

MS/006.3-35/01

Université Abou Bekr Belkaid



جامعة أبي بكر بلقايد

تلمسان الجزائر

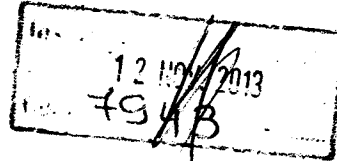
République Algérienne Démocratique et Populaire

Université Abou Bakr Belkaid- Tlemcen

Faculté des Sciences

Département d'Informatique

Ins:	
Date:	17 DEC. 2016
Code:	212



Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

Option: M.I.D

*Thème*

**Segmentation d'Image par la Méthode de Croissance de Région**

Réalisé par :

- Mme REZIGUE WAFAA

Présenté le 19 Septembre 2013 devant le jury composé de MM.

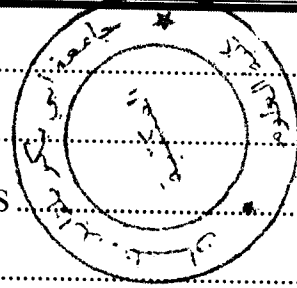
- Mr Benazzouz Mortada (Président)
- Mr Benziane yaghmorassen (Encadreur)
- Mr benaissa mohamed (Examineur)
- Mr Belabed amine (Examineur)



Année universitaire : 2012-2013

# TABLE DES MATIERES

LA LISTE DES FIGURES.....	4
INTRODUCTION GENERALE .....	7
CHAPITRE I : INTRODUCTION AUX TRAITEMENTS D'IMAGES .....	9
1. Introduction.....	9
2. Historique .....	10
3. Définition de l'image .....	11
3.1. Représentation des images numériques.....	11
3.2. Caractéristiques d'une image numérique .....	12
3.2.1. Pixel.....	12
3.2.2. Dimension .....	13
3.2.3. Résolution .....	13
3.3. La taille d'une image .....	14
3.4. Bruit.....	14
3.5. Histogramme.....	14
3.6. Contours et textures .....	15
3.7. Luminance .....	15
3.8. Images à niveaux de gris.....	15
4. Traitement d'image .....	16
4.1. Type de données manipulées.....	16
4.2. Acquisition d'une image .....	16
4.3. Opérateurs de traitement d'images.....	17
4.3.1. Opérateurs point à point.....	18
4.3.2. Opérateurs locaux .....	18
4.3.3. Les filtres usuels en traitement d'images.....	19
4.3.4. Amélioration d'une image: les filtres passe-bas.....	23
4.4. Construction d'une application de traitement d'images.....	25
5. Histogramme et Binarisation.....	25



# TABLE DES MATIERES

---

5.1. Types d'histogrammes .....	25
5.2. Égalisation d'histogramme .....	26
6. Le seuillage d'une image .....	26
7. La multi résolution .....	27
8. Conclusion .....	28
CHAPITRE II : LES METHODES DE SEGMENTATION D'IMAGES .....	28
1. Introduction .....	28
2. Segmentation d'images .....	29
3. Les Méthodes de Segmentation d'Image .....	30
3.1. Segmentation par approche « région » .....	30
3.1.1. Croissance de régions (growing-region) .....	31
L'algorithme de la croissance est un procédé itératif: .....	32
3.1.2. Ligne de partage des eaux (watershed) .....	34
3.2. Segmentation par approche « frontière » .....	38
3.3. Segmentation par classification ou seuillage .....	39
4. Conclusion .....	39
CHAPITRE III : CHAPITRE D'IMPLEMENTATION .....	40
1. Introduction .....	40
1.1. Définition du langage de Programmation Matlab .....	40
2. Traitement d'image .....	40
2.1. Chargement d'une image .....	40
2.2. Enregistrement d'une image .....	40
2.3. Affichage d'image .....	41
2.4. Conversion d'image RGB en une image d'intensité .....	41
2.5. Calcule et affichage d'histogramme .....	42
2.6. Egalisation d'image (contraste) .....	42
2.7. Ajustement du contraste d'une image .....	43

# TABLE DES MATIERES

2.8.	Le Seuillage .....	44
2.9.	Etiquetage.....	44
2.10.	Filtrage.....	45
2.10.1.	Le filtrage Gaussienne.....	45
2.10.2.	Le filtrage passe bas.....	46
2.10.3.	Le filtrage passe haut « MEILLEUR ».....	47
2.10.4.	Filtre passe bas de Butterworth.....	48
2.10.5.	Filtre passe haut de Butterworth.....	48
2.11.	Extraction de tous les régions d'une image .....	49
2.12.	Application de la méthode de croissance de région .....	49
2.12.1.	Le premier exemple .....	50
2.12.2.	Le deuxième exemple .....	50
2.13.	La multi Resolution .....	51
2.13.1.	Extraction de région avec image1/2 .....	51
2.13.2.	Extraction de région avec image1/4 .....	51
2.13.3.	Application de la méthode de croissance de région sur image1/2 .....	52
2.13.4.	Application de la méthode de croissance de région sur image1/4 .....	52
2.14.	Exemple de la méthode de ligne de partage des eaux « WATERSHED » .....	53
2.14.1.	Algorithme de ligne de partage des eaux « Watershed » .....	53
3.	Conclusion .....	55
	CONCLUSION GENERALE .....	56
	REFERENCES BIBLIOGRAPHIQUES.....	57

# La liste des Figures

---

## Chapitre I

**Figure 1.1 : Représentation d'un groupe de pixels**

**Figure 2.1 : Exemple d'histogramme d'une image**

**Figure 3.1 : Image originale**

**Figure 4.1 : Seuillage d'une image**

**Figure 5.1 : Zoom d'une partie d'image**

**Figure 6.1 : profil avec les 3 composantes R,V,B**

**Figure 7.1 : Filtrage passe bas moyenneur**

**Figure 8.1 : Le profil d'image en diagonale**

**Figure 9.1 : Suppression du bruit par un filtrage passe-bas médian**

**Figure 10.1 : Figure 10.1 : Le profil d'image en diagonale**

## Chapitre II

**Figure 1.2 : Points de départ : centre des blocs de la décomposition en blocs homogènes**

**Figure 2.2 : Croissance progressive d'une région**

**Figure 3.2 : Résultat final de la croissance de régions (827 régions)**

**Figure 4.2 : Représentation d'une image sous forme de terrain 3D et remplissage des bassins avec de l'eau**

**Figure 5.2 : Ligne de partage des eaux entre 2 bassins**

**Figure 6.2 : Carte d'élévation. L'intensité reflète l'altitude (noir=le plus bas, blanc=le plus haut)**

**Figure 7.2 : Résultats intermédiaires du remplissage des bassins**

**Figure 8.2 : Résultat final de la ligne de partage des eaux (806 régions)**

# La liste des Figures

---

## Chapitre III

- Figure 1.3 : Affichage d'une image RGB**
- Figure 2.3 : Conversion d'une image RGB vers une image d'intensité**
- Figure 3.3 : Affichage de l'histogramme**
- Figure 4.3 : Egalisation d'Image et Histogramme Egaliser**
- Figure 5.3 : Ajustement du contraste d'une image**
- Figure 6.3 : Le seuillage d'une image**
- Figure 7.3 : Application du filtrage Gaussienne**
- Figure 8.3 : Application du filtrage passe bas**
- Figure 9.3 : Application du filtrage passe haut « MEILLEUR »**
- Figure 10.3 : Application du Filtre passe bas de Butterworth**
- Figure 11.3 : Application du Filtre passe haut de Butterworth**
- Figure 12.3 : Extraction des régions d'une image**
- Figure 13.3 : La région sélectionnée par region growing**
- Figure 13.4 : La région sélectionnée par region growing**
- Figure 15.3 : La région sélectionnée par region growing**
- Figure 16.3 : La région sélectionnée par la méthode de croissance de région**
- Figure 17.3 : Extraction de région en utilisons la multi résolution Image1/2**
- Figure 18.3 : Extraction de région en utilisons la multi résolution Image1/4**
- Figure 19.3 : croissance de région sur une image1/2**
- Figure 20.3 : croissance de région sur une image1/4**
- Figure 21.3 : comparaison entre image originale & image négative**
- Figure 22.3 : watershed & transformée de distance/ superposé watershed et l'image originale (bassins)**

# La liste des Figures

---

**Figure 23.3: watershed- norme gradient / superposée watershed et l'image originale (bassins)**

**Figure 24.3: Watershed-les marqueurs controller , watershed-gradient et le marqueur contrôlée**

**Figure 25.3: superpose watershed et l'image originale**

**INTRODUCTION**

**GENERALE**



# Introduction Générale

---

- **Stabilité** : la segmentation obtenue ne doit pas varier beaucoup lorsque les conditions d'acquisition varie légèrement (bruit, illumination, point de vue,...)
- **Régularité** : les régions obtenues doivent être simples à manipuler (taille suffisante, forme régulière,...) [2]

## Problématique

Dans le cadre de ce projet, nous nous concentrons sur la segmentation d'image. Le système proposé est de faire la segmentation d'image sur des images différents et de différents résolutions.

## Contribution

Notre solution est constituée de deux phases :

Le traitement d'image

Utilisation de la méthode de croissance de région « Région growing »

## Plan du mémoire

Notre mémoire est constitué des chapitres :

Le premier chapitre est une présentation générale du traitement d'image telle que la définition d'image avec ses différents caractéristiques ainsi que les différents méthodes de traitements d'image (filtrage, le seuillage, étiquetage...).

Dans le second chapitre nous avons expliqué les principales méthodes de segmentation d'image.

Dans le troisième chapitre nous allons présenter les différentes étapes suivies pour la segmentation d'image. En l'occurrence nous citons autres méthodes d'extraction de région et d'objets contenant l'image. Nous montrons aussi les résultats obtenus durant la segmentation.

# CHAPITRE I

## **1. Introduction**

Le traitement d'images est une discipline de l'informatique et des mathématiques appliquées qui étudie les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information.

Il s'agit d'un sous-ensemble du traitement du signal dédié aux images et aux données dérivées comme la vidéo (par opposition aux parties du traitement du signal consacrées à d'autres types de données : son et autres signaux monodimensionnels notamment), tout en opérant dans le domaine numérique (par opposition aux techniques analogiques de traitement du signal, comme la photographie ou la télévision traditionnelles).

Dans le contexte de la vision artificielle, le traitement d'images se place après les étapes d'acquisition et de numérisation, assurant les transformations d'images et la partie de calcul permettant d'aller vers une interprétation des images traitées. Cette phase d'interprétation est d'ailleurs de plus en plus intégrée dans le traitement d'images, en faisant appel notamment à l'intelligence artificielle pour manipuler des connaissances, principalement sur les informations dont on dispose à propos de ce que représentent les images traitées (connaissance du « domaine »).

La compréhension du traitement d'images commence par la compréhension de ce qu'est une image. Le mode et les conditions d'acquisition et de numérisation des images traitées conditionnent largement les opérations qu'il faudra réaliser pour extraire de l'information. En effet, de nombreux paramètres entrent en compte, les principaux étant :

- la résolution d'acquisition et le mode de codage utilisé lors de la numérisation, qui déterminent le degré de précision des éventuelles mesures de dimensions,
- les réglages optiques utilisés, (dont la mise au point) qui déterminent par exemple la netteté de l'image,
- les conditions d'éclairage, qui déterminent une partie de la variabilité des images traitées,
- le bruit de la chaîne de transmission d'image. [3]

# CHAPITRE I : INTRODUCTION AU TRAITEMENT D'IMAGES

---

On désigne par traitement d'images numériques l'ensemble des techniques permettant de modifier une image numérique dans le but de l'améliorer ou d'en extraire des informations. [a]

Une image est avant tout un signal 2D, mais Souvent, cette image représente une scène 3D.

D'un point de vue mathématique :

- ✓ une image est une matrice de nombres représentant un signal.
- ✓ plusieurs outils permettent de manipuler ce signal

D'un point de vue humain :

- ✓ une image contient plusieurs informations sémantiques.
- ✓ il faut interpréter l'information au-delà de la valeur des nombres. [a]

D'autre part le traitement d'images est défini comme suit :

- ✓ Amélioration, restauration, d'images.
- ✓ Extraction d'information, reconnaissance, des formes.
- ✓ Visualisation, reconstruction 3D, image de synthèse. [3]

## 2. Historique

Le traitement d'images 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 temps de transfert passe ainsi de plus d'une semaine à moins de trois heures. Il n'y a pas vraiment d'évolution par la suite jusqu'à la période d'après-guerre. [c]

Le traitement du signal prend de l'importance vers la fin de la Seconde Guerre mondiale avec l'arrivée du radar. La prospection pétrolière participe aussi beaucoup au développement des techniques de traitement du signal.

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. Peu après, la redécouverte de la transformée de Fourier rapide (FFT) révolutionne le domaine, en rendant possible les manipulations du contenu fréquentiel des signaux sur ordinateur. Cependant, l'essentiel des recherches porte encore, à cette époque, sur l'amélioration des images et leur compression.

# CHAPITRE I : INTRODUCTION AU TRAITEMENT D'IMAGES

En 1980, David Marr formalise le premier la détection de contours de manière précise (D. Marr et E. Hildreth : *Theory of Edge Detection*, Proc. R. Soc. London, B 207, 187-217, 1980). Au cours des années 1980, un véritable engouement se fait jour pour le traitement de l'image et surtout pour la compréhension de l'image par des systèmes experts. Les ambitions étaient beaucoup trop grandes, l'échec fut d'autant plus cuisant.

Les années 1990 sont témoin de l'amélioration constante des opérateurs. La recherche médicale devient un très gros demandeur en traitement d'images pour améliorer les diagnostics faits à partir des nombreuses techniques d'imagerie médicale, la technique reine étant l'IRM. Les publicitaires, puis le grand public se familiarisent avec la retouche d'image grâce au logiciel Photoshop, et le traitement d'images dans un objectif esthétique se répand avec l'apparition d'autres logiciels dédiés (The Gimp, Paint Shop Pro). Enfin, la décennie s'achève sur l'engouement pour les ondelettes et les images multimodales. [3]

## 3. Définition de l'image

L'image est une représentation d'une personne ou d'un objet par la peinture, la sculpture, le dessin, la photographie, le film, etc.

C'est aussi un ensemble structuré d'informations qui, après affichage sur l'écran, ont une signification pour l'œil humain. [c]

### 3.1. Représentation des images numériques

une image numérique est une matrice de pixels repérés par leur coordonnées  $(x,y)$ .

S'il s'agit d'une image couleur, un pixel est codé par 3 composantes  $(r,g,b)$  (chacune comprise au sens large entre 0 et 255), représentant respectivement les "doses" de rouge, vert et bleu qui caractérisent la couleur du pixel.

S'il s'agit d'une image en niveau de gris, il est codé par 1 composante comprise au sens large entre 0 et 255, représentant la luminosité du pixel. [5]

Une image est un signal bidimensionnel ou tridimensionnel. Or, afin de pouvoir réaliser des traitements informatiques sur une image, celle-ci doit être absolument numérique ou numérisée.

La numérisation d'une image consiste à convertir les valeurs continues du signal de cette dernière  $i$  (son état analogique) en des valeurs discontinues  $I$  qui correspond à une structure de données informatiques. Par exemple, pour un objet plan, carré, de 3x3 cm de côté, nous

# CHAPITRE I : INTRODUCTION AU TRAITEMENT D'IMAGES

considérons sa décomposition en 9 petits carrés élémentaires de  $1 \text{ cm}^2$ , appelés pixels. Pour chacun de ces derniers une valeur entière  $n$  est déterminée. Les 9 valeurs de  $I$  obtenues sont disposées en une structure finie appelée matrice, repérée par leur position à l'intersection d'une ligne et d'une colonne formant une image. Ainsi, une image numérique est définie par: le nombre de pixels qui la composent en largeur et en hauteur. la valeur que peut prendre chaque pixel. Elle est représentée par un scalaire dans le cas d'images en niveau de gris et par un vecteur à trois composantes Rouge, Vert et Bleu dans le cas d'images couleur. Ces valeurs sont incluses dans  $N$ .

Cette représentation est certes approchée mais parfaitement adaptée aux possibilités de traitement mathématique qu'apportent les ordinateurs. En ce qui concerne le présent travail, nous nous intéressons aux images numériques couleurs dans l'espace RVB<sup>1</sup> 2D qui sont le résultat d'une acquisition numérique avec une caméra CCD<sup>2</sup>. [6]

## 3.2. Caractéristiques d'une image numérique

### 3.2.1. Pixel

Contraction de l'expression anglaise " Picture Elements ": éléments d'image, le pixel est le plus petit point de l'image, c'est une entité calculable qui peut recevoir une structure et une quantification. Si le bit est la plus petite unité d'information que peut traiter un ordinateur, le pixel est le plus petit élément que peuvent manipuler les matériels et logiciels d'affichage ou d'impression. La lettre A, par exemple, peut être affichée comme un groupe de pixels dans la figure ci-dessous :



Figure 1.1 : Représentation d'un groupe de pixels

La quantité d'information que véhicule chaque pixel donne des nuances entre images monochromes et images couleurs. Dans le cas d'une image monochrome, chaque pixel est codé sur un octet, et la taille mémoire nécessaire pour afficher une telle image est directement liée à la taille de l'image.

<sup>1</sup> [http://fr.wikipedia.org/wiki/Espace\\_colorimétrique](http://fr.wikipedia.org/wiki/Espace_colorimétrique)

<sup>2</sup> [http://fr.wikipedia.org/wiki/Capteur\\_photographique](http://fr.wikipedia.org/wiki/Capteur_photographique)

# CHAPITRE I : INTRODUCTION AU TRAITEMENT D'IMAGES

Dans une image couleur (R.V.B.), un pixel peut être représenté sur trois octets : un octet pour chacune des couleurs : rouge (R), vert (V) et bleu (B). [7]

## **3.2.2. Dimension**

C'est la taille de l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image. [7]

## **3.2.3. Résolution**

La résolution est définie par un nombre de pixels par unité de longueur de l'image à numériser en dpi (dots per inch) ou ppp (points par pouce)]. On parle de définition pour un écran et de résolution pour une image.

Plus le nombre de pixels est élevé par unité de longueur de l'image à numériser, plus la quantité d'information qui décrit l'image est importante et plus la résolution est élevée (et plus le poids de l'image est élevé).

La résolution d'une image correspond au niveau de détail qui va être représenté sur cette image. Pour la numérisation il faut considérer les 2 équations suivantes :

$$(X * \text{résolution}) = x \text{ pixels}$$

$$(Y * \text{résolution}) = y \text{ pixels}$$

Où X et Y représentent la taille (pouce ou cm, un pouce=2,54 centimètres) de la structure à numériser, où résolution représente la résolution de numérisation, et où x et y représentent la taille (en pixels) de l'image. [7]

## **3.3. La taille d'une image**

Pour connaître la taille d'une image, il est nécessaire de compter le nombre de pixels que contient l'image, cela revient à calculer le nombre des cases du tableau, soit la hauteur de celui-ci que multiplie sa largeur. La taille de l'image est alors le nombre des pixels que multiplie la taille (en octet) de chacun de ces éléments.

Exemple : pour une image de 240 X 420 en True Color :

Nombre de pixels :

$$240 \times 420 = 100800$$

Taille de chaque pixel : 24 bits / 8 = 3 octets

Le poids de l'image est ainsi égal à :

$$100800 \times 3 = 302.400 \text{ égal } 302.400/1024 = 295 \text{ Ko. [7]}$$

### 3.4. Bruit

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur. [7]

### 3.5. Histogramme

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image. Pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image, on modifie souvent l'histogramme correspondant. Il permet de donner un grand nombre d'information sur la distribution des niveaux de gris (couleur) et de voir entre quelles bornes est répartie la majorité des niveaux de gris (couleur) dans les cas d'une image trop claire ou d'une image trop foncée. Il peut être utilisé pour améliorer la qualité d'une image (Rehaussement d'image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci.[g]

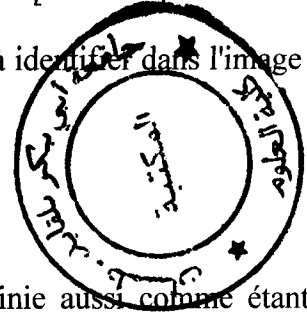


Figure 2.1 : Exemple d'histogramme d'une image



## 3.6. Contours et textures

Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative [GRA 91]. Les textures décrivent la structure de ceux-ci. L'extraction de contour consiste à identifier dans l'image les points qui séparent deux textures différentes. [h]



## 3.7. Luminance

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet. Une bonne luminance se caractérise par :

- ✓ Des images lumineuses (**brillantes**);
- ✓ Un bon contraste.
- ✓ L'absence de parasites.

## 3.8. Images à niveaux de gris

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter les images à niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise par exemple entre 0 et 255. Chaque pixel n'est donc plus représenté par un bit, mais par un octet. Pour cela, il faut que le matériel utilisé pour afficher l'image soit capable de produire les différents niveaux de gris correspondant. Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la "couleur" de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux possibles sont nombreux. [7]

## 4. Traitement d'image

### 4.1. Type de données manipulées

La mesure de la mesure spatiale est la taille du plus petit élément (pixel), mais aussi l'inter corrélation de deux éléments voisins : plus cette inter corrélation est faible, meilleure est l'image. [3,j]

### 4.3. Opérateurs de traitement d'images

Par analogie avec les opérateurs mathématiques, on appelle opérateurs de traitement d'images des traitements plus ou moins complexes prenant en entrée une image ou un ensemble d'informations relatif à une image, et produisant une image ou un ensemble d'informations relatif aux données initiales.

# CHAPITRE I : INTRODUCTION AU TRAITEMENT D'IMAGES

On classe généralement les opérateurs en différentes familles, en fonction des informations qu'ils acceptent en entrée et qu'ils fournissent en sortie, et en fonction des transformations qu'ils font subir aux données. Ainsi, par exemple, on distingue (cette liste est loin d'être exhaustive) :

Opérateurs image  $\rightarrow$  image :

- opérateurs de modifications pixel à pixel (aussi appelés opérateurs point à point) : changement de la dynamique de l'image, opérateurs binaires pixel à pixel (et, ou, xor, etc.) ;
- opérateurs locaux (traitent les pixels en fonction de leur voisinage) : opérateurs de flou, opérateurs morphologiques (érosion, dilatation, squelette), opérateurs de détection de contours ;
- opérateurs dans l'espace fréquentiel : opérateurs de réduction du bruit, filtres passe-bande (souvent utilisés en première approche pour améliorer l'image, on les appelle alors des opérateurs de pré - traitement) ;
- opérateurs globaux : calcul des distances.

Opérateurs image  $\rightarrow$  ensemble d'informations :

- opérateurs de segmentation en frontières, en régions ;
- opérateurs de classification de pixels ;
- opérateurs de calcul de paramètres.

Opérateurs ensemble d'informations  $\rightarrow$  image

- constructeurs d'image à partir d'une carte de régions ou d'une liste de frontières.

Les parties suivantes s'attachent à détailler les différents opérateurs et leurs applications habituelles, puis à présenter la manière dont ils sont combinés pour construire une application de traitement d'images. [3]

## **4.3.1. Opérateurs point à point**

Cette amélioration peut servir dans un premier temps à faciliter la visualisation de l'image sur un écran d'ordinateur. Les capacités de vision de l'être humain étant limitées, il est indispensable d'adapter la dynamique de l'image à notre vision.

# CHAPITRE I : INTRODUCTION AU TRAITEMENT D'IMAGES

On parle souvent de « Lookup Table » (ou LUT) que l'on retrouve également dans les FPGA. Il s'agit de l'opérateur le plus simple qu'on puisse trouver puisqu'en chaque pixel de l'image on modifie le niveau de gris à l'aide d'une fonction. Ainsi, pour éclaircir une image, on applique la fonction  $\log()$  à chaque niveau de gris. Au contraire pour rendre plus sombre une image un peu trop saturée, on applique une fonction exponentielle. On peut remarquer que le seuillage (binaire) n'est rien d'autre qu'une table d'affichage particulière, celle qui associe le noir à tous les niveaux inférieurs à un certain seuil et le blanc à tous les autres. Il s'agit d'un opérateur très simple et particulièrement utilisé mais qui cache une grande difficulté, trouver le seuil adéquat et de manière automatique !

Ces opérations point à point, qualifiées ainsi car elles ne travaillent que sur un pixel (et non sur un voisinage), ont un effet bien limité. En présence de bruit elles ne sont d'aucune utilité. [3]

## **4.3.2. Opérateurs locaux**

Il faut alors utiliser des opérateurs de traitement plus complexes scindés bien souvent en deux sous-catégories :

- les opérateurs linéaires ;
- les filtres non linéaires.

La première sous-catégorie comprend tous les opérateurs pouvant exprimer leur résultat comme une combinaison linéaire des niveaux de gris d'un voisinage de l'image. Ces filtres possèdent des caractéristiques spectrales, on parle ainsi de filtre passe-bas (l'image devient floue) ou de filtre passe-haut (les contours ressortent).

La deuxième sous-catégorie comprend le domaine de la morphologie mathématique, ainsi que d'autres traitements comme les détecteurs de points caractéristiques, l'opérateur de Di-Zenzo (détecteur de contour généralisé au cas couleur), le filtre Retinex, ainsi que les opérateurs homomorphiques (ceux qui travaillent sur le logarithme de l'image), mais aussi tous les opérateurs permettant d'extraire par exemple des informations sur la texture de l'image (matrice de cooccurrence, indice fractal, longueur de plage...).

On a souvent l'habitude de voir un détecteur de contours s'appliquer après un filtre linéaire passe-bas... qui rend l'image floue ! La plupart du temps il faut combiner astucieusement filtre

# CHAPITRE I : INTRODUCTION AU TRAITEMENT D'IMAGES

non linéaire et filtre linéaire afin de détecter ce que l'on souhaite tout en faisant abstraction du bruit.

Une fois le bruit éliminé et l'image restaurée afin de compenser les déformations introduites par le milieu de transmission et l'optique d'acquisition, on peut passer à l'étape de segmentation qui doit permettre de réaliser une partition de l'image en ensembles connexes homogènes.

Il existe deux grandes catégories de segmentations :

- la segmentation de région ;
- la segmentation de contour ; on se trouve alors confronté à un problème de représentation du résultat par des primitives simples.

La segmentation orientée contour connaît de nombreux progrès autour de l'utilisation de contours actifs ou des ensembles de niveaux. L'introduction d'aspects probabilistes (chaîne de Markov et champs de Markov) a permis de travailler en réduisant la connaissance a priori nécessaire pour obtenir un traitement satisfaisant.

Dans cette étape on retrouve souvent une partie de classification des pixels en classes. On essaye de regrouper au sein d'un même ensemble, aussi appelé classe, les pixels présentant une même caractéristique : niveau de gris compris dans un certain intervalle ou dérivée seconde supérieure à un certain seuil. [3]

## **4.3.3. Les filtres usuels en traitement d'images**

### **a. Définition du 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.

N'entrent pas dans la catégorie du filtrage toutes les transformations de l'image d'origine : zoom, découpage, projections, ...[8]

#### **➤ Filtrage Global**

Dans le filtrage global, chaque pixel de la nouvelle image est calculé en prenant en compte la totalité des pixels de l'image de départ. Dans cette catégorie on trouve, par exemple, les

# CHAPITRE I : INTRODUCTION AU TRAITEMENT D'IMAGES

opérations sur les histogrammes ou les opérations qui nécessitent de passer dans l'espace de Fourier. [8]

## ➤ Filtrage Local

Dans le filtrage local, chaque pixel de la nouvelle image est calculé en prenant en compte seulement un voisinage du pixel correspondant dans l'image d'origine. Il est d'usage de choisir un voisinage carré et symétrique autour du pixel considéré. Ces voisinages sont donc assimilables à des tableaux à deux dimensions (matrices) de taille impaire. [8]

### Gestion des bords

Lorsque le pixel considéré est proche du bord de l'image, certains points du voisinage sont en dehors de l'image d'origine. Il convient alors de choisir une stratégie pour gérer ces pixels extérieurs. Les stratégies couramment employées :

- Mise à zéro : Si un pixel du voisinage est en dehors de l'image d'origine, sa valeur est considérée comme nulle. C'est à dire :  $Image[-1][y]=0$
- Continuité : Si un pixel du voisinage est en dehors de l'image d'origine, sa valeur est celle du pixel le plus proche qui est dans l'image d'origine. C'est à dire :  $Image[-1][y]=Image[0][y]$
- Miroir : Si un pixel du voisinage est en dehors de l'image d'origine, sa valeur est celle du pixel symétrique par rapport au bord de l'image. C'est à dire :  $Image[-1][y]=Image[1][y]$
- Sphérique : Si un pixel du voisinage est en dehors de l'image d'origine, sa valeur est celle du pixel correspondant si l'image était projetée sur une sphère. C'est à dire :  $Image[-1][y]=Image[Largeur-1][y]$ . [8]

### 1. Filtrage local linéaire

Le filtre local est dit linéaire si la valeur du nouveau pixel est une combinaison linéaire des valeurs des pixels du voisinage.

Combinaison linéaire des pixels du voisinage

$$NouvelleValeur_{x,y} = \sum_{i,j} A_{i,j} * P_{x+i,y+j}$$

avec  $i,j$  variant entre  $-h$  et  $+h$ , la demi taille du voisinage (pour  $3 \times 3$   $h=1$ , pour  $5 \times 5$   $h=2$ , ...)  
et  $A_{i,j}$  = valeur, entière ou réelle, spécifique au filtre linéaire.

## Normalisation

Si la valeur obtenue n'est dans les limites imposées par le format d'image (entier(s) entre  $0 \dots 255$ ), alors la valeur doit être normalisée. Le facteur de normalisation peut être facilement calculé en cherchant la valeur maximale (positive) et minimale (négative) que peut atteindre la combinaison linéaire.

La valeur maximale est atteinte lorsque tous les pixels associés aux coefficients positifs ont la valeur maximale (255) et ceux associés aux coefficients négatifs ont la valeur minimale (0). Il suffit donc de faire la somme des coefficients positifs pour connaître le facteur de normalisation positif.

Facteur de normalisation positif

$$FacteurPositif = \sum_{i,j} A_{i,j}, \text{ si } A_{i,j} > 0$$

Idem pour le facteur de normalisation négatif, qui est la somme des coefficients négatifs.

Facteur de normalisation négatif

$$FacteurNegatif = \sum_{i,j} A_{i,j}, \text{ si } A_{i,j} < 0$$

Le facteur de normalisation est le plus grand (en valeur absolu) de ces deux facteurs.

Facteur de normalisation du filtre linéaire

$$Facteur = \max(|FacteurPositif|, |FacteurNegatif|)$$

# CHAPITRE I : INTRODUCTION AU TRAITEMENT D'IMAGES

Une fois la combinaison linéaire calculée, il suffit de diviser le résultat par le facteur de normalisation (et de convertir en entier, si besoin). La valeur obtenue peut être négative.

Une fois la normalisation effectuée, les valeurs obtenues sont dans la plage [-255,+255]. La conversion finale à effectuer (pour être dans la plage [0,255]) dépend du type de filtre et du résultat visuel souhaité : seuillage à zéro, mise à l'échelle, valeur absolue, ...

## Noyau

Il est d'usage de présenter les coefficients sous forme d'une matrice (appelée noyau de convolution) facilitant ainsi la mise en correspondance avec les valeurs du voisinage.

$$\begin{pmatrix} A_{-1,-1} & A_{0,-1} & A_{1,-1} \\ A_{-1,0} & A_{0,0} & A_{1,0} \\ A_{-1,1} & A_{0,1} & A_{1,1} \end{pmatrix}$$

noyau de convolution 3x3

$$\begin{pmatrix} P_{r-1,q-1} & P_{r,q-1} & P_{r+1,q-1} \\ P_{r-1,q} & P_{r,q} & P_{r+1,q} \\ P_{r-1,q+1} & P_{r,q+1} & P_{r+1,q+1} \end{pmatrix}$$

voisinage 3x3

multiplication du noyau et du voisinage

$$\begin{pmatrix} A_{-1,-1} * P_{r-1,q-1} & A_{0,-1} * P_{r,q-1} & A_{1,-1} * P_{r+1,q-1} \\ A_{-1,0} * P_{r-1,q} & A_{0,0} * P_{r,q} & A_{1,0} * P_{r+1,q} \\ A_{-1,1} * P_{r-1,q+1} & A_{0,1} * P_{r,q+1} & A_{1,1} * P_{r+1,q+1} \end{pmatrix}$$

La nouvelle valeur du pixel  $P_{x,y}$  est alors la somme des éléments de la dernière matrice (au facteur de normalisation près). [8]

## 2. Filtrage local non-linéaire

Si le filtre ne peut pas être exprimé par une combinaison linéaire, il est appelé " non-linéaire ". Les filtres non-linéaires sont plus complexes à mettre en œuvre que les filtres linéaires. Cependant les résultats obtenus avec les filtres non-linéaires sont très souvent de meilleure qualité que ceux obtenus par les filtres linéaires. [8]

## 4.3.4. Amélioration d'une image: les filtres passe-bas

L'image suivante est une image scannée à partir d'un document imprimé. Elle est très bruitée car on aperçoit le grain d'impression : [k]

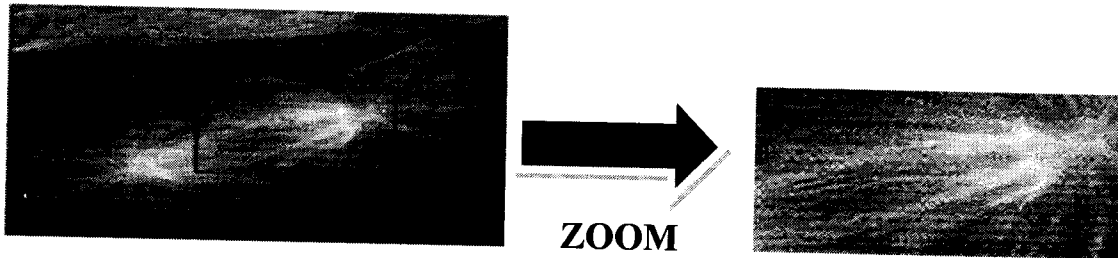


Figure 5.1 : Zoom d'une partie d'image

Son profil réalisé en diagonale dans les 3 composantes R,V,B nous le prouve :

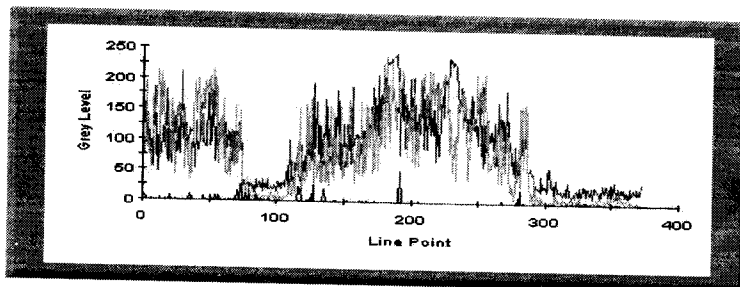


Figure 6.1 : profil avec les 3 composantes R,V,B

Après un filtrage passe-bas moyennneur, elle l'est déjà moins :

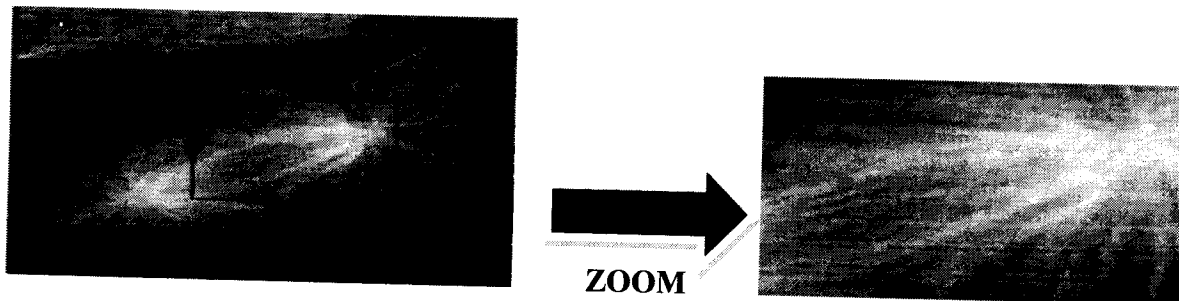


Figure 7.1 : Filtrage passe bas moyennneur

Son profil en diagonale en atteste :

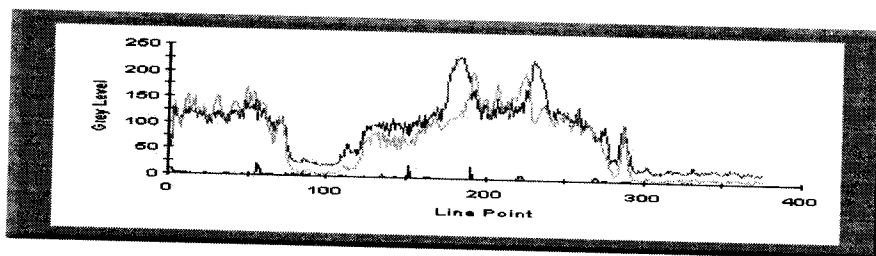
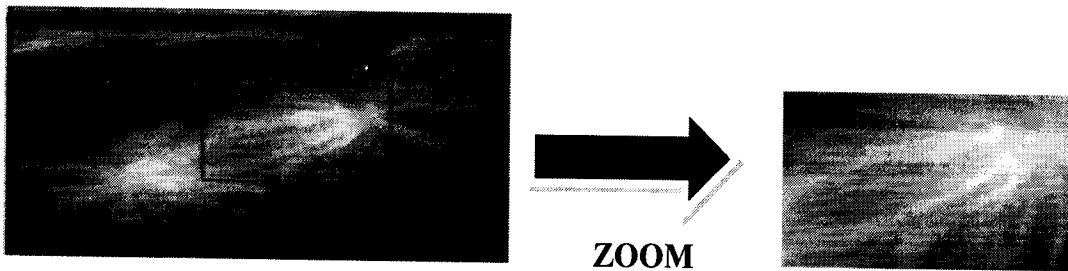


Figure 8.1 : Le profil d'image en diagonale



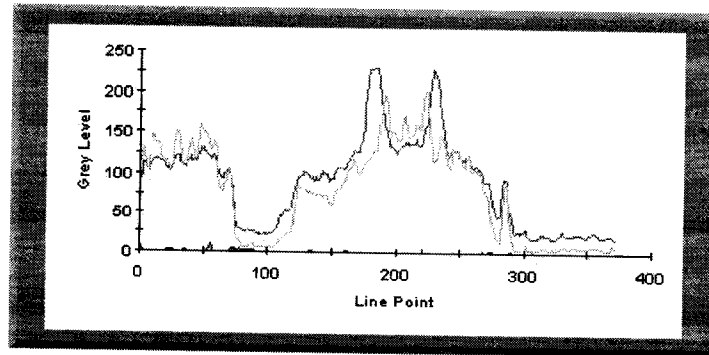
# CHAPITRE I : INTRODUCTION AU TRAITEMENT D'IMAGES

Pour supprimer encore plus de bruit, on peut faire un filtrage passe-bas médian sur cette image déjà filtrée, ce qui donne :



**Figure 9.1 : Suppression du bruit par un filtrage passe-bas médian**

Avec un profil en diagonale correspondant :



**Figure 10.1 : Le profil d'image en diagonale**

## 4.4. Construction d'une application de traitement d'images

Les objectifs des applications peuvent être de différentes natures :

- détecter la présence d'un objet ou son absence ;
- calculer les caractéristiques d'un ou de plusieurs éléments de l'image.

Dans tous les cas, l'idée est, en partant d'une image initiale, d'en extraire des informations. Pour cela, on va utiliser les opérateurs à la manière de « briques logicielles », en les combinant et en les enchaînant. Ces techniques sont la base des systèmes de vision industrielle.

De nombreuses briques sont disponibles permettant de créer des applications complexes et évoluées. [3]

## 5. Histogramme et Binarisation

L'histogramme permet d'obtenir le nombre de pixel de la même couleur dans une image.[7]

En imagerie numérique, l'histogramme représente la distribution des intensités (ou des couleurs) de l'image. C'est un outil fondamental du traitement d'images, avec de très nombreuses applications. Les histogrammes sont aussi très utilisés en photographie et pour la retouche d'images. [4]

Les histogrammes sont en général normalisés, en divisant les valeurs de chaque classe par le nombre total de pixels de l'image. La valeur d'une classe varie alors entre 0 et 1, et peut s'interpréter comme la probabilité d'occurrence de la classe dans l'image. L'histogramme peut alors être vu comme une densité de probabilité. [4,f]

### 5.1. Types d'histogrammes

- Histogramme pondéré
- Histogramme conjoint
- Histogramme intégral

### 5.2. Égalisation d'histogramme

L'égalisation d'histogramme est une méthode d'ajustement du contraste d'une image 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.

L'égalisation d'histogramme permet de mieux répartir les intensités sur l'ensemble de la plage de valeurs possibles (0 à 255 en général pour des images 8 bits), en « étalant » l'histogramme. Théoriquement (cas continu) l'histogramme devrait même être plat (loi uniforme). Toutefois à cause de la discrétisation, ceci est rarement le cas.

L'égalisation est intéressante pour les images dont la totalité, ou seulement une partie, est de faible contraste (pixels d'intensité proches). La méthode est rapide, facile d'implémentation, et complètement automatique (i.e. pas de réglages). [4,f]

## 6. Le seuillage d'une image

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 (monochrome).

Le seuillage d'image remplace un à un les pixels d'une image à l'aide d'une valeur seuil fixée (par exemple 123). Ainsi, si un pixel a une valeur supérieure au seuil (par exemple 150), il prendra la valeur 255 (blanc), et si sa valeur est inférieure (par exemple 100), il prendra la valeur 0 (noir). [9]



Figure 3.1 : Image originale



Figure 4.1 : Seuillage d'une image

## 7. La multi résolution

Résolution et échelle sont deux notions proches, mais distinctes. La résolution est une caractéristique d'une opération de mesure, d'observation, alors que l'échelle est un attribut d'une description (ou modélisation). Les théories scale-space sont bien des théories multi-échelles, dans le sens où elles proposent de décrire la structure d'une image par une famille d'images simulant tout un continuum de perceptions, allant de la perception des structures les plus fines – ou locales - aux plus grossières - ou globales. Les structures présentes dans l'analyse deviennent de plus en plus grossières au fur et à fur que le paramètre d'échelle augmente mais la définition des images reste celle de l'image d'origine. Les formes à échelle grossière sont donc définies à la résolution d'origine (ce point est indépendant du problème de délocalisation des contours dans le scale-space gaussien).

A contrario, les analyses multi-résolution, dont l'idée remonte à Tanimoto et Pavlidis (1975), proposent de simuler les images que l'on aurait pu acquérir avec des capteurs moins résolus. Une pyramide multirésolution est une séquence  $(I_0; I_1 : : I_p)$  d'images dont la taille décroît

# CHAPITRE II

## 1. Introduction

La segmentation d'images est une étape fondamentale et importante dans beaucoup d'applications de vision par ordinateur. C'est une étape primordiale pour l'interprétation d'images.

Beaucoup de méthodes de segmentation ont été développées, mais il n'y a pas encore de mesures de performance satisfaisante. Concevoir une bonne mesure pour la qualité de segmentation est un problème dur. Quelques chercheurs pensent que c'est impossible.

La segmentation, comme la cuisine, est un art. Chaque personne, comme un cuisinier, a une recette différente.

Malgré que ce soit impossible d'évaluer automatiquement la segmentation, nous pouvons définir des critères sur le résultat d'un algorithme de segmentation pour avoir une idée d'évaluation. [m,n]

## 2. Segmentation d'images

Un bon résultat de segmentation ne permet pas forcément une bonne interprétation, mais nous ne pouvons pas obtenir une bonne interprétation à partir d'un mauvais résultat de segmentation.

C'est pourquoi une définition de la segmentation est vraiment indispensable. Cependant, la segmentation d'images n'a pas de définition officielle. Elle peut être définie ainsi : « La segmentation est un traitement de bas niveau qui consiste à créer une partition de l'image  $A$  en sous-ensembles  $R_i$ , appelés régions tels qu'aucune région ne soit vide, l'intersection entre deux régions soit vide et l'ensemble des régions recouvre toute l'image. Une *région* est un ensemble de pixels connexes ayant des propriétés communes qui les différencient des pixels des régions voisines ».

Il y a certaines autres définitions similaires que nous pouvons trouver dans divers autres articles. Le but de la segmentation est d'extraire les entités d'une image pour y appliquer un traitement spécifique et interpréter le contenu de l'image. La segmentation est normalement basée sur les discontinuités (les arêtes, les changements abruptes, etc.), les similitudes (intensités, couleurs, textures, etc.). [n]

## **Chapitre II : Les Méthodes de Segmentation d'Image**

La segmentation est donc le découpage d'une image en différentes régions et/ou frontières. Il *existe une dualité entre régions et frontières : une région est délimitée par un contour, un contour sépare deux régions*. A partir d'un résultat de segmentation en régions, nous pouvons obtenir un résultat de détection de frontières.

Nous voyons que la segmentation seule ne peut pas trouver tous les composants de l'image tel que nous les interprétons. Gestalt a proposé une approche humaine de la segmentation qui s'inspire de la psychologie et de l'observation de l'humain et de son environnement. Selon cette approche, nous regroupons les objets ou les courbes de l'image à partir de ses informations : la proximité, la similarité, la tendance commune, la région commune, la parallélisme, la fermeture, la symétrie, la continuité et la configuration familière. Malgré certains essais, cette approche reste surtout théorique.

Concernant la segmentation d'images, il y a toujours des difficultés à cause de la complexité des images naturelles et de la définition du niveau de précision du résultat .

A ce jour, il existe de nombreuses méthodes de segmentation d'images que l'on peut regrouper en quatre principales classes d'algorithmes :

- Segmentation basée sur les régions (*region-based segmentation*). On y trouve par exemple la croissance de région (*region-growing*), décomposition/fusion (*split and merge*).
- Segmentation basée sur les contours (*edge-based segmentation*).
- Segmentation basée sur une approche globale de l'image, par exemple : seuillage (*thresholding*), histogramme, approches basées sur le nuage couleur.
- Segmentation basée sur la coopération entre les trois premières segmentations. [m]

### **3. Les Méthodes de Segmentation d'Image**

#### **3.1. Segmentation par approche « région »**

Les méthodes appartenant à cette famille manipulent directement des régions. Soit elles partent d'une première partition de l'image, qui est ensuite modifiée en divisant ou regroupant des régions, et on parle alors de méthodes de type décomposition/fusion (ou *split and merge* en Anglais) ; soit elles partent de quelques régions, qui sont amenées à croître par incorporation de pixels jusqu'à ce que toute l'image soit couverte, et on parle alors de méthodes par *croissance de régions*. Des méthodes

## Chapitre II : Les Méthodes de Segmentation d'Image

---

fondées sur la modélisation statistique conjointe de la régularité des régions et des niveaux de gris de chaque région existent également.

Les algorithmes de type décomposition/fusion exploitent les caractéristiques propres de chaque région (surface, intensité lumineuse, colorimétrie, texture, etc.). On cherche des couples de régions candidates à une fusion et on les note en fonction de l'impact que cette fusion aurait sur l'apparence générale de l'image. On fusionne alors les couples de régions les mieux notés, et on réitère jusqu'à ce que les caractéristiques de l'image remplissent une condition prédéfinie : nombre de régions, luminosité, contraste ou texture générale donnée, ou alors jusqu'à ce que les meilleures notes attribuées aux couples de régions n'atteignent plus un certain seuil (dans ce dernier cas, on parle d'un algorithme avec *minimisation de fonctionnelle*).[n]

Les algorithmes par croissance de régions partent d'un premier ensemble de régions, qui peuvent être calculées automatiquement (par exemple, les minima de l'image), ou fournies par un utilisateur de manière interactive. Les régions grandissent ensuite par incorporation des pixels les plus similaires suivant un critère donné, tel que la différence entre le niveau de gris du pixel considéré et le niveau de gris moyen de la région. Les algorithmes de segmentation par « ligne de partage des eaux », développés dans le cadre de « la morphologie mathématique », appartiennent à cette catégorie.

Les algorithmes fondés sur une modélisation statistique conjointe des régions et des niveaux de gris, notamment ceux s'appuyant sur les Champs de Markov Cachés, reposent sur la minimisation d'une fonction de vraisemblance (ou énergie). Cette fonction prend simultanément en compte la vraisemblance de l'appartenance du pixel à une région considérant son niveau de gris, et les régions auxquelles appartiennent les pixels voisins. Cette fonction effectue un compromis entre la fidélité à l'image initiale et la régularité des régions segmentées. [10]

### 3.1.1. Croissance de régions (growing-region)

Cette technique consiste à faire progressivement grossir les régions autour de leur point de départ.

## Chapitre II : Les Méthodes de Segmentation d'Image

L'algorithme se compose de deux étapes:

1. Trouver les points de départ des régions .
2. Faire grossir les régions par agglomérations des pixels voisins.

### Points de départ (seeds)

Le choix des points de départ est l'étape est la partie critique de l'algorithme. En effet, l'étape de croissance va utiliser une mesure de similarité pour choisir les pixels agglomérer. Si le point de départ est situé dans une zone non homogène, la mesure de similarité va produire de fortes variations et la croissance va s'arrêter très tôt.

Par conséquent, il convient de choisir les points de départ dans des zones les plus homogènes possibles.

Pour trouver ces zones, on peut réutiliser le principe de décomposition utilisé dans l'algorithme split/merge. Comme la décomposition finale nous donne une liste de blocs homogènes, il suffit de choisir le centre des plus gros blocs pour avoir de bons points de départ.



**Figure 1.2 : Points de départ : centre des blocs de la décomposition en blocs homogènes**

### Croissance (growing)

Cette étape a pour objectif de faire grossir une région en agglomérant des pixels voisins. Les pixels sont choisis afin de maintenir l'homogénéité de la région. Pour cela, nous devons définir un indicateur d'homogénéité. Les pixels voisins sont ajoutés à la région si l'indicateur



## Chapitre II : Les Méthodes de Segmentation d'Image

d'homogénéité reste vrai. La croissance s'arrête lorsqu'on ne peut plus ajouter de pixels sans briser l'homogénéité. [11]

L'algorithme de la croissance est un procédé itératif:

Créer la liste « [S] » des points de départ (triée pour avoir le centre des plus gros blocs d'abord).

Pour chaque pixel « P » dans la liste « [S] »

Si le pixel « P » est déjà associé à une région, alors prendre le pixel « P » suivant dans la liste « [S] »

Créer une nouvelle région « [R] »

Ajouter le pixel « P » dans la région « [R] »

Calculer la valeur/couleur moyenne de « [R] »

Créer la liste « [N] » des pixels voisins du pixel « P »

Pour chaque pixel « Pn » dans la liste « [N] »

Si (« Pn » n'est pas associé à une région ET « R + Pn » est homogène) Alors

Ajouter le pixel « Pn » dans la région « [R] »

Ajouter les pixels voisins de « Pn » dans la liste « [N] »

Recalculer la valeur/couleur moyenne de « [R] »

Fin Si

Fin Pour

Fin Pour. [b]

L'indicateur d'homogénéité peut être construit à partir des mesures de similarité.

Indicateur : " vrai " si  $\text{Homogénéité}(R) \leq \text{SEUIL}$ , " faux " sinon.

En utilisant la définition de la variance et la formule du calcul de la distance entre 2 couleurs, on obtient :

Homogénéité( R ) = Variance( pixels de R ) + Moyenne( distance( pixels de R , Moyenne( pixels de R ) ) ^2 ).[b]

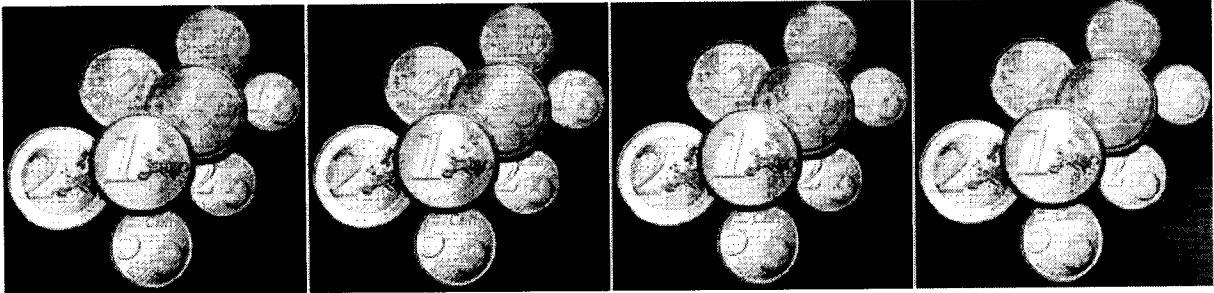


Figure 2.2 : Croissance progressive d'une région

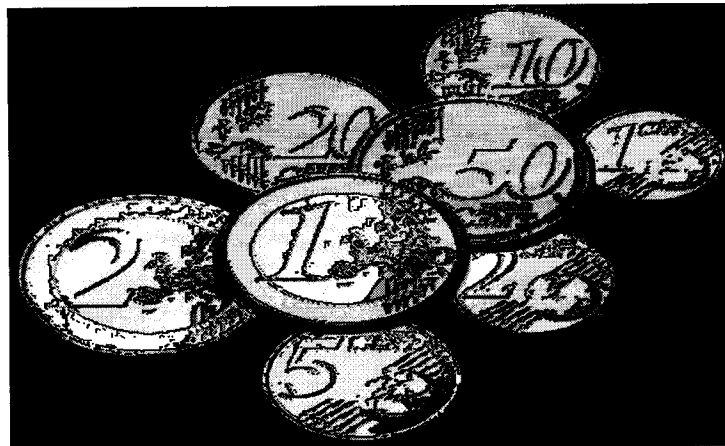
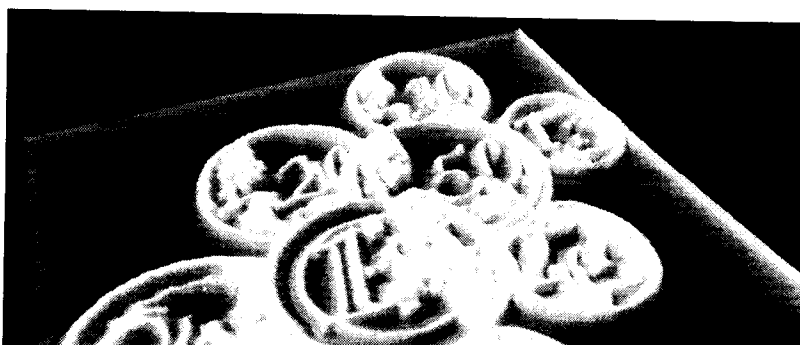


Figure 3.2 : Résultat final de la croissance de régions (827 régions)

### 3.1.2. Ligne de partage des eaux (watershed)

Cette technique consiste à faire grossir simultanément toutes les régions jusqu'à ce que l'image soit entièrement segmentée.

Cette technique tire son nom d'une analogie avec la géophysique. On peut en effet considérer les valeurs d'intensité des pixels d'une image comme une information d'altitude. Dans ce cas on peut représenter cette image (appelée carte d'élévation) comme un terrain en 3 dimensions. Le principe est alors de remplir progressivement d'eau chaque bassin du terrain. Chaque bassin représente une région.

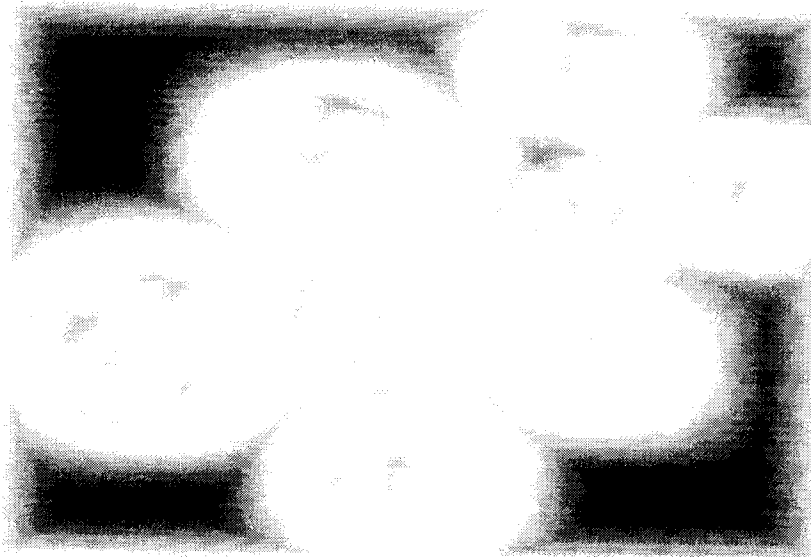


## Chapitre II : Les Méthodes de Segmentation d'Image

de crête : plus on s'éloigne de la ligne de crête, plus l'altitude doit décroître. La formule pour calculer la valeur d'altitude d'un pixel est donc :

$$\text{Altitude(Pixel)} = \text{HMAX} - \text{distance(Pixel, ligne\_de\_crête)}$$

Remarque : on peut calculer la distance entre un pixel et la ligne de crête en utilisant les masques de " Chamfer ".



**Figure 6.2 : Carte d'élévation. L'intensité reflète l'altitude (noir=le plus bas, blanc=le plus haut)**

### **Remplissage des bassins par infiltration (flooding)**

Il existe de nombreux algorithmes permettant de réaliser le remplissage des bassins (cf " The Watershed Transform: Definitions, Algorithms and Parallelization Strategies " , Roerdink and Meijster, 2001 ).

Nous détaillons ici un algorithme utilisant la technique de croissance de région vue précédemment. Dans cet algorithme, le remplissage des bassins est un procédé itératif qui consiste à élever progressivement le niveau de l'eau de zéro à HMAX. A chaque itération, on doit remplir les bassins existants (extension des régions) et éventuellement créer les nouveaux bassins (nouvelles régions).

Pour NIVEAU de 0 à HMAX

/\*\* Action 1 : Etendre les régions existantes \*\*/

Pour chaque Région « [R] » dans la liste « [Régions] »  
Faire croître « [R] » jusqu'à l'altitude NIVEAU  
Fin Pour

/\*\* Action 2 : créer les nouvelles régions \*\*/

Pour chaque pixel « P » à l'altitude NIVEAU  
Si « P » n'est pas associé à une région Alors  
Créer une nouvelle région « [R] » dans la liste  
« [Régions] »  
Ajouter le pixel « P » dans la région « [R] »  
Faire croître « [R] » jusqu'à l'altitude NIVEAU  
Fin Si  
Fin Pour  
Fin Pour

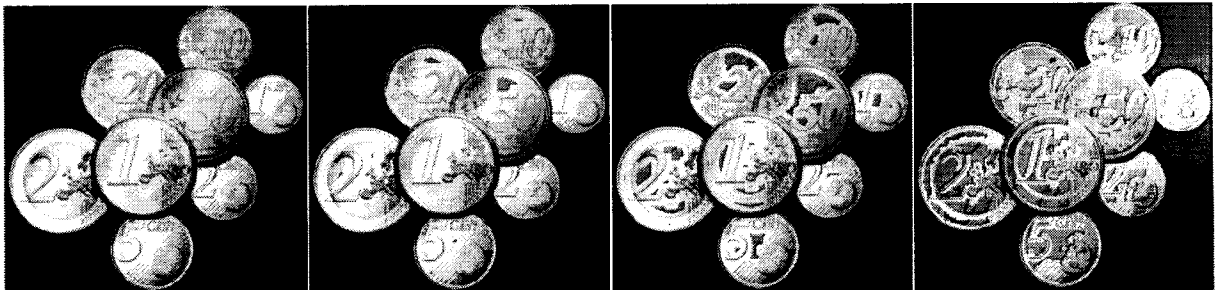


Figure 7.2 : Résultats intermédiaires du remplissage des bassins

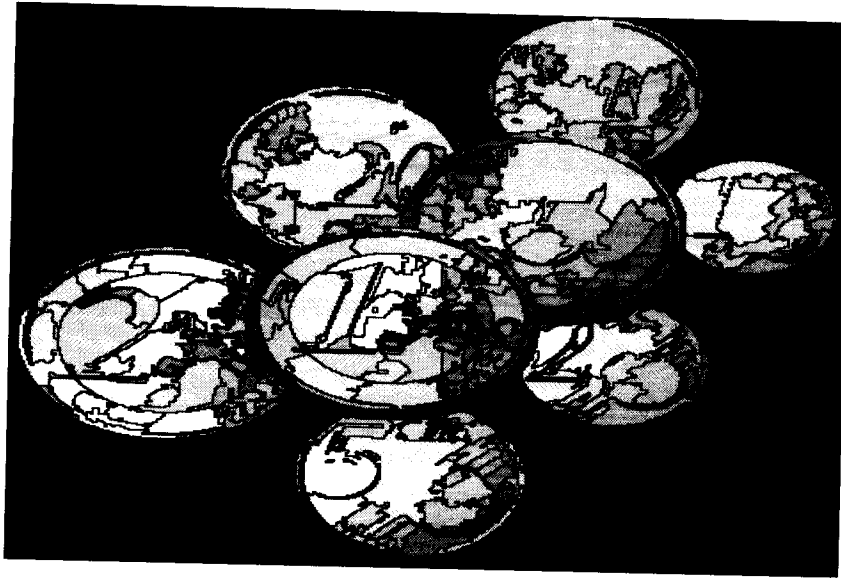


Figure 8.2 : Résultat final de la ligne de partage des eaux (806 régions)

### 3.2. Segmentation par approche « frontière »

Cette approche cherche à exploiter le fait qu'il existe une transition détectable entre deux régions connexes.

Les méthodes les plus anciennes utilisent des opérateurs de traitement d'images, tels que le filtre de Canny, pour mettre en évidence les pixels qui semblent appartenir à un contour. La construction d'une partition est alors souvent difficile.

On peut aussi faire intervenir des modèles déformables à l'aide de courbes paramétriques (courbe de Bézier, spline...) ou de polygones (par exemple algorithme à bulle).

Pour initier le processus, on recherche des points remarquables de l'image, tels que des points à l'intersection de trois segments au moins. De tels points sont appelés des *graines* (*seeds* en anglais).

L'intérêt principal des méthodes de segmentation selon l'approche frontières est de minimiser le nombre d'opérations nécessaires en cas d'itération du processus sur des séries d'images peu différentes les unes des autres (cas des images vidéo notamment). En effet, une fois que les contours des régions ont été trouvés dans la première image, l'application du modèle déformable à l'image suivante est plus efficace que de tout recalculer, si la différence entre les images est peu importante. [10]

### **3.3. Segmentation par classification ou seuillage**

On part ici d'un rapport qu'entretient chaque pixel individuellement avec des informations calculées sur toute l'image, comme par exemple la moyenne des niveaux de gris de l'ensemble des pixels, ou la médiane, permettant de construire n classes d'intensité. Lorsque les classes sont déterminées par le choix d'un seuil, on parle de seuillage. Les pixels appartenant à une même classe et étant connexes forment des régions de l'image. [10]

### **4. Conclusion**

Dans ce chapitre nous avons défini la segmentation ainsi que nous avons cité les techniques et les méthodes de segmentation d'image tel que la méthode de croissance de région et la ligne de partage des eaux...etc.

Et dans le chapitre suivant nous allons appliquer la méthode de croissance de région sur une image pour mieux comprendre la segmentation.



# CHAPITRE III



### 1. Introduction

#### 1.1. Définition du langage de Programmation Matlab

MATLAB (« matrix laboratory ») est un langage de programmation de quatrième génération et un environnement de développement ; il est utilisé à des fins de calcul numérique. Développé par la société The MathWorks, MATLAB permet la manipulation de matrice, afficher des courbes et des données, mettre en œuvre des algorithmes, créer des interfaces utilisateurs, et peut s'interfacer avec d'autres langages comme le C, C++, Java, et Fortran. Les utilisateurs de MATLAB (environ un million en 2004) sont de milieux très différents comme l'ingénierie, les sciences et l'économie dans un contexte aussi bien industriel que pour la recherche. Matlab peut s'utiliser seul ou bien avec des toolbox (« boîte à outils »). [1]

### 2. Traitement d'image

Dans cette partie nous allons montrer quelques traitements sur des images différents tel que la conversion d'une image RGB vers une image d'intensité, histogramme, égalisation d'image et d'histogramme, le seuillage, étiquetage, ainsi que le filtrage....etc.

#### 2.1. Chargement d'une image

Pour le chargement d'une image sous **MATLAB** il suffit d'utiliser 'imread'

Exemple :

```
I = imread('D:\final applications\image matlab \preview.jpeg');
```

##### 2.1.1. Enregistrement d'une image

l'enregistrement d'une image se fait comme suit:

```
imwrite(I,'C:\Users\wafaa\Desktop\pfe\PFE\image.jpg');
```

##### 2.1.2. Affichage d'image

Pour afficher une image il suffit de la charger puis on utilise « imshow(img) » pour l'afficher

```
img = imread('C:\Users\wafaa\Desktop\pfe\PFE\image\images.jpeg');  
Pour afficher l'image RGB acquise  
figure(2),subplot(2,2,1:2),  
imshow('C:\Users\wafaa\Desktop\pfe\PFE\image\images.jpeg'), title ('image RGB')
```

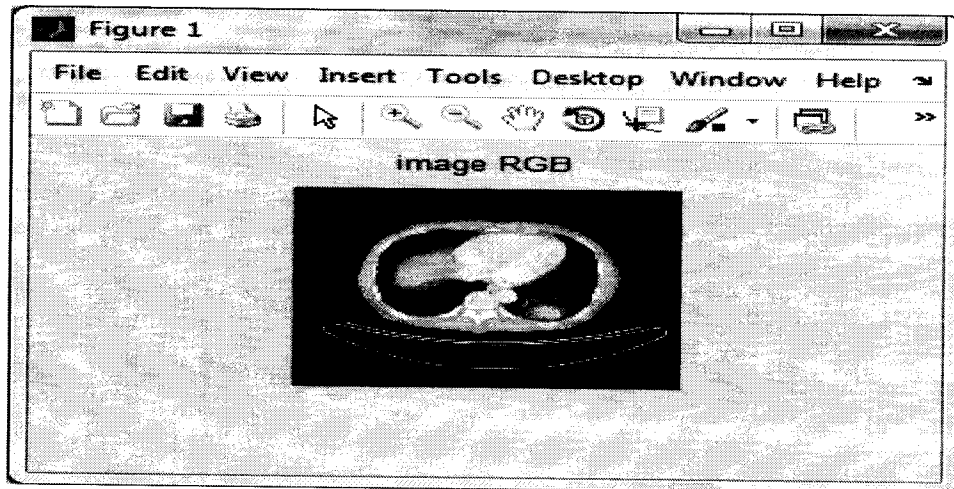


Figure 1.3: Affichage d'une image RGB

### 2.2. Conversion d'image RGB en une image d'intensité

Pour convertir l'image RGB acquise en une image d'intensité, on utilise la commande suivante :

```
>> I=rgb2gray(img);  
% afficher l'image d'intensité  
>> subplot(2,2,3), imshow(I), title ('image d'intensité')
```

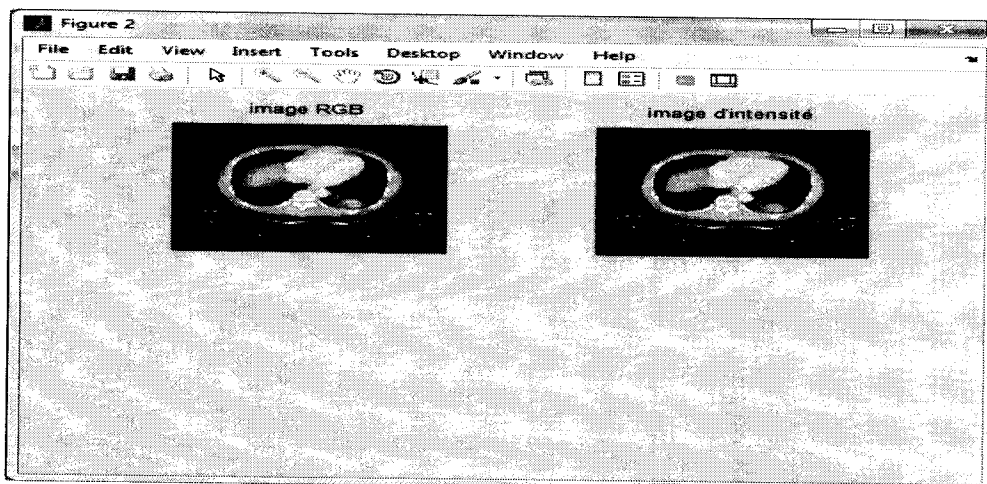


Figure 2.3 : Conversion d'une image RGB vers une image d'intensité

## 2.3. Calcule et affichage d'histogramme

Pour calculer et afficher l'histogramme on utilise « imhist »

```
>> subplot(2,2,4), imhist(I,256),title ('histogramme')
```

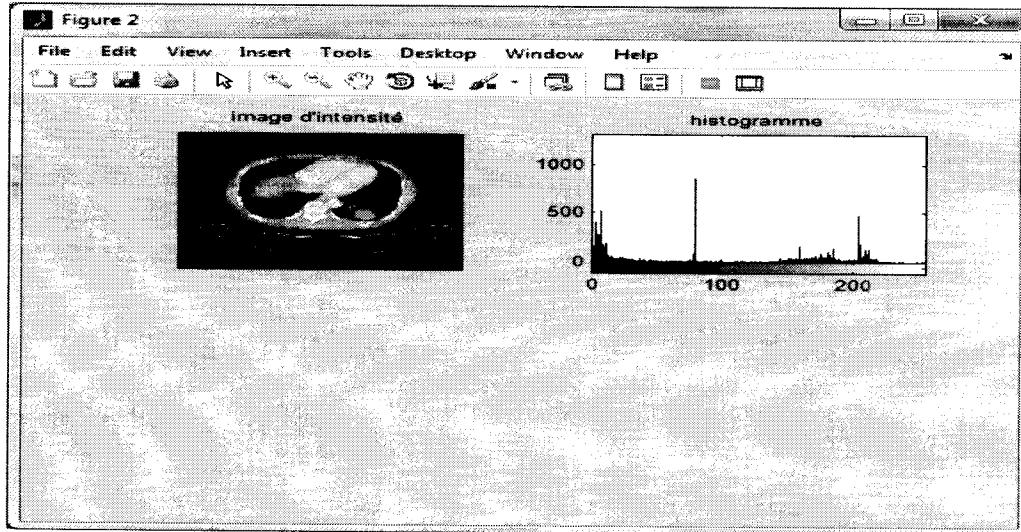


Figure 3.3 : Affichage de l'histogramme

## 2.4. Egalisation d'image (contraste)

Egaliser l'image (contraste)

```
I2 = histeq(I);
```

```
figure, imshow(I2)
```

```
figure, imhist(I2)
```

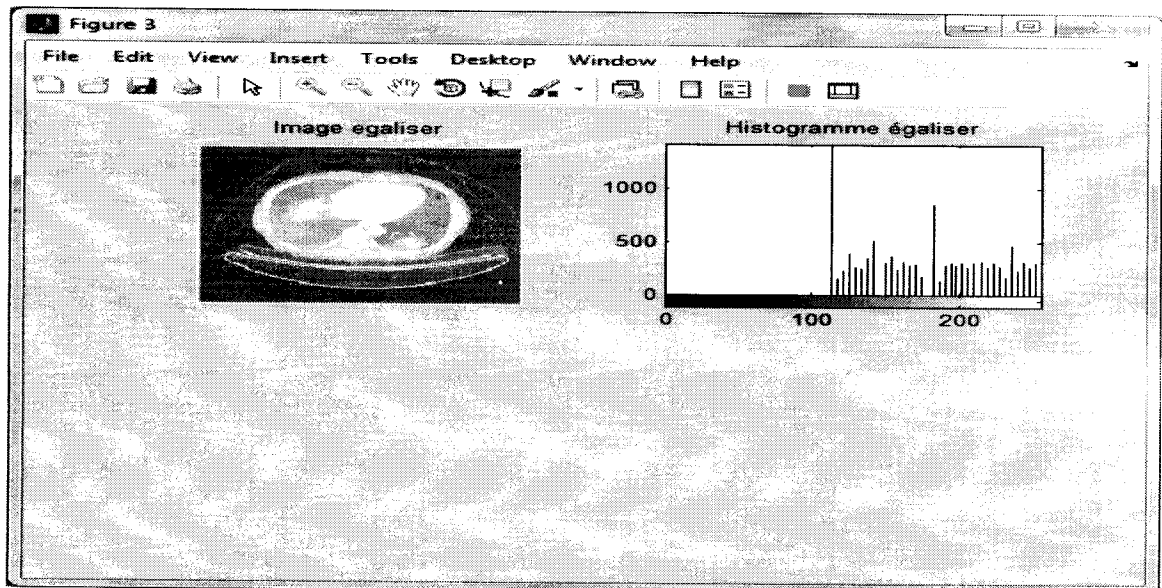


Figure 4.3 : Egalisation d'Image et Histogramme Egaliser

## 2.5. Ajustement du contraste d'une image

```
I3 = imadjust(I2, stretchlim(I2), [0 1]);  
>> subplot(2,2,4), imshow(I3)
```

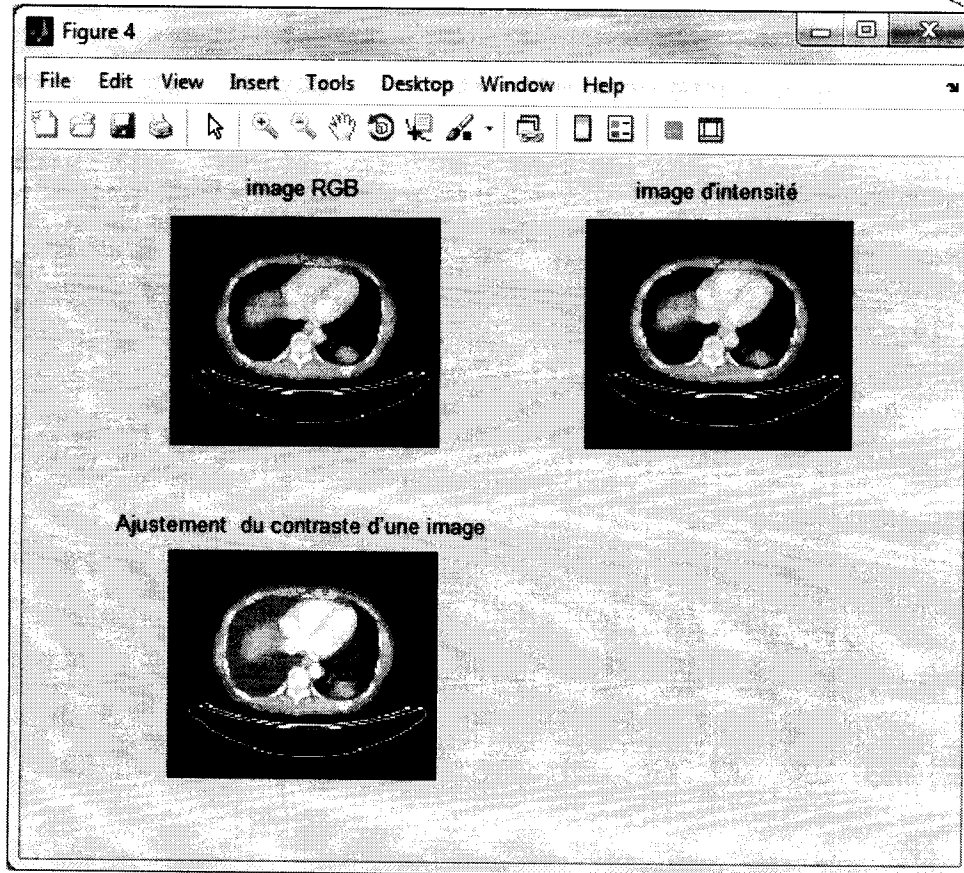
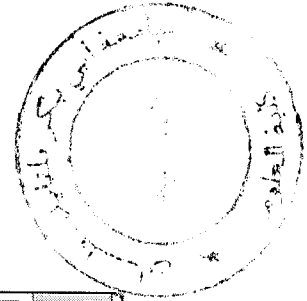


Figure 5.3: Ajustement du contraste d'une image

## 2.6. Le Seuillage

Le seuillage est une méthode simple et très populaire pour le traitement des images numériques.

Application du seuil d'une image:

```
level = graythresh(I);  
>> bw = im2bw(I, level);  
>> figure, imshow(bw)
```

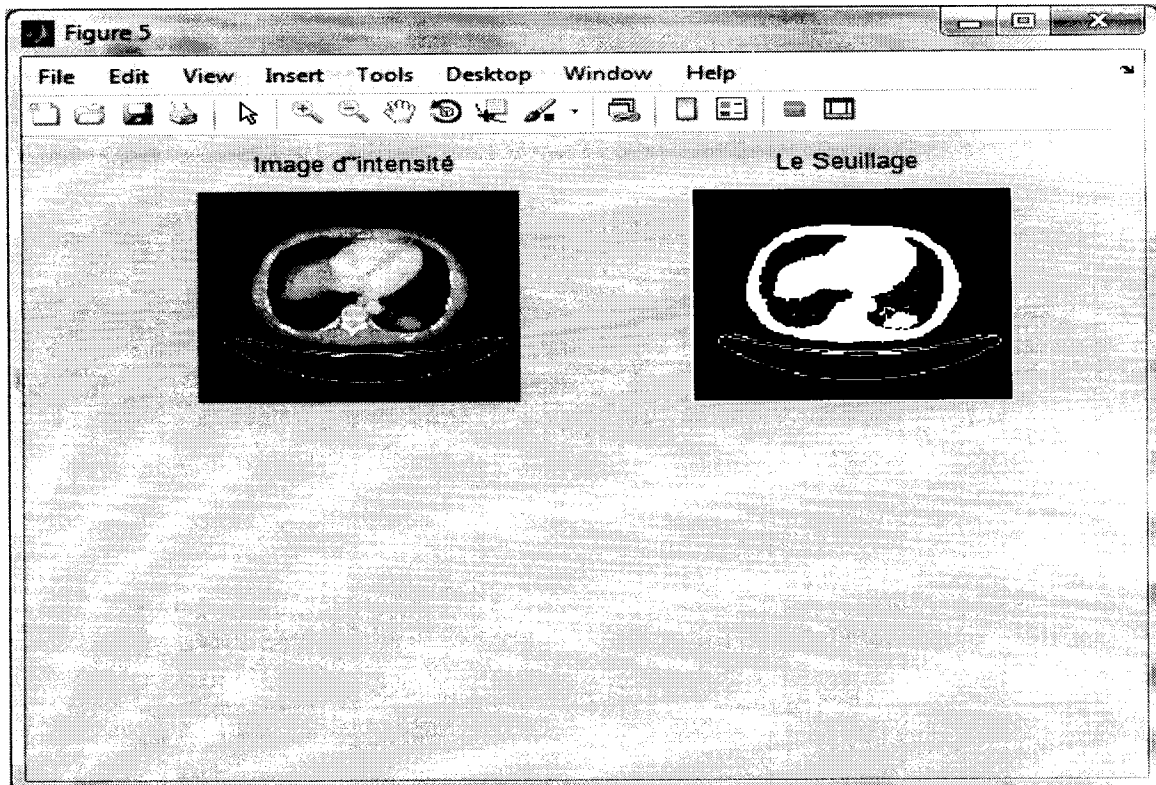


Figure 6.3 : Le seuillage d'une image

### 2.7. Etiquetage

- ◆ Détermine le nombre d'objets dans une image

```
[labeled, numObjects] = bwlabel(bw, 4);  
max(labeled(:))  
ans =104
```

Pour cet exemple d'image le nombre d'objets étiquetés est égale a 104  
et sa va s'afficher dans le résultat d'exécution comme suit :

```
ans =104
```

### 2.8. Filtrage

Il existe plusieurs types de filtrage tel que : gaussienne, passe bas, passe haut...etc

Nous allons montrer dans quelques exemples la différences entre chaque type de filtre sur la même image.

### 2.8.1. Le filtrage Gaussienne

```
t=[9 9];%taille du filter
I=imread('C:\Documents and Settings\reda\Bureau\prj pfe\cœur-clerpée.jpg');
H = fspecial('Gaussian',t,1);
>> ImageG=imfilter(I,H);
>> imwrite(uint8(ImageG),'image1.jpg','jpg');
>> i12=imread('image1.jpg');
>> figure, imshow(i12);
```

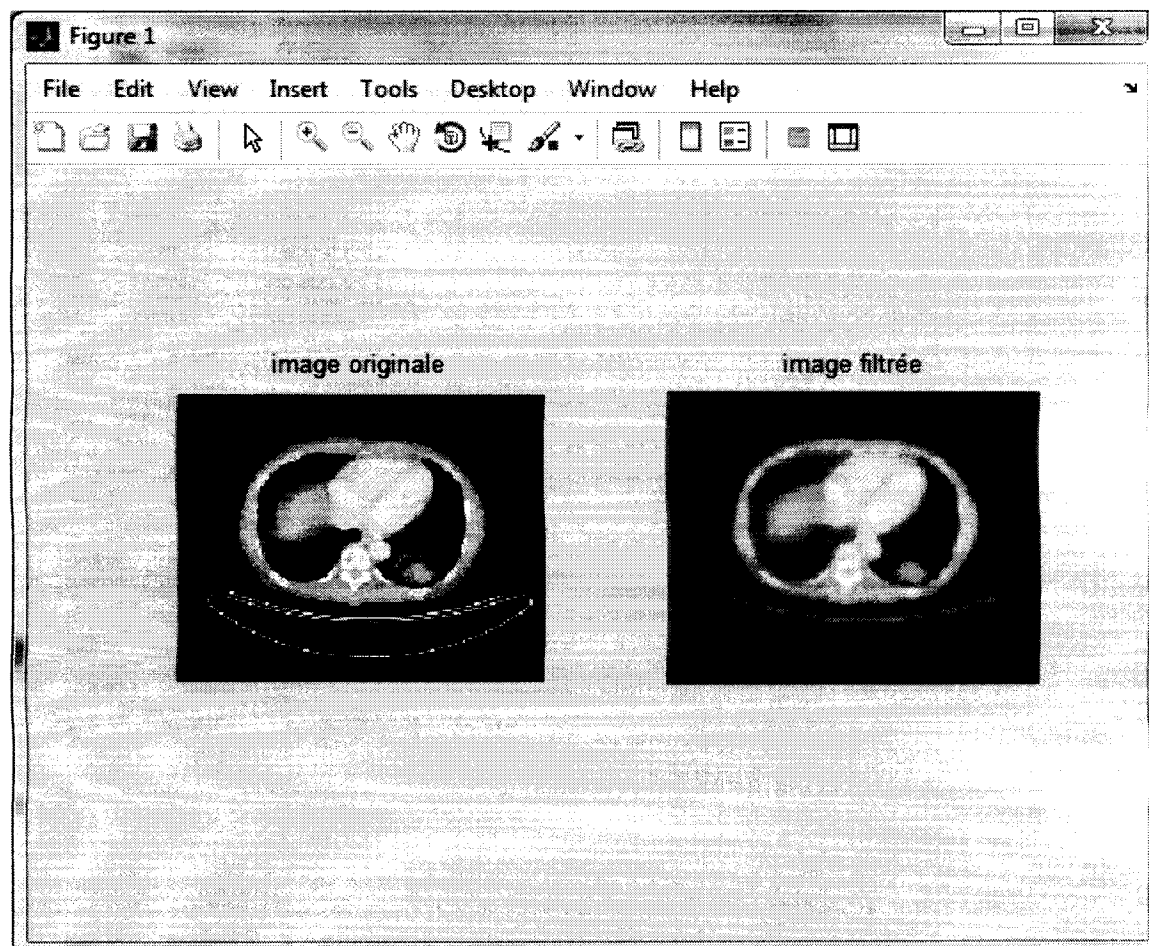


Figure 7.3 : Application du filtrage Gaussienne

### 2.8.2. Le filtrage passe bas

Application du filtrage passe bas sur une image

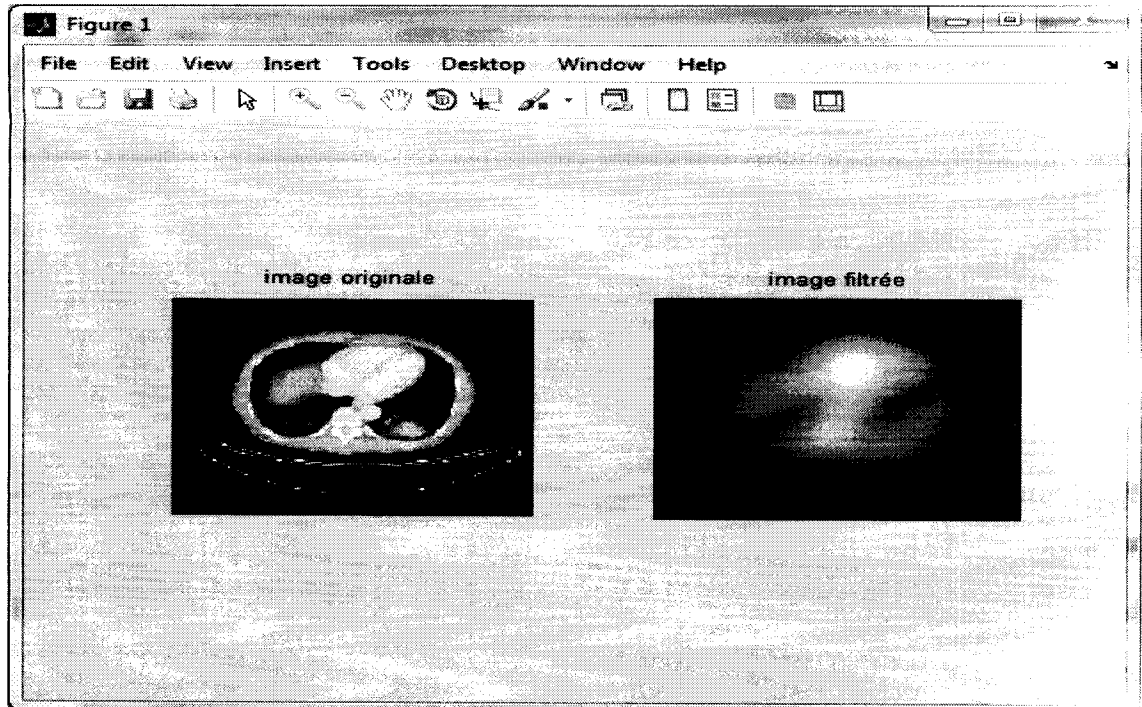


Figure 8.3 : Application du filtrage passe bas

### 2.8.3. Le filtrage passe haut « MEILLEUR »

Application du filtrage passe haut « meilleur » sur une image

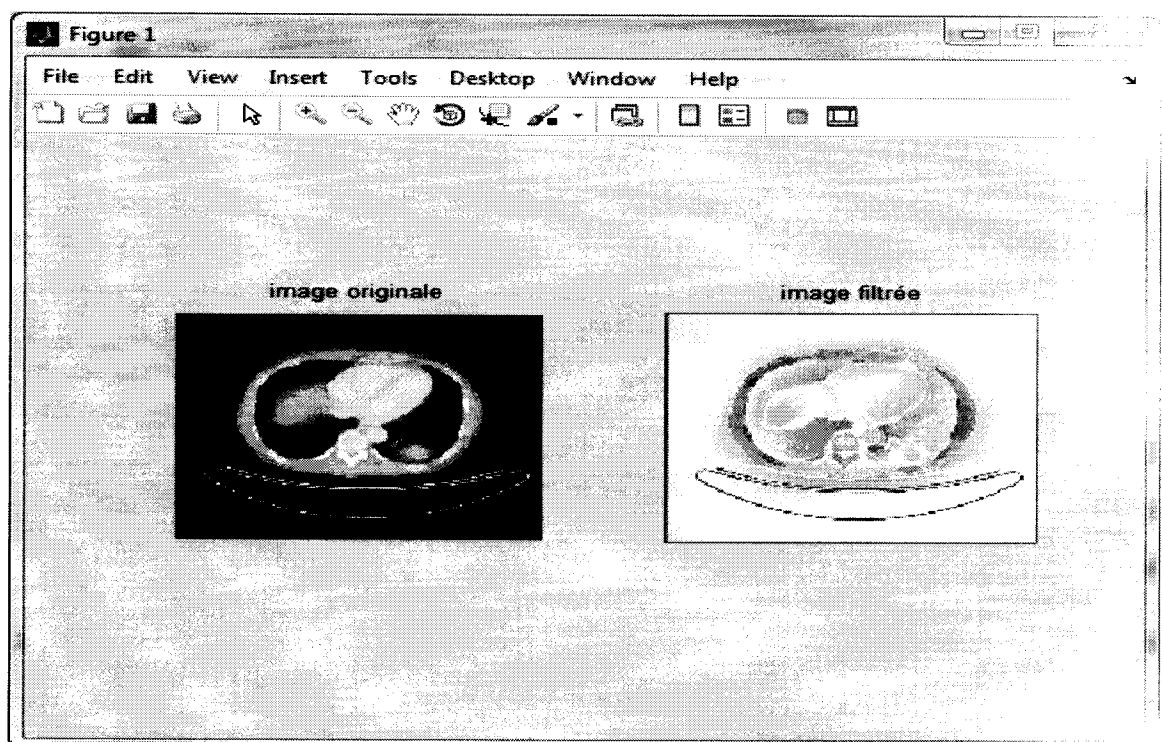


Figure 9.3 : Application du filtrage passe haut « MEILLEUR »

### 2.8.4. Filtre passe bas de Butterworth

Application du filtrage passe bas de « Butterworth » sur une image

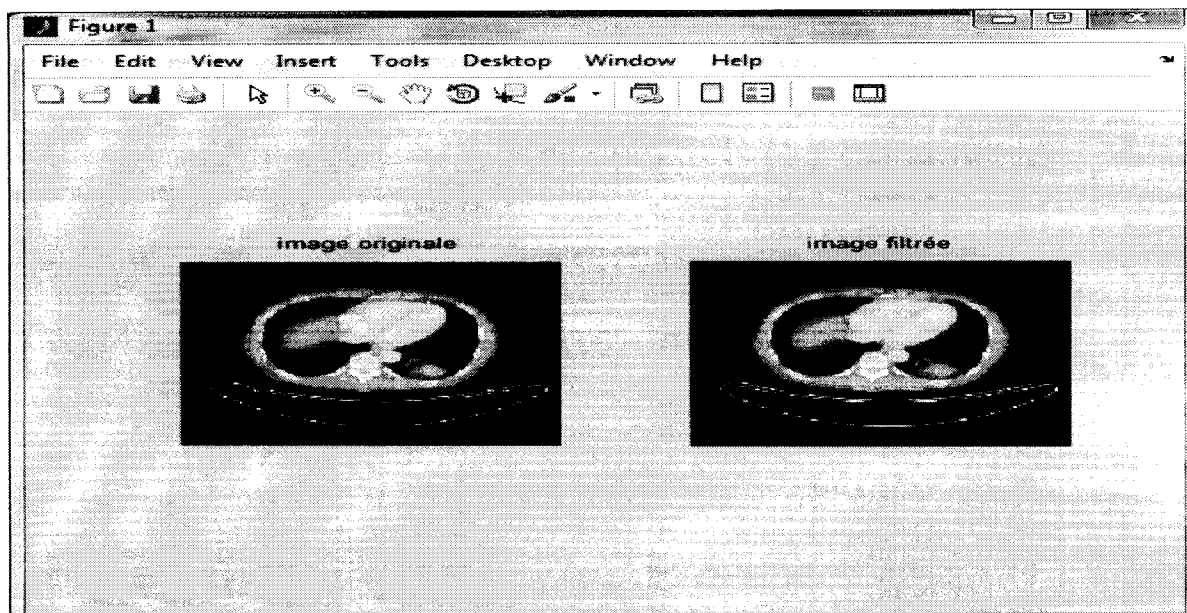


Figure 10.3 : Application du Filtre passe bas de Butterworth

### 2.8.5. Filtre passe haut de Butterworth

Application du filtrage passe haut de « Butterworth » sur une image

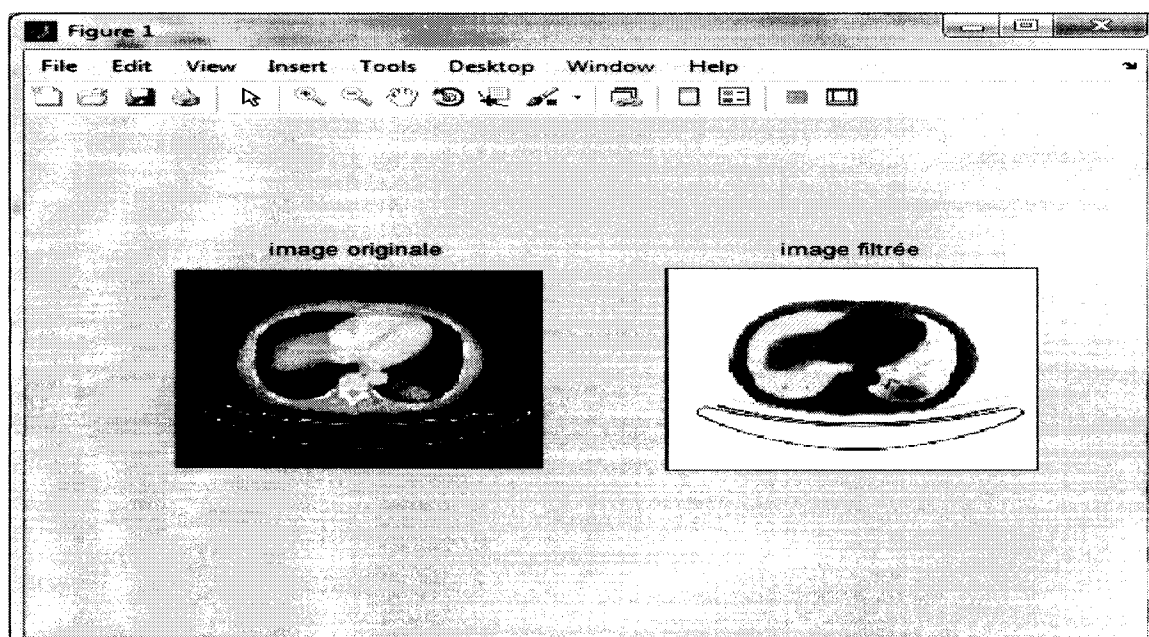


Figure 11.3 : Application du Filtre passe haut de Butterworth

## 2.9. Extraction de tous les régions d'une image

La segmentation est le découpage d'une image en différentes régions et/ou contours



La fonction **regionprops** pour extraire les régions d'une image et le résultat s'affichera dans la figure 12.3 si dessous

```
>> region_extrait = regionprops(labels,'BoundingBox');
```

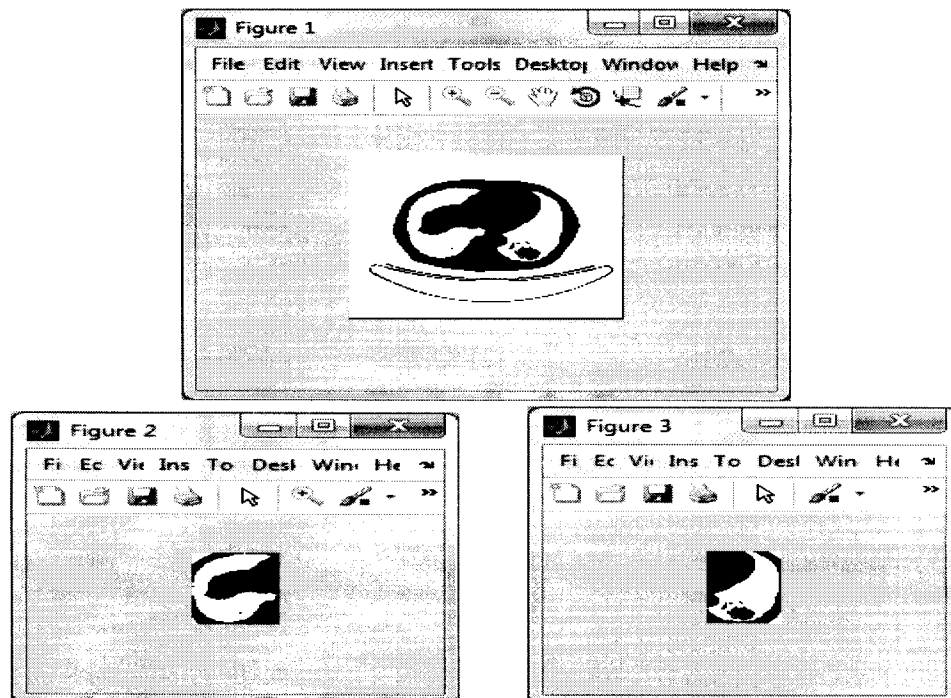


Figure 12.3 : Extraction des régions d'une image

### 2.10. Application de la méthode de croissance de région

Le but de la segmentation et d'Extraire (séparer) les entités d'une image

- Pour y appliquer un traitement spécifique
- Pour interpréter le contenu de l'image

Pour cela nous avons choisit la méthode de croissance de région

Dans la méthode de croissance de régions nous allons débiter avec un pixel, et on « ajouter » les pixels voisins qui répondent à un critère d'appartenance :

- Variance faible
- Niveau de gris répondant un seuil

Les pixels initiaux sont appelés « germes », « amorces » ou « semences ».

La région « grandit » à partir de son germe.

On part d'un germe (seed) et on l'étend en ajoutant les pixels voisins qui satisfont le critère d'homogénéité.

Le germe peut être choisi soit par un humain, soit de manière automatique en évitant les zones de fort contraste (gradient important).

### 2.10.1. Le premier exemple

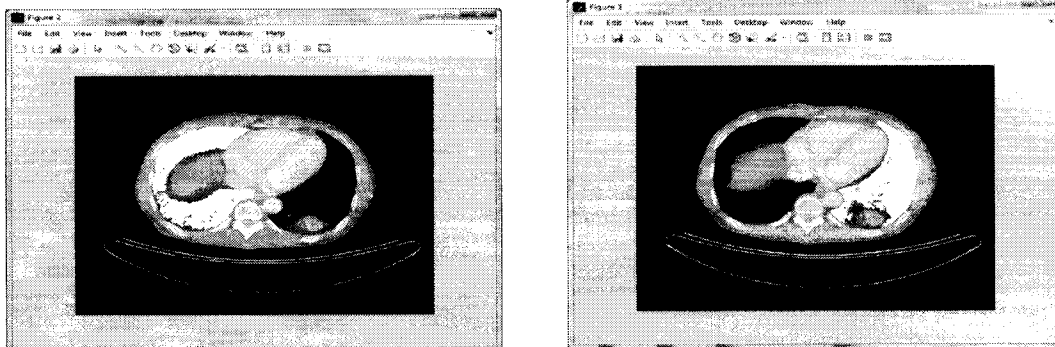


Figure 13,14 .3 : Les régions sélectionnées par region growing



Figure 15 .3 : La région sélectionnée par region growing

### 2.10.2. Le deuxième exemple

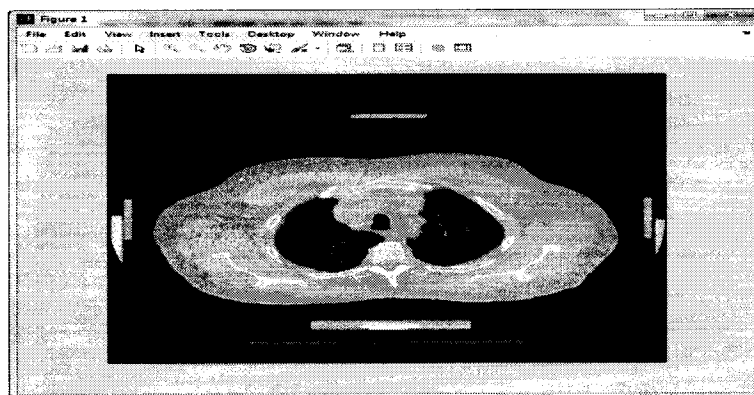


Figure 16.3 : La région sélectionnée par la méthode de croissance de région

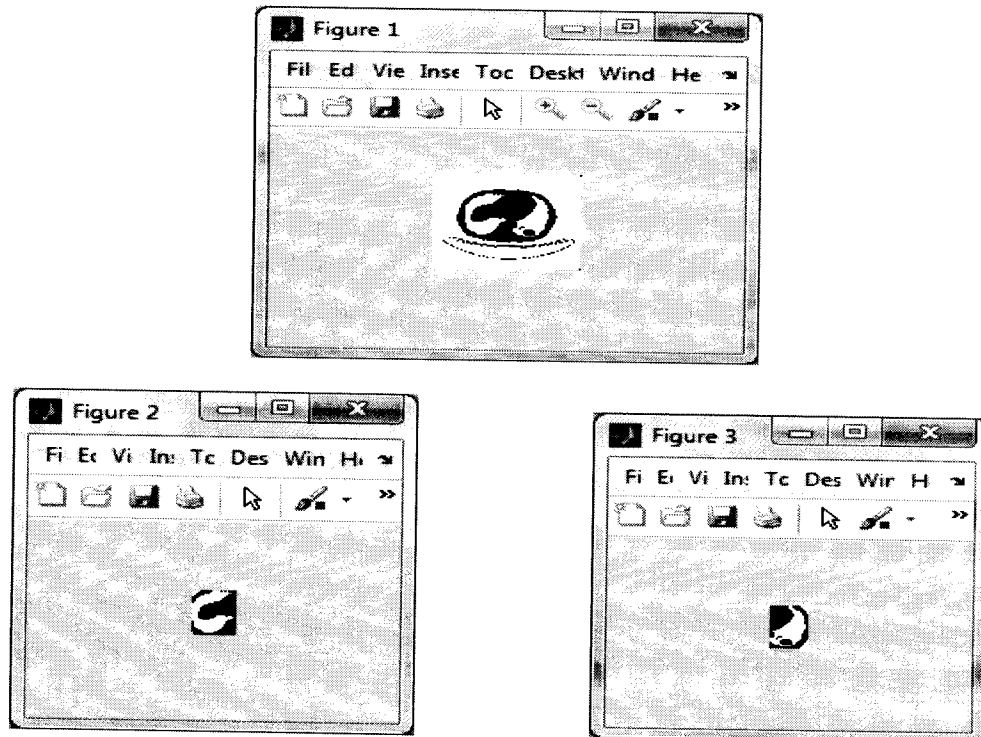
**Description de l'exemple selon la région sélectionnée :**

RegionGrowing Opening: Initial position (255|257|1) with 596 as initial pixel value!

RegionGrowing Ending: Found 361 pixels within the threshold range (71 polygon vertices)!

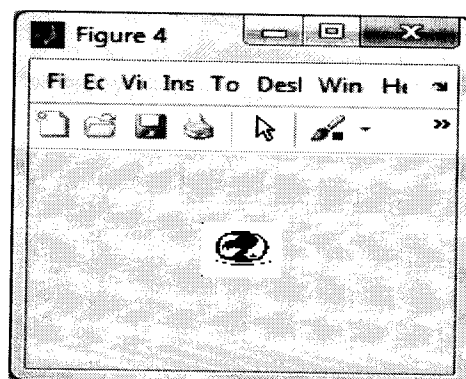
## 2.11. La multi Resolution

### 2.11.1. Extraction de région avec image1/2



**Figure 17.3 : Extraction de région en utilisons la multi résolution Image1/2**

### 2.11.2. Extraction de région avec image1/4



**Figure 18.3 : Extraction de région en utilisons la multi résolution Image1/4**

2.11.3. Application de la méthode de croissance de région sur image1/2

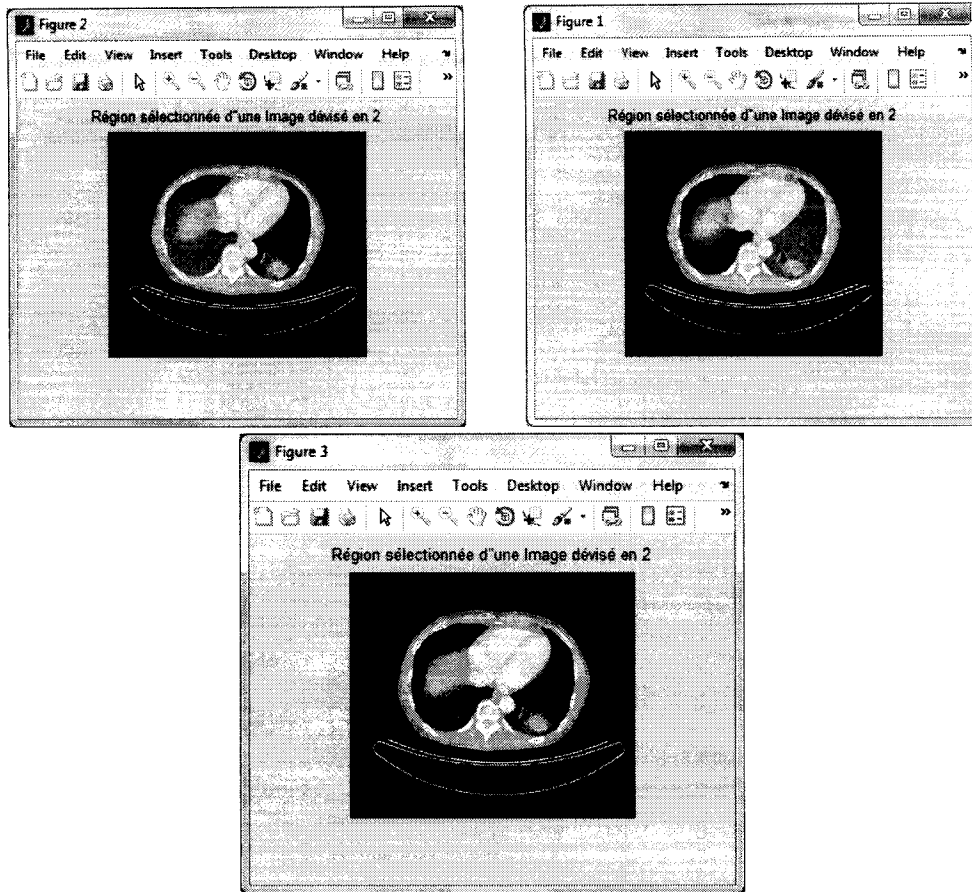


Figure 19.3 : croissance de région sur une image1/2

2.11.4. Application de la méthode de croissance de région sur image1/4

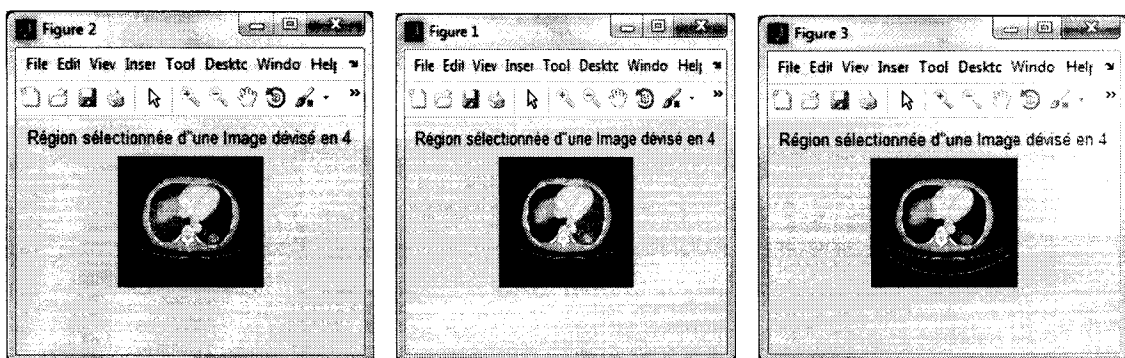


Figure 20.3 : croissance de région sur une image1/4

### 2.12. Exemple de la méthode de ligne de partage des eaux « WATERSHED »

L'idée de la méthode de ligne de partage des eaux est la Détection des bassins d'écoulement sur l'image de la norme du gradient.

#### 2.12.1. Algorithme de ligne de partage des eaux « Watershed »

- Trier les pixels par ordre croissant d'altitude
- Par groupe de pixels ( $p_i$ ) de même altitude
  - ✓ traiter les pixels par distance à un bassin existant
  - ✓ examiner le voisinage de chaque  $p_i$  pour l'étiqueter
  - ✓ Si un  $p_i$  non étiqueté (nouveau bassin), affectation d'une nouvelle étiquette et propagation de cette étiquette à ses voisins non étiquetés

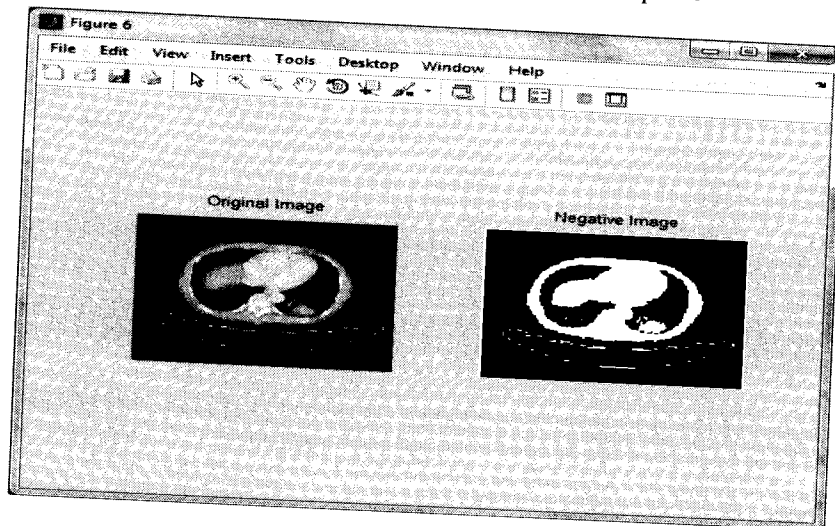


Figure 21.3 : comparaison entre image originale & image négative

La méthode de ligne de partage des eaux est caractérisée par la sensibilité au bruit.

Beaucoup de minima) → beaucoup de petites régions (on peut lisser l'image de la norme du gradient pour limiter ce défaut).

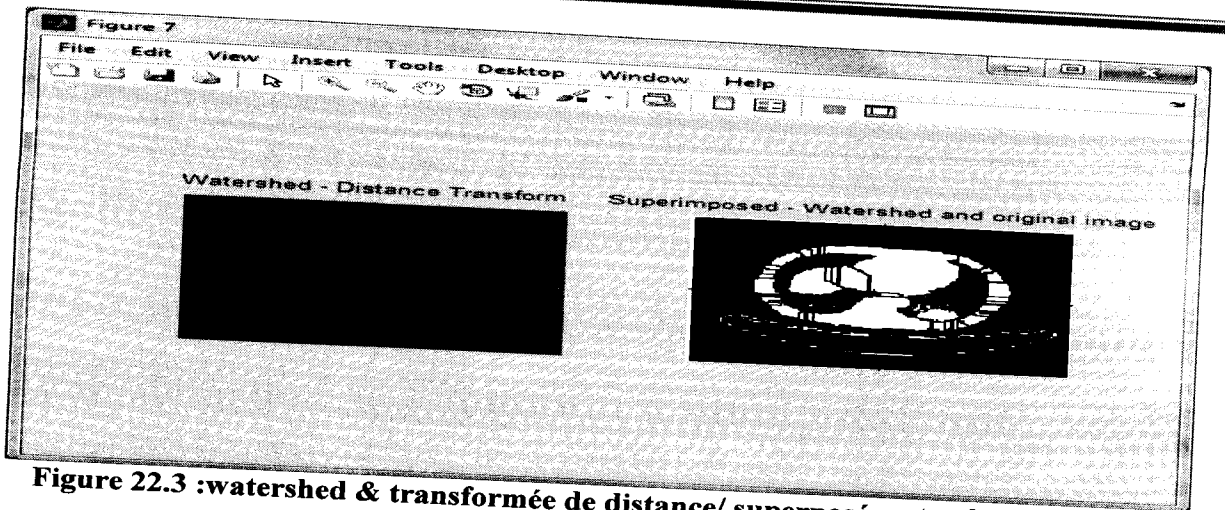


Figure 22.3 : watershed & transformée de distance/ superposé watershed et l'image originale (bassins)

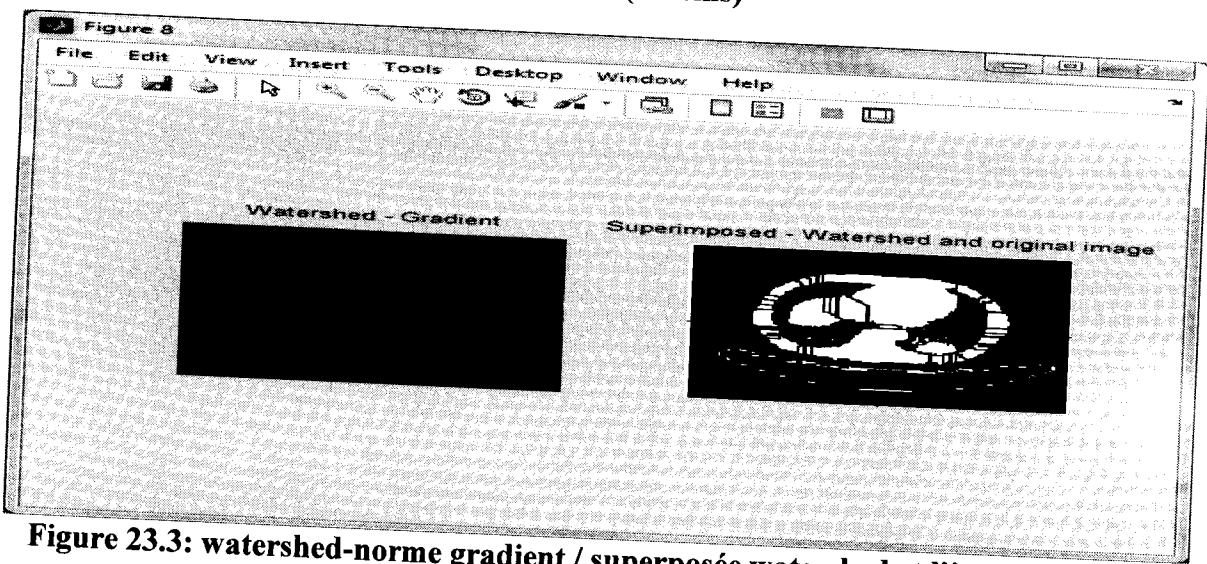


Figure 23.3: watershed-norme gradient / superposée watershed et l'image originale (bassins)

Des marqueurs donnés par un utilisateur permettent d'éviter la sur-segmentation.

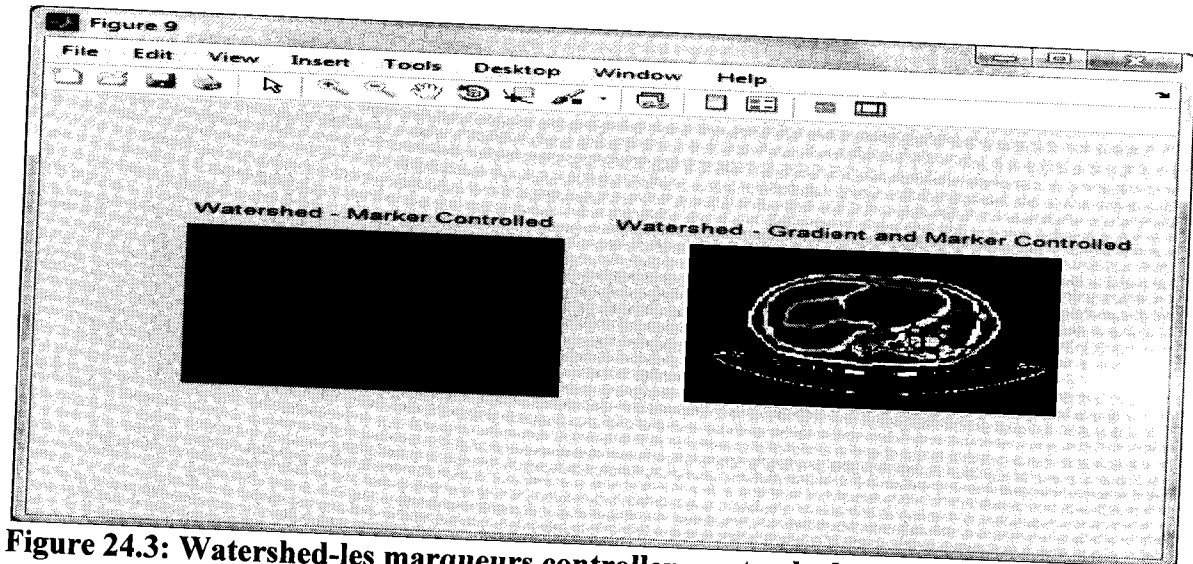


Figure 24.3: Watershed-les marqueurs contrôler , watershed-gradient et le marqueur contrôlée

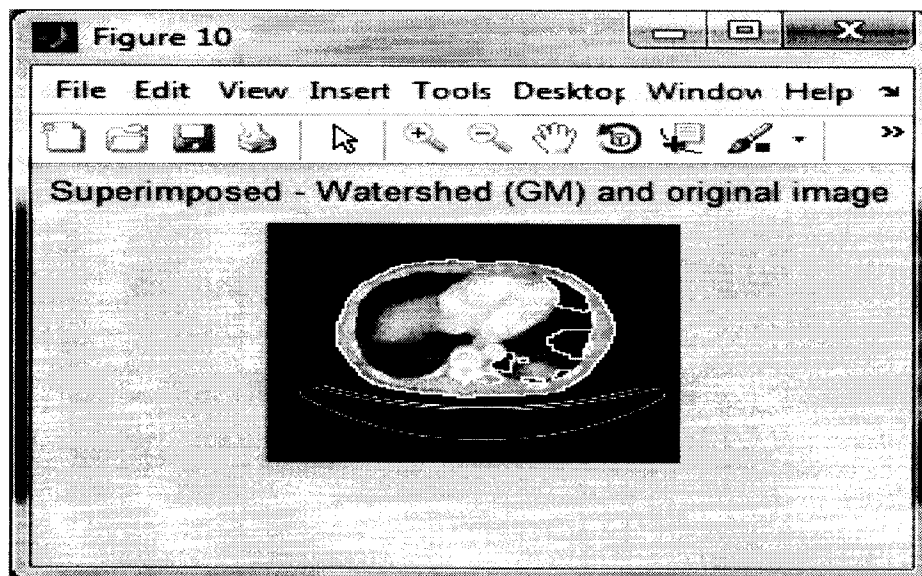


Figure 25.3: superpose watershed et l'image originale

### 3. Conclusion

D'après les résultats obtenus à partir de la segmentation d'image nous pourrions constater que les techniques de segmentation présentées ne génèrent pas une solution optimale.

Nous allons proposer d'utiliser d'autres techniques de segmentation plus complexes permettant d'obtenir de meilleurs résultats. Citons notamment :

- La décomposition adaptative (Adaptive Split).
- La recherche de formes géométriques (Model based segmentation).
- L'approche probabiliste (relaxation).
- Le regroupement (k-mean, mean shift).

**CONCLUSION**

**GENERALE**



# CONCLUSION GENERALE

---

Dans ce mémoire nous avons proposé une méthode d'extraction de région d'une image.

Pour atteindre cet objectif nous avons traité des différents images par plusieurs techniques telles que la conversion d'image RGB vers une image d'intensité, légalisation d'image et d'histogramme, le seuillage d'image, la filtrage (plusieurs types de filtrage).

Ensuite nous avons utilisé des méthodes de segmentation d'image en région on se basant su la méthode de croissance de région.

Comme perspectives a ce travail, nous proposons de

1. Utiliser plusieurs images.
2. Utiliser la Multi Résolution
3. Modifier les opérateurs de filtrage
4. Utiliser une autre méthode de segmentation
5. Comparer les résultats obtenus avec les autres méthodes de segmentation

# Références Bibliographiques

---

## Livres & Mémoires

[a] : Introduction au traitement d'images, Traitements de base

Nicholas Journet, 12 janvier 2011

[b] : Traitement d'images, Caroline Petitjean

[c] : Gonzalez, Woods (2008), Introduction p. 3-4

[d] : Master AG2I – Option SID – Traitement d'Images – F. Cabestaing

[e] : M1STIC Université de Bourgogne 2010-2011 LAMBERT VELLER Sylvain  
LECHEVALIER David

[f] : Gonzalez, Woods (2008), Intensity Transformations and Spatial Filtering p. 120

[g] : A. d'HARDANCOURT, « Fou du multimédia » Sybex 1995

[h] : M. KUNT, « Traitement numérique des images », vol.2, 1993

[i] : SAMAMBA Tony, Reconnaissance des formes comme outil d'aide aux  
traitements d'image. Cas des empreintes digitales, mémoire ISP/Bukavu, 2005-2006.

[j] : Gonzalez, Woods (2008), Introduction aux traitements d'images

[k] : Filtrage des images, Isabelle Bloch

Ecole Nationale Supérieure des Télécommunications - CNRS UMR 5141 LTCI Paris  
France

[l] : Segmentation 2D et 3D par Systèmes Multi-Agents, master thesis.pdf

[m] : Travail d'Intérêt Personnel Encadré Évaluation de la segmentation d'images

Institut de la francophonie pour l'informatique (IFI)

Bat. D, ruelle 42, rue Ta Quang Buu, Hanoi, Vietnam <http://www.ifi.auf.org>

[n] : Etudiant : DO Minh Chau , Promotion 11, IFI

Superviseur : Alain BOUCHER

Hanoi, janvier 2007

---

## Site Web

[1] : <http://fr.wikipedia.org/wiki/MATLAB>

[2] : <http://www.commentcamarche.net/contents/video/traitimg.php3>

