

## **Résumé :**

Dans ce mémoire nous allons traiter le thème de transcription de manuscrits arabes anciens, la transcription est l'action de reproduire un texte se trouvant dans une image dans un éditeur de texte, mais avant d'arriver à cette étape nous allons passer par plusieurs opérations de prétraitement qui sont : mise en niveau de gris, filtrage et binarisation, ensuite les étapes de segmentations :

- Segmentation en contours
- Segmentation en composantes connexes
- Filtrage des composantes connexes correspondant à des signes diacritiques
- Détection de la bande de base
- Squelettisation
- Segmentation en graphèmes
- Segmentation en mots

Après avoir segmenté l'image du manuscrit en mots bien distincts un transcritteur va reproduire le texte sur un éditeur de texte adapté que nous aurons réalisé après le texte en question sera enregistré sous format XML pour être ensuite transformé vers le format PDF ou HTML au choix.

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
UNIVERSITE MOULOU D MAMMERI DE TIZI OUZOU  
FACULTE GENIE ELECTRIQUE ET INFORMATIQUE  
DEPARTEMENT INFORMATIQUE



# MEMOIRE MASTER2

Spécialité : Système Informatique

## Thème

Transcription de manuscrits arabes anciens

Réalisée par :

M<sup>elle</sup> MEHADDI Lynda

Proposé par :

M<sup>r</sup> SOUALAH Med Ourabah

2010 - 2011

## *Remerciements*

*J'exprime mes sincères remerciements et ma gratitude a mon directeur de mémoire M<sup>R</sup> SOUALAH d'abord pour m'avoir proposé ce thème qui ma permis d'apprendre et de découvrir un domaine qui m'était inconnue ainsi que pour le temps qu'il ma accordé et l'aide, les conseils et l'orientation qu'il ma apporté tout long de la réalisation de ce modeste travail.*

*Mes remerciements vont aussi a mon frère FARID qui ma aidé tout au long de la réalisation de ce mémoire et a mes amis HOUME, AMIR et HADIA qui m'ont soutenu et encouragé.*

*Sans oublié de remercier mon fiancé SOFIANE qui ma aidé, soutenu, encouragé tout le long de la réalisation de ce travail.*

*Enfin j'adresse mes plus sincères remerciements aux membres du jury qui ont accepté de lire ce document.*

*Dédicaces*

*Je dédie ce modeste travail*

*À mes très chers parents qui m'ont toujours soutenu et encouragé*

*À mes deux frères Rabah et Farid*

*À mes deux sœurs Sonia et Katia*

*À mon Fiancé Sofiane*

*À tous mes amis : Hadia, Hocine, Amir, Rosa, Sabrina, Sarah.*

# Table des Matières

<b>Introduction Générale.....</b>	<b>001</b>
-----------------------------------	------------

## **Partie 1 : Problématique**

### **Chapitre1 : Position du problème et solution proposée**

1. Introduction.....	004
2. Problématique.....	004
3. Solution propose.....	004
4. Conclusion.....	005

### **Chapitre 2 : Concepts intervenant dans la solution**

1. Introduction.....	006
2. Notions sur l'image.....	006
2.1.Définition d'une image.....	006
2.2.Types d'images.....	006
2.2.1. Image numérique.....	006
2.2.1.1.Définition.....	006
2.2.1.2.Le pixel.....	007
2.2.1.3.Le contour.....	007
2.2.1.4.La dimension.....	008
2.2.1.5.La résolution.....	008
2.2.1.6.La région.....	008
2.2.1.7.La luminance.....	008
2.2.1.8.Le contraste.....	008
2.2.1.9.Le bruit.....	009
2.2.2. Image analogique.....	009
2.2.3. Image binaire.....	009
2.2.4. Image en niveau de gris.....	010
2.2.5. Image en couleur.....	010
3. Formats de fichiers images.....	013
4. Histogramme d'image.....	014
4.1.Modification d'histogramme.....	015
4.1.1. Egalisation d'histogramme.....	015
4.1.2. Etirement d'histogramme.....	016
4.2.Seuillage.....	016
4.3.Binarisation.....	018
5. les filtres.....	019
6. Processus de traitement d'images.....	019
7. Domaine d'application.....	019
8. Conclusion.....	020

### **Chapitre3 : Les manuscrits arabes anciens**

1. Introduction.....	021
2. Les manuscrits arabes anciens.....	021

2.1. Qu'est ce qu'un manuscrit ?	021
2.2. Caractéristiques des manuscrits arabes anciens	021
2.3. Alphabet de lettres Arabe	022
2.3.1. Un radical et une terminaison	022
2.3.2. Des formes identiques aux points prés	022
2.3.3. Les ligatures verticales	023
3. Conclusion	023

## Partie 2 : Etat de l'art

### Chapitre1 : Filtrage

1. Introduction	025
2. Méthodes Appliquées	025
2.1. Quelques définitions	025
2.1.1. Le voisinage d'un pixel	025
2.1.2. Le produit de convolution	026
a) Définition d'un produit de convolution	026
2.1.3. Définition d'un filtre linéaire	026
a) Définition d'un masque de convolution	026
b) Le produit de convolution et les filtres linéaires	027
2.2. Les filtres passe-bas	027
2.2.1. Le filtre moyennneur	027
3. Conclusion	028

### Chapitre2 : Binarisation

1. Introduction	029
2. Les méthodes de binarisation	029
2.1. Méthodes globales ET Méthodes Locales	030
2.1.1. Seuillage fixe	030
3. Conclusion	031

### Chapitre3 : Segmentation

1. Introduction	032
2. Définition	032
3. Méthode de Canny et Etiquetages des composantes connexes	033
3.1. Méthode de Canny	033
a) L'approche contour	033
b) Opérateur de Canny	035
3.2. Méthodes d'étiquetage des composantes connexes	036
3.2.1. L'approche région	036
3.2.2. Classification	038
a) Principe	038
4. Filtrages des composantes connexes correspondant à des signes diacritiques	040
4.1. Détection des signes diacritiques	040
4.2. L'algorithme	041
5. Extraction de la bande de base	043
5.1. Les boucles	043
5.2. Histogramme de projection horizontale	044
a) Principe	044

6. La squelettisation.....	044
6.1.Définition.....	045
6.2.Algorithme de Hilditch.....	047
a) Définition de quelques fonctions.....	047
b) Propriétés de l'algorithme de Hilditch.....	048
7. Segmentation en graphemes.....	048
7.1.Segmentation de l'écriture cursive.....	048
7.2.Procédure de segmentation en graphèmes.....	048
a) Groupement des petits segments.....	052
7.3.Segmentation à partir du squelette.....	052
8. Segmentation en mots.....	053
a) Procédure de segmentation en mots.....	053
9. Conclusion.....	057

## **Chapitre 4 : Transcription**

1. Introduction.....	059
2. Définition .....	059
3. Description de l'outil de transcription à mettre en place.....	060
3.1.Structure de l'outil de transcription.....	060
3.2.Fonctionnalités de l'outil de transcription.....	060
4. Règles de transcription.....	060
5. L'outil TEI.....	060
5.1.Les enjeux de la TEI « Text Encoding for Interchange ».....	060
5.2.Les buts de la TEI.....	061
5.3.Qu'est ce que la TEI aujourd'hui.....	061
5.4.Organisation de TEI.....	061
5.5.TEI système modulaire.....	062
5.6.L'envergure TEI.....	062
6. La transcription du manuscrit avec TEI.....	062
6.1.Le fichier XML/TEI à utiliser, sa structure.....	062
6.2.Première étape : reconnaître la structure logique du texte.....	063
6.3.Deuxième étape : transcrire le texte brut.....	063
6.4.Traitement des abréviations.....	063
6.5.Traitement des additions, suppressions et substitutions.....	063
6.6.Encodage des mains intervenues dans le manuscrit.....	064
6.7.Mise en page du texte.....	064
6.8.La signature.....	065
7. Conclusion.....	065

## **Partie 3 : Conception et Réalisation**

### **Chapitre1 : Conception : Algorithme et structure de données**

1. Introduction.....	067
2. Algorithme de mise en niveau de gris.....	067
2.1.Conversion d'une image couleur en image en niveau de gris.....	067
a) Principe.....	067
b) L'algorithme.....	067
3. Filtrage d'une image avec le filtre moyenneur.....	068
a) Exemple introductif.....	068
b) Principe.....	068
c) Algorithme.....	068
4. Algorithme de binarisation : La méthode du seuil fixe.....	069

a) Le principe de fonctionnement.....	069
b) L'algorithme .....	070
5. Segmentation.....	070
5.1.Algorithme de Canny.....	070
5.2.Algorithme d'étiquetage des composantes connexes.....	071
6. Approche coopérative séquentielle.....	074
a) Principe.....	074
7. Algorithme de filtrage des composantes connexes correspondant à des signes diacritiques.....	075
8. Détection de la bande de base.....	076
8.1.Algorithme de l'histogramme de projection horizontale.....	076
8.2.Détection des boucles dans chaque composante connexe.....	076
9. Algorithme de squelettisation de Hilditch.....	077
10. L'algorithme de segmentation en graphème.....	080
11. Algorithme de segmentation en mots.....	080
12. Conclusion.....	081
<b>Chapitre2 : Réalisation</b>	
1. Introduction.....	082
2. Présentation de l'outil de transcription réalisé.....	082
2.1.Interface principale.....	082
2.2.Traitement d'images.....	082
2.3.Transcription.....	083
3. Résultat obtenus pour les traitements des images de manuscrits.....	084
3.1.Image de manuscrit mise en niveau de gris.....	084
3.2.Image de manuscrits obtenus avec le filtre moyenneur.....	085
3.3.Images de manuscrits binarisée avec la méthode du seuil fixe.....	085
3.4.Image de manuscrits segmentés.....	088
a) En contour avec la méthode de Canny.....	089
b) En régions avec la méthode d'étiquetages des composantes connexes.....	090
4. Résultat obtenus pour la Transcription.....	091
5. Conclusion.....	093
<b>Conclusion Générale.....095</b>	
<b>Annexe A : Langage et environnement de développement</b>	
1. Environnement de travail.....	098
1.1.Langage de programmation Java.....	098
1.2. Système d'exploitation.....	098
1.3. IDE éclipse.....	098
2. Java et le traitement d'image.....	099
2.1.Le package java.awt.....	099
2.2.Le package javax.imagio.....	100
<b>Annexe B : Réseau de neurones :</b>	
1. Introduction.....	102
2. Caractéristiques des réseaux de neurones artificiels.....	106
2.1.Poids (w) et biais (b) d'un neurone.....	106
2.2. La fonction de sommation.....	106
2.3.La fonction de transfert.....	107
2.4.Réseaux statiques.....	108

**Bibliographie.....110**

# Liste des Figures

Fig.1.2.1.Représentation d'une en pixel.....	007
Fig1.2.2. (a)8-Voisinage (b) 4-voisinage.....	007
Fig.1.2.3. Image Binaire.....	009
Fig.1.2.4. Image en niveau de gris.....	010
Fig.1.2.5.Image couleur.....	010
Fig.1.2.6.Image montrant l'espace RVB.....	011
Fig.1.2.7. Histogramme.....	015
Fig.1.2.8.Egalisation d'histogramme.....	016
Fig.1.2.9. Etirement d'histogramme.....	016
Fig.1.2.10.Seuillage.....	017
Fig.1.2.11. Schéma du processus de traitement numérique d'image.....	019
Fig.1.3.1. Manuscrit arabe ancien.....	022
Fig.1.3.2. Ligatures verticales (sans les points) utilisées dans notre alphabet.....	023
Fig.2.1.1. Représentation d'un pixel et son voisinage.....	026
Fig.2.1.2.Exemple de masques de convolution.....	027
Fig.2.2.1. Le choix de seuil dans un histogramme bimodal.....	030
Fig.2.2.2.Image binarisée avec un seuil 127.....	031
Fig.2.3.1.Contour.....	034
Fig.2.3.2. Dérivée Première.....	034
Fig.2.3.3. Dérivée seconde.....	034
Fig.2.3.4décompositions successives des blocs.....	036
Fig.2.3.5.Représentation sous forme QuadTree.....	037
Fig.2.3.6. Croissance progressive d'une région.....	037
Fig.2.3.7. Les pixels voisins.....	039
Fig.2.3.8. Exemple d'image avec rectangles englobant.....	039
Fig.2.3.9. Procédure de filtrage des signes diacritiques .....	041
Fig.2.3.10. Pic maximal de l'histogramme (ligne jaune) et seuils pour déterminer une Approximation de la ligne haute et de la ligne basse .....	042

Fig.2.3.11.Présence d'un signe diacritique collé au corps du texte.....	042
Fig.2.3.12. Utilisation des boucles pour déterminer une zone de recherche du maximum de l'histogramme de projection horizontale.....	043
Fig.2.3.13. Motif en forme T et le squelette.....	045
Fig.2.3.14. squelettisation d'une forme quelconque.....	046
Fig.2.3.15. Axe médian d'une forme continue (trait pointillé).....	046
Fig.2.3.16. Utilisation des points singuliers du squelette pour trouver une évaluation plus fine de la bande de base.....	047
Fig.2.3.17. Détection fine de la bande de base.....	047
Fig.2.3.18. exemples de segmentations en graphèmes.....	049
Fig.2.3.19. Un exemple de chèque français sur le quel sera appliquer la segmentation.....	050
Fig.2.3.20. Procédure de segmentation du mot trente en graphèmes.....	050
Fig.2.3.21. L'image originale du mot « trente ».....	051
Fig.2.3.22. Résultat du processus segmentation. Chaque segment est illustré en couleur différente.....	051
Fig.2.3.23. Algorithme du Two-cost-grouping.....	051
Fig.2.3.24. Le mot « trente » après le processus de regroupement des segments.....	052
Fig.2.3.25. Segmentation à partir du squelette.....	052
Fig.2.3.26.a. Région dans un document manuscrit.....	054
Fig.2.3.26.b. Composants reliés intérieures et découpes extérieures.....	054
Fig.2.3.26. Composants reliés d'une petite section de l'image.....	054
Fig.2.3.27.Echantillons de caractère Alef.....	055
Fig.2.3.28. N'est pas un espace de mots.....	056
Fig.2.3.29. Est un espace de mots.....	056
Fig.2.3.30. Exemples de 4 paires de faisceaux.....	056
Fig.2.3.31. Image de document segmenté en mots par le réseau neurologique.....	057
Fig.2.4.1. Organisation de la TEI.....	061
Fig.3.1.1. Principe de la coopération séquentielle.....	075
Fig.3.1.2.Exemple correspondant a l'algorithme.....	076
Fig.3.2.1.Interface principale.....	083

Fig.3.2.2.Menu et sous menu.....	083
Fig.3.2.3.Interface principale de l'éditeur de texte.....	083
Fig.3.2.4.Image de manuscrits arabe numérisé.....	084
Fig.3.2.5.Image de manuscrits mis en niveau de gris.....	084
Fig.3.2.6. Même image après filtrage ayant été mise au préalable en niveau de gris.....	085
Fig.3.2.7. Manuscrit1 : image originale.....	085
Fig.3.2.8. Manuscrit 1 : Image de manuscrit binarisée avec la méthode du seuil fixe.....	086
Fig.3.2.9. Manuscrit2 : Image originale.....	086
Fig.3.2.10. Manuscrit2 : Image de manuscrit binarisée avec la méthode du seuil fixe.....	087
Fig.3.2.11.Manuscrit3 : Image originale.....	087
Fig.3.2.12. Manuscrit3 : Image de manuscrit binarisée avec la méthode du seuil fixe.....	088
Fig.3.2.13. Image de manuscrit arabe.....	089
Fig.3.2.14. Résultat avec opérateur de Canny.....	089
Fig.3.2.15. Résultat avec de segmentation en composantes connexes.....	090
Fig.3.2.16. Image manuscrit originale.....	091
Fig.3.2.17. Exemple de manuscrit reproduit sur l'éditeur de texte.....	092
Fig.3.2.18. Résultat du manuscrit transcrit en HTML.....	092
Fig.4.1.Interface principale d'éclipse.....	099

# Liste des Tableaux

Tableau 1.2.1. Tableau des formats de fichiers images généraux.....	014
Tableau.1.3.1. Lettres arabe : début/milieu vers isolée/fin.....	023
Tableau.1.3.2. Quelques lettres arabes et leur forme sans diacritiques.....	023

# Introduction Générale

## **Introduction Générale :**

Bien que l'arabe soit une langue parlée par plus de 250 millions de personnes dans le monde, il n'y a pas à ce jour d'outil de transcription de l'écriture arabe manuscrite. Les champs d'applications sont pourtant très nombreux. De nombreux travaux ont été menés au cours des dernières années, mais ce sujet reste un domaine de recherche actif.

Les manuscrits sont les livres des records de l'histoire. Les civilisations anciennes du monde entier ont utilisé les manuscrits pour consigner les événements importants de l'histoire, ou même simplement enregistrer les transactions au jour le jour de la gouvernance locale. Notre compréhension de notre passé dépend en grande partie de ces documents. Les chercheurs utilisent ces manuscrits pour produire une recherche qui va finalement dans les livres d'histoire. De toute évidence l'importance de ces manuscrits ne peut être estimée. Toutefois ces manuscrits ont tendance à se détériorer rapidement après manipulations répétées en raison de leur âge et de leur fragilité. Les archives du monde entier ont donc essayé de passer à l'ère numérique, en faisant des versions numérisées de ces manuscrits, mais le travail ne s'arrête pas là, car même numérisés sous forme d'images, ces manuscrits sont inaccessible en plein texte pour des recherche en sein d'une base de donnée dans une bibliothèque ou sur internet dans un moteur de recherche et c'est a cet instant qu'intervient la transcription.

La transcription : c'est l'opération qui permet de reproduire le texte se trouvant dans l'image d'un manuscrit sur un document texte.

Les manuscrits transcrits vont permettre :

- De préserver la qualité et la durée de vies des manuscrits papiers,
- De promouvoir l'accès à ces derniers par le billet du web, ou en sein des bibliothèques,
- La possibilité de recherche plein texte.
- La possibilité d'affichage sous divers formats.

Etant donnée que notre support de départ sont des manuscrits numérisés, donc des images de manuscrits qui vont être traitées avant d'être transcrites dont le but de faciliter cette dernière opération, car les écritures arabes anciennes ne ressemblent en rien a l'écriture imprimer afin de pouvoir aider le transcripateur dans son travail, nous devons d'abord effectuer une segmentation de l'image du manuscrits en mots bien distinct (chaque mots sera d'une couleur différente de son voisin), mais avant d'arriver a la segmentation en mots nous devons effectuer plusieurs étapes de prétraitement (mise en niveau de gris, filtrage, binarisation) afin

d'améliorer la qualité du manuscrit numérisé, ensuite l'image du manuscrit doit passer par plusieurs phases de segmentation (segmentation en composantes connexes, filtrage des composantes connexes correspondant à des signes diacritiques, détection de la bande de base, squelettisation, segmentation en graphèmes ) pour arriver à segmenter en mots.

A partir de là le transcritteur aura une image résultat où il sera très facile de reproduire le texte sur un outil de traitement de texte, pour pouvoir ensuite éditer ce document en TEI/XML, puis pouvoir l'afficher dans divers formats grâce aux feuilles de styles XSL(XSLT, XSLT-FO). La solution qui facilitera grandement l'accès au contenu du document.

Cette solution est présentée dans ce mémoire de la manière suivante :

❖ **Partie 1 : Problématique** qui est divisée en deux chapitres

- **Chapitre1:** Position du problème et Solution proposée: dans ce chapitre nous présentons la problématique de notre thème et la solution que nous proposons.
- **Chapitre2:** Concepts intervenants dans la solution: dans ce deuxième chapitre nous parlons plutôt de généralités sur le traitement d'image pour introduire le lecteur dans le domaine du traitement d'image.

❖ **Partie 2: Etat de l'art** qui se compose de trois chapitres, qui traitent de tout les concepts intervenant dans la solution proposée.

- **Chapitre1:** Filtrage, ce chapitre concerne le filtrage c-à-d débarrassé l'image du manuscrit de toutes impuretés.
- **Chapitre2 :** Binarisation, la binarisation veut dire transformer une image couleur ou en niveau de gris en image noir et blanc
- **Chapitre3 :** Segmentation, ce chapitre traite les différentes étapes de segmentation :
  - Segmentation en composantes connexes
  - Filtrages des composantes connexes correspondant à des signes diacritiques
  - Squelettisation
  - Segmentation en graphèmes
  - Segmentation en mots
- **Chapitre 4 :** Transcription

❖ **Partie 3 : Conception et Réalisation**

- **Chapitre1 :** Conception Algorithme et structure de données
- **Chapitre2 :** Réalisation : Présentation de l'outil de transcription réalisé

# Partie 1 : Problématique

### 1. Introduction:

Comme décrit dans l'introduction générale ce mémoire traitera de la transcription de manuscrits arabes anciens, dans cette partie nous allons décrire le problème ainsi que la solution que nous avons proposée.

### 2. Problématique :

La complexité de l'écriture manuscrite arabe rend sa lecture très difficile pour les non initiés. De plus cette complexité se trouve accrue par le caractère variable de la calligraphie arabe. En effet, nous recensons plusieurs styles d'écriture, nous citons entre autre: le style "*Maghribi*", le style "*Koufi*", le style "*nouskhi*",...etc.

Par ailleurs, un grand intérêt est porté à ce type de document qui commence de plus en plus à conquérir le domaine du Web. Il est donc impératif de proposer une solution qui puisse rendre sa lecture facile et aisée, voire proposer une solution qui assiste le lecteur dans son accès aux divers contenu du document manuscrit ancien.

La transcription est une solution efficace qui apporte une réponse réelle au problème de lecture du manuscrit arabe plus. Il s'agira de mettre à la disposition du transcripateur un outil qui respecte d'une manière intégrale la structure du document à transcrire. De ce fait, nous devrions identifier :

- La zone de texte et sa dispersion sur le document,
- les zones images se trouvant sur le manuscrit numérisé,
- Les diverses annotations présentes sur le document.

La complexité de l'écriture manuscrite arabe rend sa lecture très difficile pour les non initiés.

### 3. Solution proposée :

La solution que nous proposons est la suivante :

Dans un premier temps l'image du manuscrit numérisé sera segmentée en mots bien distincts. Par ailleurs, des prétraitements seront effectués au préalable sur l'image du manuscrit, avant d'arriver d'entamer les procédures de segmentation intermédiaires.

La segmentation en mots permet de faciliter la prochaine étape de la solution qui est la transcription.

Un transcripteur va reproduire le texte de l'image segmentée tel qu'il est grâce à un éditeur adapté, à partir de là nous allons effectuer une transformation de cette reproduction qui est sous format TEI/XML en fichier PDF ou HTML.

#### **4. Conclusion :**

Notre solution est une réponse à un problème pragmatique épineux et difficile. En effet, si l'accès aux images des manuscrits arabes numérisé commence à devenir une réalité quotidienne du web, l'accès à leur contenu demeure toujours réservé aux experts et aux paléographes. Par conséquent, notre solution sera d'un apport appréciable aux non initiés et aux chercheurs, qui s'intéressent à ce type de document.

## **1. Introduction:**

L'image constitue l'un des outils les plus importants qu'utilise l'être humain pour communiquer avec autrui. C'est un moyen de communication universel dont la richesse du contenu permet aux individus de tous âges et de toute culture de se comprendre.

Chacun peut alors analyser l'image à sa manière pour en dégager une impression et en extraire des informations précises. De ce fait, le traitement d'images est l'ensemble des méthodes et techniques permettant d'améliorer l'aspect visuel d'une image.

Ainsi, dans cette partie on va présenter les notions générales sur l'image et les concepts qui s'y rattachent.

## **2. Notions sur l'image:**

### **2.1. Définition d'une image:**

Une image est représentation d'une personne ou d'une chose par la peinture, la sculpture, le dessin, la photographie, le film ou divers procédés numériques de visualisation [1].

Une image est également considérée comme un signal variant dans l'espace suivant les directions, horizontal X et vertical Y. Le traitement s'effectue en appliquant tout les outils mathématiques de traitement du signal à deux dimensions [2].

### **2.2. Les types d'images:**

#### **2.2.1. Image numérique :**

##### **2.2.1.1. Définition :**

L'image dont la surface est divisée en éléments de tailles fixes appelés cellules ou pixels, ayant chacun une caractéristique à un niveau de gris ou de couleur prélevé sur l'emplacement correspondant dans l'image réelle, ou calculé à partir d'une description interne de la scène à représenter [3]

##### **2.2.1.2. Le pixel :**

Un pixel est l'abréviation de l'expression anglaise « Picture Element » qui est le plus petit élément constitutif d'une image numérique. C'est la plus petite unité d'information que peut manipuler les matériels et les logiciels d'affichage ou d'impression. C'est une unité de surface permettant de mesurer une image numérique [4].



Fig.1.2.1.Représentation d'une image en pixel

### ❖ Notion de voisinage d'un pixel :

Dans une image un pixel  $p(x, y)$  possède quatre voisins (horizontaux et verticaux) qui fournissent ce qu'on appelle le 4-voisinage de  $p(x, y)$ . Si on considère un pixel comme un carré élémentaire, les pixels présentant un coté commun avec  $p(x, y)$  sont appelés le 4-voisins de  $p(x, y)$ . Le pixel  $p(x, y)$  possède aussi quatre voisins diagonaux, ce sont les pixels qui ont un sommet commun avec  $p(x, y)$ . L'ensemble des huit voisins du pixel  $(x, y)$  représente ce qu'on appelle les 8-voisins (8-voisinage). Ces huit voisins forment la fenêtre 3x3 du pixel  $p(x, y)$ .

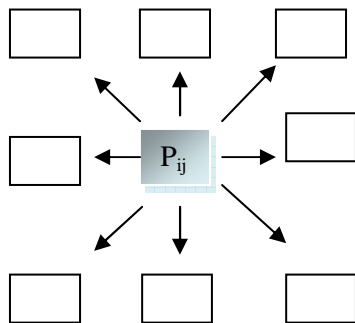


Fig.1.2.2. (a) :8-voisinage

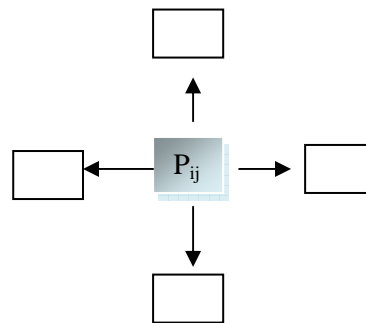


Fig.1.2.2. (b) : 4-voisinage

### 2.2.1.3. Le contour :

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 [5].

**2.2.1.4. La dimension :**

Définit la taille d'une image qui se présente sous forme d'une matrice dont les éléments représentent des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonne nous donne le nombre total de pixels dans une image [6].

**2.2.1.5. La résolution :**

La résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre).

On utilise aussi le mot résolution pour désigner le nombre de pixels affichable horizontalement ou verticalement sur un moniteur, plus grand est ce nombre, meilleur est la résolution [6].

**2.2.1.6. La région :**

Ensemble connexe de pixels ayant une ou plusieurs propriétés communes.

**2.2.1.7. La luminance :**

La luminance 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'air apparente de celle-ci.

Une bonne luminance est caractérisé par :

- Des images lumineuses (brillantes).
- Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir. Ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- L'absence de parasites [6].

**2.2.1.8. Le contraste :**

Le contraste est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'image. Une augmentation du contraste a pour effet que les zones sombres deviennent encore plus sombres et les couleurs claires encore plus claires.

Les différences entre les couleurs claires et les couleurs sombres sont donc accentuées.

La loi de Weber-Fechner définit le contraste  $C_w$  comme :

$$C_w = \frac{L - L_{fond}}{L_{fond}}$$

L et  $L_{\text{fond}}$  désignent respectivement la luminance de l'objet et du fond directement aux abords de l'objet. Ce contraste mesure la perception que l'on a d'un objet de luminance L placé sur un fond de luminance  $L_{\text{fond}}$  [7].

Le contraste de Michelson  $C_m$  (compris entre 0 et 1) est défini comme suit :

$$C_m = \frac{L_{\text{max}} - L_{\text{min}}}{L_{\text{max}} + L_{\text{min}}} = \frac{I_{\text{max}} - I_{\text{min}}}{I_{\text{max}} + I_{\text{min}}}$$

Où L désigne la luminance et I l'intensité lumineuse [7].

#### 2.2.1.9. Le bruit :

Le bruit (parasite) dans une image est considéré comme un phénomène de brusques variations de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électronique du capteur [6].

#### 2.2.2. Image analogique :

Une image analogique peut être représentée mathématiquement comme une distribution continue d'intensités lumineuses dans un plan xOy [2].

#### 2.2.3. Image binaire :

L'image binaire est représentée sous forme d'une matrice dont les éléments valent 0 ou 1. Le niveau 0 représente le noir absolu et le niveau 1 représente le blanc. Dans ce cas un seul bit est utilisé pour coder le niveau du gris.



Fig.1.2.3. Image Binaire

#### 2.2.4. Image en niveau 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 finis de niveaux intermédiaires. Pour représenter les images en niveau de gris, nous pouvons attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur est comprise entre 0 et 255 [8].



Fig.1.2.4. Image en niveau de gris

#### 2.2.5. Image couleur :

Ce sont des images codées en utilisant, en général, le codage des trois couleurs fondamentales (Rouge, Vert, Bleu), on parle alors d'images RVB. Une image couleur RVB contient trois plans de couleurs : rouge, vert, bleu. Chaque plan est codé comme une image en niveau de gris avec des valeurs allant de 0 à 255.



Fig.1.2.5. Image couleur

### ■ Types d'images couleurs :

Il existe différents types d'images couleurs en fonction du nombre de bits utilisés pour le stockage de l'information couleur :

#### ❖ Image 24 bits :

Le codage de la couleur est réalisé sur 3 octets, chaque octet représente la valeur d'une composante couleur par un entier de 0 à 255. Ces trois valeurs codent généralement la couleur dans RVB.

#### ❖ Image à 256 couleurs :

Pour que l'information couleur soit codée sur un octet (pour gagner de la place) au lieu de trois, on utilise une palette (table) de couleurs attachée à l'image. Chaque pixel véhiculera un chiffre compris entre 0 et 255. A chacun de ses chiffres va correspondre une couleur, définie par son code RVB et stockée dans une palette.

Lors de la visualisation de l'image, la correspondance se fait entre le numéro de la couleur affectée à chaque pixel et le code couleur RVB correspondant.

### ■ Les espaces couleurs :

Il existe de nombreuses façons de percevoir les couleurs d'une image, l'espace de représentation le plus connu est l'espace RGB (Red, Green, Blue dont la notation en français correspondante est RVB).

#### ❖ L'espace RVB :

L'espace RVB est l'espace le plus répandu et qui est utilisé, entre autres par la plupart des caméras et des écrans vidéo. Cet espace tridimensionnel est construit à partir d'une base de trois stimuli couleurs primaires que sont le rouge, le vert et le bleu.

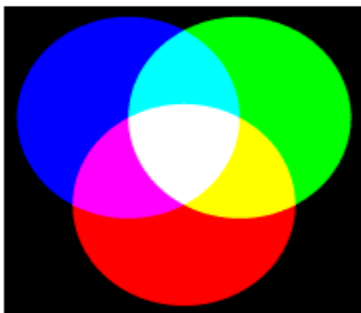


Fig.1.2.6.Image montrant l'espace RVB

### ❖ L'espace XYZ:

L'espace XYZ s'appuie sur des couleurs primaires imaginaires afin d'élargir l'espace colorimétriques, afin de ne plus avoir à manipuler des coordonnées négatives. Elles sont notées X pour l'axe du rouge et Y pour l'axe du vert puis Z pour l'axe du bleu et sont obtenues par une transformation linéaire, ce qui signifie que le nouvel espace est proportionnel à l'espace RVB d'origine. L'espace XYZ n'est rien d'autre qu'un modèle particulier de RVB.

#### ■ Conversion d'une image couleur en niveau de gris :

La conversion d'une image couleur en une image en niveau de gris se fait par suite à l'application de la fonction suivante :

$$\text{Avec : } I(i, j) = \frac{[R(i, j) + V(i, j) + B(i, j)]}{3}$$

$I(i, j)$  : Niveau de gris du pixel situé à la ligne  $i$  et à la colonne  $j$ .

$R(i, j)$  : Intensité de la couleur rouge du pixel  $(i, j)$ .

$V(i, j)$  : Intensité de la couleur verte du pixel  $(i, j)$ .

$B(i, j)$  : Intensité de la couleur bleu du pixel  $(i, j)$ .

La C.I.E (commission international de l'éclairage) propose, de caractériser l'information de luminance (la valeur de gris) d'un pixel par deux formules :

Dans sa recommandation 709, qui concerne les couleurs « vraies » ou naturelles :

$$\text{Gris} = 0.2125 \cdot \text{Rouge} + 0.7154 \cdot \text{Vert} + 0.0721 \cdot \text{Bleu}$$

Dans sa recommandation 601 pour les couleurs non linéaires, c'est-à-dire avec correction du gamma (image vue à partir d'un écran vidéo) :

$$\text{Gris} = 0.299 \cdot \text{Rouge} + 0.587 \cdot \text{Vert} + 0.114 \cdot \text{Bleu}$$

Ces formules rendent compte de la manière dont l'œil humain perçoit les trois composantes, rouge, vert et bleu, de la lumière. Pour chacune d'elles, la somme des trois coefficients vaut 1. On remarquera la forte inégalité entre ceux-ci : une lumière verte apparaît plus claire qu'une lumière rouge, et encore plus qu'une lumière bleue [8].

### 3. Les formats des fichiers images:

En général, l'objectif principal de la numérisation des manuscrits est leur conservation.

L'évolution rapide des technologies de l'information et de la communication ne joue pas en faveur de la pérennité des données. En effet, bien souvent il arrive qu'une nouvelle version d'un logiciel d'un éditeur donné, n'arrive pas à lire un fichier d'une version antérieure.

Le choix d'un format de numérisation doit être motivé par la pérennité des données : Cet aspect se retrouve souvent fournit par les formats ouverts, qui ne dépendent pas de la plate forme logicielle.

Il est fortement recommandé de numériser les manuscrits en utilisant les formats ouverts.

Néanmoins, il convient toujours de s'assurer de l'interopérabilité des versions : Ce problème se retrouve souvent dans les formats de fichiers qui passent du monde propriétaire au monde du logiciel libre, exemple : le format TIFF, qui à l'origine était la propriété d'Aldus Corporation et est passé au monde du logiciel libre en devenant outil d'Acrobat.

Les formats de fichiers images se répartissent entre les formats ouverts dont les spécifications sont données par les constructeurs, qu'il est recommandé d'utiliser et les formats propriétaires qu'il est vivement déconseillé d'utiliser, au moins pour les versions d'archive. Par ailleurs, le format de diffusion doit tenir compte du taux de compression.

Le tableau suivant donne les formats d'images les plus utilisés ainsi que leurs caractéristiques :

	TIFF 6.0 (Tagged Image File Format)	GIF 89a (Graphics Interchange Format)	JPEG (Joint Photographic Expert Group)/JFIF (JPEG File Interchange Format)	PNG 1.2 (Portable Network Graphics)	PDF 1.4 (Portable Document Format)
Extension(s)	.tif .tiff	.gif	.jpeg, .jpg, .jif, .jif	.png	.pdf
Profondeur de Bit	Bitonal 1 bit; niveaux de gris ou palette couleur 4 or 8 bits; couleur jusqu'à 64 bits.	bitonal, niveaux de gris, ou couleur : 1 - 8 bits	8 bits en niveaux de gris; couleur 24 bits	1- 48-bits; palette couleur ou niveaux de gris 1/2/4/8 bits, 16 bits en niveaux de gris, true color 24/48 bits	4 bits en niveaux de gris; couleur 8 bits; couleur jusqu'à 64 bits
Compression	Non compressée Non destructive : ITU-T.6, LZW, ...etc. Destructive : JPEG	Non destructive : LZW.	Destructive : JPEG  Non destructive	Non destructive	Non compressée, Non destructive : ITU-T.6, LZW, JBIG Destructive : JPEG
Standard/ Propriétaire	standard	standard	JPEG : ISO 10918-1/2 JFIF: standard.	ISO 15948.	standard.
Supporté par le Web	Plug-in ou application externe	Native	Native	Natif (mais toujours incomplet)	Plug-in ou application externe
Commentaires	Supporte des images/fichiers multiples.	Peut être remplacé par le PNG; Entrelacement et transparence supportés par la plupart des navigateurs Web	JPEG progressif largement supporté par les navigateurs Web.	Pour remplacer le GIF, bien que la pénétration du marché reste disparate.	À préférer pour l'impression et le visionnage des documents multi- pages; usage fortement répandu.

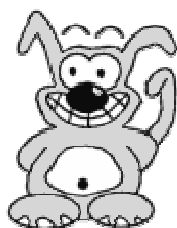
Tableau.1.2.1. Tableau des formats de fichiers images généraux [9]

#### 4. Histogramme d'image :

L'histogramme d'une image représente le nombre de pixels en fonction du niveau de gris dans l'image, c'est-à-dire la distribution des valeurs de niveau de gris dans une image.

L'histogramme est défini comme une fonction discrète qui associe à chaque valeur d'intensité le nombre de pixels prenant cette valeur dans l'image.

Pour les images en couleur, plusieurs histogrammes sont nécessaires. Par exemple, pour une image en RVB (Rouge, Vert, Bleu), trois histogrammes représentant respectivement la distribution des valeurs respectives des composantes rouge, verte, bleue.



L'histogramme et la palette associés à cette image sont respectivement les suivants :

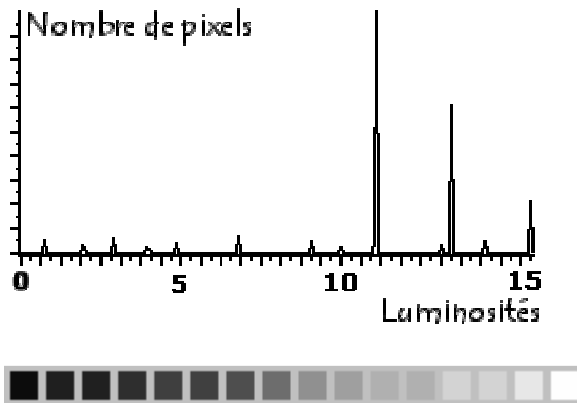


Fig.1.2.7. Histogramme [10]

#### 4.1. Modification d'histogramme :

Généralement le nombre de niveau de gris différent présent dans une image est tellement faible que celle-ci apparaît sombre ou trop claire, ce qui rend la distinction des objets difficile.

La modification d'histogramme est une opération qui consiste à donner à l'histogramme une forme bien déterminée dans le but d'améliorer l'image.

En outre sa modification ne modifie pas les informations contenues dans l'image mais les rend plus au moins visibles [10].

##### 4.1.1. Egalisation d'histogramme :

L'égalisation d'histogramme a pour but d'harmoniser la répartition des niveaux de luminosité de l'image, de telle manière à tendre vers un même nombre de pixel pour chacun des niveaux de l'histogramme. Cette opération vise à augmenter les nuances dans l'image [10].



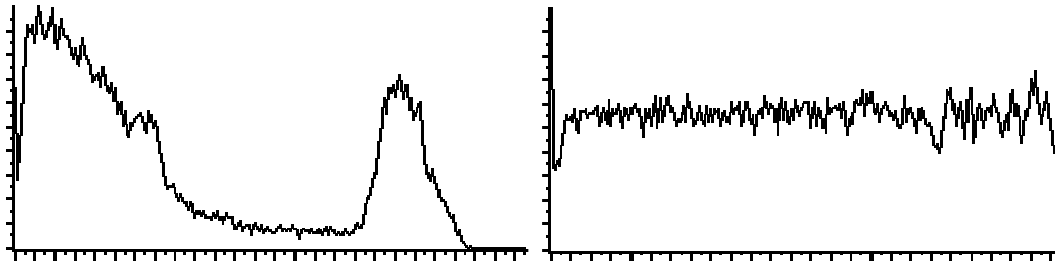


Fig.1.2.8. Egalisation d'histogramme [10]

#### 4.1.2. Etirement d'histogramme :

La normalisation d'histogramme est une transformation des niveaux de gris des pixels de telle sorte que l'image utilise toute la plage des niveaux de gris d'histogramme afin que la valeur de l'intensité la plus faible soit à zéro et que la plus haute soit à la valeur maximale, donc un histogramme bien réparti permet d'obtenir un bon contraste [10].

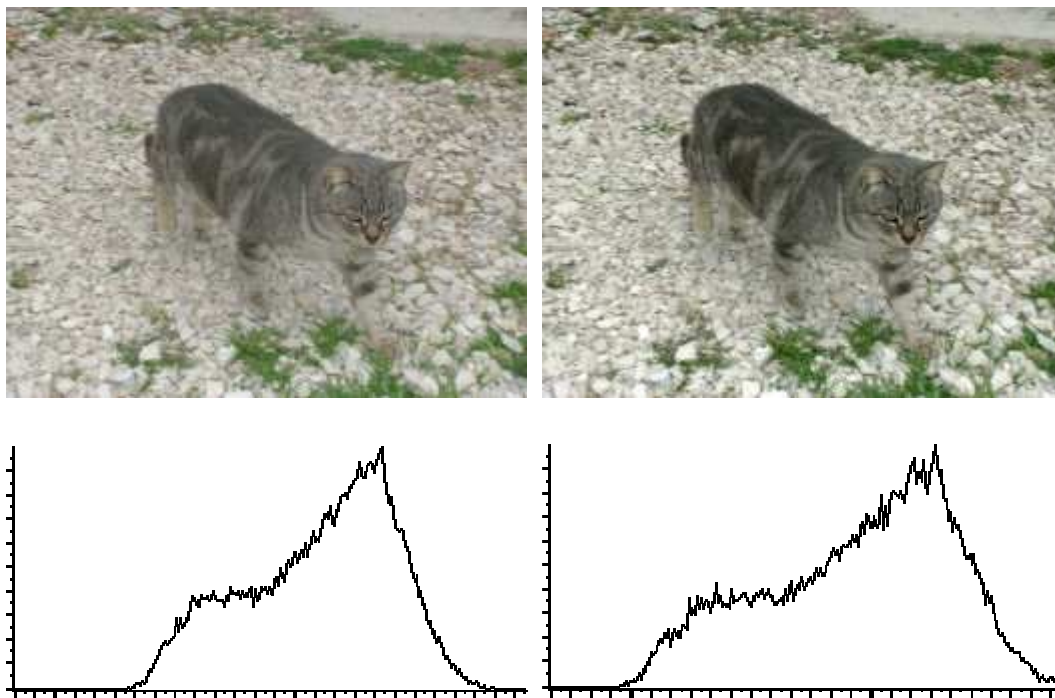


Fig.1.2.9. Etirement d'histogramme [10].

#### 4.2. Seuillage :

L'opération dite de "seuillage simple" consiste à mettre à zéro tous les pixels ayant un niveau de gris inférieur à une certaine valeur (appelée *seuil*, en anglais *threshol*d) et à la valeur maximale les pixels ayant une valeur supérieure. Ainsi le résultat du seuillage est une image binaire contenant des pixels noirs et blancs, c'est la raison pour laquelle le terme de

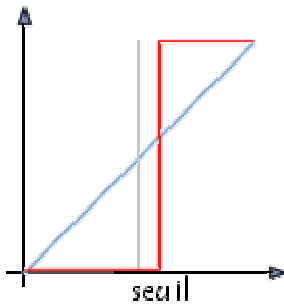
*binarisation* est parfois employé. Le seuillage permet de mettre en évidence des formes ou des objets dans une image. Toutefois la difficulté réside dans le choix du seuil à adopter [11].

Voici une image en 256 niveaux de gris et le résultat d'une opération de seuillage avec les valeurs respectives de seuil de 125 et 200 :

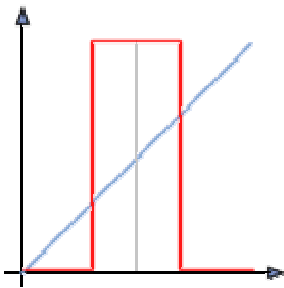


Fig.1.2.10. Seuillage [11]

La courbe tonale de l'opération de seuillage est la suivante [11] :



Par opposition au "seuillage simple" il est possible de définir deux valeurs de seuil, respectivement borne inférieure et borne supérieure, afin de mettre à la valeur maximale les pixels ayant une valeur comprise entre les bornes et à zéro l'ensemble des autres valeurs [11] :



#### 4.3. Binarisation :

La binarisation consiste à obtenir à partir d'une image codée sur plusieurs niveaux de gris (généralement 256), une image codée sur deux niveaux de gris (noir et blanc ou bien objet et fond). Elle a comme but de diminuer la quantité d'informations présentes dans l'image, et de garder que les informations pertinentes [8].

Le principe de la binarisation consiste à déterminer un seuil  $S$  appelé seuil de binarisation, puis modifier les valeurs des pixels de l'image comme suit :

Soit  $I$  une image et  $I(x, y)$  un point de coordonnées  $(x, y)$ .

$$I(x, y) = \begin{cases} 0 & \text{si } I(x, y) < S & \text{(Noir)} \\ 255 & \text{non} & \text{(Blanc)} \end{cases}$$

## 5. Les Filtres :

Afin d'améliorer la qualité visuelle de l'image, on doit éliminer les effets des bruits (parasites) en lui faisant subir un traitement appelé filtrage.

On distingue généralement les types de filtres suivants :

- Les filtres passe-bas (lissage) :

Ce type de filtrage consiste à atténuer les composantes de l'image ayant une fréquence haute (pixels fonces), il est utilisé généralement pour atténuer le bruit de l'image. C'est la raison pour la quelle on parle habituellement de lissage.

- Les filtres passe-haut (accentuation) : a l'inverse des passe-bas, ils atténuent les composantes des basses fréquences de l'image et permettent, notamment d'accentuer des contours et leur extraction. C'est la raison pour laquelle on parle habituellement d'accentuation.

## 6. Processus de traitement d'une image :

Un système de traitement numérique d'image est composé de plusieurs étapes à savoir l'acquisition, prétraitement, segmentation, reconnaissance et transcription et enfin stockage et transmission.

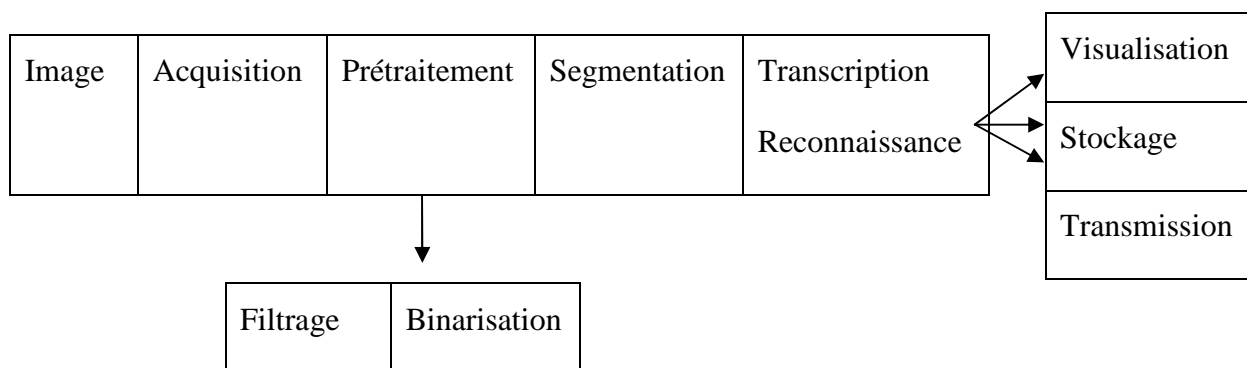


Fig.1.2.11. Schéma du processus de traitement numérique d'image.

## 7. Domaine d'application :

Le traitement d'images possède l'aspect multidisciplinaire. On trouve ses applications dans des domaines très variés tels que :

- ❖ Médecine (Radiographies, Scanners, ...)
- ❖ Biologie

- ❖ Météorologie
- ❖ Astronomie
- ❖ Géologie
- ❖ Physique (spectroscopie, physique des plasmas, ...)
- ❖ Applications militaires
- ❖ Applications industrielles (Robotique, surveillance de qualité, ...)
- ❖ Photographie
- ❖ Publicité.
- ❖ Autres ...

### **8. Conclusion :**

Dans ce chapitre nous avons présenté la majorité des concepts qu'on retrouve généralement dans le domaine du traitement d'image ce qui va permettre de mieux introduire le sujet pour les non initiés, ainsi qu'un aperçu général sur l'image et sur les différents traitements qui lui sont appliqués afin d'exploiter de façon optimale les informations qu'elle contient.

Dans ce qui suit nous allons nous focaliser sur les manuscrits arabes et l'écriture arabes afin de montrer les difficultés que nous pouvons retrouver dans un système de traitement d'images numérisées de manuscrits.

## **1. Introduction :**

Dans ce chapitre nous allons décrire les manuscrits arabes anciens ainsi que les différentes écritures arabes afin de voir la complexité et la difficulté de traitements de ces manuscrits.

## **2. Les manuscrits Arabe numérisés :**

### **2.1. Qu'est ce qu'un manuscrit ?**

Un manuscrit (du latin scriptus) est un texte écrit à la main [12]. Le manuscrit est le véritable témoin de la présence d'un texte à une époque donnée. Il véhicule les connaissances de cette ère. Il serait donc intéressant de pouvoir accéder au contenu de ces manuscrits. Pour ce faire, il faudrait mettre en place un système de descriptions adéquat des manuscrits.

Notons par ailleurs, qu'un manuscrit est tout d'abord une œuvre produite d'une manière artisanale, utilisant des matériaux rares et couteux. Il est possible de ce fait d'approché un document archéologique à travers ses caractéristiques matérielles et son histoire.

### **2.2. Caractéristiques des manuscrits arabes anciens :**

Les documents anciens sont d'une valeur inestimable, ils montrent souvent des dégradations plus ou moins importantes à cause des conditions de préservation et risquent ainsi de devenir inexploitable [13]. Les caractéristiques des manuscrits arabes peuvent être résumées comme suit [14] :

- Début: des manuscrits au verso du premier feuillet, alors que le recto est réservé à l'inscription du nom de l'auteur, au commanditaire de l'œuvre et parfois au cachet.
- Le début du texte peut être accompagné d'un décor particulier et représente souvent, le début de chaque chapitre, section ou sourate quand il s'agit d'une œuvre coranique.
- Le texte est écrit en longues lignes, à l'exception du texte poétique.
- Des règles d'usage de l'encre sont observées : la couleur rouge est souvent utilisée pour l'écriture des noms propres, des nombres et des citations.
- L'usage du texte encadré dans les manuscrits coraniques.
- Présences du texte dans les marges.



Fig.1.3.1.Manuscrit arabe ancien [15]

### 2.3. Alphabet de lettres Arabe

Cet alphabet repose sur trois constatations :

- la plupart des lettres arabes s'écrivent sous la forme radical + terminaison
- certaines lettres ne se différencient que par le nombre et/ou la position des points
- les ligatures verticales sont complexes à segmenter [16].

#### 2.3.1. Un radical et une terminaison

On a souvent coutume de dire que les lettres arabes peuvent prendre 4 formes différentes en fonction de leur position dans le mot (début, fin, milieu et isolée). C'est vrai pour les lettres

(ه ه ه ه) et (ع ع ع ع)

c'est faux dans le cas général.

Pour la plupart des lettres de l'alphabet arabe, les formes début et milieu sont identiques (à la ligature avec la lettre précédente près). Il en va de même pour les formes fin et isolée. De plus, les formes fin/isolée sont souvent construites à partir des formes début/milieu auxquelles on rajoute une "jambe" (voir tableau 2) [16].

#### 2.3.2. Des formes identiques aux points près

Certains groupes de lettres ne se différencient que par le nombre et/ou la position de leurs signes diacritiques (voir tableau 3). [16]

Jambe 1 : ل	Jambe 2 : ن	Jambe 3 : ح
ب ت ث → ل	ل → ل	ح → ح
و → و	س → س	ع → ع
	ص → ص	ع → ع
	ز → ز	ع → ع

Tableau.1.3.1.Lettres arabe : début/milieu vers isolée/fin [16]

{ د ذ } → د	{ ب ت ث ز ي } → ر	{ س ش } → س
{ ص ض } → ص	{ ر ز } → ر	{ ج ح خ } → ح
{ و ف } → و	{ ط ظ } → ط	{ ع غ } → ع
{ ع } → ع	{ ي } → ي	{ ه ة } → ه

Tableau.1.3.2. Quelques lettres arabes et leur forme sans diacritiques [16]

### 2.3.3. Les ligatures verticales

Les "ligatures verticales" sont des superpositions verticales de lettres (voir figure 3). Ces symboles peuvent être reconnus tels quels.



Fig.1.3.2. Ligatures verticales (sans les points).

### 3.Conclusion:

Dans ce chapitre nous avons présenté d'une manière générale les manuscrits arabes et la complexité de l'écriture arabes, ce qui représente un grand défi pour le travail que nous souhaitons réalisé.

# Partie2 : Etat de l'art

## 1. Introduction:

Le filtrage des images numérisées est une étape primordiale dans le processus de traitement d'image, car l'image à beaucoup de bruit qui peut influencer négativement sur les étapes ultérieures d'analyse d'images, telle que la segmentation.

Les manuscrits arabes anciens sont des documents historiques, écrits pendant ce que l'on appelle, généralement, l'âge d'or de la civilisation arabo-islamique. Ces derniers ; ne sont plus comme dans leur états d'origine, en effet, les documents sont dégradés, ou modifiés par différents facteurs externes parmi lesquels nous citons : l'humidité, la poussière, les taches...etc. Ainsi, l'objectif est de débarrasser l'image des manuscrits arabes numérisés de toutes ces impuretés ou bruit afin d'améliorer la qualité de l'image numérique.

## 2. Méthodes Appliquées:

Pour le filtrage de manuscrits arabes nous avons choisis d'utilisé la méthode du filtre moyennneur car le résultat qu'elle donne sur les manuscrits arabes anciens sont assez bons, d'abord elle permet un lissage de l'image et distingue mieux l'écriture par rapport au font mais comme toute opération de filtrage elle a des inconvénients elle rend l'image un peu flou.

### 2.1 Quelques Définitions :

#### 2.1.1. Le voisinage d'un pixel:

Le filtrage d'images numériques s'appuie fondamentalement sur le traitement à l'aide de voisinages d'un pixel. Cela signifie que le traitement effectué en un endroit donné correspondant à un pixel, dépend non seulement de ce pixel mais aussi, de pixels appartenant à son voisinage.

Nous illustrons ce concept par l'exemple suivant :

Considérons un pixel P dont la position dans l'image est donnée par les coordonnées (m, n).

Un voisinage de P, noté  $V(P)$ , se définit comme un ensemble de pixels P' connectés à P. la figure 2.1.1 illustre clairement ce concept.

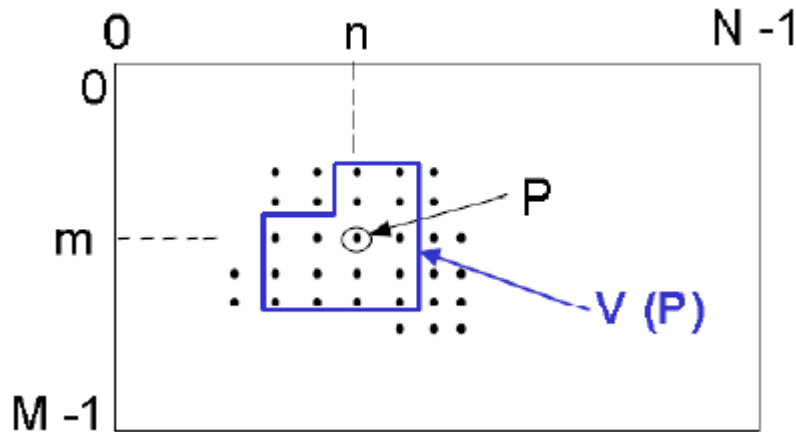


Fig.2.1.1.1. Représentation d'un pixel et son voisinage.

Le pixel P, cerclé dans la figure précédente, appartient à son propre voisinage  $V(P)$ .

Le voisinage d'un pixel peut être approché selon deux aspects : 4-voisinage ou 8-voisinage (défini précédemment partie1, chapitre2, paragraphe 2.2.1.2, page 6).

### 2.1.2. Le produit de convolution:

a) **Le produit de convolution noté  $L*M$  entre les deux matrices L et M se calcule comme suit :**

$$L*M = (P1*M1) + (P2*M2) + (P3*M3) + (P4*M4) + (P5*M5) + (P6*M6) + (P7*M7) + (P8*M8) + (P9*M9)$$

b) **Définition d'un produit de convolution [17]:**

Le produit de convolution de deux matrices  $A = (a_{ij})$  et  $B = (b_{ij})$  à  $n+1$  lignes et  $p+1$  colonnes, est donné par :

$$A*B = \sum_{i=0}^n \sum_{j=0}^p a_{ij} \times b_{(n-i)(p-j)}$$

### 2.1.3. Définition d'un filtre linéaire:

Le filtrage linéaire d'une image est déterminé par le remplacement de chaque pixel de cette image, par la convolution d'une matrice définie appelée masque ou noyau de convolution et le voisinage de ce pixel.

a) **Définition du masque de convolution:**

Le masque de convolution est une matrice carrée, généralement, de valeurs entières, caractérisant chaque filtre de convolution linéaire. La plupart des masques de convolution sont représentés par des matrices  $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$  et  $9 \times 9$ .

Le masque de convolution est encore appelé noyau du filtre.

Masque de convolution 3x3  $\begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & * & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix}$  Masque de convolution 5x5

·	·	·	·	·
·	·	·	·	·
·	·	*	·	·
·	·	·	·	·
·	·	·	·	·

Fig.2.1.2. Exemple de masques de convolution

Nous notons par ailleurs, que l'emplacement du pixel représenté par une étoile (\*) dans chaque masque de convolution constitue l'emplacement du pixel central.

### b) Le produit de convolution et les filtres linéaires:

Le principe est de construire à partir d'une première image  $I_e$  une seconde image  $I_s$  de même taille. Cela on remplaçant chaque pixel de l'image d'entrée  $I_e$ , par le produit de convolution d'un masque de convolution et le voisinage du pixel (de même taille que le masque).

Tous les filtres linéaires fonctionnent selon ce principe. Par ailleurs le noyau diffère d'un filtre à un autre.

Il existe principalement deux classe les filtres passe-bas et les filtre passe-haut, dans ce qui suit nous allons présenter les filtres passe-bas aux quels appartient le filtre moyennneur.

### 2.2.Les filtres passe-bas:

Les filtres passe-bas ont pour effet de lisser l'image et de limiter les variations brusques d'intensité. On les utilise typiquement pour réduire les effets de bruit.

Par conventions tous les noyaux de filtre passe bas doivent respecter les deux caractéristique suivante :

Les noyaux de filtre passe-bas doivent respecter les deux caractéristiques suivantes :

Soit  $h(i, j)$  le noyau d'un filtre passe-bas :

- La somme de tous les éléments du noyau =1,  $\sum h(i, j) =1$ , ce qu'on appel aussi l'opération de normalisation [18].
- Tous les éléments de  $h$  sont positifs ou nul,  $\forall i, j : h(i, j) \geq 0$  [19]

#### 2.2.1. Le filtre moyennneur:

Le filtre moyennneur appartient à la famille des filtres linéaires, il consiste à remplacer le niveau de gris de chaque pixel par la moyenne arithmétique de ce dernier.

Autrement dit, cela revient à remplacer chaque pixel par le produit de convolution du noyau  $h$  et le voisinage de ce pixel de taille  $n \times n$ , tel que  $h$  est défini :

$$\forall i, j: h(i, j) = \frac{1}{n^2}$$

- **Exemple :**

Soit le masque  $3 \times 3$  défini par la matrice  $h_1$

$$h_1 = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

**3. Conclusion:**

L'objectif de ce chapitre consiste à développer des méthodes informatiques permettant de débarrasser, les images des manuscrits arabes numérisés, de toute impureté ou bruit et d'améliorer la qualité de l'image numérique. Ces derniers ont été causés par les différents facteurs externes par exemple, l'humidité, la poussière ...etc.

Dans ce chapitre nous avons proposé, des méthodes de filtrage d'image numérique, qui permettent la suppression des différents bruits. Ces méthodes permettent de modifier les pixels de l'image selon leurs voisinages, afin de pouvoir éliminer les bruits.

Les images obtenues après les différents filtrages pour l'élimination des bruits, peuvent à présent, être utilisées dans la suite des autres étapes de binarisation et segmentation.

## 1. Introduction :

La binarisation est une technique très utilisée en traitement d'image. Elle consiste à partitionner l'image en couleur ou en niveau de gris, en régions homogènes correspondant aux objets et au fond. Elle a comme but de diminuer la quantité d'informations présentes dans l'image et de ne garder que les informations pertinentes. Cet aspect nous permet d'utiliser des méthodes d'analyse simples vis-à-vis des images.

Les performances des étapes suivantes dans le système d'analyse de documents, dépendent fortement des résultats de l'algorithme de binarisation utilisé, c'est pour cette raison que l'étape de binarisation doit être effectuée le plus fidèlement possible. Elle doit d'une part, conserver le maximum d'informations et détails présents dans l'image en entrée, et d'autre part, elle doit éliminer le bruit présent dans l'image.

Le problème de binarisation de documents est très ancien, un très grand nombre de techniques ont été proposées dans la littérature dont chacune d'entre elles est appropriée à un type d'images. En effet, plusieurs techniques ont été proposées ces deux dernières décennies pour la binarisation de documents en niveau de gris, mais malheureusement, aucune d'entre elles ne se montre puissante pour la binarisation des documents anciens dégradés, caractérisés par une structure complexe, à savoir : une variété d'illumination, la présence de tâches et de trous, les effets de transparence... etc.

Pour la binarisation de manuscrits arabes anciens nous avons choisi d'utiliser une méthode globale qui est la méthode du seuil fixe car elle donne des résultats satisfaisants sur ce type de documents.

## 2. Les méthodes de binarisation :

Les techniques de binarisation d'images peuvent être classées en deux catégories : le seuillage global, où un seul seuil est utilisé pour toute l'image et le seuillage local, où les valeurs de seuils sont déterminées localement, pixel par pixel ou bien région par région (cette dernière ne sera pas utilisée dans notre travail).

Avant de commencer toute opération de binarisation, il est impératif de convertir l'image couleur en niveau de gris. Cette opération aura pour effet d'affecter à chaque pixel de l'image une seule valeur comprise entre 0 et 255.

## 2.1. Méthodes Globales et Méthodes Locales :

Les méthodes globales consistent à calculer un seul seuil pour toute l'image. Ces méthodes reposent sur l'exploitation de l'histogramme de toute l'image qui caractérise la distribution de niveau du gris. Les niveaux du gris de tous les pixels de l'image sont comparés au seuil déjà calculé, puis nous affectons à chacun d'eux une valeur selon la position de son niveau du gris par rapport au seuil.

Dans les méthodes locales les valeurs des seuils sont déterminées localement, pixel par pixel où bien région par région. Le seuil est calculé pour chaque pixel de l'image, les résultats d'application de ces méthodes ont montré qu'elles donnent beaucoup de détails, donc présentent des fonds fortement bruités, parfois même cache complètement les écritures.

C'est pour cette cause que notre choix de la méthode de binarisation c'est portée sur une méthode globale la méthode du seuil fixe.

### 2.1.1. Seuillage fixe :

Dans cette méthode le calcul de seuil  $T$  se fait manuellement par l'utilisateur après consultation de l'histogramme des niveaux de gris. Cette méthode de seuillage n'est pas sans risque d'erreur, hormis dans le cas d'un histogramme bimodal idéal, où les deux modes traduisent parfaitement les deux classe de l'image. Dans ce cas, le meilleur seuil  $T$  est le milieu de la vallée, située entre les deux modes, le niveau de gris de chaque pixel est ensuite, comparé au seuil  $T$  (exemple = 127).

Soit  $I$  une image et  $I(x, y)$  un point de l'ide coordonnées  $(x, y)$

$$\begin{cases} 0 & \text{si } I(x, y) < T & (\text{noir}) \\ 255 & \text{sinon} & (\text{blan}) \end{cases}$$

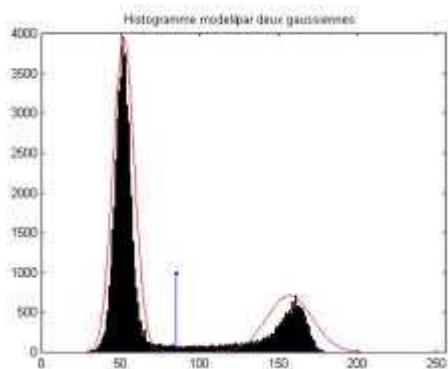


Fig.2.2.1. Le choix du seuil dans histogramme bimodal

Dans la pratique il est rare de trouver un histogramme qui présente deux modes distincts.

En effet, la plupart des images présentent des histogrammes avec des formes aléatoires, d'où la difficulté de choisir le seuil manuellement. C'est la raison pour laquelle le seuil automatique est utilisé.

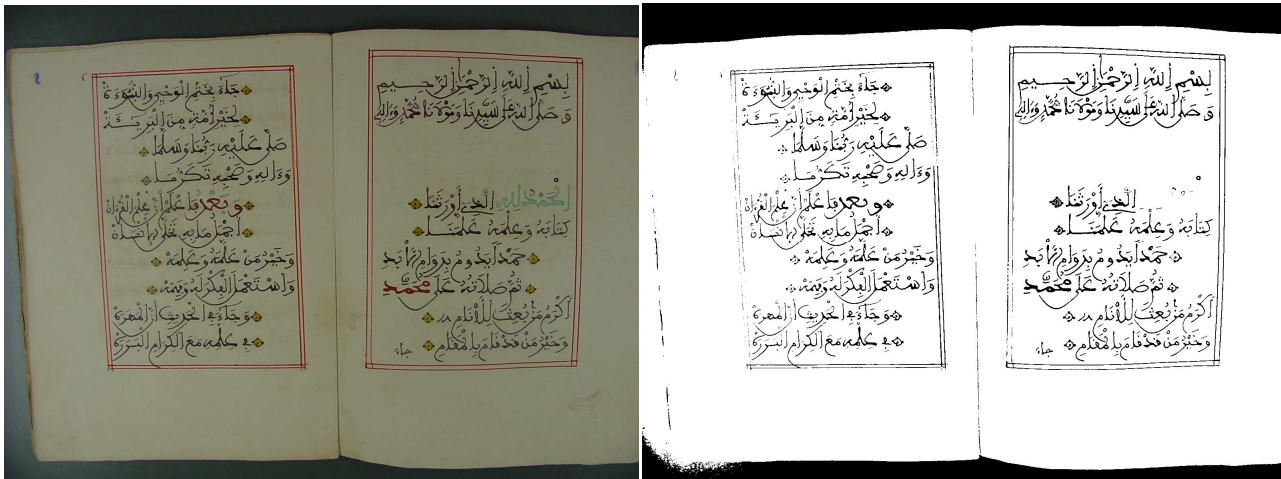


Fig.2.2.2. Image binarisée avec un seuil 127[20]

Dans les méthodes globales nous appliquons un seul seuil pour toute l'image.

### 3. Conclusion :

Dans ce chapitre nous avons présenté une méthode globale de binarisation qui est la méthode du seuil fixe, des exemples ont été donnés afin de montrer les résultats de cette méthode sur les manuscrits arabes numérisés. En se basant sur l'appréciation visuelle, nous constatons que la méthode du seuil fixe donne des résultats appréciables, en effet elle présente une bonne séparation entre les objets et leurs fonds.

## 1. Introduction :

La segmentation est une partie primordiale dans le traitement d'image. A ce jour il existe de nombreuses méthodes de segmentation, en se limitant a la référence que fait la segmentation aux notions de dissimilarités et de similarité perçues par le système visuel humain nous ramène a regrouper ces différentes approches de segmentation en deux grandes approches : *approches contours* et *approches régions*, nous pouvant également dire qu'il ya l'approche coopérative c-à-d utilisation des deux méthodes précédemment citées simultanément soit séquentiellement soit de manière hybride, et non coopératives c-à-d utilisation de chacune des méthodes a part.

Dans notre cas de figure nous allons utiliser une approche coopérative séquentielle c-à-d d'abord une méthode de détection de contours sera appliquée sur l'image puis suivra la méthode d'étiquetage des région connexes en suites en se servira du résultats de cette dernière pour le filtrage des signes diacritique puis il y'aura détection de la bande de base grâce a l'histogramme de projection horizontale de chaque composante connexe, ensuite application de l'algorithme de Hilditch pour extraire le squelette qui permettra une détection plus fine de la bande de base pour chaque ligne d'écriture puis segmentation en graphème et enfin segmentation en mots.

## 2. Définition :

La segmentation a pour but de trouver des régions homogènes et contours de l'image. Les régions doivent correspondre le mieux aux parties significatives des objets du monde réel, et leurs contours apparents. Les pixels sont regroupés en régions  $R_i$  qui constituent une partition ou pavage de l'image  $I$  tel que :

- |    |                             |                    |
|----|-----------------------------|--------------------|
| 1. | $\cup R_i = I;$             | $\forall i = 1..n$ |
| 2. | $R_i \neq \emptyset;$       | $\forall i = 1..n$ |
| 3. | $R_i \cap R_j = \emptyset;$ | $\forall i, j$     |

1. La première condition implique que tous les pixels de l'image appartiennent à une et une seule région (segmentation complète). Cela signifie que l'algorithme de segmentation ne doit pas terminer avant d'avoir traité tous les points.
2. La seconde implique que toute région doit être connexe (tous les pixels appartenant a la même région doivent être connectés).
3. La troisième condition implique que toutes les régions doivent être disjointes

L'approche région citée plus haut s'attache à faire des régions homogènes, alors que l'approche contours tente de trouver des contours ou frontières de régions. Les deux approches sont complémentaires et aucune d'elle n'a prouvé sa priorité par rapport à l'autre. Chacune ayant ses avantages, ses inconvénients et ses domaines d'applications.

Pour les autres applications de segmentation qui vont suivre (filtrage des signes diacritiques, détection de la bande de base, squelettisation, segmentation en graphèmes et segmentation en mots), elles sont généralement utilisées dans le cadre d'un système de reconnaissance ou pour l'aide à la transcription et c'est le cas de notre travail.

### **3. Méthode de Canny et Etiquetages des composantes connexes :**

#### **3.1. Méthode de Canny :**

##### **a) L'approche contour :**

Un contour est en général une bordure qui sépare des zones d'images adjacentes ayant des luminances (ou des couleurs) distinctes. Les techniques de détection de contours analysent souvent une image dans sa globalité sans tenir compte des spécificités des différentes régions de l'image.

L'objectif de l'opération est de transformer l'image en une autre dans laquelle les contours apparaissent par convention en blanc sur un fond noir. Les contours dans une image sont caractérisés par des discontinuités de la fonction d'intensité dans l'image. Le principe de la détection de contours repose sur l'étude des dérivées de la fonction d'intensité dans l'image.

Il existe un grand nombre de méthodes de détection de contours mais la plupart d'entre elles peuvent être regroupés en deux catégories :

- La première cherche les extremums de la dérivée première, en général les maximums locaux de l'intensité de gradient.
- La seconde cherche les annulations de la dérivée seconde, en général les annulations du Laplacien [21].
- **Norme du gradient :** Le gradient permet de détecter la présence des contours, ces derniers sont indiqués par des maximums de ce gradient, dans le cadre d'une image le gradient est bidimensionnel. Il peut être calculé par la formule suivante [21] :

$$\|\nabla I(x, y)\| = \sqrt{\frac{\partial I^2}{\partial x} + \frac{\partial I^2}{\partial y}}$$

- **Opérateur Laplacien** : C'est l'opérateur dérivatif de deuxième ordre, les points de contours représentant un maximum local dans le cas du gradient, correspondent à des passages par zéro pour la seconde dérivée [22]. Avec la méthode du Laplacien, les contours peuvent être détectés et localisés par le passage par zéro du Laplacien. Le Laplacien d'une image d'intensité est défini par :

$$\Delta I(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad I(x, y) \text{ l'image à traiter}$$

- **Le filtre de Gaussienne** : Coefficients calculés à l'aide d'une fonction gaussienne [23] :

$$G(x) = \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

Filtre séparable : gaussienne 2D = produit de deux Gaussiennes 1D

- **Filtres bidimensionnels** : La convolution de filtres rectangulaires (resp. triangulaires) selon les deux axes de l'image conduit à un filtre uniforme (resp. pyramidal) représenté par une matrice carrée, les deux directions étant d'une certaine manière privilégiées.

Au contraire le filtre de Gauss à deux variables isotropes. Il est utilisé en particulier dans le filtre de Canny [24], comme on va le présenter dans le paragraphe suivant.



Fig.2.3.1 : Contour

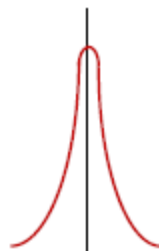


Fig.2.3.2 : Dérivée Première



Fig.2.3.3 : Dérivée seconde

### b) Opérateur de Canny : Opérateur dérivatif de deuxième ordre :

Canny a proposé en 1983 une étude théorique sur la détection de contour, il a conçu son algorithme dont le but d'optimiser les trois critères suivants :

- ✓ Bonne détection: faible taux d'erreur dans la signalisation des contours,
- ✓ Bonne localisation : minimisation des distances entre les contours détectés et les contours réels.
- ✓ Clarté de la réponse : une seule réponse par contour et de faux positifs.

Canny propose de créer un filtre dans la direction orthogonale au contour que l'on cherche à détecter et à un filtre passe-bas dans la direction du contour, afin d'assurer la continuité de celui-ci. Pour simplifier les calculs, il choisit pour ce filtre passe-bas une gaussienne de même largeur que la dérivée de gaussienne utilisée pour détecter les contours. Cette opération est effectuée grâce à une convolution de l'image par une gaussienne en 2 dimensions.

$$G(x, y) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

$$dG(x) \otimes I(x) = (G(x) \otimes I(x))'$$

Dans la pratique, il n'est pas nécessaire d'effectuer cette dérivation dans toutes les directions de l'image pour détecter les contours dans chacune d'entre elles : il suffit de l'appliquer dans deux directions orthogonales.

Cela se résume donc à appliquer un filtre gaussien bidimensionnel à l'image puis dériver le résultat dans deux directions orthogonales. Le maxima de ces deux nouvelles images correspondent aux contours dans deux directions orthogonales.

Le filtre de Canny est apprécié pour sa simplicité et sa rapidité d'exécution. Il permet d'éliminer les 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. 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. Par la suite on applique le seuil bas, en ne conservant que les composantes connexes qui contiennent un point au dessus de  $S_b$ , on suit un chemin constitué de points-au-dessus de  $S_b$ , ce chemin est le contour recherché.

### 3.2. Méthodes d'étiquetage des composantes connexes:

#### 3.2.1. L'approche région :

Cette approche consiste en la division de l'image en régions distinctes en cherchant plutôt les similarités. Ces régions sont composées de l'ensemble des pixels connexes possédant les mêmes propriétés au sens d'un prédicat d'homogénéité donné.

Dans l'approche région, c'est la similitude des points connexes qui est favorisée. Les points connexes ayant des propriétés (attributs) similaires : intensité du niveau de gris, texture, couleur, vont être réunis dans le même ensemble [21]. Le problème fondamental associé à cette approche est lié à la définition du critère d'homogénéité dans le processus de segmentation de l'image.

Ces approches fournissent une carte de région fermée. Cependant, la localisation de frontière reste généralement peu précise. Parmi les approches régions on distingue les approches par fusion, par division, par division-fusion.

#### ■ Les approches par division (split) :

La méthode couramment utilisée consiste à faire une dichotomie par blocs de l'image. Pour cela, on commence par définir un bloc de la taille de l'image, puis on examine le contenu de ce bloc. Si le bloc est homogène (le bloc contient exclusivement des pixels similaires) alors la décomposition est arrêtée, sinon, le bloc est découpé en 4 sous-blocs. Par la suite, le contenu de chaque sous-bloc est examiné [25]. L'opération est répétée pour chaque bloc jusqu'à l'homogénéité de tous les blocs ainsi découpés.

Le résultat obtenu est donc un ensemble jointif de blocs de différentes tailles qui recouvrent entièrement l'image.

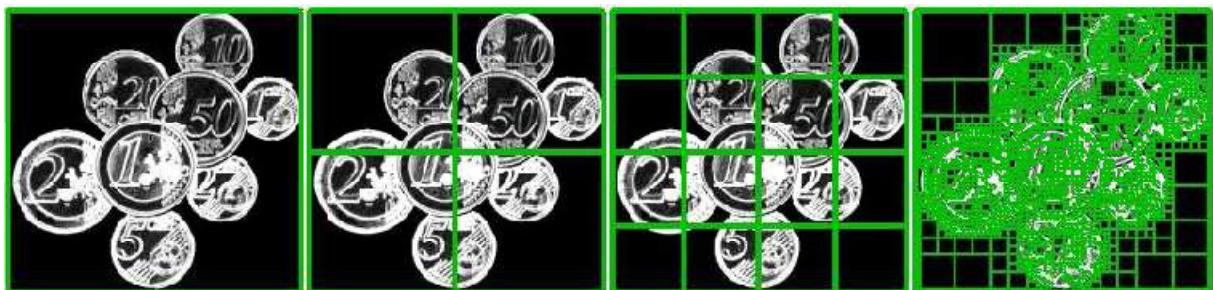


Fig.2.3.4. Décompositions successives des blocs [25]

L'implémentation la plus simple pour cette méthode consiste à définir une structure d'arbre appelée QuadTree. C'est un arbre dans lequel chaque nœud représente un bloc. Chaque nœud possède donc 0 sous-nœud (bloc homogène) ou 4 sous-nœuds (bloc homogène).

La décomposition finale est définie par les blocs associés aux feuilles de l'arbre. On obtient ainsi une décomposition finale par les blocs associés aux feuilles de l'arbre. On obtient ainsi une liste de blocs de différentes tailles et positions. Si la structure du QuadTree permet une navigation aisée entre bloc conteneur (parent) et sous-blocs (enfants), elle ne permet pas de naviguer facilement entre des blocs voisins. Pour cela, il est préférable de construire et d'utiliser le graphe d'adjacence [25].

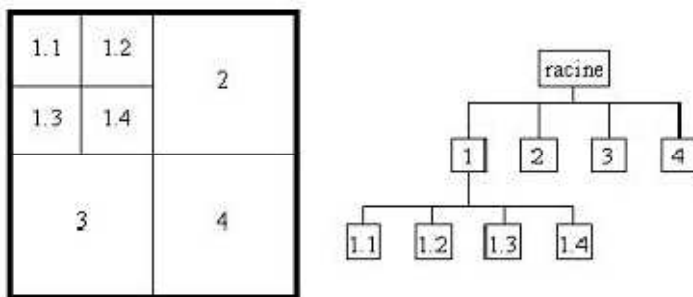


Fig.2.3.5. Représentation sous forme QuadTree [25].

#### ■ Les approches par fusion (croissance de régions) :

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 [25]. Les pixels voisins sont ajoutés à la région si l'indicateur d'homogénéité reste vrai. La croissance s'arrête lorsqu'on ne peut plus ajouter de pixels sans briser l'homogénéité.

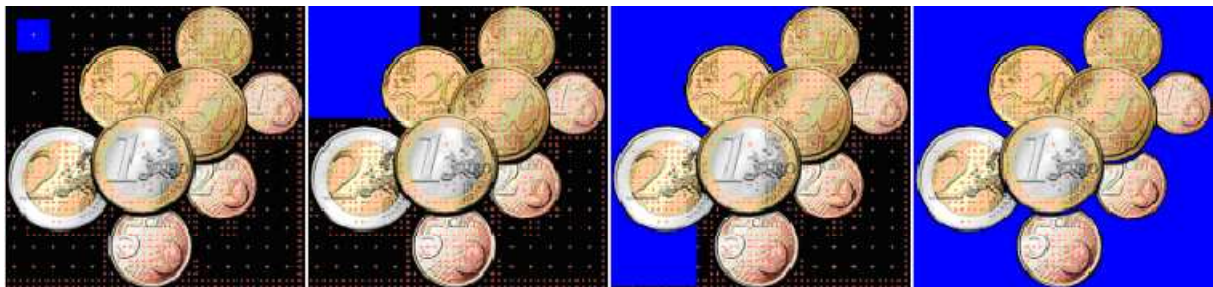


Fig.2.3.6. Croissance progressive d'une région [25]

### ■ Les approches par division-fusion :

Cette méthode combine deux méthodes. La division de l'image en des petites régions homogènes, puis la fusion des régions connexes et similaires.

Ces méthodes font appel à la théorie des graphes. Ainsi, elles peuvent être classées selon la structure du graphe utilisé. Plusieurs variantes de cette approche existent dans la littérature, on trouve par exemple les approches pyramidales, ou le processus de segmentation utilise la structure pyramidale du QuadTree.

### 3.2.2. Classification :

Parmi les techniques de segmentation d'images, la classification (étiquetage des régions connexes) est l'une des procédures les plus utilisées. La classification d'une image consiste à partitionner cette image en un ensemble de K classes disjointes. A chaque point de l'image on attribue une étiquette parmi les K étiquettes qui correspondent chacune à une classe [26]

a) **Principe** : Une première passe consiste :

- L'image binarisée est balayée horizontalement de la gauche vers la droite, du haut vers le bas.
- Un masque se déplace sur l'image des contours et s'applique à tous les points (contours ou non), son but est l'attribution et la gestion des étiquettes.

Lors de la seconde passe les étiquettes sont mise à jour à l'aide du tableau de correspondance des étiquettes : lorsqu'une entrée dans la table de correspondance existe l'étiquette est par la valeur dans la table.

1	1	1	0	1	1	0	1
1	1	0	1	1	0	1	1
1	0	0	0	0	1	1	1

Image d'origine : image binaire

1	1	1	0	2	2	0	3
1	1	0	4	2	0	5	3
1	0	6	4	0	7	5	3

Image des étiquettes

1	1	1	0	1	1	0	1
1	1	0	1	1	0	2	2
0	0	1	1	0	2	2	2

Image des étiquettes après mises à jour

L'étiquetage en composantes connexes (Composante connexe = ensemble de pixels connexes (voisins) appartenant à une même entité, une image segmentée n'est qu'une succession de pixels) est la base de la majorité des chaînes algorithmiques en traitement d'image dues à l'analyse des régions dans une scène.

Ceci crée d'abord l'image d'étiquettes, à partir de l'image originale (on va effectuer un parcours de l'image pour affecter un numéro unique (étiquette) pour chaque région, tous les pixels d'une même région doivent avoir le même numéro (étiquette)), on utilisant l'approche pixel. Cette dernière consiste à faire une comparaison de l'étiquette courante avec l'étiquette voisine et rechercher les segments adjacents pour déterminer la connectivité des segments.

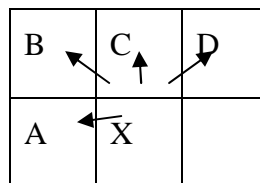


Fig.2.3.7. Les Pixels voisins

Après avoir créé l'image d'étiquettes, celle-ci sera parcourue en largeur afin de rechercher les coordonnées min et max de chaque composante connexe dans le but de la mettre dans un rectangle et renvoyer l'image comme résultat finale.

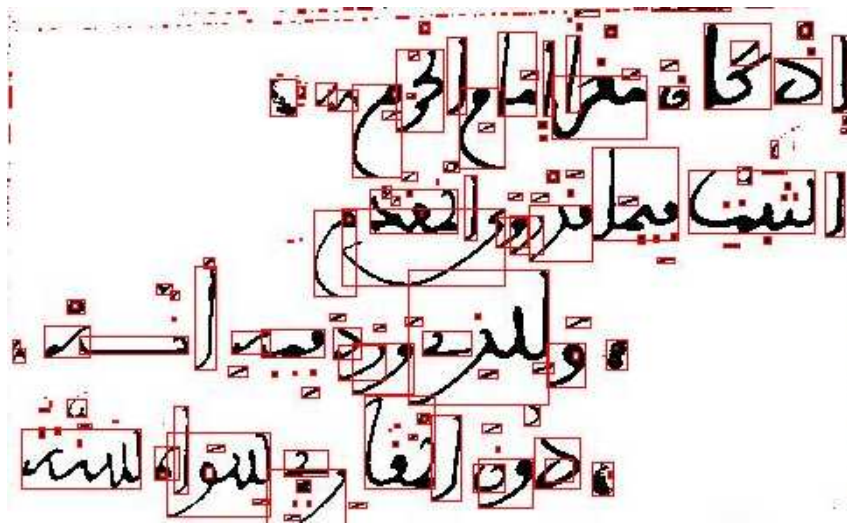


Fig.2.3.8.Exemple d'image avec rectangles englobant.

#### **4. Filtrages des composantes connexes correspondant à des signes diacritiques :**

##### **4.1. Détection des signes diacritiques :**

Pour la détection de signes diacritiques jusqu'à la segmentation en graphèmes nous nous sommes basés sur les recherches effectuées dans ce domaine par M. Chriet.

La détection des signes diacritiques a deux utilités :

- D'une part, un trop grand nombre de signes diacritiques risque de perturber l'histogramme de projection horizontale, qui sert à évaluer la bande de base.
- Et d'autre part, les Signes diacritiques ainsi détectés pourront par la suite être utilisés pour améliorer les résultats de la segmentation.

La détection des signes diacritiques se fait en deux étapes.

La première étape consiste à effectuer un filtrage des composantes connexes en s'appuyant sur des critères assez simples : taille des boîtes englobantes, aire, superposition verticale.

L'objectif est de rejeter la plupart des signes diacritiques, sans rejeter de composante connexe correspondant à un corps de lettre ou à un pseudo-mot. La figure 14 décrit cette procédure de filtrage. Les seuils sont fixés empiriquement [16].

## 4.2.Algorithme :

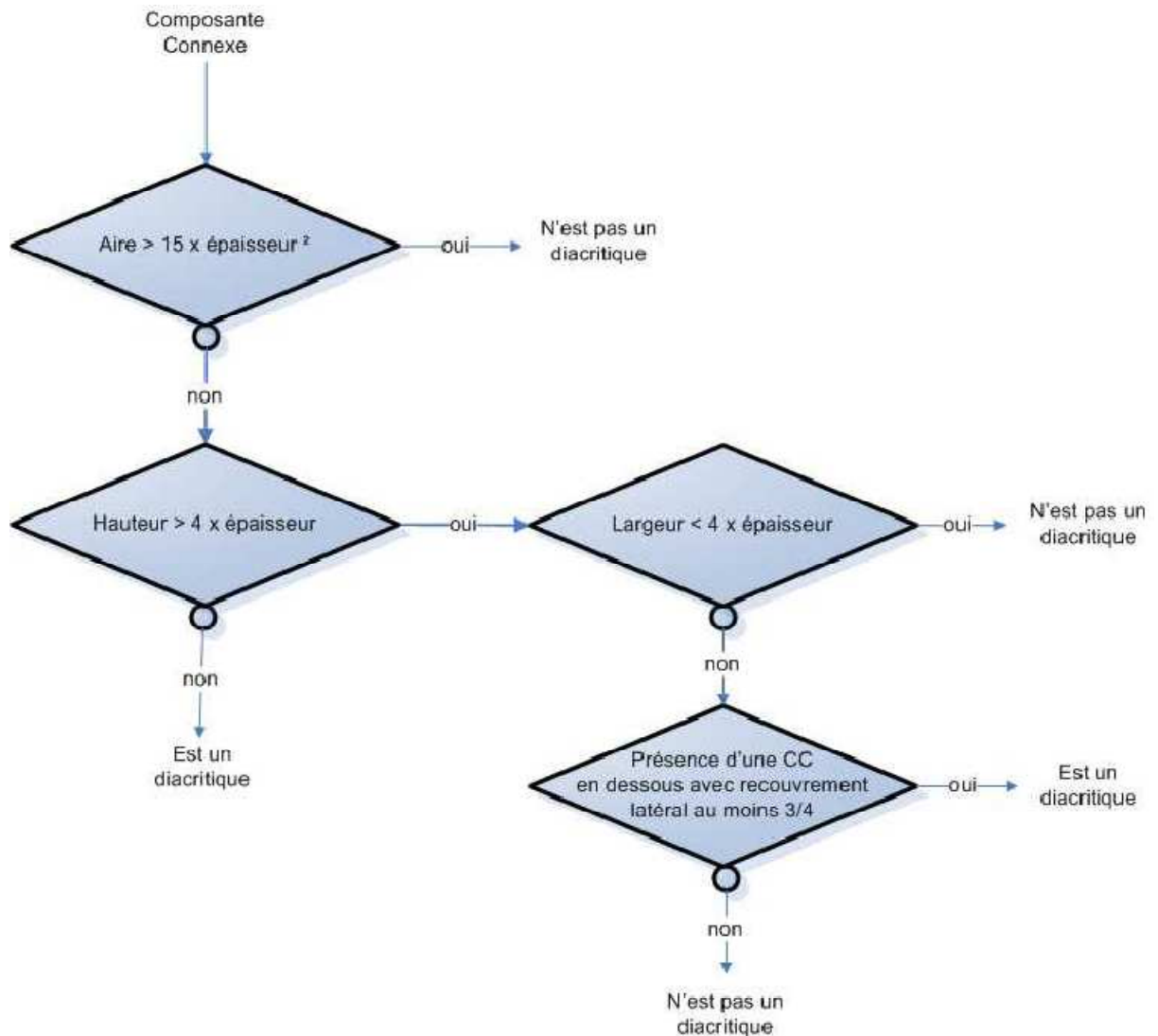


Fig.2.3.9. Procédure de filtrage des signes diacritiques : des seuils heuristiques définis à partir de l'épaisseur du tracé permettent d'effectuer un premier filtrage grossier [16].

- Le premier test se base sur l'aire : les composantes connexes trop grosses ne peuvent pas être des signes diacritiques.
- Les deuxième et troisième tests permettent de conserver les barres d'alif (Ī), qui contiennent peu de pixels et sont étendues verticalement.
- Le 4ème test permet de conserver les caractères ▸ isolés, qui sont parfois facilement assimilables à des signes diacritiques par leur forme. Cette règle repose sur la position relative de la composante connexe et de ses voisines : si une composante connexe C1

d'aire réduite est située au-dessus d'une autre composante connexe C2, et que C2 recouvre verticalement C1 à plus de 75%, alors C1 est un signe diacritique. Ce premier filtrage permet de détecter et d'écartier un certain nombre de composantes connexes. On peut alors réaliser une approximation de la bande de base sur l'image ainsi filtrée. Cette estimation de la bande de base permet d'effectuer un second filtrage plus précis. Ce second filtrage prend en compte le fait que les signes diacritiques se situent soit en dessous, soit au-dessus de la bande de base. Par exemple, dans la figure 2.3.10, une chadda n'a pas été filtrée lors de la première étape. Elle n'est pas supprimée car son recouvrement vertical n'est que partiel (inférieur au seuil fixé empiriquement) avec la lettre  $\text{ح}$  à laquelle elle est associée. Cette composante connexe a donc été conservée lors de la première étape. Mais après l'évaluation de la bande de base, il apparaît clairement qu'il s'agit bien d'un signe diacritique, puisque cette composante connexe est entièrement au-dessus de la bande de base. On peut donc l'ajouter à la liste des signes diacritiques détectés lors de la première étape [16].

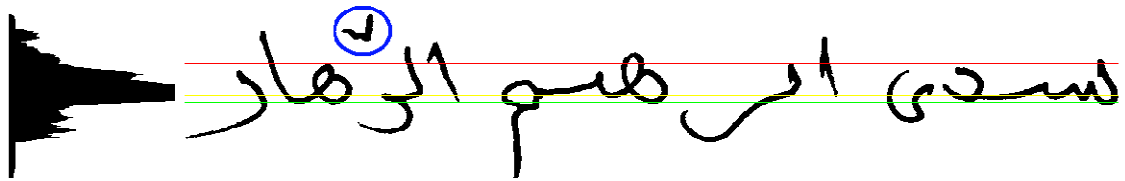


Fig.2.3.10. Pic maximal de l'histogramme (ligne jaune) et seuils pour déterminer une approximation de la ligne haute et de la ligne basse. On se sert de cette approximation de la bande de base pour retirer les signes diacritiques résiduels (ici une chadda entourée en bleu) [16].

Dans certains cas, il arrive que les signes diacritiques ne forment pas des composantes connexes séparées du corps du texte. Un exemple est donné figure 2.3.11. Ces cas nécessitent la mise en œuvre de mécanismes plus complexes, qui permettraient de générer plusieurs alternatives. Ce type de problème, relativement marginal, n'est pas traité ici [16].



(a) Image initiale (b) Image obtenue après application de l'algorithme de suppression des signes diacritiques.



trois fois la hauteur maximale des boucles, centrée sur les boucles. La recherche du maximum dans cette zone permet d'éviter le pic du bas, qui est dû à une succession de descendants [16].

## 5.2. Histogramme de projection horizontale :

### a) Principe:

A partir du maximum de l'histogramme, on applique un seuil haut et un seuil bas, qui permettent de déterminer une approximation de la bande de base (lignes rouge et verte de la figure 2.3.10).

Empiriquement, ces seuils sont fixés à :

- Seuil haut = 20% de la valeur du pic principal
- Seuil bas = 45% de la valeur du pic principal

Cette approximation de la bande de base permet d'effectuer un deuxième filtrage des composantes connexes pour achever la détection des signes diacritiques résiduels (deuxième étape de filtrage des signes diacritiques, décrite précédemment, un exemple est donné dans la figure 2.3.10). La hauteur de l'écriture (qui sera la hauteur de la bande de base précise) est supposée constante sur l'ensemble de la ligne. Elle est déterminée par la moyenne des distances verticales entre les extremums locaux du contour situés dans cette première estimation de la bande de base (entre les lignes rouge et verte de la figure 2.3.10) [16].

La technique des histogrammes de projections est une méthode de caractérisation globale d'une forme qui renseigne sur son épaisseur dans plusieurs directions. Chaque histogramme est calculé en comptant le nombre de pixels de la forme dans une direction  $\delta$ :  $HP(\delta) = \sum FI_{\delta}(x, y)$ . Cela revient à *projeter* les pixels de la forme dans une direction et à regarder les variations de la distribution marginale.

## 6. Squelettisation :

La *squelettisation* est une classe d'algorithmes utilisée en analyse de formes. Elle consiste à réduire une forme en un ensemble de courbes, appelées *squelettes*, centrées dans la forme d'origine. La squelettisation est un outil d'analyse de forme non-scalaire, qui conserve les propriétés topologiques de la forme d'origine ainsi que les propriétés géométriques, selon la méthode employée.

En termes simples, la squelettisation consiste à *amaigrir* une forme jusqu'à obtenir un ensemble de courbes centrées. L'ensemble obtenu est alors appelé *squelette* ou *axe médian* (*medial axis*, en anglais) [28].

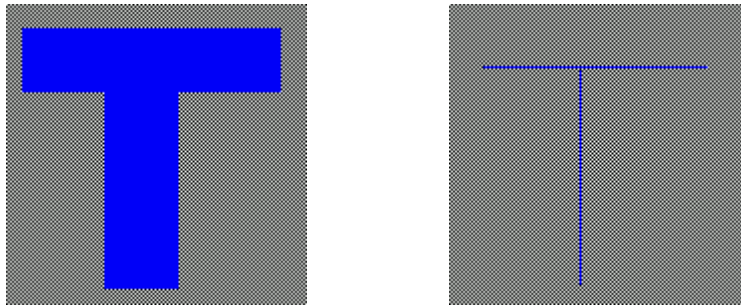


Fig.2.3.13.Motif en forme de T

Le squelette [28]

### 6.1.Définitions :

Il existe différentes définitions de la squelettisation :

La squelettisation est une opération qui permet de passer d'une image à sa représentation en "fil de fer". Le squelette à un pixel d'épaisseur. C'est une manière de représenter l'information indépendamment de l'épaisseur initiale de l'écriture.

Il permet d'extraire des caractéristiques importantes, comme les intersections et le nombre de tracés, leurs positions relatives. Il est également possible de renormaliser l'épaisseur de l'écriture à partir du squelette. Il n'existe pas de définition unique du squelette. Le squelette doit seulement remplir trois conditions :

- Il doit être aussi fin que possible (typiquement, 1 pixel d'épaisseur)
- Il doit respecter la connexité
- Il doit être centré dans la forme qu'il représente

A chaque composante connexe de la forme, le squelette correspondant ne doit être composé que d'une seule composante connexe d'un pixel d'épaisseur incluse dans la première (voir figure 2.3.14). Il existe de nombreuses méthodes de squelettisation. L'une des manières d'évaluer le squelette est de calculer l'axe médian de la forme, comme l'ensemble des centres des boules maximales de cette forme (voir figure 2.3.15) [28].

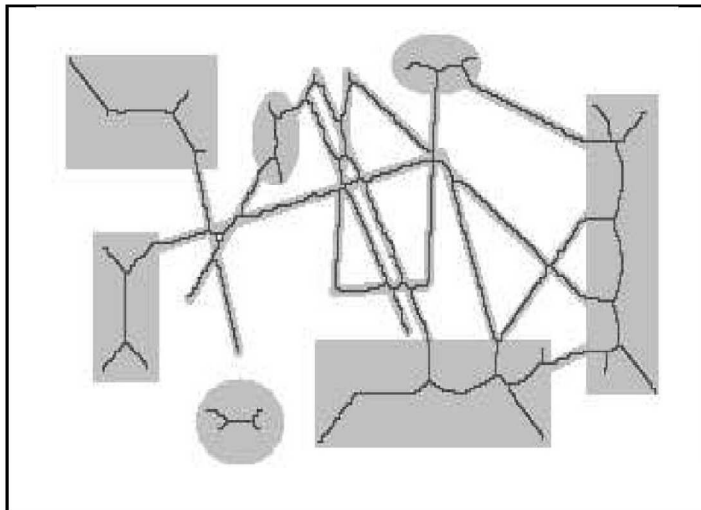


Fig.2.3.14. Squelettisation d'une forme quelconque [28]

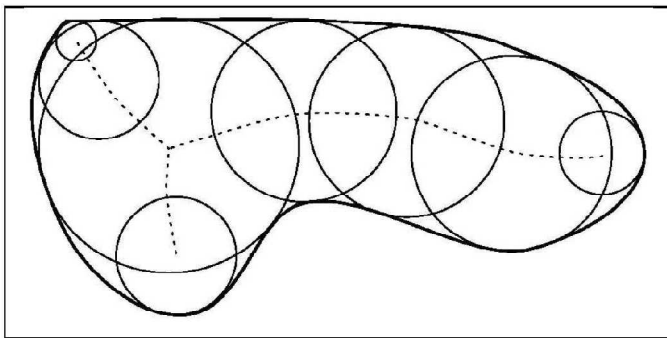


Fig.2.3.15. Axe médian d'une forme continue (trait pointillé) dont seul le contour est représenté [28]

Un autre moyen d'obtenir le squelette est de procéder par affinages successifs : la forme est "épluchée" (peeling) de manière itérative, en maintenant valide le critère de connexité, l'algorithme de Hilditch s'inspire de ce principe.

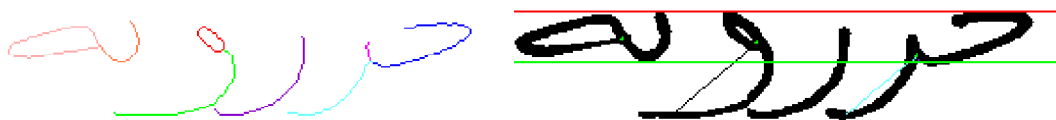
Dans notre cas de figure le squelette est extrait à l'aide d'une version modifiée de l'algorithme de Hilditch, utilisée pour l'écriture latine. Une image du squelette est donnée figure 2.3.16(a).

Les points singuliers du squelette sont les points dont le voisinage est supérieur à 2. Ce sont les points d'embranchement, points de croisement ou boucles. Ils sont marqués en vert dans la figure 2.3.16(b). Certains de ces points singuliers servent d'estimateurs locaux plus précis de la position de la ligne d'écriture (ligne de base).

Les minima globaux des arcs de squelette sont également extraits. Les points singuliers retenus sont ceux qui se situent dans la bande de base, et dont l'un des arcs de squelette auxquels ils sont reliés a un minimum global qui se situe sous la bande de base.

Un exemple est donné figure 2.3.16(b). Dans cet exemple, trois points singuliers sont candidats.

- Celui de gauche appartient à une boucle. Il n'est pas retenu, car les arcs de squelette auxquels il est relié ne vont pas sous la bande de base.
- Celui du milieu appartient également à une boucle. Il est retenu, un trait noir est marqué pour le relier au minimum global de l'arc de squelette considéré.
- Celui de droite appartient à un embranchement. Il est retenu, un trait bleu est marqué pour le relier au minimum global de l'arc de squelette considéré. Les minima locaux du contour inférieur sont également retenus s'ils se situent dans la bande de base. Ces minima locaux et les points singuliers retenus sont de bons estimateurs locaux de la position de la ligne de base. La ligne de base affinée est donc construite comme une interpolation linéaire à partir de ces points retenus. La partie supérieure de la bande de base est donnée par la hauteur du tracé [16], supposée constante sur toute la ligne.



(a) Extraction du squelette et des traits,  
pour déterminer les points singuliers (b) Détection des descendants

Fig.2.3.16. Utilisation des points singuliers du squelette pour trouver une évaluation plus fine de la bande de base [16].

Au final, on obtient une détection de la ligne de base qui s'adapte bien aux légères variations de l'inclinaison de l'écriture au sein d'une même ligne (voir figure 2.3.16) [16].

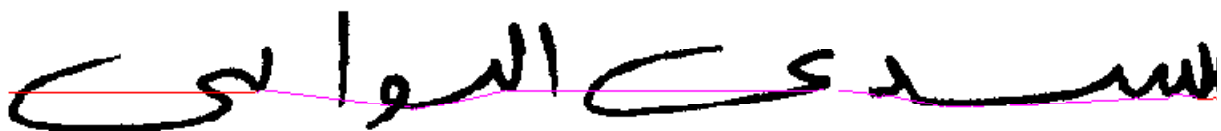


Fig.2.3.17. Détection fine de la bande de base [16].

## 6.2.Algorithme de Hilditch

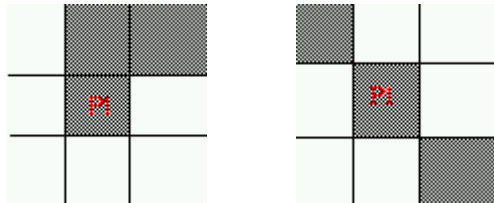
### a) Définition de quelques fonctions [28]:

Considérez ce qui suit 8-voisinage d'un pixel  $p_l$

P9	P2	P3
P8	P1	P4
P7	P6	P5

Nous voulons décider de décoller p1 ou le garder comme une partie du squelette qui en résulte. A cet effet, nous organisons les 8 voisins de p1 dans un ordre aiguilles d'une montre et nous définissons les deux fonctions:

$B(p1)$  = nombre de non-zéro voisins de p1 et un certain nombre  $A(p1)$  = de 0,1 motifs dans la séquence de P2, P3, P4, P5, P6, P7, P8, P9, p2



$B(p1) = 2, A(p1) = 1$

$B(p1) = 2, A(p1) = 2$

### b) Propriétés de l'algorithme de Hilditch [28] :

Il s'agit d'un algorithme parallèle séquentiel. Il est parallèle parce qu'à un seul passage tous les pixels sont vérifiés en même temps et si les décisions sont prises pour supprimer chacun des pixels enregistrés. Il est séquentiel, car cette étape vient de mentionner est répété plusieurs fois (jusqu'à plus de changements sont effectués). Cependant, l'algorithme de Hilditch s'est avéré ne pas être parfait pour l'algorithme de squelettisation, car il ne fonctionne pas sur tous les modèles. En fait, il existe des modèles qui sont complètement effacés par l'algorithme.



Les modèles que l'algorithme Hilditch efface complètement.

## 7. Segmentation en graphèmes :

La segmentation en graphèmes utilisée dans ce système est la même que celle utilisée pour l'écriture cursive latine. Des exemples de segmentation en graphèmes sont donnés figure 2.3.18 [16].

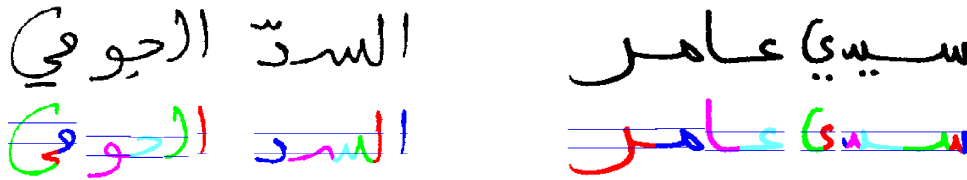


Fig.2.3.18. Exemples de segmentations en graphèmes [16].

### 7.1.Segmentation de l'écriture cursive:

La segmentation en graphèmes permet de délocaliser le problème de segmentation des mots au niveau des lettres. Segmenter l'image d'un mot sans la découper au préalable est une méthode limitée à des problèmes restreints où le nombre de mots est faible comme l'écriture d'un nombre en lettres. Les graphèmes sont des images extraites de l'image à segmenter. Passer d'une seule image à une séquence de graphèmes pose le problème de la taille de ces éléments. Ils ne doivent pas être trop petits afin d'être statistiquement significatifs, et pas trop gros afin de ne pas dépasser la taille d'une lettre. Il est en effet important qu'un graphème donné soit une sous-partie d'une seule lettre ou une lettre : cette condition est nécessaire pour construire un modèle de mot comme étant la concaténation de modèles de lettres [29].

### 7.2.Procédure de segmentation en graphèmes:

L'algorithme qu'on va présenter est une nouvelle technique dans la construction d'objets appelés graphèmes à partir de plus petits segments. L'idée principale est basée sur la concurrence entre chaque segment pour devenir un graphème ou d'adhérer à un graphème existant.

La segmentation des mots écrits à la main est l'une des tâches les plus difficiles dans le traitement de manuscrits, en vue de la grande variabilité dans les styles d'écriture.

Cette approche consiste à segmenter les mots en petits morceaux basée sur la régularité et la courbure de l'écriture. Certains segments sont parfois très petits. Donc Par conséquent, un nouvel algorithme a été développé appelé Two-cost grouping (Deux coût de regroupement), pour grouper un ou plusieurs petits segments dans des objets plus gros, appelé graphèmes.

Le processus de segmentation coupe en petits segments les mots manuscrits basé sur la régularité et courbure de ces derniers. Ces segments sont parfois trop petits et distribués en

dimensions (2D) de l'espace. Ainsi, un processus de regroupement de segment est nécessaire pour grouper plusieurs segments en un ensemble qui va former un objet plus informatif, appelé graphème [30].



Fig.2.3.19. Un exemple de chèque français sur le quel sera appliquer la segmentation [30].



Fig.2.3.20. Procédure de segmentation du mot trente en graphèmes [30]. L'idée principale derrière ce processus est de couper un mot en petits morceaux en fonction de leur régularité et leur courbure. Un Canny-Deriche détecteur de pointe est utilisée pour extraire les arêtes d'un mot. Une fois que tous les bords sont détectés, un processus d'appariement est effectué. Chaque ensemble de pixel représentant une arête est apparié avec le meilleur correspondant en pixel de bord de l'arête opposée. Quand aucune correspondance n'est possible, alors, le pixel appartient à une région singulière. Il découle de cette procédure un ensemble de listes de sections, qui représente un moyen de traçabilité, une liste particulière correspondant au traçage des intersections entre les points de deux régions singulières ou un

point d'intersection et un point terminal. La figure 2.3.21 montre l'image initiale en 256 niveaux de gris du mot «trente». Après la segmentation, le résultat est illustré à la figure 2.3.22 [30].

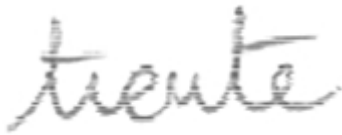


Fig.2.3.21.L'image originale du mot « trente » [30]



Fig.2.3.22.Résultat du processus de segmentation. Chaque segment est illustré en couleur différente [30].

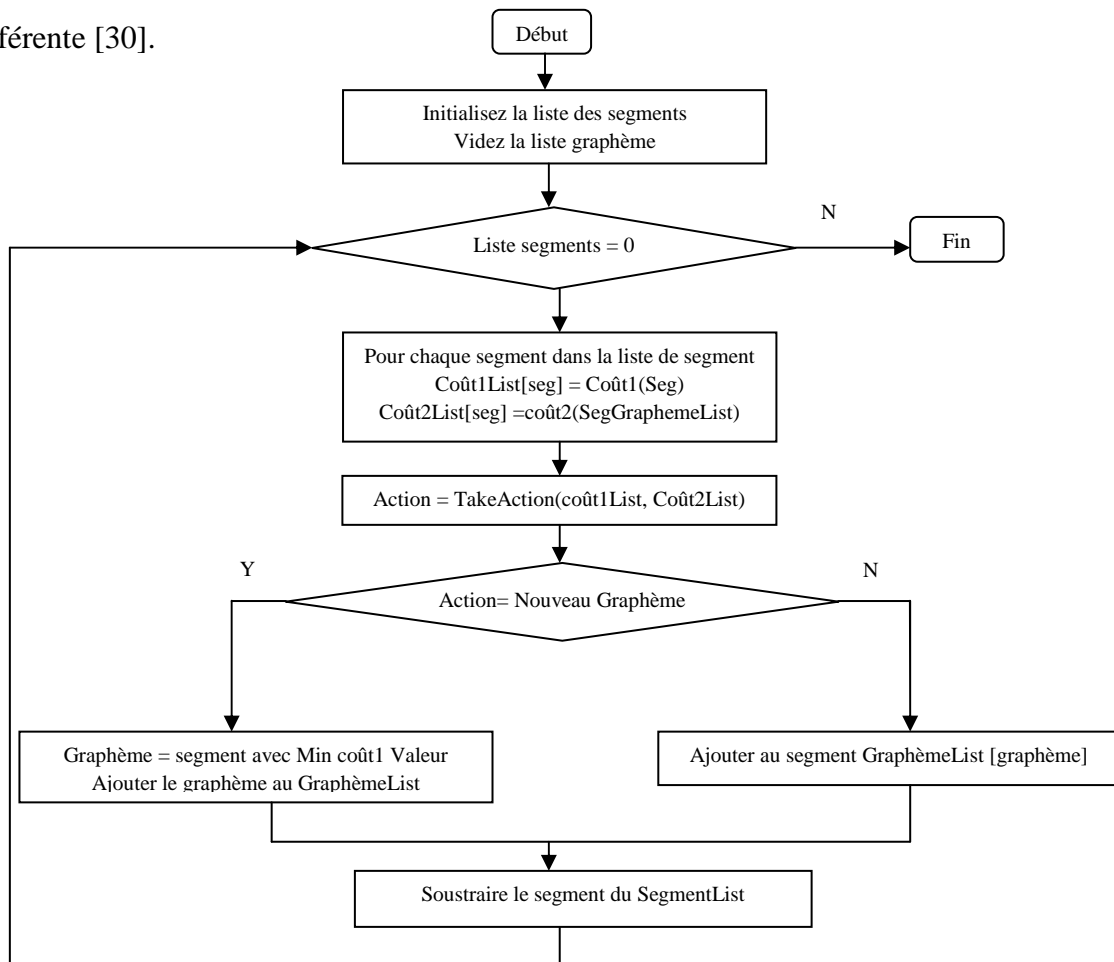


Fig.2.3.23.Algorithme du Two-cost-grouping [30]



Fig.2.3.24. Le mot « trente » après le processus de regroupement des segments [30].

#### a) Groupement des petits segments :

Le processus de segmentation génère une Distribution 2D des petits segments. En conséquence, un processus de regroupement des segments est mis en œuvre Afin de regrouper les petits segments appartenant au même objet, appelé graphème. Les graphèmes seront ensuite classés de gauche à droite, résultant en des signaux 1D. Après chaque graphème sera constitué d'un ou plusieurs segments, ils fournissent des informations plus structurales que les segments. Donc pour résoudre ce problème de regroupement, on va utiliser un algorithme nommé Two-cost grouping. Le principe de fonctionnement de cet algorithme est de grouper les segments de manière itérative en utilisant leurs caractéristiques géométriques (comme la hauteur, la largeur, etc) et leur contexte (la distance de leurs graphèmes voisins, la hauteur et largeur de leur graphèmes voisins, etc.) Pour chaque segment, nous calculons deux fonctions de coût, à savoir Coût1 et Coût2. Coût1 est défini comme le coût d'un segment qui va devenir un graphème en soi, et Coût2 est défini comme le coût d'un segment qui va rejoindre l'un des graphèmes déjà existants. Ce processus itératif calcule les deux coûts pour chaque segment. A la fin de chaque itération, une action est prise. L'action entreprise est déterminée en comparant la valeur minimum de Coût1 avec la valeur minimum Coût2 de tous les segments. Cette action est soit de faire d'un segment un graphème en soi, ou de joindre le segment dans un graphème existant. Après cela, le processus recalcule les deux coûts des valeurs pour les segments restants et le processus continue jusqu'à ce qu'il n'y ait aucun segment qui demeure [30].

#### 7.3.Segmentation à partir du squelette:

Pour notre cas de figure on va appliquer le même processus de segmentation donné précédemment sur un mot après être rendu à l'état de squelette. La figure suivante est un exemple de ce qui est obtenu.



Fig.2.3.25. Segmentation à partir du squelette [29].

### 8. Segmentation en mots :

Dans les segmentations en mots basé sur celles en graphèmes il s'agit de déterminer les graphèmes consécutifs qui appartiennent à deux mots différents. Par exemple, la figure précédente contient 17 graphèmes, soit au plus seize coupures entre deux mots. Le principe consiste à affecter à chacune de ces coupures une probabilité de séparer deux mots, celle-ci dépend de paramètres tels que la distance entre les deux graphèmes qui l'entourent, leurs tailles, leurs formes... S'il y a  $N$  graphèmes, on obtient  $N - 1$  probabilités de césure  $(P_1, \dots, P_{N-1})$ . A chaque point de césure, on associe la variable aléatoire  $Y_i \in \{0, 1\}$  vérifiant  $P(Y_i = 1) = p_i$ . Une segmentation en mots est alors complètement décrite par la donnée de  $(Y_1, \dots, Y_{N-1})$ . Comme ces variables aléatoires sont indépendantes, la probabilité associée à cette segmentation est :

$$\mathbb{P}(Y_1, \dots, Y_{N-1}) = \prod_{i=1}^{N-1} p_i(\theta)^{Y_i} (1 - p_i(\theta))^{1-Y_i}$$

Chaque  $p_i(\theta)$  est fonction à valeur dans  $[0, 1]$  et qui dépend de caractéristiques  $\theta$  extraites de l'image. Cette fonction peut être par exemple un réseau de neurones estimé en maximisant la vraisemblance par rapport à  $\theta$  sur une base d'images pour laquelle les valeurs  $(Y_i)_i$  sont connues. Une fois cette fonction apprise, cette écriture permet de trouver la segmentation en mots la plus probable. Il est également parfois utile de conserver les segmentations les plus probables lorsque l'écriture à découper est ambiguë [29].

Aujourd'hui la segmentation en mots repose sur un réseau de neurones qui apprend si oui ou non il existe une séparation inter-mots entre deux composantes connexes.

#### a) Procédure de segmentation en mots:

La segmentation automatique en mots est donc basée sur :

- Prendre plusieurs dispositifs de chaque côté d'un point potentiel de segmentation et employer un réseau neurologique pour décider si la segmentation est entre deux mots distincts.

La différence entre la segmentation en mots de manuscrit arabes avec la celle de manuscrits latin est : la présence des points multiples (signes diacritiques de manière générale) au-dessus

et au-dessous du corps principale du mot qu'on aura filtrer au préalable pour adapter cette solution a notre cas de figure (expliquer précédemment dans le paragraphe 5), ainsi que l'absence des lettres majuscules au début des phrases situation qui est prise en compte par cette solution dans les manuscrit latin et dont on a pas besoin pour notre solution de segmentation de manuscrits arabes.

Le processus de segmentation en mot commence par l'obtention l'ensemble de composants reliés pour chaque ligne dans l'image du document. La figure 2.3.26 montre une région d'une image de manuscrit où les composants reliés (découpes extérieures et intérieures) d'une petite section [31].

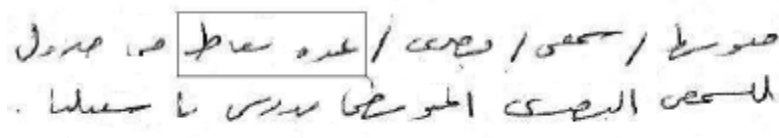


Fig.2.3.26.a. Région dans un document manuscrit [31]



Fig.2.3.26.b. Composants reliés intérieures et découpes extérieurs [31]

Fig.2.3.26. Composants reliés d'une petite section de l'image [31]

Les découpes ou les boucles d'intérieur dans un composant sont ignorées afin de les reliés, sont groupés dans des faisceaux (clusters). En fusionnant les composants important. Egalement une des propriétés de la langue arabe peut être utilisée, la plupart des mots commence par le caractère Alef. La présence de Alef est un indicateur fort qu'il peut y a avoir un espace de mot entre les paires de faisceaux. La taille et la largeur du composant sont des paramètres employés pour vérifier si le composant est le caractère Alef. La figure 2.3.27 montre des échantillons du caractère arabe Alef [31].



Fig.2.3.27. Echantillons de caractère Alef [31]

Chacune de ces paires de faisceaux adjacents sont des candidats pour des séparations entre de mots. 9 dispositifs sont extraits pour ces paires de faisceaux et un réseau neurologique est employé pour déterminer si l'espace entre les paires est un espace de mot. Les 9 dispositifs sont :

- La largeur du premier faisceau
- La largeur du deuxième faisceau
- Différence entre la boîte de bondissement des deux faisceaux,
- Drapeau a placé à 1 ou 0 selon la présence ou l'absence du caractère Alef dans le premier faisceau,
- Le même drapeau pour le deuxième faisceau,
- Le nombre de composants dans le premier faisceau
- Le nombre de composants dans le deuxième faisceau
- La distance minimum entre coques de corps convexe enfermant les deux faisceaux
- Le rapport entre, la somme des secteurs a enfermé par les coques convexes des différents faisceaux adjacents a déterminé des séparations entre deux mots.

Un réseau neurologique a été formé en utilisant ces neuf dispositifs avec le vecteur de dispositif marqué si c'est un espace de mot ou pas [31].

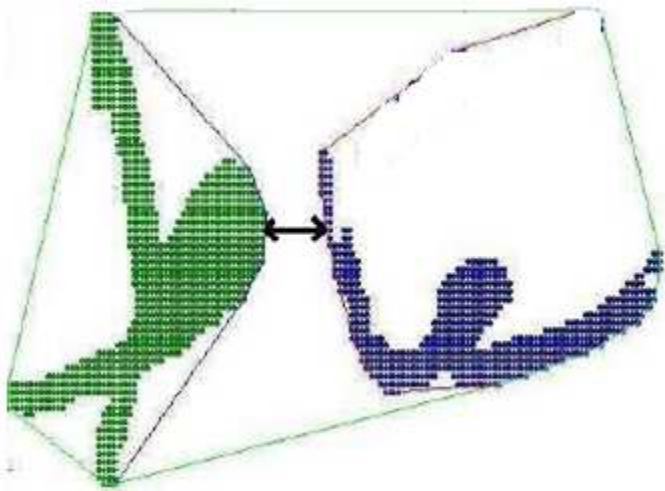


Fig.2.3.28.N'est pas un espace de mots [31]

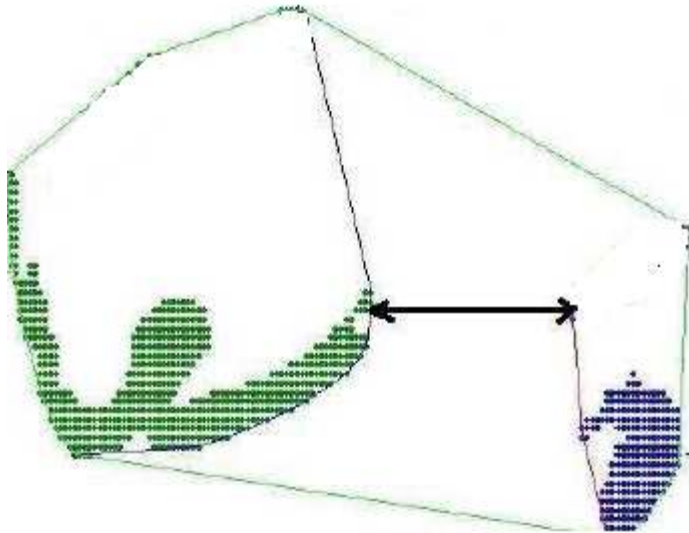


Fig.2.3.29.Est un espace de mots [31]

La figure suivante montre les 4 premières paires de faisceaux dont les dispositifs sont extraits et alimenté au réseau neurologique. Ces 4 paires de faisceaux correspondent aux mots à partir de la gauche sur la première ligne d'un document manuscrit [31].

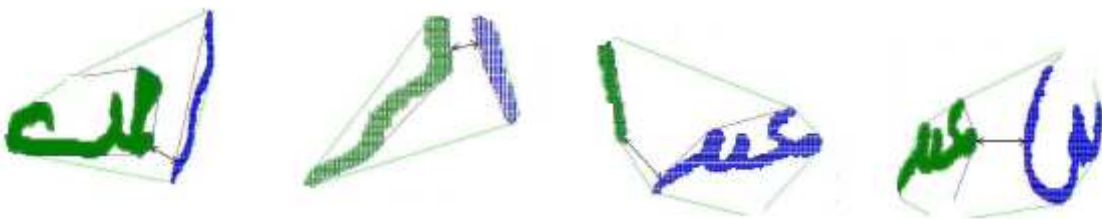


Fig.2.3.30.Exemples de 4 paires de faisceaux [31]

Le rendement du réseau neurologique indique qu'une présence d'espace de mot et d'un résultat négatif indique l'absence d'un espace de mot.

Il est évident que bien que la distance entre faisceaux pour les paires 1 et les paires 3 soient presque identique, le réseau neurologique donne la classification correcte. C'est du à des présences Alef qui indique fortement la présence d'un espace de mot.

La figure 2.3.31 montre un document segmenté mots par le réseau neurologique [31].



Fig.2.3.31. Image de document segmenté en mots par le réseau neurologique [31]

## 9. Conclusion :

Dans ce chapitre on a présenté la segmentation de manuscrits arabe anciens, d'abord on présenté les méthodes de filtrage, ensuite de binarisation qui sont les prétraitements nécessaire avant toute opération de segmentation, puis on utilisé plusieurs procédures pour arriver à la segmentation dont on besoins pour transcription qui sera présentée dans la partie suivante, ces procédures sont les suivantes appliquées de manière séquentielles l'une après l'autres :

- Segmentation en contours avec l'opérateur de Canny
- Segmentation en régions connexe avec l'étiquetage des composantes connexes de manières séquentielle après l'opérateur de Canny
- Filtrage des composantes connexes correspondant à des signes diacritiques
- Détection de la bande de base
- Squelettisation

- Segmentation en graphèmes
- Segmentation en mots

Le résultat de cette partie nous permettra d'avoir des images sur les quelles chaque peut être reconnu par le scripteur sans difficulté et donc peut être transcrit facilement.

## 1. Introduction:

Les manuscrits arabes constituent un trésor universel pour l'humanité, ils se retrouvent constamment menacés par un effritement à cause de la précarité et du manque de moyens dans les lieux de conservation. La manipulation de ces œuvres constitue un danger supplémentaire de détérioration. Par conséquent, la transcription constitue une solution idéale qui permettra à la fois de préserver les manuscrits dans de meilleurs états et de donner la possibilité d'accès distant aux chercheurs, aux étudiants et aux amateurs en générale, et c'est ce dont on parler dans cette dernière partie.

Le transcripteur aura comme tâche de reproduire le texte d'un manuscrit segmenter en mots sur un éditeur, ensuite se dernier sera enregistrer en format XML qui sera par la suite convertit en HTML ou PDF au choix grâce aux feuilles de styles (XSLT, XSL-FO ...), pour décrire le manuscrit de manière complète et uniforme on va utiliser la TEI qui sera prochainement décrit dans cette partie.

## 2. Définition :

Transcrire veut dire : Copier, reproduire par le moyen de l'écriture.  
Ou : Transposer dans une écriture différente [32].

La transcription diplomatique consiste à transcrire le contenu du manuscrit tout en respectant la disposition des lignes [33].

La transcription est un travail de paléographe qui doit être fait au crayon lorsque le manuscrit pose des problèmes de graphie. Cette étape peut être évitée si les qualités du paléographe ou la lisibilité du manuscrit sont suffisantes. La transcription peut alors se faire directement sur l'ordinateur en codant et balisant lexicalement le document [34].

La transcription des manuscrits arabes anciens est une solution pragmatique, au vu de la difficulté posée par la reconnaissance optique des caractères, la plupart des projets de numérisation des manuscrits anciens optèrent pour une transcription pure et simple des manuscrits afin d'obtenir des versions en mode texte. Dans notre cas, il s'agit de mettre en place un système de transcription diplomatique des manuscrits arabes. Une relation bijective sera établie entre chaque ligne de la transcription à la position de l'image du manuscrit.

Un système de sélection de lignes dans l'image du manuscrit devra être mis en œuvre. Il permettra de relier d'une manière biunivoque le texte transcrit et la partie sélectionnée de

l'image du manuscrit. Ce système permettra le passage du mode image au mode texte et vis versa grâce à des liens hypertextes [33].

### **3. Description de l'outil de transcription à mettre en place :**

#### **3.1. Structure de l'outil de transcription :**

- L'outil de transcription des manuscrits arabes anciens se basera sur l'identification des divers objets sélectionnés par le transcripneur.
- Lors de la saisie du texte, à chaque mot saisi correspond son équivalent présélectionné dans l'image du manuscrit.
- Les différentes classes d'objets (enluminures, texte, ...etc.) seront repérées, extraites et placées dans la partie correspondante du document transcrit.
- Les manuscrits étant dans leur majorité des œuvres littéraires, ce qui milite pour l'utilisation de la TEI (Text Encoding Initiative) comme format d'encodage [33].

#### **3.2. Fonctionnalités de l'outil de transcription :**

L'outil de transcription des manuscrits arabes anciens devra être capable d'exécuter les fonctions de base d'un traitement de texte (Gras, italique, souligné, couleur,...etc.).

Grâce à la structure de la TEI, une table de matière peut être générée automatiquement pour chaque manuscrit transcrit. Des feuilles de styles seront mises en place, ce qui permettra de générer des documents sous divers formats en appliquant la transformée appropriée (XSLT, XSL-FO,...) [33].

### **4. Règles de transcription :**

La transcription diplomatique, c'est-à-dire respectueuse de la topographie de l'écriture, ainsi que du tout détail se trouvant sur le manuscrit comme le positionnement des lignes.

### **5. L'Outil TEI :**

#### **5.1. Les enjeux de la TEI "Text Encoding for Interchange"**

Faciliter la création, l'échange, et l'intégration des données :

-Textuelles informatisées, toute sorte de texte, toutes les langues, toute origine temporelle ou culturelle.

La TEI s'adresse tout autant...

-Aux débutants, cherchant des solutions bien connues et consensuelles, mais aussi aux experts, cherchant à créer de nouvelles solutions [34].

## 5.2. Les buts de la TEI

-Faire des recommandations qui se basent sur un consensus existant

-Préférer les solutions générales à celles spécifiques à une discipline en même temps permettre la spécialisation et l'extension [34].

## 5.3. Qu'est-ce que c'est que la TEI aujourd'hui:

Un ensemble des *Guidelines* (lignes directrices) : peu prescriptives représentant un consensus au sujet des distinctions significatives dans un vaste ensemble de matériaux textuels qui s'expriment en deux gros volumes de prose et un ensemble de définitions formelles ces définitions peuvent être exprimées en plusieurs langages de schéma:

-TEI P1-P3 (1991-1999) : en SGML DTD

-TEI P4 (2000) : soit en SGML soit en XML DTD

-TEI P5 (2005-) soit en XML DTD, en W3C Schema, ou en RelaxNG derrière cet ensemble se trouve un modèle formel ("conceptual schema") de plus en plus élaboré en système de classes, voire en ontologie [34].

## 5.4. Organisation de la TEI

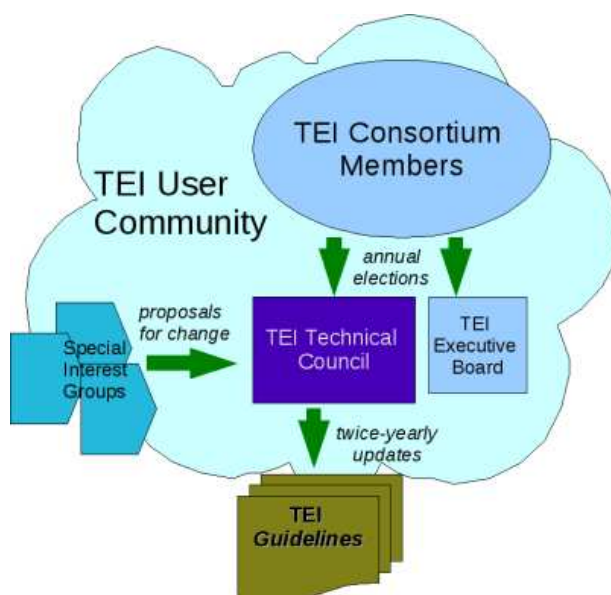


Fig.2.4.1. Organisation de la TEI [34].

### 5.5. TEI système modulaire:

TEI est un système *modulaire*. On s'en sert pour créer un modèle d'encodage selon ses propres besoins, en sélectionnant des *modules* spécifiques. Chaque module définit un groupe d'éléments (et leurs attributs) on peut sélectionner les éléments voulus, et même en changer des propriétés on peut y ajouter des éléments nouveaux, issus soit de TEI soit d'autres standards [34].

### 5.6. L'envergure TEI :

- Structuration basique des textes continus
- Transcription diplomatique, images, multimédia, annotations
- Données formelles : dates, noms de lieux ou de personnes...
- Données paratextuelles et "meta"
- Analyses linguistiques à tout niveau
- Documentation de balisage [34].

## 6. La transcription du manuscrit avec TEI :

Après la transcription vient l'encodage de celle-ci.

Cependant, pour simplifier les choses, on fournit la transcription seule dans un fichier texte brut [35].

### 6.1. Le fichier XML/TEI à utiliser, sa structure :

- Le fichier a pour élément racine TEI et que l'espace de noms par défaut est l'espace de noms TEI (ce qui est une information très importante, à conserver absolument).
- La suite du fichier consiste en un élément <teiHeader>
- Puis on trouve un élément <text>, lequel contient <body>. Cette structure simple est très courante pour les fichiers TEI. Elle est tout à fait adaptée à notre besoin, dans le cas d'une transcription d'un seul manuscrit.
- S'il s'agit d'une transcription d'une série de manuscrits homogènes, en faisant précéder l'édition des manuscrits d'une introduction générale, et en construisant (éventuellement) des index et une liste de notes communs à tous ces manuscrits, nous pourrions utiliser une structure dans laquelle <text> contiendrait <front> (pour les préliminaires communs), puis <group>, et enfin éventuellement <back> (pour les notes et index). Dans <group>, chacun des manuscrits serait édité en utilisant un élément <text>.

-Si c'était l'édition d'un corpus de manuscrits plus hétérogènes, chacun d'entre eux méritant des métadonnées, une introduction et un paratexte spécifiques, nous pourrions choisir d'utiliser <teiCorpus> comme élément racine de notre fichier, puis après un élément <teiHeader> général, autant d'éléments <TEI> que nécessaire pour les manuscrits composant le corpus.

-On va transcrire le texte du manuscrit dans l'élément <div> prévu à cet effet; cet élément possède un attribut @type qui en spécifie la fonction, ce qui permettra ultérieurement de le distinguer d'autres éléments de même nom, afin par exemple d'en produire une version HTML [35].

## **6.2. Première étape : reconnaître la structure logique du texte :**

Avant tout, on va avoir besoin de comprendre la structure logique du texte inscrit sur la page.

Il faut absolument, pour le faire correctement, comprendre le texte lui-même.

Il restera à déterminer quels éléments TEI utiliser pour encoder cette structure logique, et à encoder cette ossature, qui va rester stable dans le fichier XML - l'élément <div> déjà encodé servira de conteneur pour cette ossature, qui englobera toute la transcription [35].

## **6.3. Deuxième étape : transcrire le texte brut :**

On va se contenter dans un premier temps d'encoder la transcription telle quelle, sans entrer dans les détails, dans la structure XML. Il y a plusieurs manières de faire. On peut par exemple copier/coller les segments de texte directement au bon endroit dans le fichier TEI.

Penser à vérifier que le fichier est conforme à XML. Ou tout simplement saisir le texte et l'enregistrer comme fichier XML (méthode qu'on va utiliser dans notre cas) [35].

## **6.4. Traitement des abréviations:**

Pour garder la possibilité de choisir ultérieurement comment restituer ces abréviations à l'écran (de manière muette, ou au contraire en utilisant des signes conventionnels tels que les parenthèses), on va baliser explicitement les segments de texte développés et les abréviations, à l'aide des éléments <choice>, <abbr>, <am>, <expan>, <ex> [35].

## **6.5. Traitement des additions, suppressions et substitutions:**

Les textes à traiter peuvent comporter plusieurs types d'additions/suppressions :

- Si on considère (comme nous le proposons) que le texte se réduisait dans un premier temps à une série de questions inscrite sur la moitié droite de la page et que tout le reste constitue des

additions successives, alors il faut baliser avec <add> le contenu de paragraphes entiers de texte. Cela donnerait par exemple, pour le premier paragraphe de la série de réponses :

- A l'intérieur du 5<sup>e</sup> paragraphe de réponses inscrit sur la partie gauche de la page, le scripteur a d'abord écrit oui , puis a raturé ce mot, et a écrit à côté non, une demie. Cette opération peut être considérée comme une substitution ; pour l'encoder vous pourriez utiliser <subst>, dans lequel il faudrait inscrire <del> et <add> [35].

### 6.6. Encodage des mains intervenues dans le manuscrit

Pour compléter le balisage structurel du texte, il est possible d'utiliser l'attribut **@hand** sur chacun des segments de texte (ici, les paragraphes ou les segments ajoutés), pour spécifier à quelle main ces segments sont dus. La valeur de **@hand** doit être un renvoi vers un élément <handNote> qui décrit la main concernée, et qui doit être doté d'un attribut **@xml:id** (attribut de l'espace de noms XML, permettant d'affecter à un élément un identifiant unique).

Les éléments <handNote> doivent donc être au nombre de trois. On peut les encoder au sein d'un élément <handDesc> ; <handDesc> se trouve dans <physDesc>, lui-même élément enfant de la description du manuscrit source <sourceDesc> de l'en-tête TEI [35].

### 6.7. Mise en page du texte :

Même s'il ne s'agit pas forcément de faire une transcription imitative, il peut être intéressant de rapporter dans la partie du fichier TEI consacrée à la transcription des informations sur la mise en page du texte. Noter par exemple les changements de page (si applicable), éventuellement les changements de ligne, ou l'emplacement relatif des ajouts, peut avoir un intérêt en soi, sans compter que cela peut s'avérer utile, notamment pour restituer ensuite adéquatement le fichier TEI en HTML ou autre format de sortie.

Ici, si l'on souhaite produire ensuite, à partir du fichier, une page HTML dans laquelle un retour à la ligne serait fait pour chaque retour à la ligne sur la page manuscrite, on doit poser des éléments <lb/>.

L'attribut **@place** peut aussi être utilisé pour noter à quel endroit sont faites les additions.

Si on souhaite garder dans le fichier XML une trace de cette caractéristique, vous pouvez utiliser l'élément <hi> avec un attribut **@rend** [35].

### 6.8. La signature :

Si à la fin du texte, on a une signature. Cette signature est d'ailleurs écrite en lettres plus grandes. On peut penser à plusieurs solutions pour baliser cette signature :

- Utiliser `<hi>` comme plus haut
- Utiliser `<seg>` avec un attribut `@type` et un attribut `@rend`
- Utiliser `<signed>`

Si on souhaite établir un lien entre la transcription du texte et l'image numérique du document, la solution la plus simple consiste à utiliser l'attribut `@facs`, de lui donner comme valeur le chemin d'accès à l'image.

Cet attribut est disponible pour de très nombreux éléments TEI. Dans notre fichier le mieux serait de le saisir sur l'élément `<div>` qui englobe toute la transcription [35].

### 7. Conclusion:

Dans cette partie on présente une solution pour la transcription des manuscrits arabes anciens, qui va être appliquée comme suit :

- Le texte d'origine sera transcrit sur un éditeur de texte
- Ensuite il sera enregistré sous formats XML en respectant les recommandations de la TEI
- En fin ce dernier pourra être converti et affiché en HTML ou PDF grâce aux feuilles de styles XSLT et XSL-FO ce qui va permettre l'accès en full-text à des manuscrits qui ont été à la base juste des images.

# **Partie3 : Conception et Réalisation**

## 1. Introduction :

Ce chapitre va traiter de la conception de l'outil de transcription dont traite notre thème, il contiendra tout les algorithmes qui permettrons de construire une application d'aide a la transcription.

## 2. Algorithme de mise en niveau de gris

Avant de commencer toute opération de filtrage, il est impératif de convertir l'image couleurs en niveau de gris. Cette opération aura pour effet d'affecter à chaque pixel de l'image une seule valeur comprise entre 0 et 255.

### 2.1. Conversion d'une image couleur en image en niveau du gris :

#### a) Principe :

Le principe est très facile, on récupère les valeurs des trois couleurs de chaque pixel de l'image Rouge, Vert et Bleu puis on calcule leur moyenne, ensuite on remplace chaque pixel par cette moyenne, enfin on obtient une image en niveau de gris où chaque pixel de l'image est représenté par une valeur comprise entre 0 et 255.

C'est cette image en niveau de gris qu'on va utiliser pour la méthode de filtrage.

#### b) L'algorithme:

```

Variable  i, j, HautImage, LargeImage, MoyenneImage, Bleu, Rouge, Vert : entier ;

          M : matrice d'entier ;

Début

    Pour i <- 0 jusqu'à (LargeImage - 1) faire

        Début

            Pour j <- 0 jusqu'à (HautImage - 1) faire

                Début

                    Rouge = Valeur_Rouge(M([i][j]) ;

                    Vert = Valeur_Vert(M([i][j]) ;

                    Bleu = Valeur_Bleu(M([i][j]) ;

                    MoyenneImage = (R+V+B)/3
  
```

```

M[i][j] = MoyenneImage ;

    Fimpour ;

Fimpour ;

Fin.

```

#### 4. Filtrage d'une image avec le filtre moyennneur :

##### a) Exemple introductif :

Soit deux matrice M et L définies par :

$$L = \begin{pmatrix} P1 & P2 & P3 \\ P4 & P5 & P6 \\ P7 & P8 & P9 \end{pmatrix} \quad M = \begin{pmatrix} M9 & M7 & M7 \\ M6 & M5 & M4 \\ M3 & M2 & M1 \end{pmatrix}$$

##### b) Principe :

Le principe de fonctionnement consiste à faire, le produit de convolution entre le masque de convolution h et le voisinage du pixel à modifier. Le voisinage est une matrice de 3x3, de ce fait, le masque conçu pour le filtre est une matrice de 3x3 qui est défini par la matrice h :

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

Le principe du filtre peu se résumer comme suit :

- Remplacer le pixel central défini dans une matrice 3x3, par la moyenne arithmétique de son voisinage.
- Utiliser l'image en niveau de gris M[i][j], pour chaque pixel de l'image. Calculer alors, la moyenne arithmétique du voisinage de chaque pixel de la matrice M. Le résultat sera stocké dans une variable nommé moyenne.
- Remplacer chaque pixel de la matrice M par la moyenne de son voisinage.

L'algorithme suivant représente l'algorithme du filtre moyennneur 3x3 d'une image en niveau de gris.

##### c) L'algorithme:

```

Variable i, j HautImage, LargeImage, Moyenne, Somme, a, b, c, d, e, f, g, h, k : entier ;

Début

```

```

a=1 ; b=1 ; c=1 ; d=1 ; e=1 ; f=1 ; g=1 ; h=1 ; k=1 ;
Somme= a + b + c + d + e + f+ g + h + k ;
  Pour i=1 jusqu'à (HautImage - 2) faire
Début
  Pour i<- jusqu'à (LargeImage - 2) faire
    Début
      Début
        Moyenne = (a* M[i+1][j+1] + b* M[i+1][j] + c* M[i+1][j-1] + d* M[i][j+1]
          + e* M[i][j]+ f* M[i][j-1]+g* M[i-1][j+1]+ h* M[i-1][j]+
          k* M[i-1][j+1])/Somme;
        M[i][j] <- Moyenne ;
        j<- j+1 ;
      Finpour ;
      i<- j+1 ;
    Finpour ;
  Fin.

```

## 5. Algorithme de binarisation : La méthode du seuil fixe :

### a) Le principe de fonctionnement :

Convertir l'image en niveau de gris.

- Parcourir tous les pixels de l'image avec lesquels le seuil T fixe donné par l'utilisateur sera comparé.
- Si la valeur du pixel est inférieure au seuil, le pixel prend alors la valeur 0, sinon il prend la valeur 255.

**b) L'algorithme:**

```

Début
Variable      i, j, HautImage, LargeImage, T : entier ; /* T : seuil
              M : matrice d'entier ;
Début
    Pour i<- 0 jusqu'à LargeImage-1 faire
        Début
            Pour j<-0jusuq'à HautImage-1 faire
                Début
                    Si M[i][j] < T alors
                        M[i][j]<- 0
                    Sinon
                        M[i][j] <- 255
                    Finsi ;
                Finpour ;
            Finpour ;
        Finpour ;
Fin.

```

**6. Segmentation****5.1. Algorithme de Canny :**

L'algorithme de Canny se présente comme suit :

- Chercher la largeur (w) et la longueur (h) de l'image originale ;
- Créer un tableau a deux dimensions (tab) de largeur (w) et de longueur (h) ;
- Faire un parcours de l'image originale pixel par pixel, en chercher les valeurs RGB de chacun des pixels, et remplir le tableau créé précédemment, ce que traduit l'algorithme suivant ;

```
Pour i ← 1 jusqu'à i < h faire
```

```
  Début
```

```
    Et pour j ← 1 jusqu'à j < w faire
```

```
      Début
```

```
        tab [i] [j] ← la valeur RGB du pixel (i, j)
```

```
      Fin
```

```
  Fin
```

- Faire un autre parcours de l'image originale, calculer X et Y les valeurs RGB des voisins du pixel courant, et le G qui est l'addition de leurs valeur absolus, puis affecter le résultat (G) obtenu a l'image résultat (Imageresultat), comme le montre l'algorithme qui suit :

```
Pour i=1 jusqu'à i < h-2 faire
```

```
  Début
```

```
    Et pour j=1 jusqu'à j < w-2 faire
```

```
      Début
```

```
        X ← - tab [i] [j] + tab [i] [j+2]
```

```
        Y ← tab [i] [j] - tab [i+2] [j]
```

```
        G ← |X| + |Y|
```

```
        Imageresultat [i][j] ← G
```

```
      Fin
```

```
  Fin
```

## 5.2.Algorithme d'étiquetage des composantes connexes :

L'algorithme d'étiquetage des composantes connexes se présente comme suit :

1. Parcourir l'image originale (image), comparer la valeur du pixel courant par rapport à ses voisins.

2. Si le pixel courant trouve qu'il porte la même valeur que se voisins il sera étiquette avec la même étiquette sinon il lui sera attribuer une nouvelle étiquette à l'aide des fonctions union, trouve et ajout comme le décrit l'algorithme suivant:

```

Copier les pixels de l'image dans roots un tableau à une dimension
Fonction union (root0 : entier, root1 : entier)
Début
    Si (root0 = root1) alors retourner (root0) Fsi
    Si (root0= -1) alors retourner (root1) Fsi
    Si (root1= -1) alors retourner (root0) Fsi
    Si (root0 < root1) alors
        roots[root1] ← root0
        retourner (root0)
    Fsi

    Sinon
        roots[root0] ← root1
        retourner (root1)
    Fin Sinon
Fin Union

Fonction trouve (K : entier)
Début
    Tant que (roots[K] ≠ K) Faire

        K ← roots[K]
    Fin tant que
    retourner (K)
Fin trouve

Fonction ajout (K : entier, entie : entier)
Début
    Si (entie= -1) alors

```

```

    roots[K] ← K
  Fin si

  Sinon
    roots[K] ← entie

  Fin sinon
Fin ajout

entie : entier
w : largeur de l'image
h : longueur de l'image

Pour i ← 1 et K=0 jusqu'à i < h faire {i ← 1 pour éviter les extrémités}
  Début
    Et pour j ← 1 jusqu'à j < w et k < (h*w) faire
      Début
        Si j > 0 et pixel courant = pixel précédent alors
          entie ← union (trouve (K-1), entie)
        Fsi
        Si j > 0 et i > 0 et pixel courant = pixel nord-ouest alors

          entie ← union (trouve (K-1-w), entie)
        Fsi
        Si i > 0 et pixel courant = pixel nord alors
          entie ← union (trouve (K-w), entie)
        Fsi

        Si j < w-1 et y > 0 et pixel courant = pixel nord-est alors
          entie ← union (trouve (K+1-w), entie)
        Fsi
      Fin
    Fin
  Fin
Fin

```

3. Parcourir l'image d'étiquette (image-étiq) obtenue, et chercher pour chaque pixel ces 4-voisinages, comme suit :

entier : nord, sud, ouest, est.

Pour  $y \leftarrow 1$  jusqu'à  $y < h$  faire

    Début

        Et pour  $x \leftarrow 1$  jusqu'à  $x < w$  faire

            Début

                Nord  $\leftarrow$  image-étiq [x] [y-1]

                Sud  $\leftarrow$  image-étiq [x] [y+1]

                Ouest  $\leftarrow$  image-étiq [x-1] [y]

                Est  $\leftarrow$  image-étiq [x+1] [y]

            Fin

    Fin

4. Comparer la valeur du pixel courant par rapport a ses 4-voisinages, si elle n'est pas identiques afficher une frontière.
5. Rechercher les coordonnées min et max de chaque composante connexe dans le but de dessiner les rectangles englobant de celles-ci.
6. Afficher les rectangles englobant dans une image.

## 6. Approche coopérative séquentielle :

### a) Principe :

Le principe générale de la coopération séquentielle est que l'une des techniques, par régions ou par contours, est réalisée en premier lieu. Son résultat sera par la suite exploiter par l'autre technique [27]. L'intégration de l'information provenant de la segmentation par contours dans une segmentation par régions est l'une des formes de coopération les plus courantes.

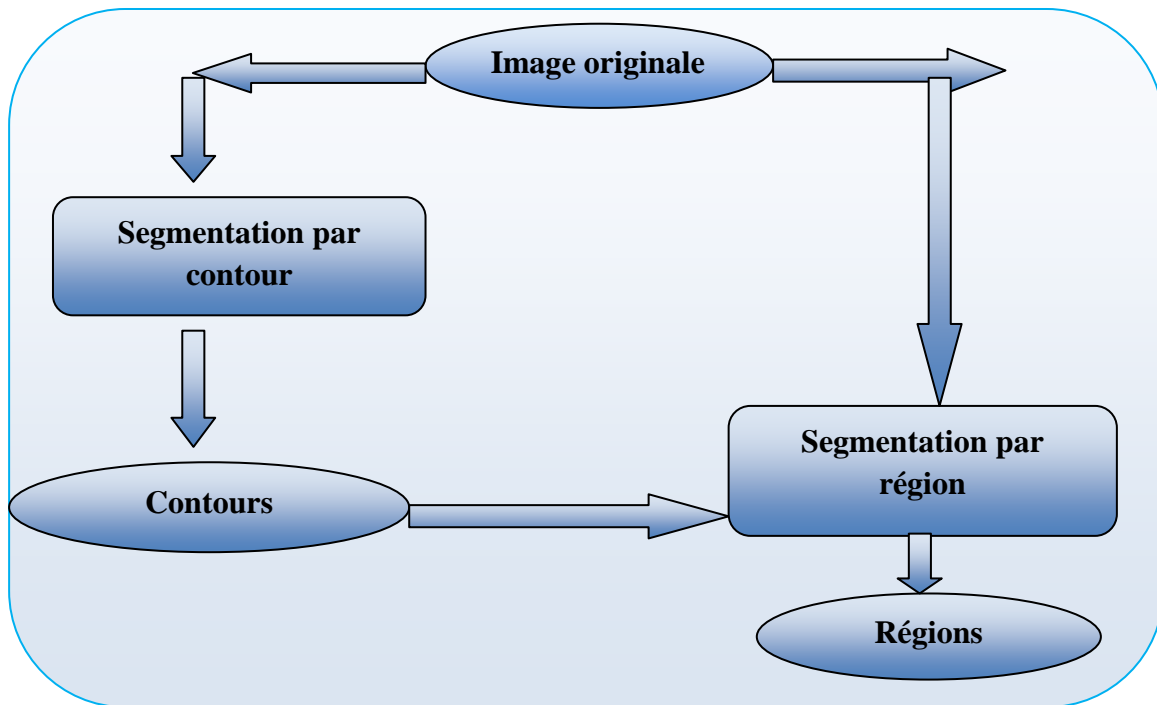


Fig.3.1.1. Principe de la coopération séquentielle.

**7. Algorithme de filtrage des composantes connexes correspondant à des signes diacritiques :**

Composante connexe

Si Aire > 15x épaisseur<sup>2</sup>

N'est pas un signe diacritique

Sinon

Si Hauteur > 4 x épaisseur

Si Largeur < 4 x épaisseur

N'est pas un signe diacritique

Sinon

Si Présence d'une CC en dessous avec recouvrement latéral au moins  $\frac{3}{4}$  alors

Est un diacritique

Sinon

N'est pas un signe diacritique

Sinon

Est un diacritique

Fin.

## 8. Détection de la bande de base :

### 8.1.Algorithme de l'histogramme de projection horizontale :

Données : Image binaire  $I$  contenant la forme  $F$

Résultat : Un tableau contenant l'histogramme de projections horizontal

Début

Initialiser le tableau *Horizontal* à 0 ;

Pour tous les *points*  $p(x, y) \in F$  Faire

Incrémenter (*Horizontal*[ $y$ ]) ;

Fin

Dans notre cas on va l'appliquer a chaque composantes connexes (boites englobantes).

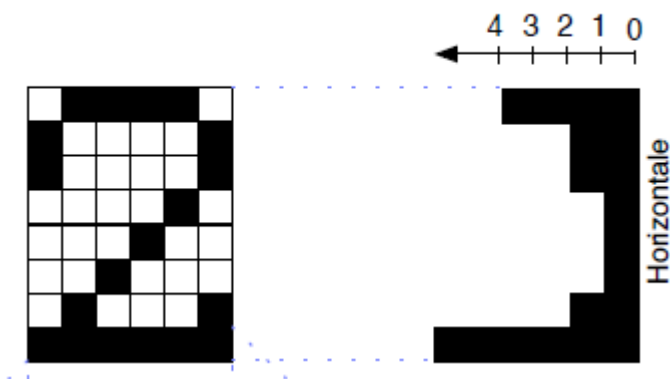


Fig.3.1.2.Exemple correspondant a l'algorithme.

### 8.2. Détection des boucles dans chaque composante connexe :

Pour la détection des boucles (و, ه, ه, ه, ه, ه, ق, ف, ع, ظ, ط, ض, ص, ة) dans chaque composante connexe, pour ça nous devons faire appelle à un système de reconnaissances qui va détecter ces lettres (boucles), à partir de là nous pouvons tester

l'histogramme de chaque composante connexe est trouver une zone élargie candidate pour être une bande de base dans chaque histogramme se sera dans cette zone qu'il faut chercher.

### 9. L'algorithme de squelettisation de Hilditch [28]:

Il existe deux versions de l'algorithme de Hilditch, l'un utilisant une fenêtre de 4x4 et l'autre en utilisant une fenêtre 3x3. Ici nous nous intéressons à la version 3x3 fenêtre.

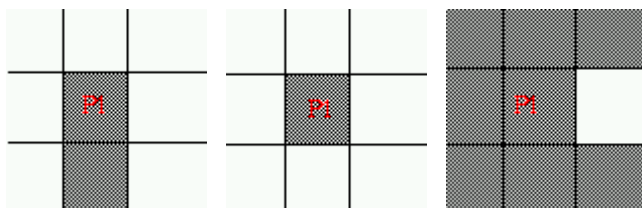
Algorithme Hilditch consiste à effectuer plusieurs passages sur la structure et à chaque passage, l'algorithme vérifie tous les pixels et décide de modifier un pixel du noir au blanc, si elle remplit les quatre conditions suivantes:

- $2 \leq B(p1) \leq 6$
  - $A(p1) = 1$
  - $p2.p4.p8 = 0$  ou  $A(p2) \neq 1$
  - $p2.p4.p6 = 0$  ou  $A(p4) \neq 1$
- Arrêtez-vous lorsque rien ne change (pas plus de pixels peut être retiré)

Laissez-nous voir chacune des conditions ci-dessus séparément.

#### ➤ Condition 1 : $2 \leq B(p1) \leq 6$

Cette condition combine deux sous-conditions, en premier lieu que le nombre de non-zéro voisins de  $p1$  est supérieur ou égal à 2 et la seconde qu'il soit inférieur ou égal à 6. La première condition assure qu'aucun pixel point final pas un cas isolé être supprimé (un pixel avec 1 voisin noir est un pixel point final), la deuxième condition assure que le pixel est un pixel de bord.



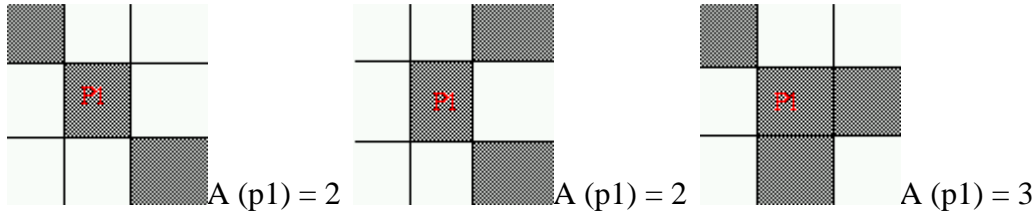
$$B(p1) = 1 \quad B(p1) = 0 \quad B(p1) = 7$$

Comme l'image, il est clair, si  $B(p1) = 1$ , alors  $p1$  est une astuce squelette-point et ne devrait pas être supprimé. Si  $B(p1) = 0$ , alors  $p1$  est un point isolé et devrait également être tenu (au cas où il ya un bruit, il n'est pas le travail du processus de squelettisation de supprimer les

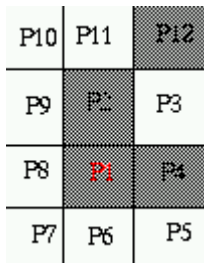
pixels (c'est le travail de l'algorithme de suppression du bruit). Si  $B(p1) = 7$ ,  $p1$  n'est plus sur la limite du motif et ne devraient donc pas être un candidat.

➤ **Condition 2 :  $A(p1)=1$**

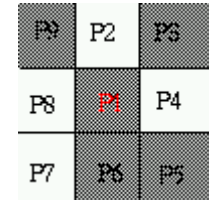
Ceci est un test de connectivité. En fait, si l'on considère les images ci-dessous où  $A(p1) > 1$ , vous pouvez voir qu'en changeant  $p1$  à 0 le gabarit est débranché.



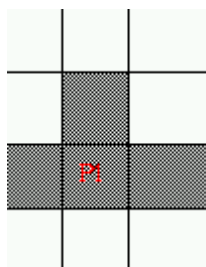
➤ **Condition 3:  $p2.p4.p8 = 0$  or  $A(p2) \neq 1$**



Voici un exemple où  $A(p2)$  n'est pas 1.

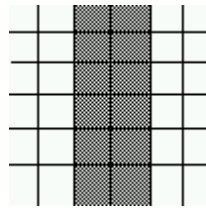


Voici un exemple où  $p2.p4.p8 = 0$ .



Voici un exemple où  $p2.p4.p8$  n'est pas 0 et  $A(p2) = 1$ .

Cette condition garantit que le 2-pixel de larges lignes verticales n'est pas complètement érodée par l'algorithme.



➤ **Condition 4:  $p_2.p_4.p_6 = 0$  or  $A(p_4) \neq 1$**

P9	P2	P3	P10
P8	<b>P1</b>	P4	P11
P7	P6	P5	P12

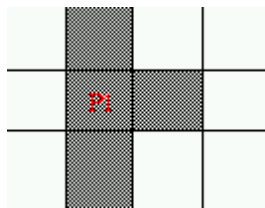
P9	P2	P3
P8	<b>P1</b>	P4
P7	P6	P5

Voici un cas ou

$A(p_4) \neq 1$ .

Voici un cas ou

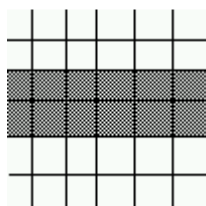
$p_2.p_4.p_6 = 0$ .



Voici un cas où

$p_2.p_4.p_6 = 0$  et  $A(p_4) = 1$ .

Cette condition garantit que le 2-pixel de larges lignes horizontales n'est pas complètement érodée par l'algorithme.



De manière générale, les méthodes d'obtention du squelette par érosions successives fonctionnent selon le principe suivant :

Répéter

Appliquer masques et règles pour déterminer les pixels noirs effaçables

Changer tous les pixels marqués comme effaçables de noir vers blanc

Jusqu'à

Aucun nouveau pixel n'a été effacé.

### 10. L'algorithme de segmentation en graphème:

Début

Initialisez la liste des segments

Videz la liste des graphèmes

Tant que segments = 0 faire

    Pour chaque segment dans la liste de segment faire

$Cout1List[seg] = Cout1(seg)$

$Cout2List[seg] = Cout2 (SegGraphemeList)$

Action = TakeAction (Cout1List, Cout2List)

    Si Action = NouveauGrapheme Alors

        Graphème = segment avec Min coût1 Valeur  
        Ajouter le graphème au GraphèmeList

    Sinon

        Ajouter au segment GraphèmeList [graphème]

Soustraire le segment du SegmentList

    Sinon

Fin.

### 11. Algorithme de segmentation en mots :

-Prendre plusieurs dispositifs de chaque côté d'un point potentiel de segmentation et employer un réseau neurologique pour décider si la segmentation est entre deux mots distincts.

-Alimenté le réseau neurologique par neuf dispositif qui permettrons de déduire si il ya une

séparation effective entre deux mots.

## **12. Conclusion:**

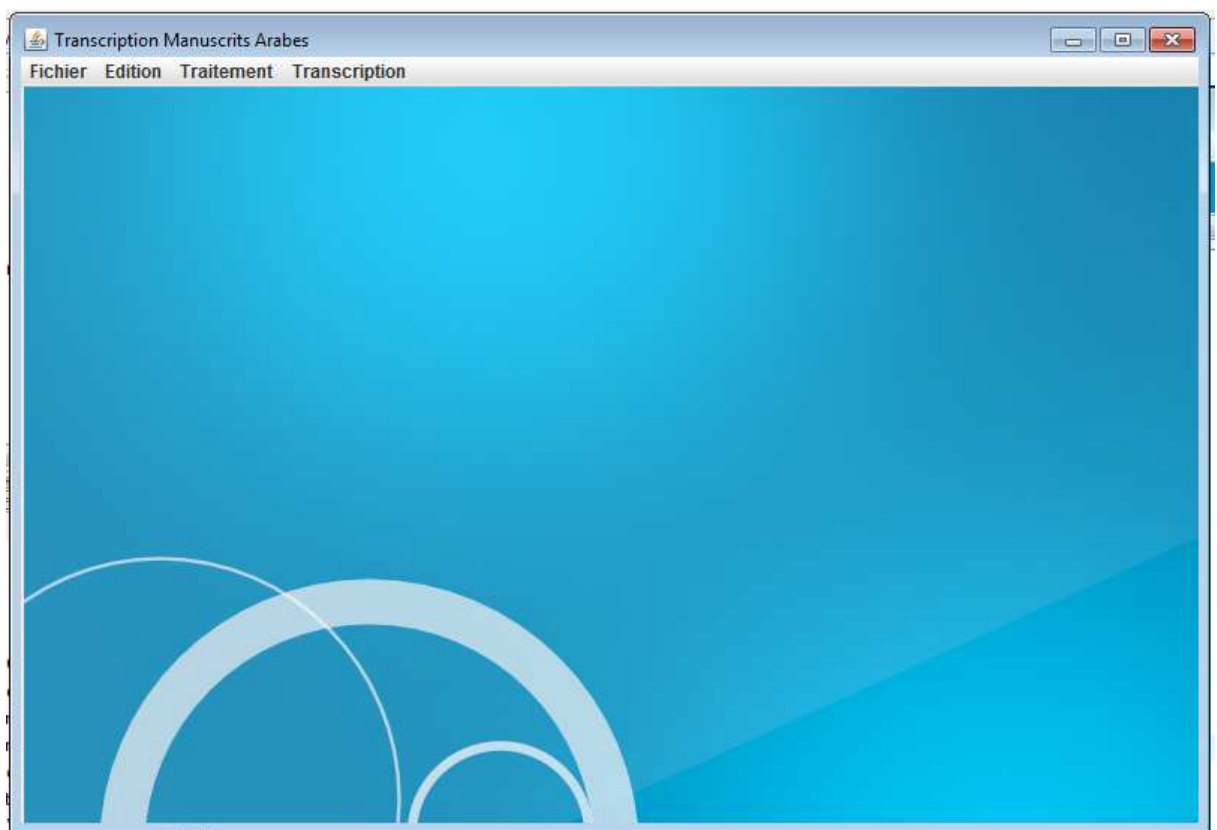
Dans ce chapitre nous avons présenté les algorithmes qui vont former l'outil de traitement de l'image du manuscrit, pour la partie transcription sera sous forme d'un éditeur ou va reprendre l'utilisateur les informations se trouvant sur l'image du manuscrit traiter précédemment, ce que nous allons expliquer dans la chapitre suivant réalisation.

## 1. Introduction:

Dans le chapitre qui va suivre nous allons expliquer le fonctionnement de l'outil de transcription réalisé ainsi que les résultats obtenus sur quelques exemples d'images de manuscrits arabes anciens numérisés extraits du corpus de la bibliothèque nationale d'EL-HAMA.

## 2. Présentation de l'outil de transcription réalisé et son fonctionnement:

### 2.1.Interface principale :



**Fig.3.2.1. Interface principale**

### 2.2.Traitement d'images :

Le système de traitement d'images que nous avons développé, permet d'effectuer plusieurs opérations sur les images de manuscrits numérisées, tel que la mise en niveau, le filtrage, la binarisation, la segmentation.

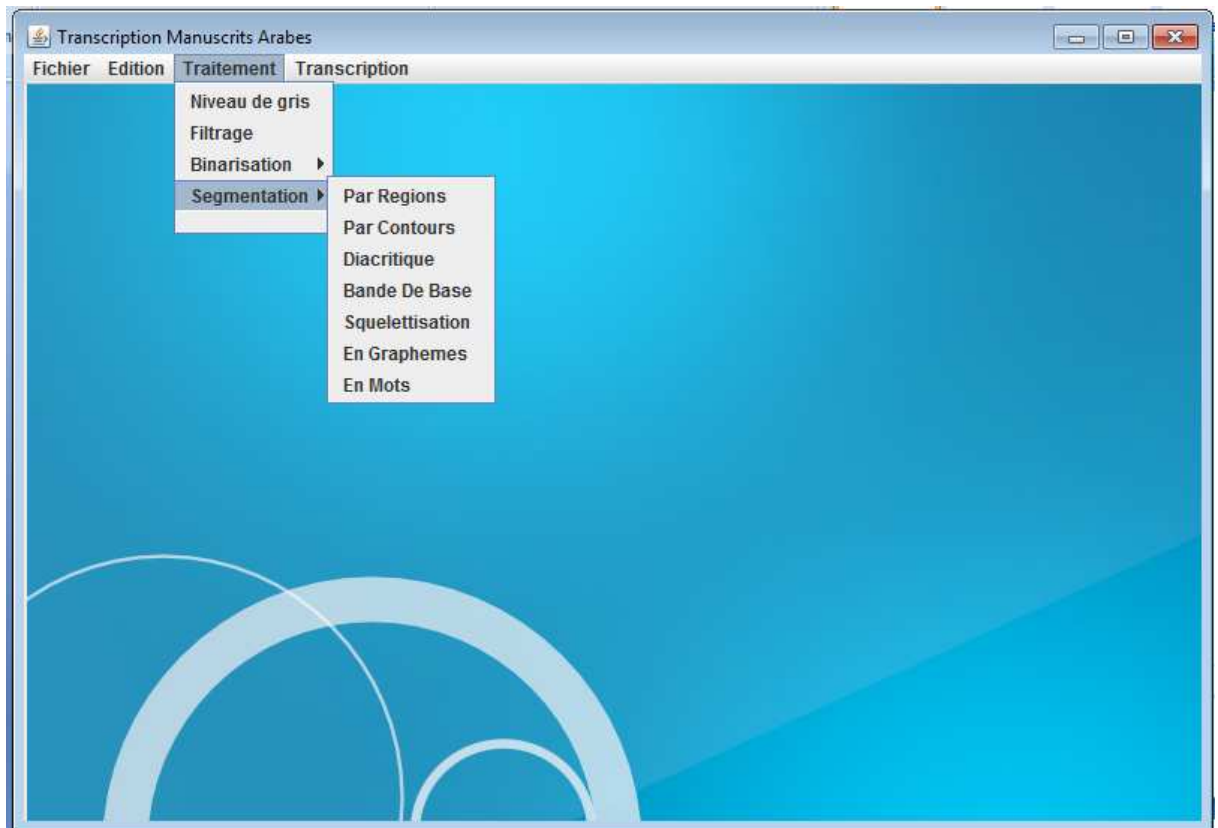


Fig.3.2.2. menu et sous menu

### 2.3.Transcription :

Pour crée le document XML/TEI vous devez cliquer sur le bouton transcription, il y aura l'apparition de cet éditeur sur le quel reprendre les informations se trouvant sur le manuscrit

Fig.3.2.3.Interface principale de l'éditeur de texte

**3. Résultats obtenus pour les traitements des images de manuscrits:**

**3.1.Image de manuscrit mise en niveau de gris :**



Fig.3.2.4 Image de manuscrits arabe numérisé [15].

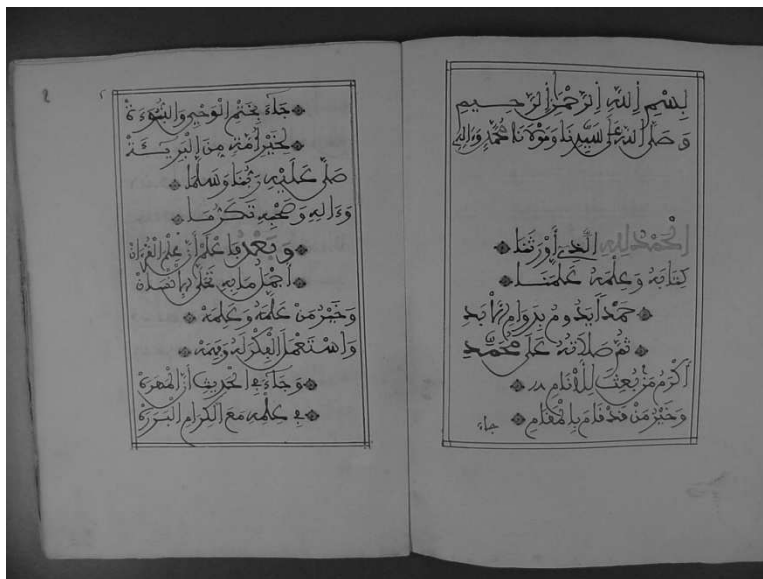


Fig.3.2.5. Image de manuscrits mis en niveau de gris [20]

3.2. Image de manuscrits obtenus avec le filtre moyenneur :

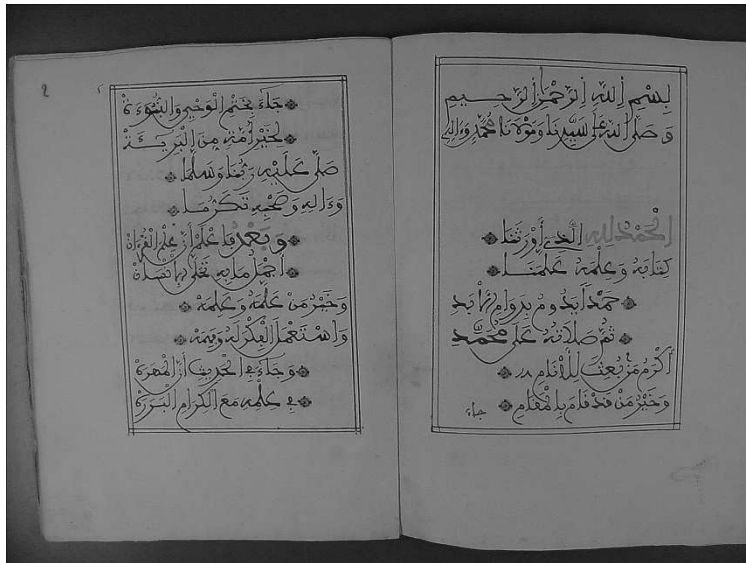


Fig.3.2. 6. Même image après filtrage ayant été mise au préalable en niveau de gris [20].

Lissage de l'image, nous distinguons mieux l'écriture par rapport au fond, et l'image est devenu un peu flou.

3.3. Images de manuscrits binarisée avec la méthode du seuil fixe :

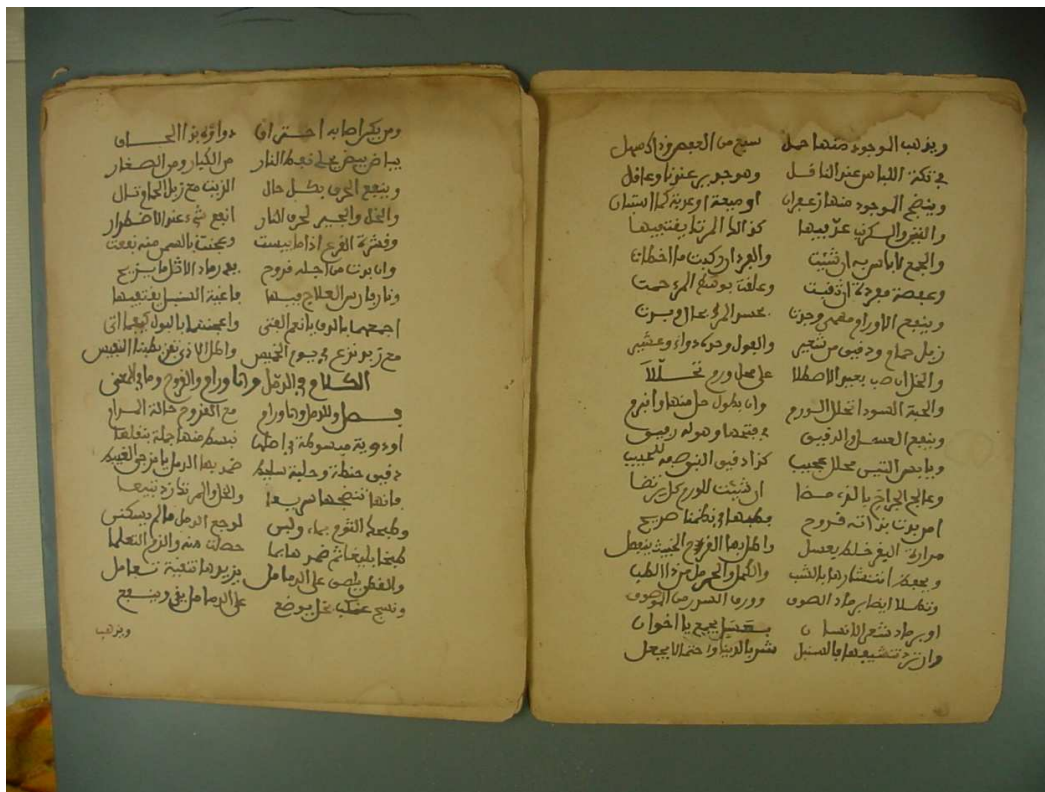


Fig.3.2. 7. Manuscrit1 : Image originale [15]



Fig.3.2.8. Manuscrit1 : Image de manuscrit binarisée avec la méthode du seuil fixe T=95[20]



Fig.3.2.9. Manuscrit2 : Image Originale [15]

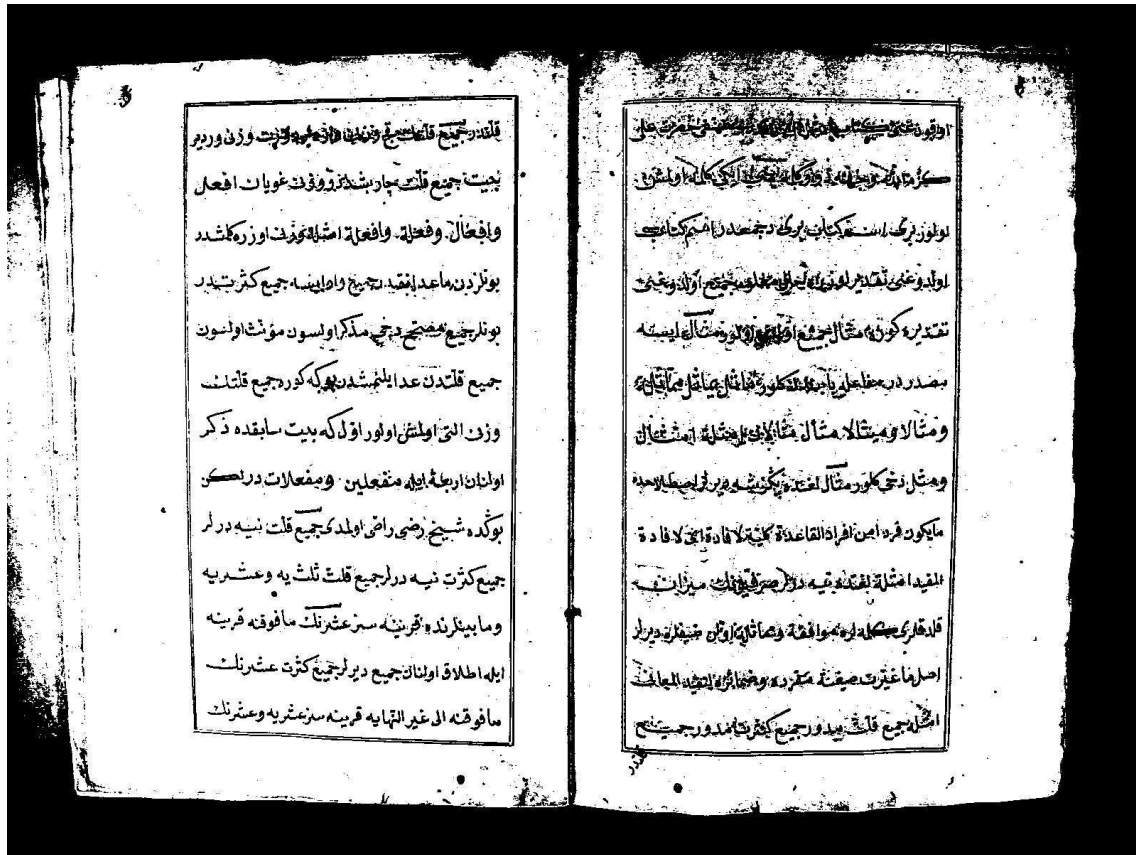


Fig.3.2.10.Manuscrit2 : Image de manuscrit binarisée avec la méthode du seuil fixe  $T=110$ [20]



Fig.3.2.11.Manuscrit3 : Image originale [15]



Fig.3.2.12. Manuscrit3 : image binarisée avec la méthode du seuil fixe  $T=110$  [20]

En se basant sur l'appréciation visuelle, nous constatons que les résultats de la méthode du seuil fixe montrent des résultats appréciables pour le manuscrit 1, 2 et En effet, elle présente une bonne séparation entre les objets et leurs fonds. En revanche, elle donne des mauvais résultats pour le manuscrit3. En effet, des détails de manuscrit sont supprimés (tel que la texture). Par conséquent, cette méthode donne des résultats satisfaisant dans le cas ou les manuscrits présentent des fonds plus au moins uniformes.

3.4. Image de manuscrits segmentés :

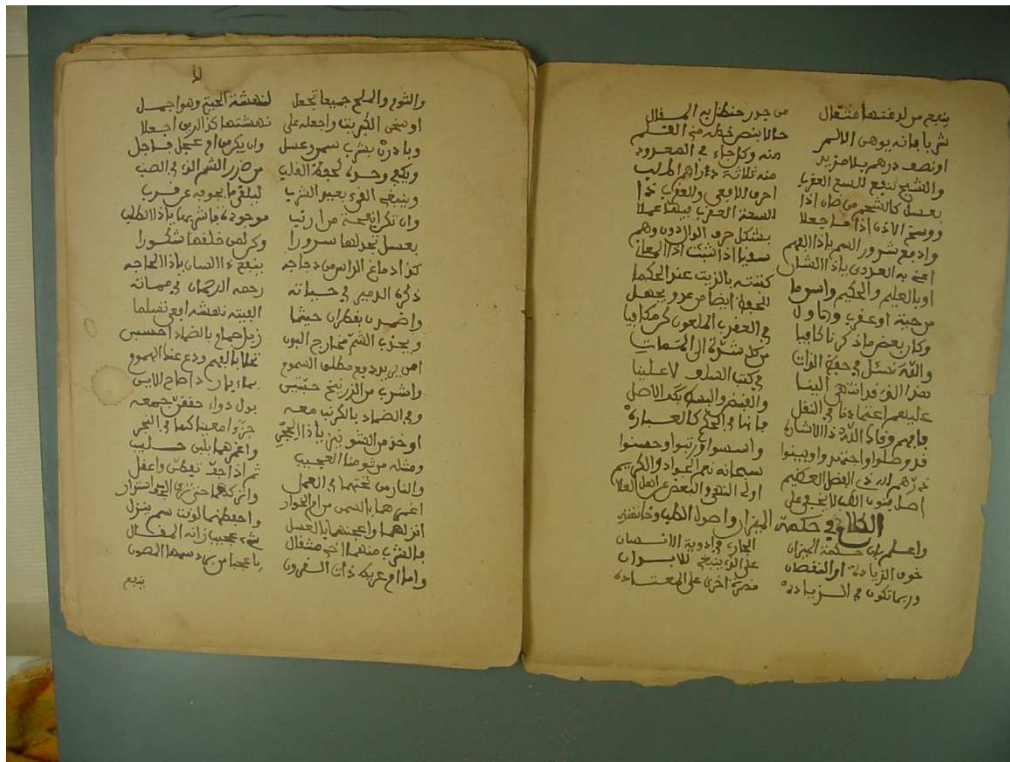


Fig.3.2.13. Image de manuscrit arabe [15]

a) En Contour avec la méthode de Canny:

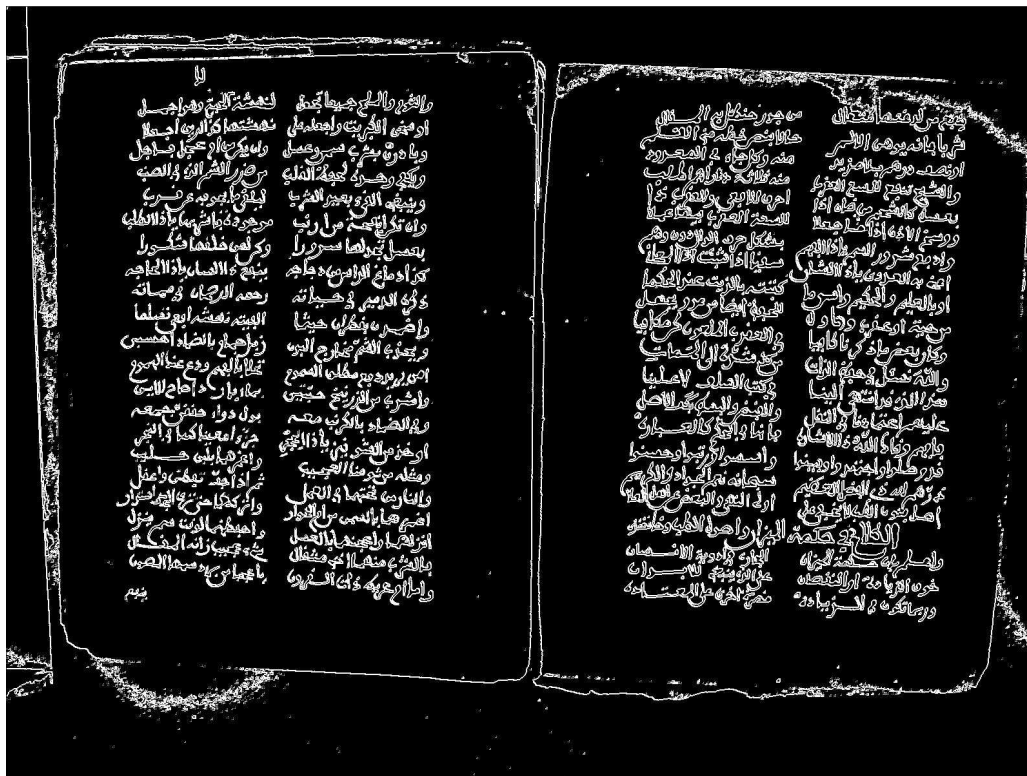


Fig.3.2.14. Résultat avec opérateur de Canny [20]

Les résultats montrent que l'opérateur de Canny produit des contours un peu épais mais une très bonne localisation de contours ainsi qu'une stabilité sur la détection de contours, ajouter à cela une résistance moyenne au bruit.

**b) En régions avec la méthode d'étiquetages des composantes connexes:**



Fig.3.2.15.Résultat avec procédure de segmentation en composantes connexes [20]

Les résultats que la segmentation par régions « Etiquetage des régions connexes », lorsqu'il y a chevauchement entre les régions, c'est-à-dire deux régions différentes sont considérées comme faisant partie d'une seule et unique région ce qui n'est pas le cas. Dans ces conditions, il faudrait trouver d'autres méthodes de segmentation plus appropriées, permettant de séparer plus exactement les régions connexes.

#### 4. Résultat obtenu pour la Transcription :

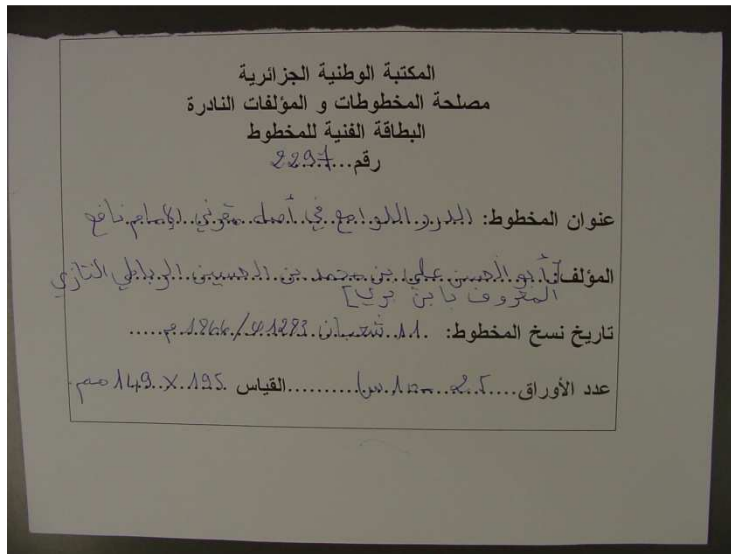


Fig.3.2.16. Image manuscrit originale

Après avoir saisi les informations sur l'éditeur, nous pouvons enregistrer le document sous format XML, nous avons aussi à notre disposition au niveau du bouton éditer le copier coller couper, également au niveau de format le gras italique si nécessaire, à partir de la nous pouvons convertir le fichier XML en HTML ou en PDF au choix.

Editeur\_XML\_TEI

Fichier Editer Format Convertir

رقم المخطوط: 105

عنوان المخطوط: الدرر اللوامع في أصل مقرني الإمام نافع

المؤلف: أبو الحسن علي بن الحسين الرباطي التازي المعروف بابن بري

تاريخ نسخ المخطوط: شعبان 11 1283 / 1966م

القياس: العرض 145، الطول 195

عدد الأوراق: 10-25

النص:

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ وَ صَلَّى اللَّهُ عَلَي سَيِّدِنَا وَ مَوْلَانَا مُحَمَّدٍ وَ سَلَّمَ  
 الْحَمْدُ لِلَّهِ الَّذِي أَوْثَقَنَا كِتَابَهُ وَ عَلَّمَهُ عَلَمَنَا  
 حَقْدًا يَدْوُمُ بِرَوَامٍ لِلْأَيِّدِ ثُمَّ صَلَاتُهُ عَلَي مُحَمَّدٍ  
 أَكْرَمَ مَنْ بَعَثَ لَأَنَامٍ وَ خَيْرَ مَنْ قَدَ فَامَ بِالْمَقَامِ  
 جَاءَ بِحَقِّهِ الْوَحْيِ وَ النَّبُوءَةُ لِحَيْرِ أُمَّةٍ مِنَ الْبَرِيَّةِ  
 صَلَّى عَلَيْهِ رَبُّنَا وَ سَلَّمَ وَ آءَالِهِ وَ صَحْبِهِ تَكْرَمًا  
 وَ بَعْدَ فَاعْلَمَ أَنَّ جِلْمَ الْقُرْآنِ أَجْمَلُ مَا بَعَثَ الْإِنْسَانَ  
 وَ خَيْرَ مَنْ عَلَّمَهُ وَ عَلَّمَهُ وَ اسْتَعْمَلَ الْفَقْرَ لَهُ وَ قِيَمَهُ  
 وَ جَاءَ فِي الْحَدِيثِ أَنَّ الْمَهْرَةَ فِي عِلْمِهِ مَعَ الْكِرَامِ الْبِرَّةُ

Fig.3.2.17. Exemple de manuscrit reproduit sur l'éditeur de texte

Firefox

Document sans nom

file:///C:/Users/mypc/Desktop/Ms\_description.html

Rechercher AVIRA WebGuard est activé Mise à jour d'Avira

المكتبة الوطنية الجزائرية

مصلحة المخطوطات و المؤلفات النادرة

الرقم: 105

عنوان المخطوط: الدرر اللوامع في أصل مقرني الإمام نافع

المؤلف: بري باين المعروف التازي الرباطي الحسين بن علي الحسن أبو

تاريخ نسخ المخطوط: م 1283 / 1966 شعبان 11

الأوراق عدد: 25-10

المقياس: 145مم x 195

النص:

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ وَ صَلَّى اللَّهُ عَلَي سَيِّدِنَا وَ مَوْلَانَا مُحَمَّدٍ وَ سَلَّمَ  
 الْحَمْدُ لِلَّهِ الَّذِي أَوْثَقَنَا كِتَابَهُ وَ عَلَّمَهُ عَلَمَنَا  
 حَقْدًا يَدْوُمُ بِرَوَامٍ لِلْأَيِّدِ ثُمَّ صَلَاتُهُ عَلَي مُحَمَّدٍ  
 أَكْرَمَ مَنْ بَعَثَ لَأَنَامٍ وَ خَيْرَ مَنْ قَدَ فَامَ بِالْمَقَامِ  
 جَاءَ بِحَقِّهِ الْوَحْيِ وَ النَّبُوءَةُ لِحَيْرِ أُمَّةٍ مِنَ الْبَرِيَّةِ  
 صَلَّى عَلَيْهِ رَبُّنَا وَ سَلَّمَ وَ آءَالِهِ وَ صَحْبِهِ تَكْرَمًا  
 وَ بَعْدَ فَاعْلَمَ أَنَّ جِلْمَ الْقُرْآنِ أَجْمَلُ مَا بَعَثَ الْإِنْسَانَ  
 وَ خَيْرَ مَنْ عَلَّمَهُ وَ عَلَّمَهُ وَ اسْتَعْمَلَ الْفَقْرَ لَهُ وَ قِيَمَهُ  
 وَ جَاءَ فِي الْحَدِيثِ أَنَّ الْمَهْرَةَ فِي عِلْمِهِ مَعَ الْكِرَامِ الْبِرَّةُ

Fig.3.2.18. Résultat du manuscrit transcrit en HTML

### **5. Conclusion:**

Dans ce qui a précédés nous avons présentés quelques exemples d'images de manuscrits numérisés et les résultats obtenus sur ces dernières avec l'outil de transcription que nous avons réalisé et nous pouvons constater qu'il donne des résultats appréciables.

# Conclusion Générale

**Conclusion Générale :**

Dans ce mémoire on a voulu présenter une solution pour la transcription de manuscrits arabes anciens, dont le but de les préserver de la détérioration et de l'effritement qui menaces leur existence.

Pour commencer ils doivent être numériser grâce aux scanner ou appareils photos numériques, ensuite plusieurs prétraitement seront effectuer sur ces images pour les débarrasser du bruit et améliorer leur qualité, ensuite on va appliquer un processus de segmentation pour segmenter le manuscrits en mots bien distincts car certains de ces manuscrits sont difficile a lire et a reproduire, la cause et du a leur styles ou calligraphie.

Après sa un transcripateur aura pour tâche de reproduire le manuscrit sur un éditeur a partir de l'image segmenter, qui sera enregistrer sous XML puis finalement sera afficher en HTML ou PDF aux choix grâce aux feuilles de styles XSLT et XSL-FO.

Bien évidemment cette solution a des avantages et des inconvénients :

**Avantages de la solution de transcription des manuscrits**

- La calligraphie de l'écriture arabe a subit beaucoup de modification. Peu de lecteurs peuvent déchiffrer le contenu de certains styles d'écriture. La transcription des manuscrits est un moyen efficace pour permettre aux lecteurs contemporains, d'accéder au contenu des manuscrits dont ils ne maîtrisent pas le style calligraphique.

Ce qui peut être un avantage supplémentaire, pour une initiation à tout type de calligraphie.

- L'accès au manuscrit grâce à une partie de son contenu est une option très recherchée.

En effet, une fois le manuscrit transcrit, un système de recherche plein texte sera facile à mettre en place.

- Le format texte des documents étant facilement téléchargeable, la transcription des manuscrits permettra de ce fait, un accès rapide au manuscrit proprement dit. En effet, la version texte peut être affichée dans un premier temps et l'image originale, pourrait être affichée dans un second temps, à la demande de l'utilisateur.

**Inconvénients de la solution de transcription des manuscrits**

Si la transcription des manuscrits permet leur valorisation, nous reconnaissons l'aspect fastidieux et long de cette solution, qui pourrait décourager le transcripateur. C'est pour cela que nous suggérons deux solutions intermédiaires en perspectives :

- Mise en place d'un outil de transcription selon un modèle collaboratif. Ce qui facilitera la tâche aux différents transcrip-teurs.
- Un système de transcription vocal pourra être envisagé. Ce qui facilitera la tâche au transcrip-teur, qui se limitera à lire le contenu du manuscrit à transcrire.

Annexe A :

Langage et Environnement de  
Développement

## **1. Environnement de travail :**

### **1.1. Langage de programmation JAVA :**

Le langage java est un langage de programmation orienté objet, mis au point par Sun Microsystems. Sa caractéristique principale est qu'il est indépendant de toute plate forme, il est possible d'exécuter des programmes java sur tous les environnements qui possèdent une java Virtual Machine (JVM), ce concept est à la base du slogan de Sun pour java : WORA (Write Once, Run Anywhere : écrire une fois, exécuter partout). Sun fournit aussi gratuitement un ensemble d'outils et d'API pour permettre le développement de programmes avec ce langage, ce kit est nommé JDK (Java Development Kit). Java est caractérisé aussi par la réutilisabilité de du code de ces programmes ainsi que la simplicité de leurs mise en œuvre.

### **1.2. Système d'exploitation :**

On a développé notre application, sous le système d'exploitation Windows 7, avec le langage de programmation orienté objet java, il peut être intégré dans n'importe quel autre système d'exploitation supportant la machine virtuel de java.

### **1.3. IDE éclipse :**

Eclipse est un projet open source fondé par Sun Microsystems, il est un environnement de développement permettant d'écrire, compiler, déboguer et déployer des programmes. Il est écrit en java, et il y'a un grand nombre de modules pour étendre l'IDE éclipse.

Pour le développement de notre application nous avons choisi Eclipse version Ganymede (son interface est donnée dans la figure 4.1). De part ces qualités en tant que langage évolué Eclipse permet un traitement simple et efficace sur des fichiers images en fournissant des API spécialisées, qui sont APIs standards et JAI (Java Advanced Imaging). Elles sont fournies dans des package comme java.awt et javax.imagio ...etc.

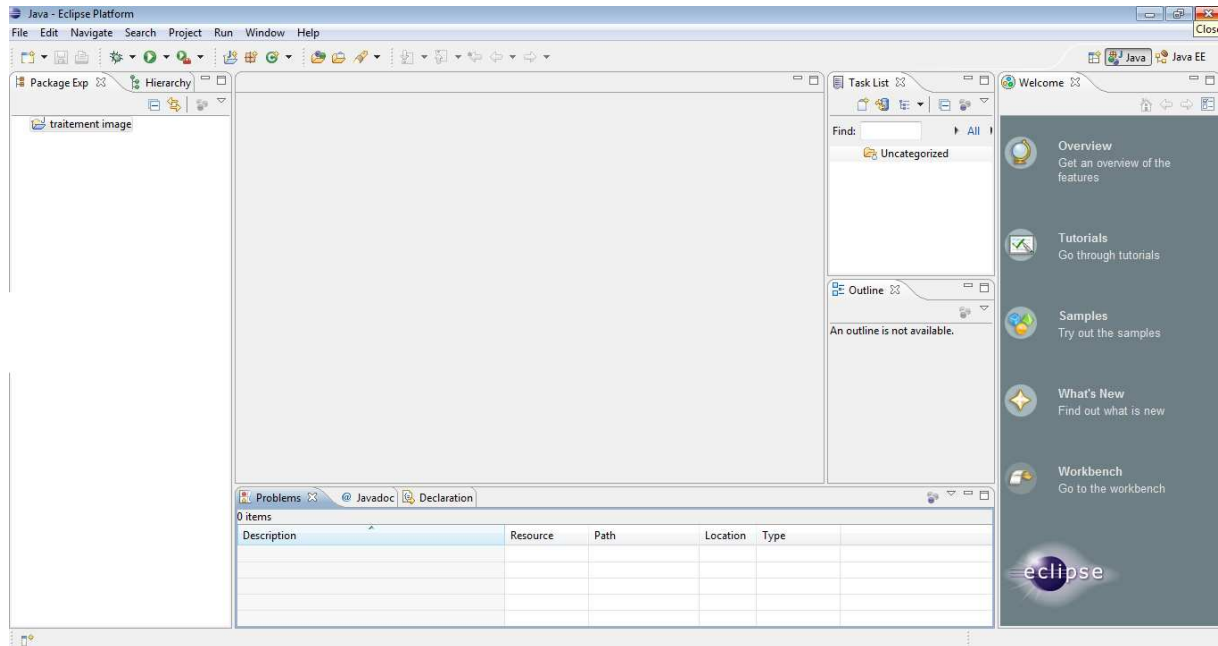


Fig.4.1.Interface principale d'éclipse

## 2. Java et le traitement d'image :

Le traitement d'image en java se fait grâce aux deux packages `java.awt` et `javax.imageio`.

### 2.1. Le Package `java.awt` :

AWT (Abstract Windowing Toolkit) est une collection de classes pour la réalisation d'applications graphiques ou GUI (Graphic User Interface).

Les composants qui sont utilisés par les classes définies dans ce package, sont des composants dits « lourds » : dépendent entièrement du système d'exploitation. D'ailleurs leur nombre est limité, car ils sont communs à plusieurs systèmes d'exploitation pour assurer la portabilité.

Cependant, la représentation d'une interface graphique avec awt sur plusieurs systèmes peut ne pas être identique.

AWT se compose de plusieurs packages dont les principaux sont :

- `java.awt` : c'est le package de base de la bibliothèque AWT
- `java.awt.image` : ce package permet la gestion des images
- `java.awt.event` : ce package permet la gestion des événements utilisateurs
- `java.awt.font` : ce package permet d'utiliser les polices de caractères

## 2.2. Le package `javax.imageio` :

Le package `javax.imageio` contient les classes de base et des interfaces pour décrire le contenu des fichiers images.

La classe `ImageIO` est une classe statique fournissant un certain nombre de services relatifs aux lectures/écritures des images. Les formats supportés par défaut sont : GIF, PNG, JPEG, BMP et WBMP. Les principales méthodes sont une série de `read(..)` et `write(..)` ainsi que `getReaderFormatNames()` et `getWriterFormatNames()` pour obtenir des informations sur les formats pris en charge, ainsi que « `IOException` » pour signaler les erreurs.

**Annexe B :**  
**Réseaux de neurones**

## 1. Introduction

Les réseaux de neurones sont composés d'éléments simples (ou neurones) fonctionnant en parallèle. Ces éléments ont été fortement inspirés par le système nerveux biologique. Comme dans la nature, le fonctionnement du réseau (de neurone) est fortement influencé par la connections des éléments entre eux. On peut entraîner un réseau de neurone pour une tâche spécifique (reconnaissance de caractères par exemple) en ajustant les valeurs des connections (ou poids) entre les éléments (neurone).

En général, l'apprentissage des réseaux de neurones est effectué de sorte que pour une entrée particulière présentée au réseau corresponde une cible spécifique. L'ajustement des poids se fait par comparaison entre la réponse du réseau (ou sortie) et la cible, jusqu'à ce que la sortie corresponde (au mieux ?) à la cible. On utilise pour ce type d'apprentissage dit superviser un nombre conséquent de pair entrée/sortie.

L'apprentissage « par paquet » (batch training) du réseau consiste à ajuster les poids et biais en présentant les vecteurs d'entrée/sortie de tout le jeu de données.

L'apprentissage « pas à pas ou séquentiel » (incremental training) consiste à ajuster les poids et biais en présentant les composantes du vecteur d'entrée/sortie les unes après les autres. Ce type d'apprentissage est souvent qualifié d'apprentissage « en ligne » (« on line » training) ou « adaptatif » (« adaptive » training).

L'apprentissage permet aux réseaux de neurones de réaliser des tâches complexes dans différents types d'application (classification, identification, reconnaissance de caractères, de la voix, vision, système de contrôle...). Ces réseaux de neurones peuvent souvent apporter une solution simple à des problèmes encore trop complexes ne pouvant être résolus rapidement par les ordinateurs actuels (puissance de calcul insuffisante) ou par notre manque de connaissances.

La méthode d'apprentissage dite superviser est souvent utilisée mais des techniques d'apprentissage non supervisé existent pour des réseaux de neurones spécifiques. Ces réseaux peuvent, par exemple, identifier des groupes de données (réseaux de Hopfield).

Les réseaux de neurones ont une histoire relativement jeune (environ 50 ans) et les applications intéressantes des réseaux de neurones n'ont vu le jour qu'il à une vingtaine d'année (développement de l'informatique).

L'objectif de ce cours (de 2 heures) est de présenter (sommairement) les grands types de réseaux de neurones, leurs applications et éventuellement les règles et algorithmes d'apprentissage. On s'attardera un petit peu plus sur le perceptron multicouche. Les figures de ce cours sont issues du livre Neural Network Toolbox fourni par le logiciel Matlab et des recherches effectuées au Laboratoire de Météorologie Physique.

### **-Historique**

Le champ des réseaux neuronaux va démarrer par la présentation en 1943 par W. McCulloch et W. Pitts du neurone formel qui est une abstraction du neurone physiologique. Le retentissement va être énorme. Par cette présentation, ils veulent démontrer que le cerveau est équivalent à une machine de Turing, la pensée devient alors purement des mécanismes matériels et logiques.

Une machine de Turing se résume à une tête de lecture comportant un nombre fini d'états internes et à un ruban. La puissance de l'analyse de Turing (1912-1954) tient au fait que sa tête de lecture ne lit qu'un seul symbole à la fois, et que cette lecture, associée à la table d'états adéquate, suffit à effectuer toutes les opérations possibles. La Machine de Turing est toutefois une machine imaginaire, abstraite, et idéale. Elle n'a pu être construite. Son fonctionnement implique en effet d'avoir un ruban extensible à volonté donc infini! La combinaison de cette mémoire infinie et d'un nombre d'états fini a cependant apporté la preuve que, si un problème est calculable, alors il existe une machine pour le résoudre. Dès lors, une MACHINE, reprenant les règles de toutes les autres, est UNIVERSELLE ».)

En 1949, D. Hebb présente dans son ouvrage « The Organization of Behavior » une règle d'apprentissage. De nombreux modèles de réseaux aujourd'hui s'inspirent encore de la règle de Hebb.

En 1958, F. Rosenblatt développe le modèle du perceptron. C'est un réseau de neurones inspiré du système visuel. Il possède deux couches de neurones : une couche de perception et une couche liée à la prise de décision. C'est le premier système artificiel capable d'apprendre par expérience.

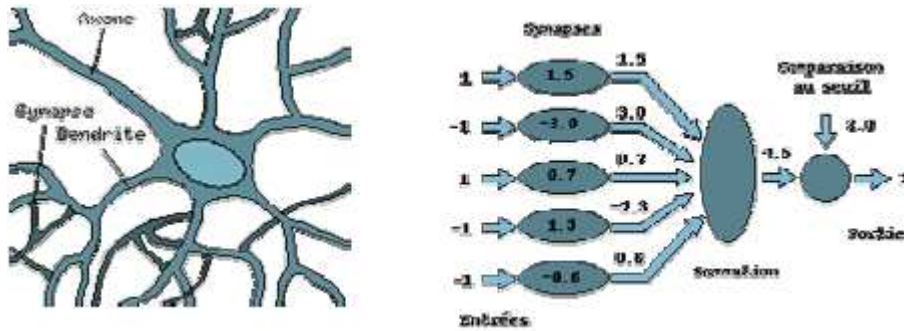
Dans la même période, Le modèle de L'Adaline (ADAPtive LINar Element) a été présenté par B. Widrow et Hoff. Ce modèle sera par la suite le modèle de base des réseaux multicouches.

En 1969, M. Minsky et S. Papert publient une critique des propriétés du Perceptron. Cela va avoir une grande incidence sur la recherche dans ce domaine. Elle va fortement diminuer

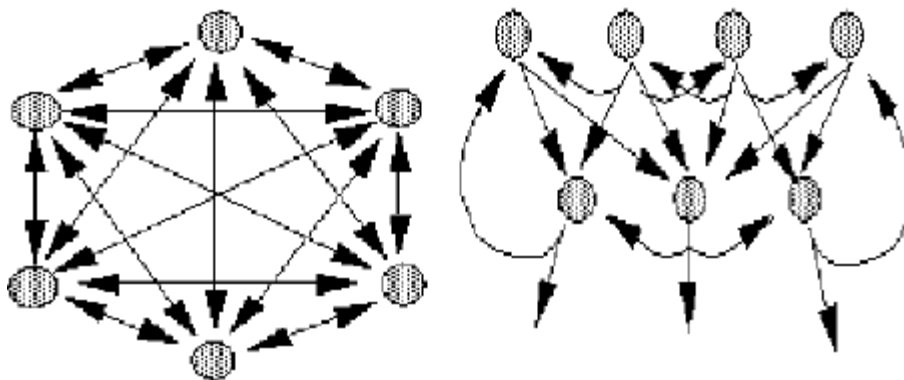
jusqu'en 1972, où T. Kohonen présente ses travaux sur les mémoires associatives et propose des applications à la reconnaissance de formes.

C'est en 1982 que J. Hopfield présente son étude d'un réseau complètement rebouclé, dont il analyse la dynamique.

**-Analogie neurone formel et neurone biologique**

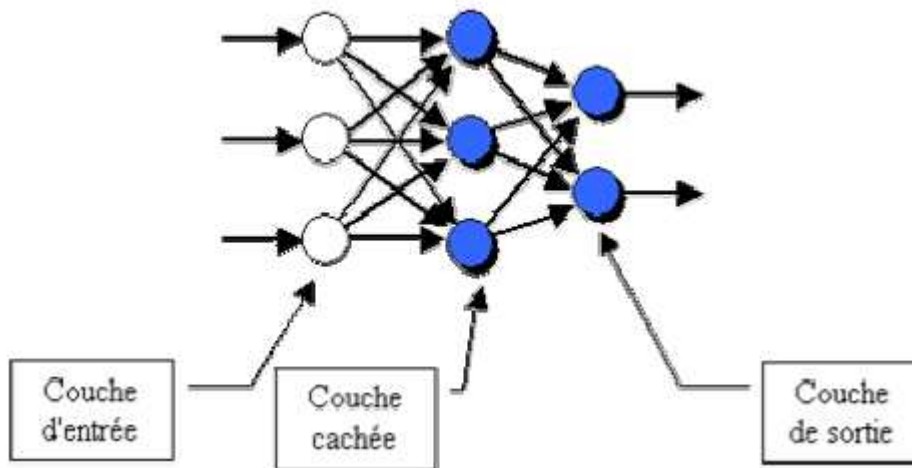


**-Exemple de réseaux de neurones :**



Réseau entièrement bouclé (carte topologique)

Réseaux avec connections arrière (réseaux récurrent ou feedback)



Réseau « ordonné » sans connexion arrière (réseau feed-forward)

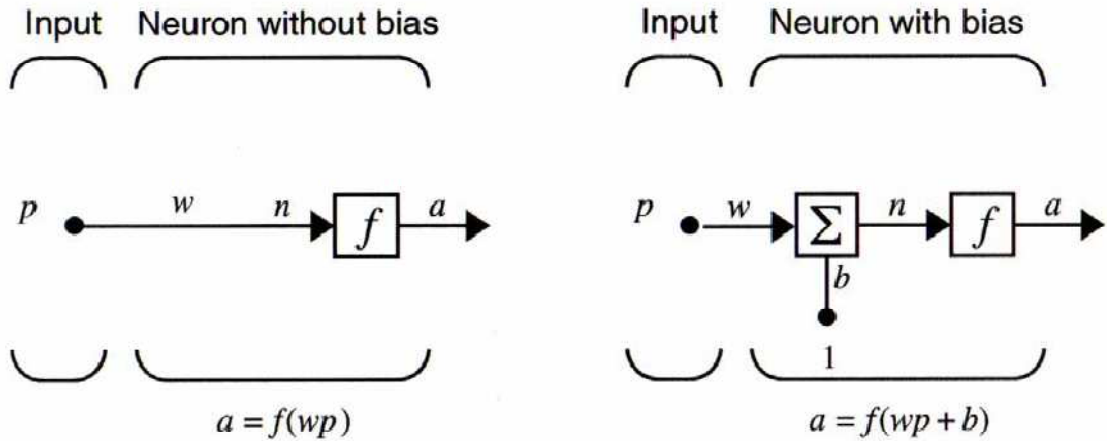
#### **-Applications**

- Aérospatial : pilotage automatique, simulation du vol...
- Automobile : système de guidage automatique,...
- Défense : guidage de missile, suivi de cible, reconnaissance du visage, radar, sonar, lidar, traitement du signal, compression de données, suppression du bruit...
- Electronique : prédiction de la séquence d'un code, vision machine, synthétiseur vocal, modèle non linéaire,...
- Finance : Prévission du coût de la vie
- Secteur médical : Analyse EEC et ECG
- Télécommunications : Compression de données ...

## 2. Caractéristiques des réseaux de neurones artificiels

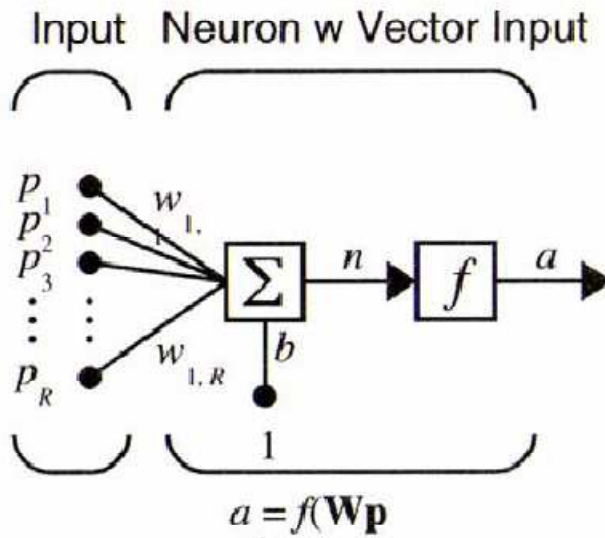
### 2.1. Poids (w) et biais (b) d'un neurone

#### Simple Neuron



### 2.2. La fonction de sommation

#### Neuron With Vector Input



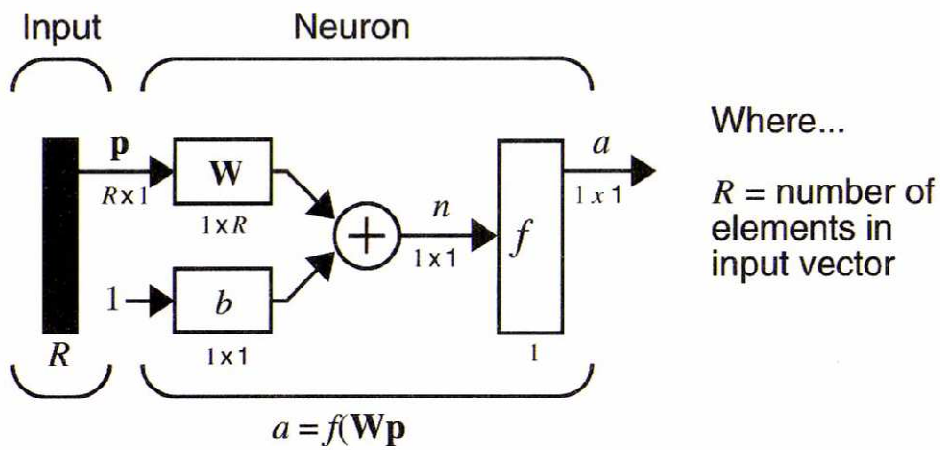
Where...

$R$  = number of elements in input vector

#### Net Input

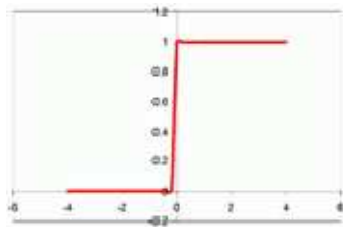
$$n = w_{1,1}p_1 + w_{1,2}p_2 + \dots + w_{1,R}p_R + b$$

## Single Neuron Using Abbreviated Notation

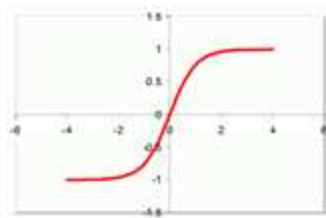


L'entrée d'un neurone contient le biais et la somme des ses poids. La sortie d'un neurone dépend de l'entrée du neurone et de sa fonction de transfert  $f$ .

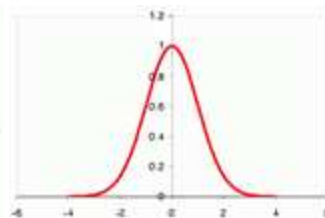
### 2.3. La fonction de transfert




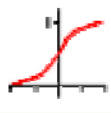

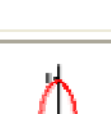
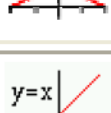
Fonction à seuil



Tangente hyperbolique



Fonction Gaussienne

Pas unitaire		$f(x) = \begin{cases} 0 & \text{if } 0 > x \\ 1 & \text{if } x \geq 0 \end{cases}$
Sigmoïde		$f(x) = \frac{1}{1+e^{-\beta x}}$
Linéaire Seuillée		$f(x) = \begin{cases} 0 & \text{if } x \leq x_{min} \\ mx+b & \text{if } x_{max} > x > x_{min} \\ 1 & \text{if } x \geq x_{max} \end{cases}$
Gaussienne		$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
Identité		$f(x) = x$

#### 2.4. Réseaux statiques (Apprentissage par « paquet ») et Réseaux dynamiques (Apprentissage « séquentiel »)

Le format des données d'entrée va affecter l'évolution des poids et biais des réseaux de neurones lors de leur apprentissage.

Un réseau dit statique est un réseau qui ne contient pas de connexion arrière (feedback or delay). Par conséquent, on peut lui présenter les données en entrée dans n'importe quel ordre, cela n'influencera pas l'évolution de ses poids lors de la phase d'apprentissage. Il est alors préférable de lui donner tout le jeu de donnée en un seul coup lors de la phase d'apprentissage. On parle alors d'apprentissage par paquet (« batch training »). Les réseaux « feedforward » ne peuvent pas simuler des processus dépendant du temps.

Par contre, si l'on veut simuler un processus qui dépend du temps, alors on pourra utiliser un réseau de neurones contenant des connexions arrières. L'ordre de présentation du jeu de données au réseau de neurone sera alors primordial. On parle alors d'apprentissage séquentiel.

# Bibliographie

# Bibliographie

- ❖ [1] : J.P.COQUEREZ, S.PHILIP. « Analyse d'image filtrage et segmentation ».Edition Masson. 1995.
- ❖ [2] : M.BERGOUNIOUX. « Méthodes mathématiques pour le traitement d'image ». Cour master université de Pierre et Marie Curie. 2009
- ❖ [3] : Antoine MANZANERA. « Les images numériques »
- ❖ [4] : Encyclopédie universelle Wikipedia. «Définition de pixel ». Disponible sur le web à l'adresse <http://fr.wikipedia.org/wiki/Pixel>. consulter Avril 2011.
- ❖ [5] : Encyclopédie universelle Wikipedia. «Définition de contour ». Disponible sur le web à l'adresse <http://fr.wikipedia.org/wiki/Contour>. consulter Avril 2011.
- ❖ [6] : C. KADDOUR, S.AISSA BRAHIM, « Compression des images Fixes par fractales basée sur la triangulation de Delaunay et la quantification vectorielle ». Université des sciences et techniques Houari Boumediene. Disponible sur <http://www.kaddour.com> .
- ❖ [7] : Encyclopédie universelle Wikipedia. «Définition du contraste ». Disponible sur le web à l'adresse <http://fr.wikipedia.org/wiki/Contraste>. consulter Avril 2011.
- ❖ [8] : Encyclopédie universelle Wikipedia. «Définition du niveau de gris ». Disponible sur le web à l'adresse [http://fr.wikipedia.org/wiki/niveau\\_de\\_gris](http://fr.wikipedia.org/wiki/niveau_de_gris). consulter Avril 2011.
- ❖ [9]Crousseau. Article disponible sur le Web à l'adresse : <http://crousseau.free.fr/imgfmt.htm> consulté Avril 2011.

- ❖ [10] : Encyclopédie universelle Wikipedia. «Histogramme d'image». Disponible sur le web à l'adresse [http://fr.wikipedia.org/wiki/Histogramme\\_d'images](http://fr.wikipedia.org/wiki/Histogramme_d'images) consulter Avril 2011.
- ❖ [11] : Abderahman Kefali, Toufik Sari, Mokhtar Sellami, « Evaluation de plusieurs techniques de seuillage d'images de documents arabes anciens » Laboratoire de recherche en informatique, Département informatique, B12, Université de Annaba, 2300, Algérie.
- ❖ [12] : Encyclopédie universelle Wikipedia. «Le manuscrit : Notion générales sur les manuscrits». Disponible sur le web à l'adresse <http://fr.wikipedia.org/wiki/manuscrit> . consulter Avril 2011.
- ❖ [13] : Prétraitement de documents anciens. Disponible sur le web à l'adresse : <http://hal.archives-ouvertes.fr/docs/00/33/44/26/PDF/paper-45.pdf> .
- ❖ [14] : Paola Orsatti. « le manuscrit islamique : caractéristiques matérielles et typologie (Ancient and medieval book materials and techniques- Città del Vaticano : Biblioteca Apostolica Vaticana), 1993, PP. 269-331
- ❖ [15] : Corpus d'image provenant de la bibliothèque nationale d'EL-HAMMA.
- ❖ [16] : Farès Menasri<sup>1</sup> – Nicole Vincent<sup>1</sup> – Emmanuel Augustin<sup>2</sup> – Mohamed Cheriet<sup>3</sup>. « Un système de reconnaissance de mots arabes manuscrits hors-ligne sans signes diacritiques ». <sup>1</sup>Université PARIS DESCARTES - UFR de Mathématiques et Informatique 45 rue des Saints Pères, 75006 Paris France. <sup>2</sup> A2iA SA 40 bis rue Fabert, 75007 Paris – France. <sup>3</sup> ETS Montréal - Laboratoire LIVIA 1100 rue Notre-Dame ouest, Montréal, QC, H3C 1K3 – CANADA
- ❖ [17] : Jean-Pierre COCQUERZ et Sylvie PHILIPP. « Analyse d'image filtrage et segmentation ». Masson. Paris 1995.

- ❖ [18] : Stéphane BRES, Jean Michel Jolion et Franck LE BOURGOIS. « Traitement et analyse d'images numériques ». Lavoisier, 2003
- ❖ [19] : Michel FOUQUS et Roger SAMY. «Algorithmes de traitement d'images et réseaux systolique ». CETIA : centre d'Etudes des Techniques de l'Informatique et d'Analyse de Systèmes.
- ❖ [20] : Résultat obtenu avec l'application Transcription Manuscrits Arabes anciens développé au cours de ce mémoire.
- ❖ [21] : Jean Jaques Rousselle « «Les contours Actifs, une méthode de segmentation, Application à l'imagerie médicale ». Thèse de doctorat2003. Université François Rabelais de Tours.
- ❖ [22] : Détection de contours disponible sur le web a l'adresse : <http://perception.inrialpes.fr/people/Boyer/Teaching/M2PGI/c3.pdf>
- ❖ [23] : Sébastien Mavromatis Olivier Coulon. « Analyse d'images Détection de contours » Laboratoire des Sciences de l'Information et des Systèmes UMR CNRS 6168.
- ❖ [24] : Filtrage bidimensionnel disponible sur le web a l'adresse [http://fr.wikipedia.org/wiki/Lissage\\_de\\_l%27image#Filtre\\_gaussien](http://fr.wikipedia.org/wiki/Lissage_de_l%27image#Filtre_gaussien)
- ❖ [25] : Segmentation en régions par Xavier Philippeau disponible sue le web a l'adresse <http://xphilipp.developpez.com/articles/segmentation/regions/>
- ❖ [26] : J.J. TOUMAZET. « Traitement de l'image sur micro-ordinateur ». Sybex 1987.
- ❖ [27]: Imane Sebari, Dong Chen He. “ Les approches Coopérative d'image Multi composantes: régions-contours”. Université de Sherbrooke Québec 2007.

- ❖ [28] : Algorithme Hilditch disponible sur le web à l'adresse : <http://cgm.cs.mcgill.ca/~godfried/teaching/projects97/azar/skeleton.html>
- ❖ [29] : Xavier Dupré. Thèse pour obtenir le grade de Docteur de l'Université René Descartes Discipline : Informatique. « Contributions à la reconnaissance de l'écriture cursive à l'aide de modèles de Markov cachés ». Thèse soutenue le 15 janvier 2004.
- ❖ [30] : □Yong Haur Tay, □Marzuki Khalid, □Stefan Knerr, □Pierre-Michel Lallican, □Christian Viard-Gaudin
  - Centre for Artificial Intelligence & Robotics (CAIRO), Universiti Teknologi Malaysia, Jalan Semarak, 54100 Kuala Lumpur, Malaysia. □yhtay@mail.cc.utm.my, □marzuki@utmnet.utm.my
  - Vision Objects SARL, 11, Rue de la Fontaine Caron 44300 Nantes, France.
  - Laboratoire SEI/EP CNRS 63, IRESTE, Université de Nantes, Rue Christian Pauc, La Chantrerie, BP 60601, 44306 Nantes Cedex 3, France. « Two-Cost Stroke Segment Grouping Mechanism for Off-line Cursive Hand-written Word Recognition »
- ❖ [31] : Sargur Srihari, Harish Srinivasan, Pavithra Babu and Chetan Bhole Center of Excellence for Document Analysis and Recognition (CEDAR) University at Bu@alo, State University of New York Amherst, New York 14228 srihari @cedar.bu@alo.edu. « Handwritten Arabic Word Spotting using the CEDARABIC Document Analysis System »
- ❖ [32] : La transcription disponible sur le web a l'adresse <http://www.mediadico.com/dictionnaire/definition/transcrire/1>
- ❖ [33] : Mémoire de Magister en Informatique Option : Ingénierie des Systèmes d'Informations et du Document Electronique ISIDE SOUALAH Mohammed Ou Rabah Thème : « Numérisation des manuscrits arabes : Catalogage et accès multilingue »
- ❖ [34] : Lou Burnard. « La TEI c'est quoi? »
- ❖ [35] : Florence Clavaud. « TP2 : transcription d'un manuscrit ». École nationale des chartes