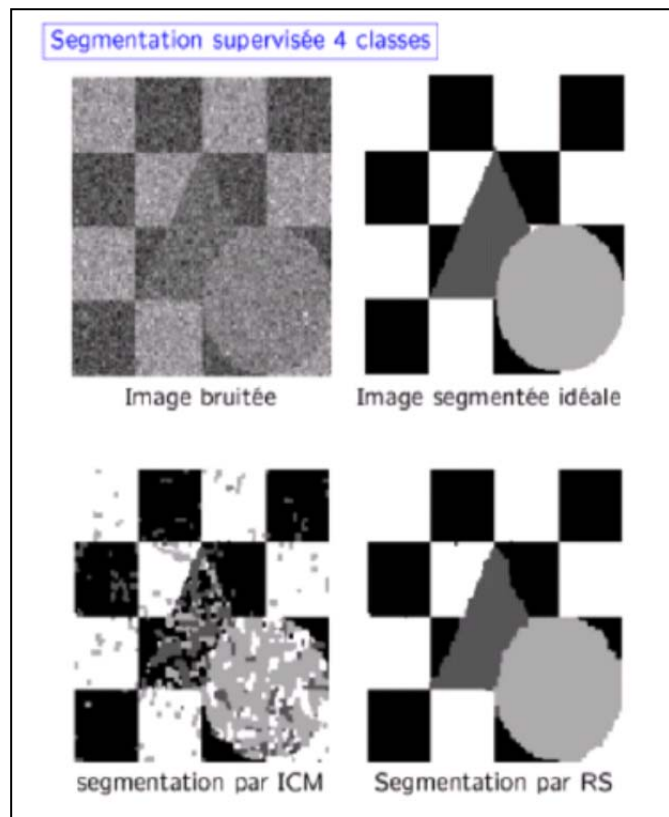


Figure2.2 : image segmentée.

## 3. Définition Seuillage :

Le seuillage est une approche simple de la segmentation [Bou 09] . Donc c'est une méthode simple et très populaire pour le traitement des images numériques.

Le seuillage sert de composante de base à des algorithmes plus complexes de segmentation. [Jou 2011]

Le seuillage peut être :

Global : un seuil pour toute l'image dépend de  $I(i,j)$ .

Local : un seuil pour une portion de l'image dépend de  $N(i,j)$ .

Adaptatif : un seuil s'ajustant selon les parties de l'image. [Bou 2009]

Seuillage de base (2 classes) :

*Si valeur (pixel) > seuil alors valeur (pixel) = 1*

*Si valeur (pixel) < seuil alors valeur (pixel) = 0*

Le résultat du seuillage est une image binaire 0 ou 1.  
Remarque : parfois en transforme 0:255 pour l'affichage)

### 3.1 Choix des seuils :

#### 3.1.1 Seuillage d'histogramme simple :

On applique ce algorithme :

$$G(x ; y) = \begin{cases} 1 & \text{si } f(x, y) \text{ supérieur ou égal de } T \text{ (T c'est le seuil)} \\ 0 & \text{si } f(x, y) \text{ inférieur de } T \end{cases}$$

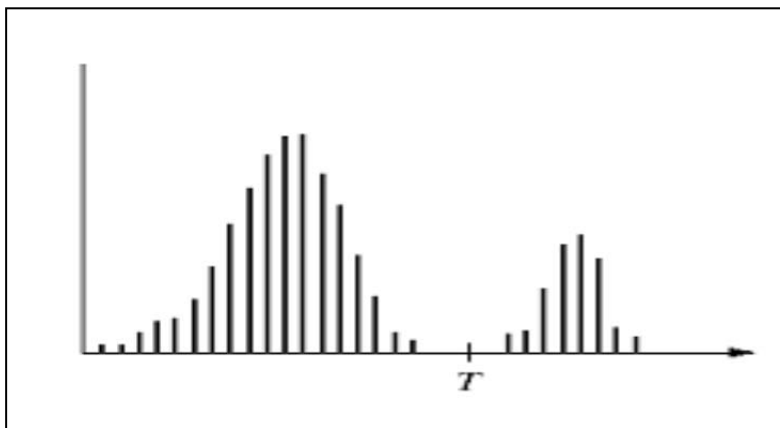


Figure 2.3 : histogramme d'un seuil simple.

Donc on souhaite voir les résultats pour appliquer cet algorithme sur des images il y a un exemple sur ceci (voir figure 2.4)

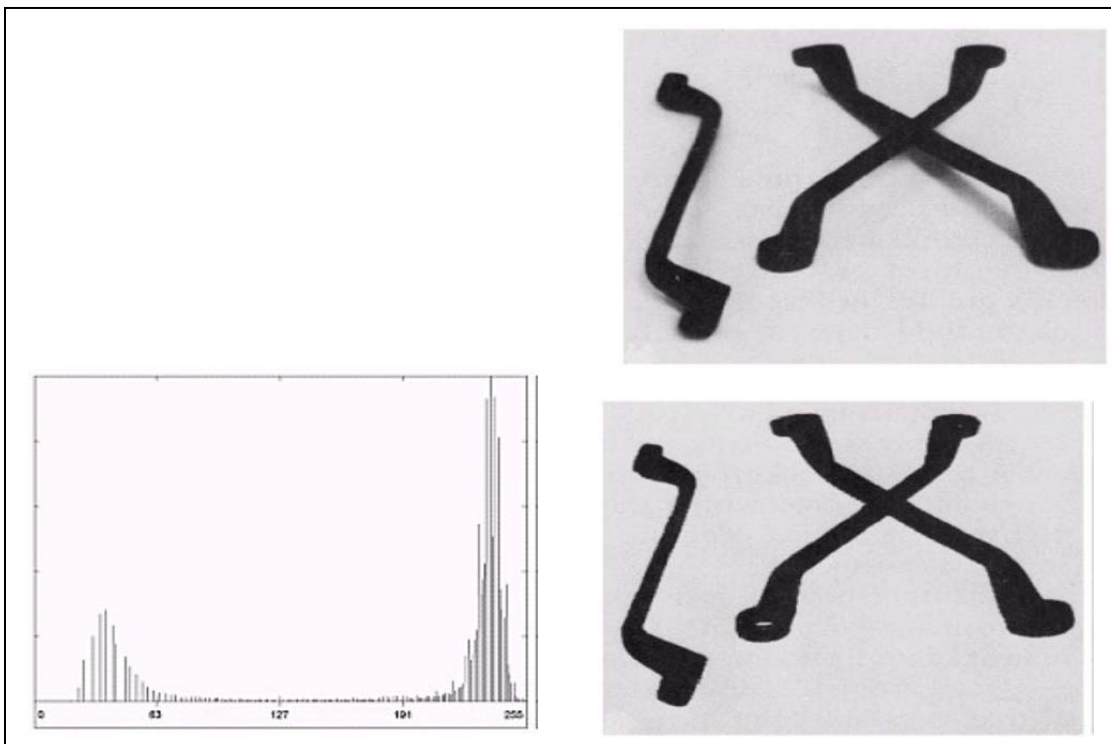


Figure2.4 : Application du seuillage

- Avantages du seuillage d'histogramme simple :
  - Universel, temps réel, simplicité.
  - Fonctionne bien sur des histogrammes multimodaux.
- Inconvénients du seuillage d'histogramme simple :
  - Connaître le nombre de classes
  - Apparition de faux éléments (aucune prise en compte de la Composante spatiale).
  - Nombre de modes souvent nombre de classes attendu [Bou 09]

### 3.1.2 Seuillage d'histogramme multiple :

$$G(x ; y) = \begin{cases} 2 & \text{si } f(x ; y) \text{ supérieur ou égal de } T_2 \\ 1 & \text{si } f(x ; y) \text{ supérieur ou égal de } T_1 \\ 0 & \text{si } f(x,y) \text{ inférieur de } T_1 \end{cases}$$

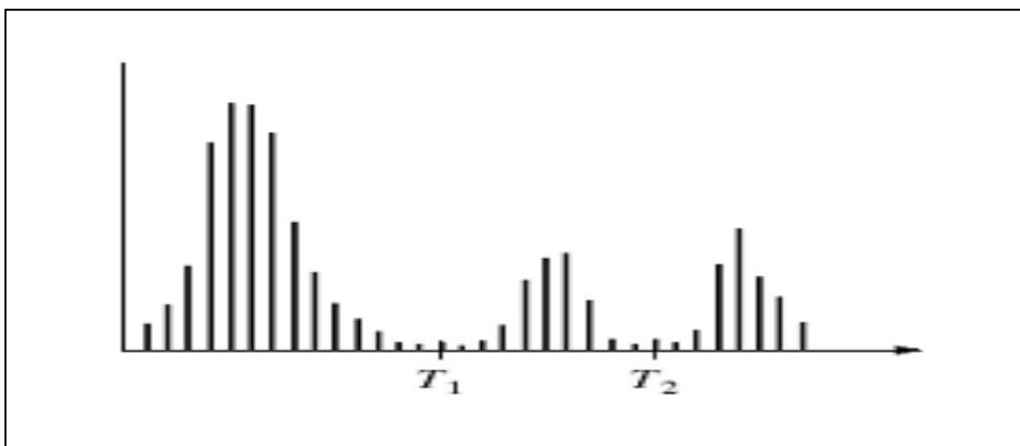


Figure 2.5 : histogramme d'un seuil multiple

### 3.2 Comment définir le seuil ?

Méthode1 : la moyenne de tous les pixels :

Voici un script qui donne un bon seuil.

```
for i=1:size(I,1)
    for j=1:size(I,2)
        seuill = seuill + Gr(i,j);
    end
end
seuill = seuill / (size(I,1)*size(I,2))
```

Méthode 2 : La valeur médiane entre le ton maximum et le ton minimum

Voici un autre script qui donne un bon seuil :

```
max = 1000;
min = 0;
for i=1:size(I,1)
    for j=1:size(I,2)
        if I(i,j)<max
            max = I(i,j);
        end
        if I(i,j)>min
            min = I(i,j);
        end
    end
end
moyenne = (max+min)/2;
```

Méthode 3 : Une valeur obtenue par tests :

On fixe un n'importe qu'elle seuil et tester pour retrouver le seuil exacte.  
Seuil=150 ou seuil = 185 ou seuil =190...etc.

## 4 Binarisation :

### 4.1 Définition :

La binarisation est basée sur un seuillage brut. Cela signifie que si un pixel de l'image a une intensité supérieure à une certaine valeur de seuil.

Il lui sera attribué la couleur blanche sinon il sera noir. Ce procédé est réalisé sur chaque pixel de l'image.

Nous obtenons donc une image comportant seulement deux niveaux (valeur 0 ou 1). C'est fait l'image est binarisée.

Pour appliquer la binarisation on applique les deux méthodes de trouver le seuil et on binarise les images qu'on nous avons (dans notre cas on utilise des pièces d'immatriculation).

### 4.2 Les techniques de binarisation :

#### 4.2.1 Méthode de la moyenne de tous les pixels :

Voici les résultats obtenus (voir figure 2.5).

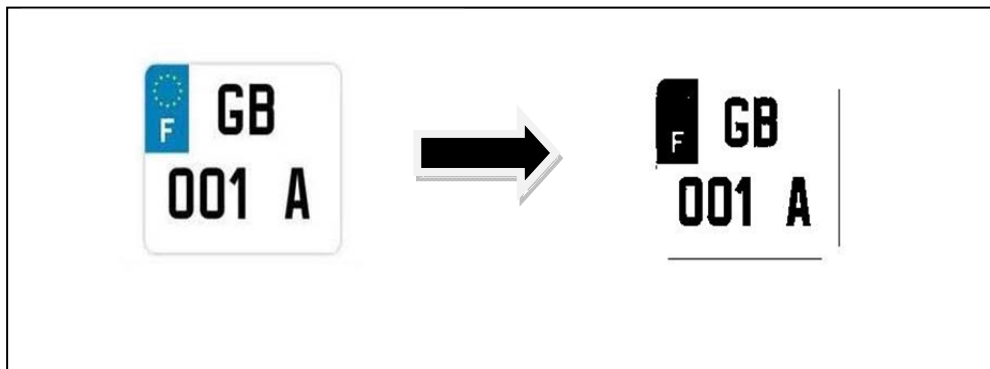


Figure 2.6 : image d'une pièce d'immatriculation binarisée par méthode 1

#### 4.2.2 Méthode La valeur médiane entre le ton maximum et le ton minimum

Voici les résultats obtenus (voir figure 2.6).

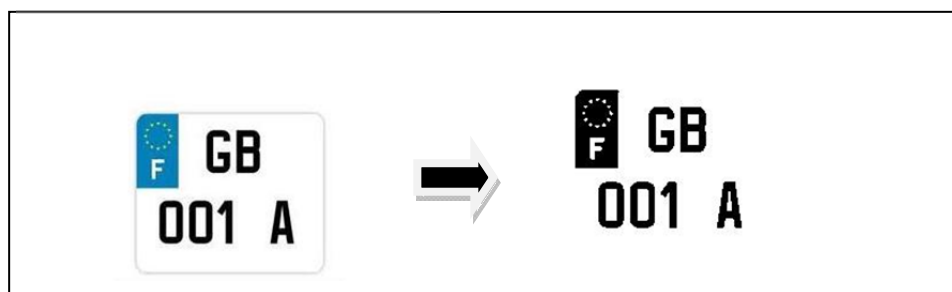


Figure 2.7 : image d'une pièce d'immatriculation binarisée par méthode 2

On remarque que les deux méthodes ne donnent pas exactement les mêmes résultats car le seuil choisi est différents.

## 5. Conclusion :

Le seuillage d'image est la méthode la plus simple de segmentation d'image. À partir d'une image en niveau de gris, le seuillage d'image peut être utilisé pour créer une image comportant uniquement deux valeurs, noir ou blanc.

La binarisation est une opération qui produit deux classes de pixels, en général, ils sont représentés par des pixels noirs et des pixels blancs.

Pour binariser n'importe quelle image il faut définir le seuil.

Il se définit On donnant :

- Une valeur constante
- Une valeur relative (% , moyenne, ou ...)
- Un algorithme pour trouver le seuil.

La différence entre la binarisation et la segmentation est que la binarisation produit toujours deux classes alors que la segmentation peut produire plusieurs classes.

**CHAPITRE 03 :**

**LE FILTRAGE**



## 1. Introduction :

Dans ce chapitre on va essayer de définir la phase de filtrage de l'image, sa démarche et son résultat, on propose des différents types des filtres principalement on vise à connaître:

C'est quoi le filtrage et c'est quoi la convolution ?

Qu'est-ce qu'un filtre ?

Quels sont les différents types des filtres ?

Est-ce qu'il existe d'autres filtres connus ?

Comment appliquer ces filtres sous Matlab ?

## 2. Définition de filtrage :

Le filtrage consiste à appliquer une transformation (appelée *filtre*) à tout ou partie d'une image numérique en appliquant un opérateur. [9]

### 2.1 Qu'est-ce qu'un filtre ?

Un filtre est une transformation mathématique (appelée *produit de convolution*) permettant, pour chaque pixel de la zone à laquelle il s'applique, de modifier sa valeur en fonction des valeurs des pixels avoisinants, affectées de coefficients. [9]

Le filtre est représenté par un tableau (matrice), caractérisé par ses dimensions impaires et ses coefficients, dont le centre correspond au pixel concerné. Les coefficients du tableau déterminent les propriétés du filtre.

Voici un exemple de filtre 3 x 3 :

1	1	1
1	<b>4</b>	1
1	1	1

Ainsi le produit de la matrice image, généralement la taille sera grande car représentant l'image initiale (tableau de pixels) par le filtre donne une matrice correspondant à l'image filtrée. [9]

Pour accélérer les traitements, il est possible de décomposer les filtres en sous-filtres équivalents qu'on passe un après l'autre. [9]

## 2.2-Principe de filtrage :

Le principe du filtrage est de modifier la valeur des pixels d'une image, généralement dans le but d'améliorer son aspect. En pratique, il s'agit de créer une nouvelle image en se servant des valeurs des pixels de l'image d'origine. [Bou1 09]

## 2.3-Pourquoi on doit filtrer une image ?

- Pour réduire le bruit dans l'image.
- Pour détecter les contours d'une image. [Bou1 09]

## 3-la convolution d'une image :

### 3. 1- définition:

C'est convoluer l'image avec la réponse impulsionnelle du filtre ou bien :  
La convolution discrète est un outil permettant la construction de filtres linéaires ou de filtres de déplacements invariants. [Bou1 09]

L'équation de convolution, notée  $g(x)$ , de la séquence  $f(x)$  avec une fonction  $h(x)$  est :

$$g(x) = f(x) * h(x) = \sum_{\text{all } k} h(x - k) f(k)$$

- $h(x)$  est appelée *masque de convolution, noyau de convolution, filtre, fenêtre, kernel, ...*

- $f(x)$  est la fonction d'origine et  $g(x)$  la fonction convoluée (résultat de la convolution)

En pratique, la convolution numérique d'une image se fera simplement par une **sommation de multiplications**. [Bou1 09]

### 3.2 Masque de convolution :

Le masque de convolution est le plus souvent :

- ✓ Carré
- ✓ De taille 3x3 ou 5x5 (ou plus, mais impair)
- ✓ Ce masque représente un filtre linéaire permettant de modifier l'image.

La plupart du temps, on divisera le résultat de la convolution par la somme des coefficients du masque :

- ✓ Pour éviter de modifier l'entropie de l'image [1], la somme des coefficients doit être égale à 1.
- ✓ Dans le cas du Laplacien [2], la somme sera égale à zéro. [Bou1 09]

### 3.3 Types de convolution :

a) Convolution linéaire : on considère que l'image est entourée de noir, donc de valeurs nulles : zéro-padding.

b) Convolution circulante : on considère que l'image est entourée d'elle-même. [Bou1 09]

## 4. Les différents types des filtres :

### 4.1 Les filtres passe-bas :

Les filtres passe-bas consistent à atténuer les composantes de l'image ayant une fréquence haute (pixels foncés). Ce type de filtrage est généralement utilisé pour atténuer le bruit de l'image

C'est la raison pour laquelle on parle habituellement de lissage (voir Figure 3.1). Les filtres moyennés sont un type de filtres passe-bas dont le principe est de faire la moyenne des valeurs des pixels avoisinants. Le résultat de ce filtre est une image plus floue. [9]



Image originale    Image filtrée passe-bas

Figure 3.1 : Filtrage par le filtre passe-bas

[1] : L'entropie d'une image : est un indicateur de sa complexité, Si l'image est uniforme et ne possède qu'une couleur, son entropie est nulle, plus l'entropie est élevé, plus l'image est "aléatoire".

[2] : Laplacien ou l'opérateur Laplacien est l'opérateur différentiel défini par l'application de l'opérateur gradient suivie de l'application de l'opérateur divergence.

## 4.2 Les filtres passe-haut :

À l'inverse des passe-bas, atténuent les composantes de basse fréquence de l'image et permettent notamment d'accentuer les détails et le contraste (voir figure 3.2), c'est la raison pour laquelle le terme de "*filtre d'accentuation*" est parfois utilisé.

- les filtres passe-bande : permettant d'obtenir la différence entre l'image originale et celle obtenue par application d'un filtre passe-bas.
- les filtres directionnels : appliquant une transformation selon une direction donnée. [9]

Image originale

Image filtrée passe-haut

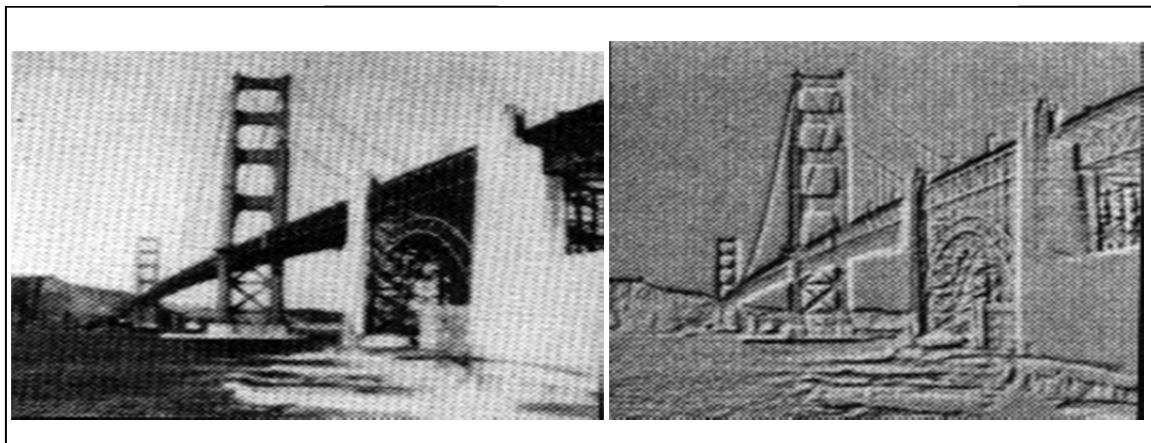


Figure3.2 : Filtrage par le filtre passe-haut

## 5. Types de filtrage :

### 5.1 Filtrage adaptatif :

C'est un filtre numérique dont les coefficients évoluent en fonction des signaux reçus. Ces coefficients seront estimés par des algorithmes récursifs, au sens d'un certain critère.

Les opérations de filtrage possédant une étape préalable de sélection des pixels. Une opération de filtrage adaptatif permet de réduire cette erreur tout en conservant les détails de la surface de l'objet. (Voir figure 3.3). [8]

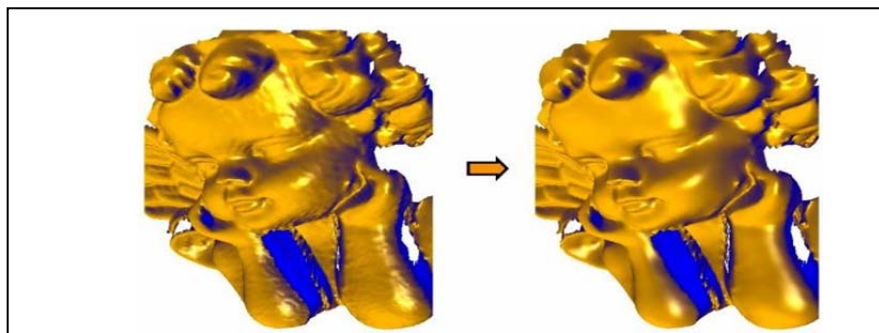


Image originale

image filtrée adaptative

Figure 3.3 : Filtrage par le filtre adaptatif

### 5.2 Filtrage des couleurs :

C'est un filtrage qui applique pour d'arrière-plan intelligent, la suppression des signatures, sont particulièrement utiles pour extraire les données de documents imprimés en couleurs ou sur des arrière-plans texturés. [5]

### 5.3 Le filtre moyenne :

Il Permet de lisser l'image (smoothing) et remplace chaque pixel par la valeur moyenne de ses voisins et réduire le bruit et les détails non-important (voir figure 3.4). [Bou1 09]

Exemple des filtres moyens :

1	1	1
1	1	1
1	1	1

Taille d'un filtre 3\*3

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Taille d'un filtre 5\*5

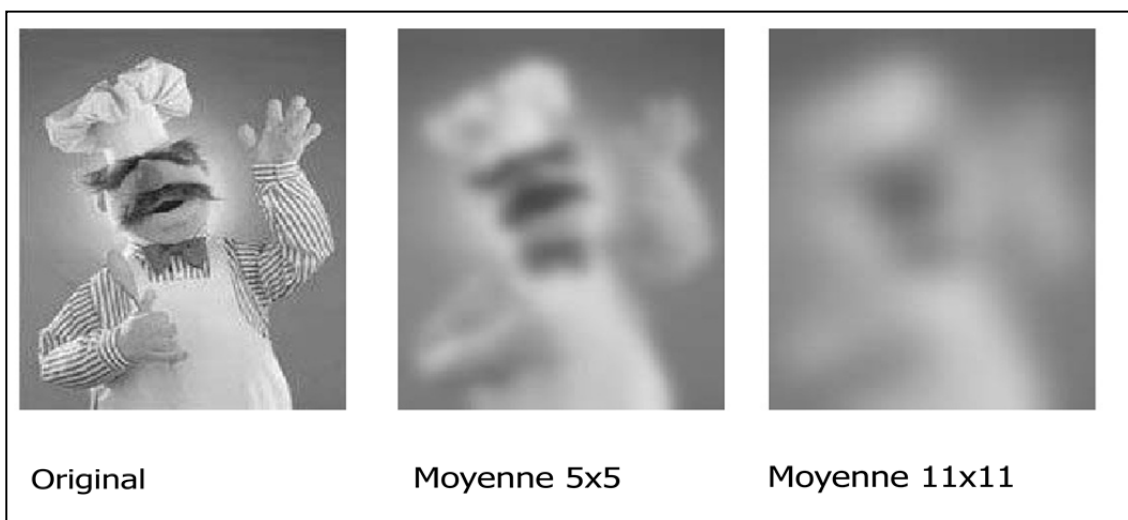


Figure 3.4 : Filtrage par le filtre moyen

### 5.4 Le filtre Gaussien :

C'est un filtre qui utilise une fonction gaussienne (Figure3.5) Le filtre gaussien donnera un meilleur lissage et une meilleure réduction du bruit que le filtre moyenne (voire figure3.6). [Bou1 09]

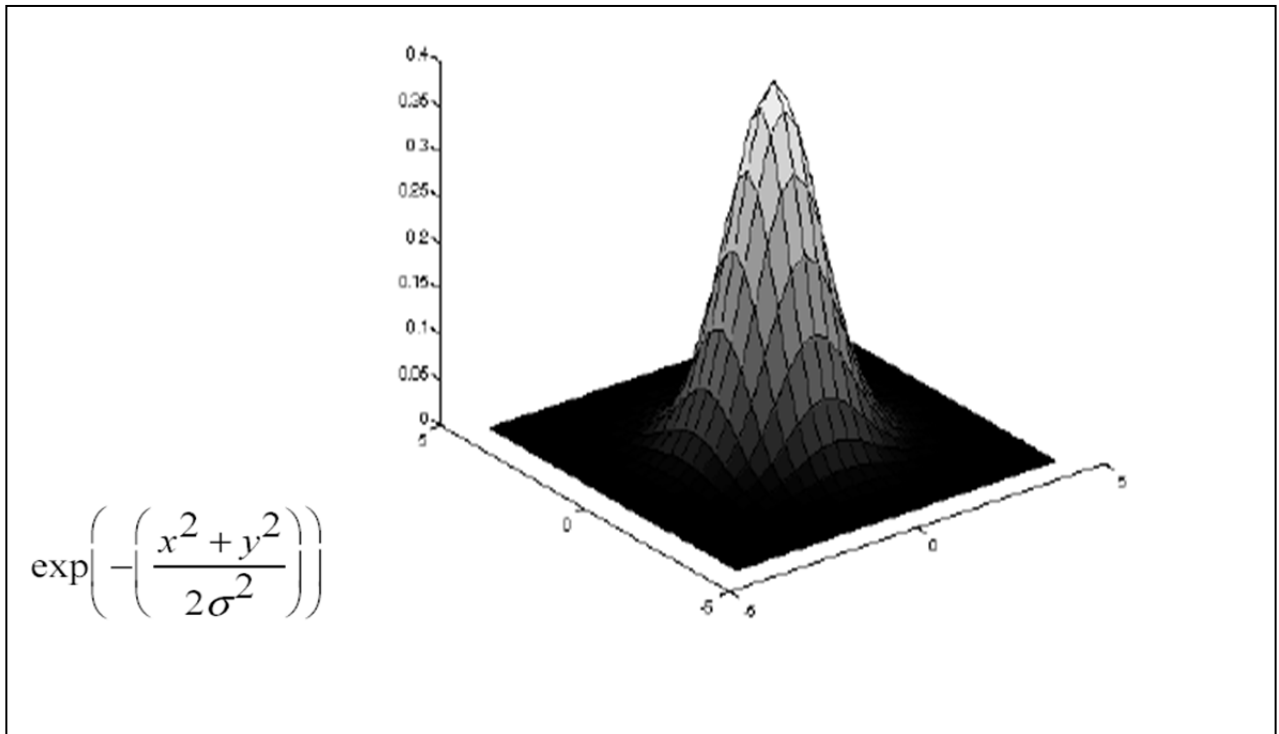


Figure3.5 : présentation de la forme gaussienne

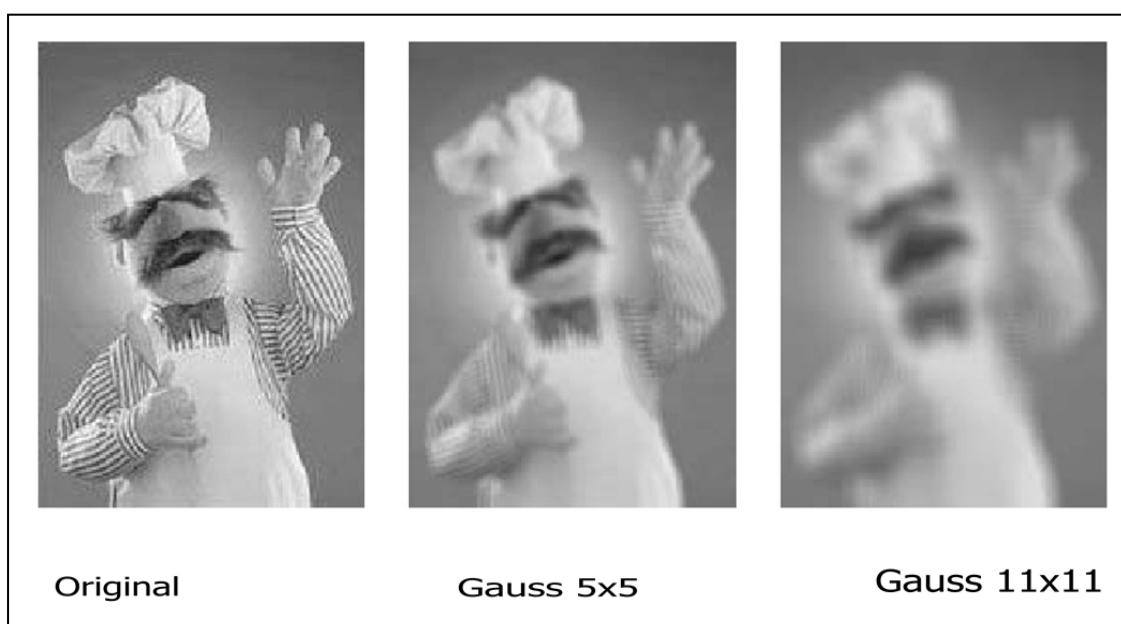


Figure 3.6 : Filtrage par le filtre gaussien

### 5.5 Le filtre médian :

Pour atténuer et réduire le bruit dans une image, il existe mieux que le filtre moyen ou le filtre gaussien. Il s'agit du filtre médian. (Voir figure 3.7)

C'est un filtre non-linéaire, qui ne peut pas s'implémenter comme un produit de convolution. On remplace la valeur d'un pixel par la valeur médiane dans son voisinage  $N \times N$  [Bou1 09]

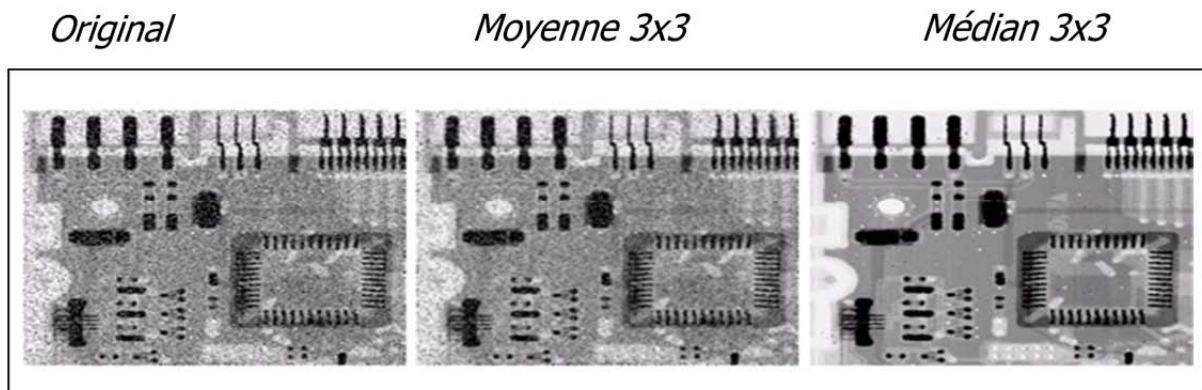


Figure3.7 : Filtrage par un filtre médian.

### 5.6 Le filtre de Canny : (ou détecteur de Canny)

Détection de contour : la détection de contour permet de repérer les différents objets que constituent la scène de l'image. Il existe de nombreuses méthodes pour trouver les contours des objets, on va discuter beaucoup plus sur le chapitre 04.

Ce filtre utilisé en traitement d'images pour la détection des contours (voire figure 3.8). Pour être optimal il faut définir clairement ces trois types critères :

1. *bonne détection* : faible taux d'erreur dans la signalisation des contours,
2. *bonne localisation* : minimisation des distances entre les contours détectés et les contours réels,
3. *Clarté de la réponse* : une seule réponse par contour et pas de faux positifs. [6]

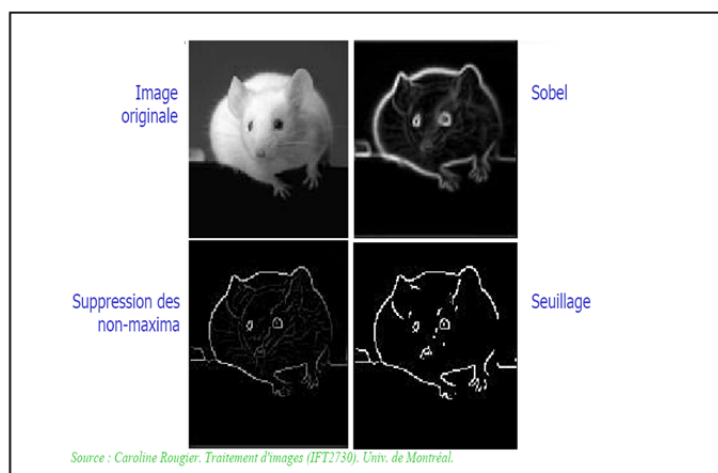


Figure 3.8 : Filtrage de sobel et canny.

Remarque : La suppression du non-maxima :\_est souvent utilisé en combinaison avec des algorithmes de détection de bord. L'image est balayée le long de la direction du gradient de l'image, et si pixels ne font pas partie des maxima locaux, ils sont mis à zéro. Cela a pour effet de supprimer toutes les informations d'image qui ne fait pas partie des maxima locaux.

## 6. Utilisation des filtres sous Matlab :

Ils existent plusieurs fonctions permettant l'opération de filtrage d'image comme : *fdesign*, *fspecial*, *imfilter*, mais avant de les appliquer on doit créer plusieurs bruits sur l'image.

Pour faire cela on doit utiliser Le bruit « poivre-et-sel », ce lui si consiste à mettre aléatoirement plusieurs pixels aux valeurs 255 ou 0 (valeurs extrêmes de l'intervalle des niveaux de gris). Ce type de bruit impulsionnelle peut apparaître par numérisation d'une image ou au cours d'une transmission.

Le scripte dessous permet de créer le bruit sur une image on utilisant la fonction *imnoise* :

```
t=[9 9];
I=imread('1.jpg');
K = rgb2gray(I);

Ib=imnoise(K, 'salt & pepper') ;

figure;subplot(1,2,1);

imshow(I);title('Image initiale');

subplot(1,2,2);imshow(Ib);title('Image bruitée');
```

Voici les résultats obtenus (voire Figure 3.9) :

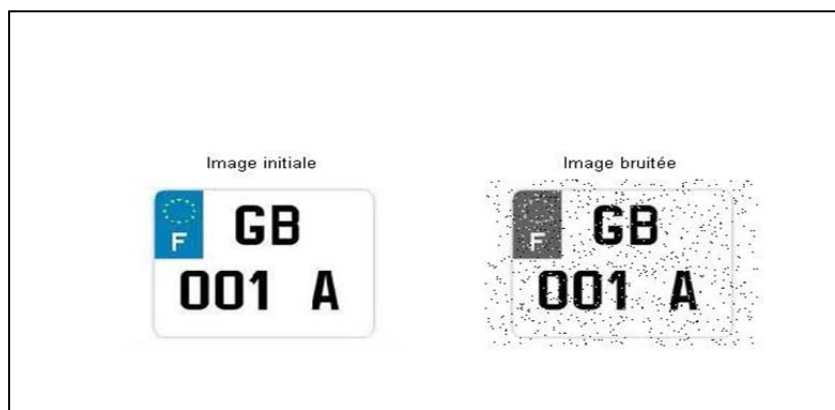


Figure 3. 9 : Application du bruit 'salt & pepper'.



Il existe plusieurs types des filtres, mais dans notre cas nous allons utiliser quatres filtres :

- 1) Filtre Gaussien
- 2) Filtre Moyenne
- 3) Filtre de sobel
- 4) Filtre de canny

## 6.1 Filtre Gaussien :

On utilise 2 fonctions : *fspecial* et *imfilter*.

La syntaxe de *fspecial* dans Matlab est :

```
H = fspecial('gaussian',HSIZE,SIGMA) .
```

Dont :

*HSIZE* : c'est la taille de filtre

*SIGMA* : c'est la tensité de bruit

Par défaut : *HSIZE* est [3 3] et *SIGMA* est 0.5.

La syntaxe de la fonction *imfilter* est :

```
B = imfilter(A,H)
```

A : l'image, H : le filtre multidimensionnel

Voilà l'exemple d'un script sur Matlab qui permet d'utiliser le filtre gaussien:

```
t=[9 9];

I=imread('1.jpg');
K = rgb2gray(I);

Ib=imnoise(K, 'salt & pepper') ;

figure;subplot(1,3,1);

imshow(I);title('Image initiale');

subplot(1,3,2);imshow(Ib);title('Image bruitée');

H = fspecial('Gaussian',t,1);
ImageG=imfilter(Ib,H);

subplot(1,3,3);
imshow(ImageG);title('filtrage Gaussien ');
```

Voici les résultats obtenus (figure 3.10 ,3.11,3.12) :

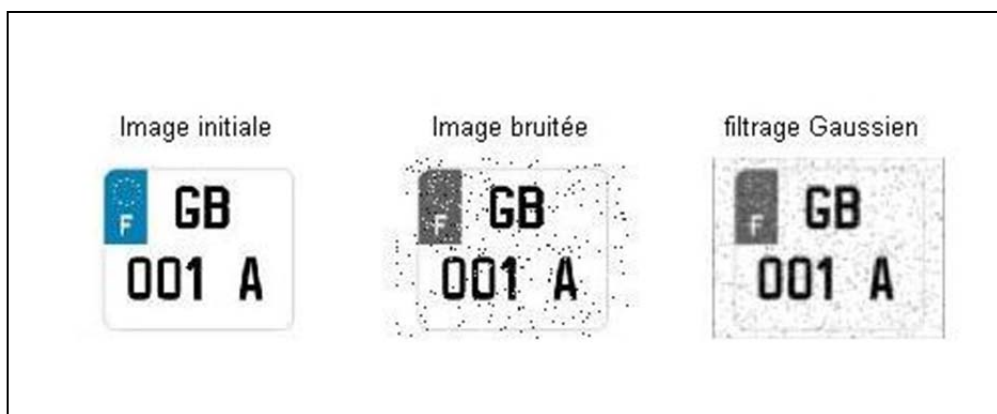


Figure 3.10 : image filtrée par le filtrage gaussien (SIGMA=1).

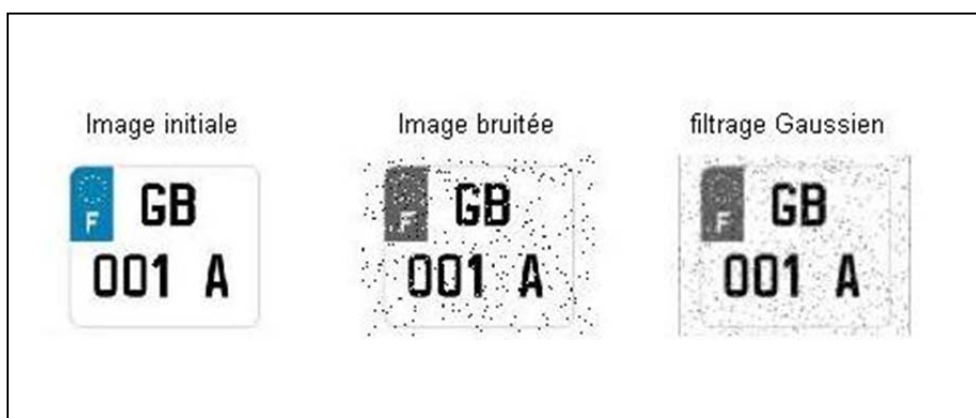


Figure 3.11 : image filtrée par le filtrage gaussien (SIGMA=0.75).

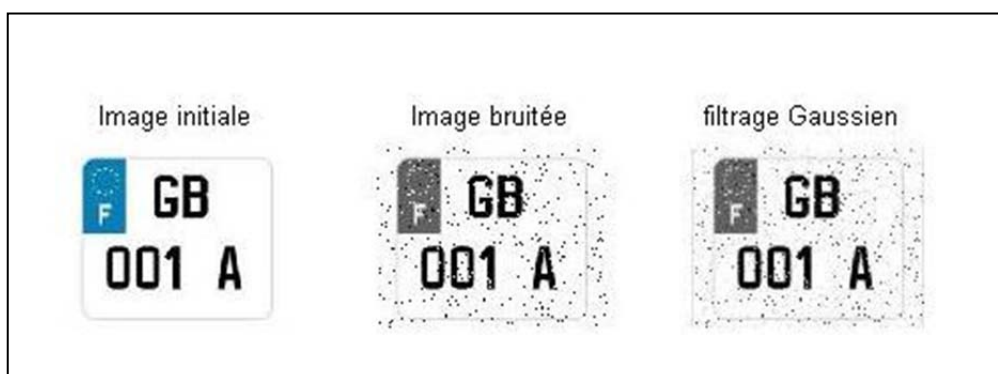


Figure 3.12 : image filtrée par le filtrage gaussien (SIGMA=0.5).

## 6.2 Filtre moyen :

Le syntaxe de ce filtre est : `H = fspecial ('average',HSIZE)`

Par exemple voilà le script sur Matlab qui permet d'utiliser le filtre moyenne:

```
t=[9 9];
I=imread('1.jpg');
K = rgb2gray(I);
Ib=imnoise(K, 'salt & pepper') ;
figure;subplot(1,3,1);

imshow(I);title('Image initiale');

subplot(1,3,2);imshow(Ib);title('Image bruitée');

H = fspecial('average',1);
ImageG=imfilter(Ib,H);

subplot(1,3,3);
imshow(ImageG);title('filtrage moyenne');
```

La figure ci-dessous illustre les résultats obtenu :

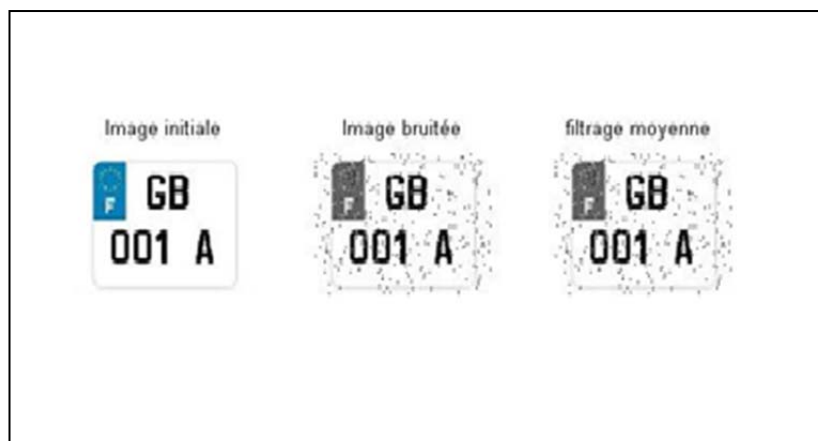


Figure 3.13 : image filtrée par le filtrage moyenne.

### 6.3 filtres de Sobel :

La commande utilisée pour ce filtre est : `H = fspecial ('sobel')`

Par exemple voilà le script sur Matlab qui permet d'utiliser le filtre Sobel:

```
t=[9 9];
I=imread('1.jpg');
K = rgb2gray(I);

Ib=imnoise(K, 'salt & pepper') ;

figure;subplot(1,3,1);

imshow(I);title('Image initiale');

subplot(1,3,2);imshow(Ib);title('Image bruitée');

H = fspecial('sobel');
ImageG=imfilter (Ib,H);

subplot(1,3,3);
imshow(ImageG);title('filtrage sobel');
```

Voici les résultats obtenus (voire figure 3.14) :

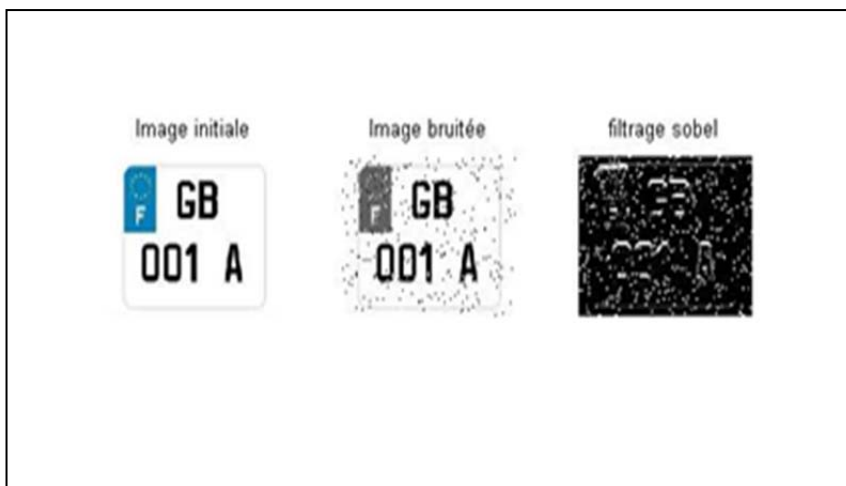


Figure 3.14 : image filtrée par le filtre de Sobel.

## 6.4 filtres de Canny:

La commande utilisée pour ce filtre est :`H= edge(Ib, 'canny');`

On utilise la fonction *edge* mais elle ne marche pas si et seulement si l'image est en niveau de gris et on ne doit pas mettre la fonction *imfilter*.

Voilà le script sur Matlab qui permet d'utiliser le filtre de Canny:

```
t=[9 9];
I=imread('1.jpg');
K = rgb2gray(I);

Ib=imnoise(K, 'salt & pepper') ;

figure;subplot(1,3,1);

imshow(I);title('Image initiale');

subplot(1,3,2);imshow(Ib);title('Image bruitée');

H = edge(Ib,'canny');

subplot(1,3,3);
imshow(H);title('filtrage canny');
```

voici les résultats qu'il a donné (voir figure 3.13):

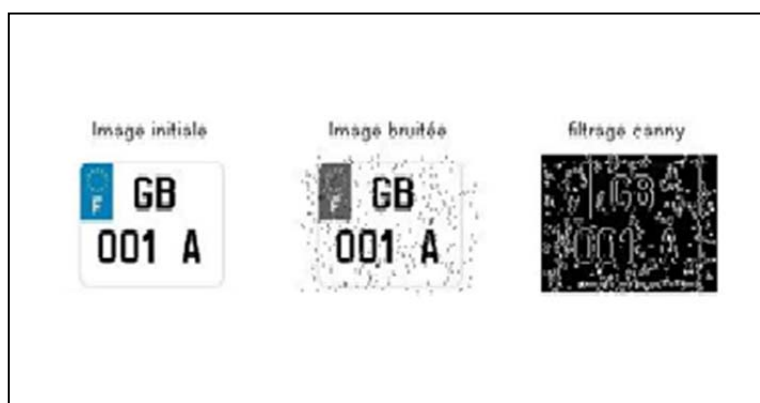


Figure 3.13 : image filtré par le filtre de Canny.

## 7. Conclusions :

Le filtrage c'est une étape très importante et nécessaire dans le traitement d'image alors il apparait tout d'abord que la qualité des filtres dépend des critères d'appréciation choisis (psycho-visuel, fréquentiel). On constate ensuite que l'usage que l'on veut faire de l'image débruitée conditionne complètement le choix du filtre (cf. exemple de la détection de contours).

On peut toutefois mettre en avant la robustesse du filtre médian qui donne de bons résultats dans tous les cas envisagés ici. On constate que c'est le seul filtre qui permet de débruiter une image corrompue simultanément.

**CHAPITRE 04 :**

**DÉTECTION**

**DE**

**CONTOUR**

## 1. Introduction :

Dans ce chapitre, nous allons présenter la détection de contour, ses méthodes et étudier comment les implémenter sous Matlab. Ce chapitre vise à répondre aux questions suivantes :

C'est quoi la détection de contours ?

Quelles sont les méthodes de détection de contours ?

Comment implémente les méthodes sous Matlab ?

## 2. Qu'est-ce qu'un contour ?

Un contour est une variation brusque d'intensité. [Bou 09].

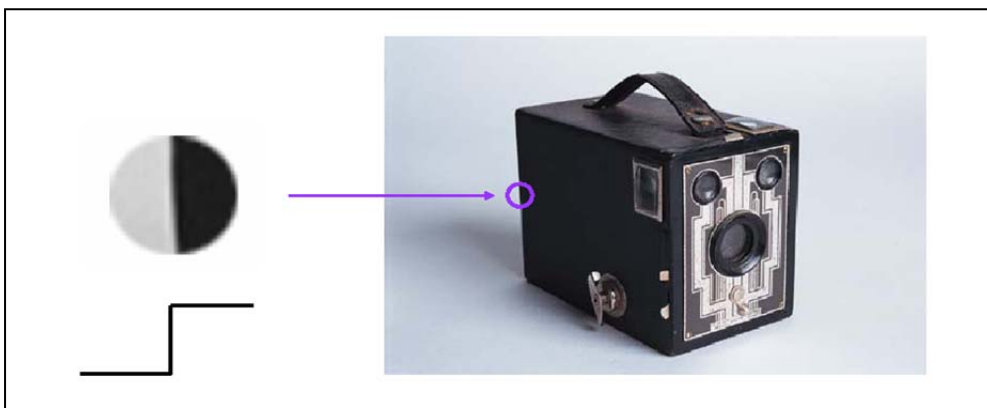


Figure 4.1 : exemple d'une détection de contour

### 2.1 Définition de la détection de contours:

La détection de contours est une technique de réduction d'information dans les images, qui consiste à transformer l'image en un ensemble de courbes, pas forcément fermées, formant les frontières significatives de l'image. [Man 10]

Si les structures extraites sont simples à manipuler (courbes fines, régulières, stables...), elles peuvent être utiles pour la mise en correspondance d'images (robotique, indexation,...). [Man 10].

### 3. Les méthodes de la détection de contours :

✚ Définition sure contour et régions :

#### a) Les discontinuités : contours

C'est les changements abrupts, frontières entre régions...

#### b) Les zones homogènes : régions

C'est les mêmes couleurs, textures, intensités, ...

- Dans les approches régions : Rechercher les zones dans l'image sur un critère d'homogénéité
- Dans les approches contours : Rechercher les discontinuités entre régions
- Dans les approches duales c'est régions et contours



Il existe de nombreuses méthodes pour trouver les contours des objets, la plupart sont basées sur les dérivées premières et secondes de l'image.

### 3.1 Dérivée première :

- Méthode de base de dérivée première :

L'opérateur de Roberts décrit dans une image rectangulaire, ayant un contour rampe, les dérivées premières elles-mêmes sont d'un intérêt limité car la pente maximale a peu de chances de se trouver sur l'une des deux directions considérées. Ce qui importe c'est la longueur du vecteur gradient dont elles sont les composantes. Cette longueur se calcule en principe par le théorème de Pythagore au prix d'un calcul sur les réels et on l'accélère considérablement en utilisant une approximation entière :

$$|G| = |E - W| + |N - S|$$

Indépendamment de la précision du calcul, il faut transformer ce résultat en un filtre numérique. La notion physique de filtre correspond à la notion mathématique de convolution. Lorsqu'il s'agit de données numérisées comme dans le cas du traitement d'images, la relation entre les valeurs des pixels de sortie et celle des pixels d'entrée est décrite par un tableau de nombres appelé matrice de convolution. [10]

#### 3.1.1 Filtre de Prewitt :

La matrice qui correspond au filtrage horizontal, faisant ressortir essentiellement les contours verticaux, selon l'opérateur de Prewitt, s'écrit  $h_x = [-1 \ 0 \ 1]$  tandis que la matrice verticale  $h_y$  est sa transposée.

Les deux convolutions avec le tableau de valeurs initiales créent deux tableaux  $G_x$  et  $G_y$  à l'origine du tableau  $G$  sur lequel on peut localiser les maximums. [10]

#### 3.1.2 Filtre de Sobel :

On obtient un meilleur résultat en remplaçant le filtre rectangulaire par un filtre triangulaire. [10]

#### 3.1.3 Filtre de Canny :

Le filtre de Canny a été bâti autour de l'algorithme de Sobel pour améliorer ses résultats. La méthode de Canny elle permet d'éliminer des faux contours. En considérant non seulement l'intensité du gradient mais aussi sa direction, il est possible d'éliminer un pixel qui pointe vers deux pixels de valeur supérieure car ce n'est pas un maximum local. Il faut ensuite effectuer un seuillage par hystérésis. Pour cela on fixe deux seuils, un seuil haut  $S_h$  et un seuil bas  $S_b$ . On commence par sélectionner les points qui dépassent le seuil haut et on applique ensuite le seuil bas en ne conservant que les composantes connexes qui contiennent un point au-dessus de  $S_h$ . En d'autres termes à partir de chaque point au-dessus de  $S_h$  on "suit" un chemin constitué de points au-dessus de  $S_b$ , ce chemin est le contour recherché. [10]

### 3.2 Dérivée secondes :

A priori, l'utilisation des dérivées secondes est plus commode que celle des dérivées premières : au lieu de chercher les maximums de l'intensité du gradient, on cherche l'annulation du Laplacien, somme des dérivées secondes dans deux directions.

Soit  $N + S + E + W - 4C$ . La contrepartie de la simplicité est l'amplification du bruit par la seconde dérivation. Ce bruit est atténué par l'adjonction d'un filtre gaussien mais les méthodes du second ordre ne semblent pas compétitives avec celles du premier ordre évoquées précédemment. [10]

### 4. implémenter le contour sous Matlab :

La détection de contours sous Matlab se fait par le la fonction *edge*, nous allons utiliser deux méthode de détection de contour sont :

- détecteur de Canny
- détecteur de Prewitt

#### 4.1 filtre de Canny :

Voici le script sous Matlab qui permet d'utiliser ce filtre pour détecter le contour :

```
I=imread('1.jpg');
subplot(1,2,1);
imshow(I);title('Image initiale');
Im=rgb2gray(I);
BW1= edge(Im,'canny');
subplot(1,2,2);
imshow(BW1);title('détecteur de canny');
```

La figure ci-dessous illustre les résultats obtenus :

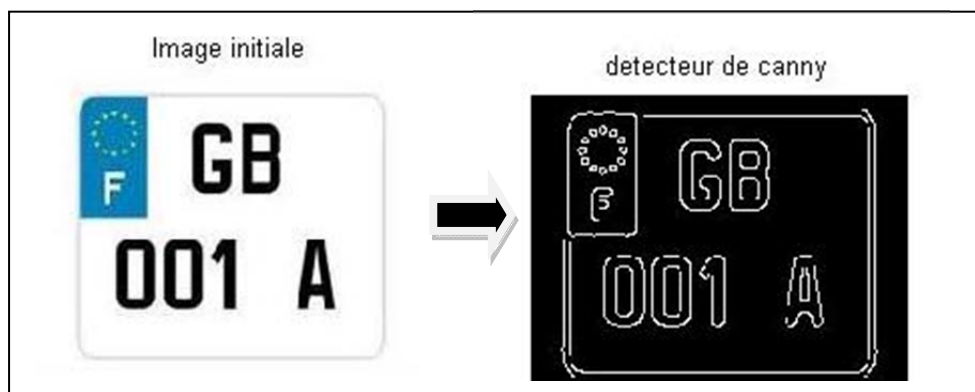


Figure 4.2 : exemple sur le détecteur de Canny

## 4.2 Détecteur de Prewitt :

Voici le script sous Matlab qui permet d'utiliser ce filtre pour détecter le contour :

```
I=imread('1.jpg');  
subplot(1,2,1);  
imshow(I);title('Image initiale');  
Im=rgb2gray(I);  
BW1= edge(Im,'prewitt');  
subplot(1,2,2);  
imshow(BW1);title('détecteur de prewitt');
```

La figure ci-dessous illustre les résultats obtenus :

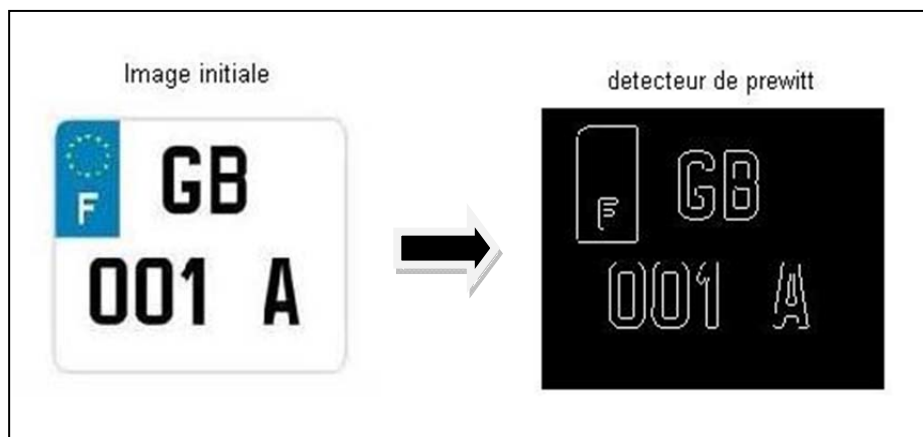


Figure 4.3 : exemple sur le détecteur de Prewitt

## 5. Conclusion :

Nous avons présenté quelques méthodes permettant d'extraire les contours d'une image, puis comment appliquer ces méthodes dans Matlab et les résultats de chaque méthode.

La détection de contour est très utile en traitement d'images, c'est par exemple une étape indispensable à la reconnaissance de formes. Ces algorithmes sont également utilisés en imagerie médicale, cartographie, ...

**RÉFÉRENCES**  
**BIBLIOGRAPHIQUES**  
**&**  
**WEBOGRAPHIQUES**

## Webographie:

- [1]:<http://fr.wikipedia.org>.
- [2]:<http://dictionnaire.sensagent.com/pr%C3%A9traitement+d+image/fr-fr/>
- [3]:[http://www2.ulg.ac.be/telecom/teaching/notes/totali/elen016/node166\\_mn.html](http://www2.ulg.ac.be/telecom/teaching/notes/totali/elen016/node166_mn.html)
- [4]:[http://lhe.epfl.ch/enseignement/index.php?option=com\\_content&view=article&id=82:traitement-dimages-avec-matlab&catid=6:matlab&Itemid=13&lang=fr](http://lhe.epfl.ch/enseignement/index.php?option=com_content&view=article&id=82:traitement-dimages-avec-matlab&catid=6:matlab&Itemid=13&lang=fr)
- [5]:<http://france.abbyy.com/flexicapture-engine/acquisition-donnees/pretraitement-images/>
- [6]:[http://fr.wikipedia.org/wiki/Filtre\\_de\\_Canny](http://fr.wikipedia.org/wiki/Filtre_de_Canny)
- [7]:[http://fr.wikipedia.org/wiki/Seuillage\\_d%27image](http://fr.wikipedia.org/wiki/Seuillage_d%27image)
- [8]:<http://www.commentcamarche.net/contents/video/filtres.php3>
- [9] <http://www.commentcamarche.net/contents/1199-les-filtres-graphiques>
- [10]:[http://fr.wikipedia.org/wiki/D%C3%A9tection\\_de\\_contours](http://fr.wikipedia.org/wiki/D%C3%A9tection_de_contours)

## Bibliographie:

- [Mic 08]: Jacques Michelet, cours du prétraitement, 2008-04-19.
  - [Bou 09] : Alain Boucher, cours traitement d'image IFI, 2009.
  - [Van1 09]: Vandebroucke Nicolas, cours Traitement d'images sous Matlab, 2009.
  - [Van2 09]: Vandebroucke Nicolas, cours Traitement d'images, 2009.
  - [Dib 13]: A. Dib, cours d'infographie 2013.
  - [Bou1 09] : Alain Boucher, traitement d'images et vision par ordinateur IFI, 2009
  - [Jou 11] : Nicholas Journet, introduction au traitement d'images, 12 janvier 2011.
  - [Dub 12] : J Séverine Dubuisson, cours filtrage d'images, 29 octobre 2012.
  - [Man 10] : Antoine MANZANERA, Cours TERI – Master 2 IAD, 2010
- Help Matlab.

## ***Résumé***

Le Travail présenté dans ce mémoire expose les techniques et les méthodes utilisées pour le traitement d'image. Ces méthodes vont être utilisées sur des images de pièces d'immatriculation. On propose beaucoup de techniques qu'utilise Matlab tout au long du processus de traitement d'images.

Cette expérience était très utile et bénéfique pour nous, elle nous a permis de tester et d'expérimenter des méthodes et les techniques de traitement d'image, comparer les résultats et pouvoir les interpréter.

Finalement, le diplôme que nous venons d'obtenir sera pour nous une ouverture sur le domaine de traitement d'image, et une expérience afin de faire part à d'autres stagiaires de notre domaine.

### ***MOTS-CLÉS :***

Prétraitement, Amélioration, Restauration, Segmentation, Binarisation, Seuil, Histogramme, Filtrage, Convolution, Image.