République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 – Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique

Mémoire de Fin d'études Master

Filière : Informatique

Option : Systèmes informatique

Thème :

Segmentation de relevés de notes du baccalauréat Algérien

Encadré Par :

Dr. Abderrahmane KEFALI

Présenté par :

Ahlem OBEIZI

Juillet 2019

Résumé

La numérisation et la dématérialisation des archives est une tendance actuelle d'un grand nombre d'organisations et d'administrations. En effet, la simple numérisation permet de transformer un document papier en une image numérique mais elle n'est pas suffisante pour répondre aux besoins des organisations. Elle doit être accompagnée par des techniques facilitant leur analyse automatique.

Le présent travail s'inscrit dans ce contexte. Nous nous intéressons ici à analyser un type particulier de documents, à savoir les relevé de notes du baccalauréat (bac) algérien, qui constituent une des pièces les plus importants dans le dossier de l'étudiant, afin de reconnaitre leur structure physique, c'est à dire d'en extraire les différentes entités et informations constituant ces documents.

Cependant, une approche de segmentation des relevés de notes du baccalauréat a été proposée ici. L'approche proposée est une approche de segmentation hybride. Elle commence par l'application de plusieurs prétraitements afin d'améliorer la qualité des documents d'entrée. Après, l'algorithme RLSA est utilisé pour séparer la bordure du relevé car il n'a pas d'importance. Ensuite, une première détection des lignes du texte est effectuée et les lignes détectées sont regroupées en blocs par application de nouveau de l'algorithme RLSA. Les blocs textuels sont ensuite identifiés en se basant sur l'analyse des profils de projection et leurs lignes sont extraites. Finalement, les blocs non textuels sont réparties en tableaux ou graphiques à l'aide de la transformée de Radon.

Plusieurs tests ont été effectués sur un corpus local d'images de test afin d'évaluer les performances du système développé et les résultats obtenus sont encourageants.

Le travail présenté dans ce mémoire fait l'objet d'une communication, et l'article a été sélectionné parmi les meilleurs papiers et suggéré pour publication dans une revue internationale

Dédicace

Je dédie ce Travail :

A ma mère qui m'a assurée un soutien et encouragement inconditionnel

sans lequel je n'aurais jamais pu terminer mes années d'études

A mon père, pour m'avoir donné la possibilité de faire ce que

Je voulais et pour son soutien, sa patience et sa compréhension tout au long de

mes études.

A mon frère ***Oussama*** *et mes sœurs* ***Soulafa*** *et* ***Alaa***

Et sur tous pour ma grande famille mes tantes, oncles et cousins et cousines

A tous mes Chère copines et plus spécialement ***Imen*** *et* ***Abir***

A tous mes camarades de la promos 2018-2019

A tous le cadre du département E8

Ahlem

Remerciements

"الشكر و الحمد لله"

Merci Allah de m'avoir donné la capacité d'écrire et de réfléchir, la force d'y croire, la patience d'aller jusqu'au bout du rêve.

Je tiens à remercier vivement mon encadreur, Dr. Abderrahmane KEFALI, de m'avoir encadré pour réaliser ce travail, pour ses précieux conseils et de m'avoir donné le meilleur de son savoir et aide.

Je remercie également les membres de jury de me faire l'honneur de juger mon travail. Je remercie profondément toutes les profs du département d'informatique et toutes les personnes qui ont contribuées à l'élaboration de ce travail.

Finalement, je remercie ma grande famille, mes amies, mes collègues de l'université 08 mai 1945 Guelma et toute la promotion 2018/2019 de l'informatique.

Ahlem

Table de matières

Table de figures

Chapitre 01

Chapitre 02

Chapitre 03

Chapitre 04

Liste des tableaux

Chapitre 04

Introduction générale

Le 21ème siècle a vu un grand développement des technologies de l'information et le besoin accru de communication. Ainsi, même si l'utilisation de l'électronique est devenu populaire, l'utilisation du papier traditionnel reste inchangée et la production de documents papiers continue d'augmenter.

Si nous jetons aujourd'hui un coup d'œil à nos institutions tels que les organismes, les bibliothèques, les administrations, et les grandes entreprises…etc., nous trouvons qu'elles traitent chaque jour un grand volume de documents imprimés ou manuscrits, émanant de leurs citoyens et de leurs clients.

Cependant, bien que le papier reste le support primordiale de transformation et de sauvegarde de l'information, son utilisation apporte certaines difficultés; le stockage des documents papiers (l'archivage) est couteux en espace, la recherche manuelle dans ces documents peut demander un nombre incalculable d'heures de travail, et au fil du temps, les documents peuvent être endommagés, perdus et déchirés et leurs informations peuvent être perdues, surtout lorsqu'ils ne soient pas préservés dans des bonnes conditions d'archivage.

Un besoin urgent pour l'humain à relever plusieurs défis pour réduire les charges, diminuer les temps de traitement, réduire les erreurs, et fournir de nouveaux services capables de partager rapidement des informations électroniques dans un contexte Internet ou mobile.

Une solution consiste à transformer le flux de papiers en un flux de documents électroniques grâce à la numérisation. Cependant la numérisation permet de transformer un document papier en une image numérique stockée en format électronique, c'est bien, mais elle ne permet pas d'accéder au contenu des images numérisées, ni de les modifier ou de rechercher des informations dedans. De ce fait la simple numérisation n'est pas suffisante, elle doit accompagnée par des techniques informatiques permettant le traitement, l'analyse et la compréhension des images numérisées.

Cela a conduit à l'apparence de nouveaux domaines de recherche tel que l'analyse et la compréhension des documents et la reconnaissance des éléments qu'ils contiennent. Parmi les applications de l'analyse, la reconnaissance, et la compréhension de documents nous citons: la reconnaissance de caractères (OCR pour Optical Character Recognition) et des mots, la localisation des tableaux, la séparation texte/graphique, l'identification de scripteur, la reconnaissance de la langue et la fonte, etc.

En effet, les éléments constituant un document sont organisés en structures qui porte des informations sur le contenu du document, et pour réaliser un système permettant l'analyse, la reconnaissance, l'indexation et la recherche, et la classification automatique de documents, il faut d'abord reconnaitre la structure de ces documents.

Le présent projet de fin d'étude s'inscrit dans ce contexte. Nous nous intéressons ici à analyser un type particulier de documents, à savoir les relevé de notes du baccalauréat (bac) algérien, qui constituent une des pièces les plus importants dans le monde universitaire, afin de reconnaitre leur structure physique et logique. Cette reconnaissance nous permettra de segmenter ses relevés de notes et d'en extraire les différentes entités et informations constituant ces documents (entête, titre, matricule de l'étudiant, branche et année d'étude, tableau de notes, …etc.).

L'objectif de notre travail à moyen terme est de séparer les différentes informations contenues dans les relevés de bac. Cela pourra être une entrée pour d'autres systèmes effectuant des traitements ou des analyses sur des parties particulières de l'image contenant les informations souhaitées au lieu de les effectuer sur l'image entière. On pourra par exemple indexer le relevé par sa matricule (après reconnaissance), extraire la moyenne de l'étudiant, classifier les relevées selon la branche d'étude, etc. L'objectif à long terme est la construction d'un réel système d'analyse et de reconnaissance des relevés de bac qui intègre plusieurs fonctionnalités: acquisition, compression, prétraitements, analyse et reconnaissance, recherche, etc. Ce système permettra sans doute de faciliter le travail des agents dans les services de scolarité et d'archive de l'université.

Le présent mémoire est organisé en quatre chapitre comme suit:

Le ***premier chapitre*** présente un vue général sur l'analyse et la reconnaissance de document. Ce chapitre commence par la définition des notions de document et de structure. Ensuite, il aborde la reconnaissance de document et les étapes qui la composent.

Le ***deuxième chapitre*** présente un état de l'art sur le domaine de segmentation ou d'analyse de la structure physique d'images de documents. Nous commençons le chapitre par quelques définitions de la reconnaissance de structure physique. Le reste du chapitre est consacré à la présentation des différentes approches et méthodes proposées, et des divers travaux effectués dans la littérature pour la reconnaissance des structures physiques.

Le ***troisième chapitre*** présente une conception générale de notre travail. Nous décrivons d'abord les caractéristiques et la structure des relevés de notes du baccalauréat utilisées tout au long de ce travail. Nous décrivons par la suite la démarche suivie tout en détaillant les différentes étapes inclues et les méthodes utilisées.

Le ***quatrième chapitre*** est consacré à l'implémentation et l'expérimentation de notre système. Nous décrivons les tests effectués et les résultats obtenus.

Chapitre 1.

Document et structures

1. Introduction

L'homme a utilisé les documents papiers depuis longtemps pour conserver et transmettre de l'information. Aujourd'hui, les types de documents sont très nombreux : les factures, les journaux, les livres,...etc. L'évolution du masse de documents papiers a causé des nouveaux besoins: la préservation des documents qui a conduit vers l'utilisation de documents numériques et à la conversion des documents existants par numérisation, et l'extraction du contenu des documents qui a donné naissance à un nouveau domaine de recherche en traitement d'images, à savoir l'ARD (Analyse et Reconnaissance de Document).

Le présent chapitre présente un vue général sur l'analyse et la reconnaissance de document. Il commence par la définition des notions de document et de structure. Ensuite, il aborde la reconnaissance de document et les étapes qui la composent, avant de conclure.

2. Vue d'ensemble du document

2.1. Qu'est ce qu'un document ?

Le domaine documentaire est très vaste et les articles qui exposent ce domaine sont nombreux, mais seulement quelques-uns d'entre eux donnent des définitions pour ce terme.

Selon le dictionnaire LE PETIT ROBERT : « un document est tout ce qui sert à instruire, à savoir : tout écrit servant de preuve ou de renseignement, tout ce qui sert de preuve de témoignage, toute pièce qui permet d'identifier une marchandise en cours de transport ».

Une définition plus générale donnée dans [HAD 06], indique qu'un document peut avoir plusieurs types (textuel, sonores, vidéo, graphique…etc.) selon le support choisi. Pour lui « Un document est le support physique pour conserver et transmettre de l'information ».

Du point de vue informatique : « un document est généralement tout ce que l'on produit, distribue, utilise ou garde lors d'un processus de communication écrite ou électronique. Par conséquent un document peut être qualifiée de physique ou électronique » [AZO 95].

2.2. Document électronique

Un document électronique est la représentation d'un document, sous la forme d'une structure de données stockée en mémoire ou sur un support informatique, transmissible entre ordinateurs. Dans un système informatique. La principale caractéristique d'un document électronique est sa facilité de modification. Tout document électronique est modifiable ; différentes opérations y sont applicables ; parmi celles-ci, nous citons : copiage d'un support à un autre (diffusion sans limite), l'impression, et l'édition [HAD 06].

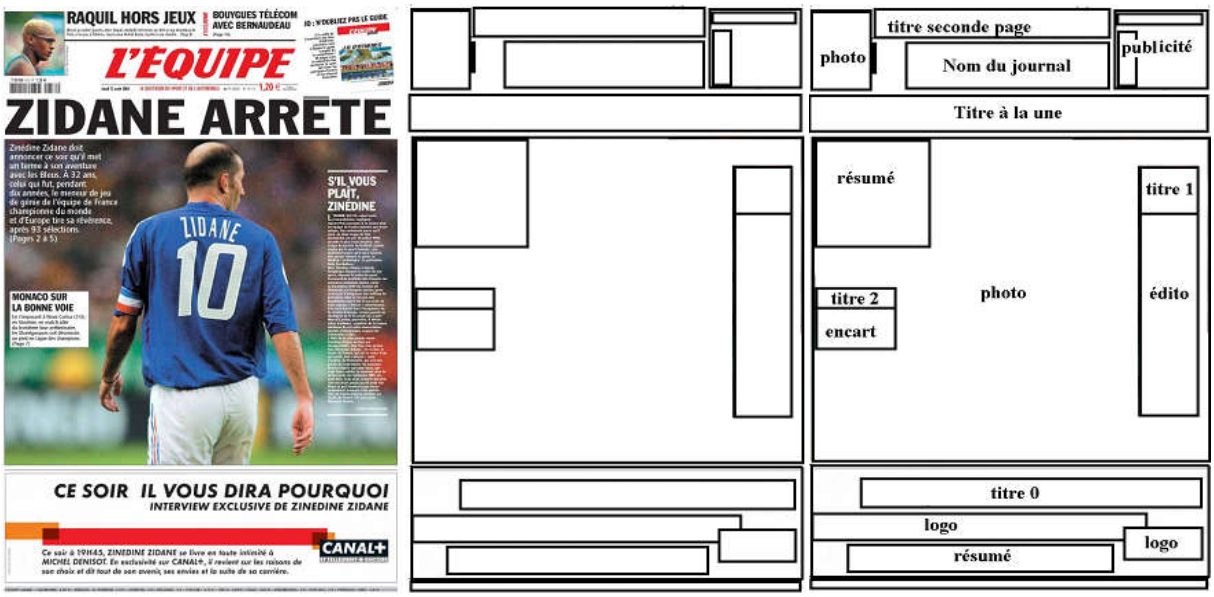
3. Notion de structure

Tout document textuel est construit selon une structure reconnue par les lecteurs grâce à des marques typographiques, des conventions de mise en page, des connaissances culturelles relatives aux informations contenues dans tout document [MIC 00].

Pour un document, le terme « structure » désigne l'organisation du document en blocs, niveaux…etc. et leurs relations. On peut ainsi indiquer que la structure d'un document se présente en deux niveaux : la structure *physique* et la structure *logique*.

3.1. Structure physique

La structure physique représente l'aspect tangible du document. C'est à dire qu'il s'agit d'une réunion d'objets physiques **[DUO 05].** Azokly [AZO 95] considère que la structure physique décrit au moyen d'entités physiques l'organisation hiérarchique des blocs (qui servent à structurer l'aspect graphique d'un document) composant les pages d'un document (entête, bloc, contenu, colonne, ligne...). La ***Figure 1.1.b*** illustre un exemple de structure physique.



(a) Page de journal *(b) sa structure physique* *(c) sa structure logique*

Figure 1.1 : *Structure physique et logique d'une page de journal [JOU 07].*

3.2. Structure logique

La structure logique d'écrit l'organisation hiérarchique du texte contenu dans un document au moyen d'entités logiques telles que les chapitres, les sections, les titres, les paragraphes, les notes, les citations, les formules, les tableaux, les cellules, ou les graphiques [AZO 95].

La structure logique découpe un document en parties, pour donner des éléments dotés d'un type. Pour chaque catégorie de documents, on trouve un ensemble de types. Pour un livre, par

exemple, les types utilisés seront "Titre", "Auteur", "Chapitre" , "Paragraphe", etc. [DUO 05].

La figure 1.1.c présente un exemple d'une structure logique.

3.3. Lien entre structure physique et logique

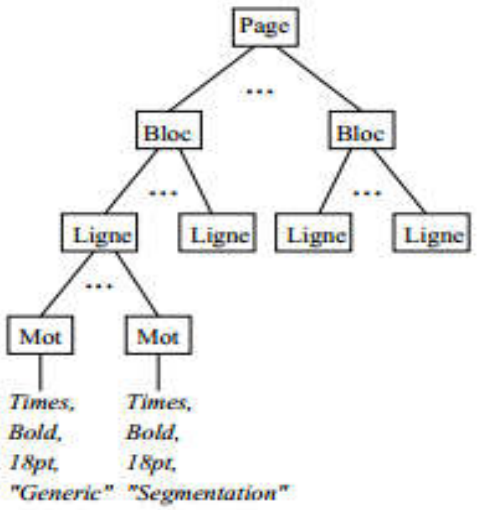
D'après [DUO 05] les structures physique et logique sont deux points de vue différents d'un document et qui n'ont aucun rapport entre eux mais dans certaine mesure ces deux structures sont mutuellement conditionnées. Il donne ainsi un exemple de l'entité logique "Chapitre", l'apparition de cette entité dans la plupart des manuels s'accompagne d'un changement de page et cela veut dire la création d'un élément physique "Page". Un simple changement dans cette règle conduit l'utilisateur à juger la mise en page du document incorrectement.

3.4. Représentation des structures

3.4.1. Représentation de la structure physique

La structure physique d'un document peut être représentée de diverses manières, indépendamment ou conjointement à la structure logique du document [TRU 05][ROB 01].

- En utilisant des paramètres de présentation du document tels que les dimensions et les distances entre les objets contenus dans le document.
- Par un arbre pour transcrire les liens hiérarchiques visibles qui existent entre les objets (*exemple* : un mot fait partie d'une ligne dans le cas de la structure physique).
- En se basant sur des grammaires formelles, ce qui permet d'offrir l'avantage de limiter les types de production qui peuvent être utilisés.
- Dans sa thèse, Robadey [ROB 01] propose de décrire la structure physique avec la norme XML qui permet de spécifier n'importe quel format désiré.




```
<page>
  <bloc>
    <ligne>
      <mot fonte="Times" graisse="Bold"
           taille="18" contenu="Generic"/>
      ...
      <mot fonte="Times" graisse="Bold"
           taille="18" contenu="Segmentation"/>
    </ligne>
    ...
  </bloc>
  ...
</page>
```

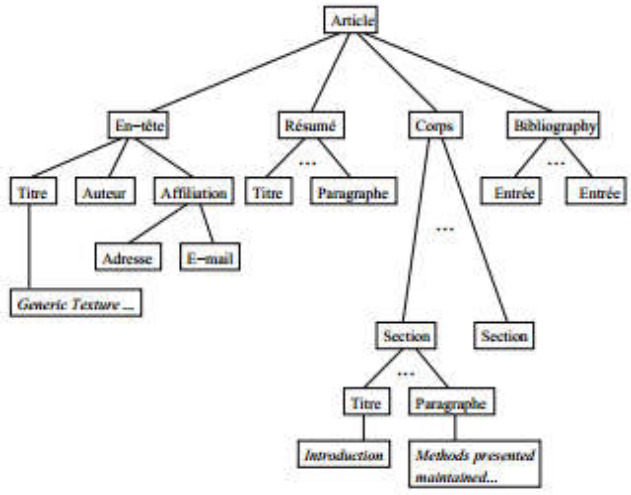
(a) *(b)*

Figure 1.2. *Représentation d'une structure physique sous forme, (a) arbre, (b) XML [ROB 01]*

La figure 1.2 met en parallèle la représentation de la structure physique d'un document sous forme d'arbre avec la proposition de représentation sous forme XML.

3.4.2. Représentation de la structure logique

La structure logique d'un document décrit son contenu sémantique, elle peut être représentée par un arbre comme la structure physique et encodée en XML [ROB 01]. La figure 1.3 met en parallèle la représentation d'une structure logique sous forme d'arbre et sous forme XML.




```
<article>
  <en-tête>
    <titre contenu="Generic Texture ..."/>
    <auteur> ... </auteur>
    <affiliation> ... </affiliation>
  </en-tête>
  <résumé> ... </résumé>
  <corps>
    <section>
      <titre contenu="Introduction"/>
        ...
      <paragraphe contenu="Methods
                  presented maintained ..."/>
    </section>
      ...
  </corps>
  <bibliographie>... </bibliographie>
</article>
```

(a) *(b)*

Figure 1.3. *Représentation d'une structure logique sous forme, (a) arbre, (b) XML. [ROB 01]*

Cependant, d'autres représentations ont été utilisées [TRU 05]

- Par un graphe orienté complet dont les sommets sont les objets physiques et les arcs sont des relations spatiales entre les objets.
- En se basant sur des grammaires formelles. Dans ce cas, le document est considéré comme une phrase ou chaîne d'étiquettes logiques.
- En se basant sur des modèles stochastiques. Ici, l'analyse de la structure logique d'un document est considérée comme un problème d'analyse stochastique.
- A l'aide des paternes. Dans cette représentation, chaque entité est représentée par un pattern comprend les propriétés physiques de cette entité ainsi que de ses voisins.

4. Reconnaissance de document

4.1. Présentation

La reconnaissance de document est une opération qui consiste à reconstruire une version électronique à partir d'un document imprimé ; c'est alors le processus inverse de la production de document [ROB 01]. Cette discipline regroupe un ensemble de techniques informatiques dont le but est de reconstituer le contenu d'un document à partir de son image [MON 11].

En réalité, il n'y'a pas de système universel capable de reconnaitre tout types de documents, compte tenue de la difficulté à décrire dans un seul modèle l'ensemble de tous les différents types de documents existants [AZO 95].

4.2. Etapes de reconnaissance de document

La reconnaissance de document implique plusieurs traitements comme il est illustré dans la figure 1.4 suivante:

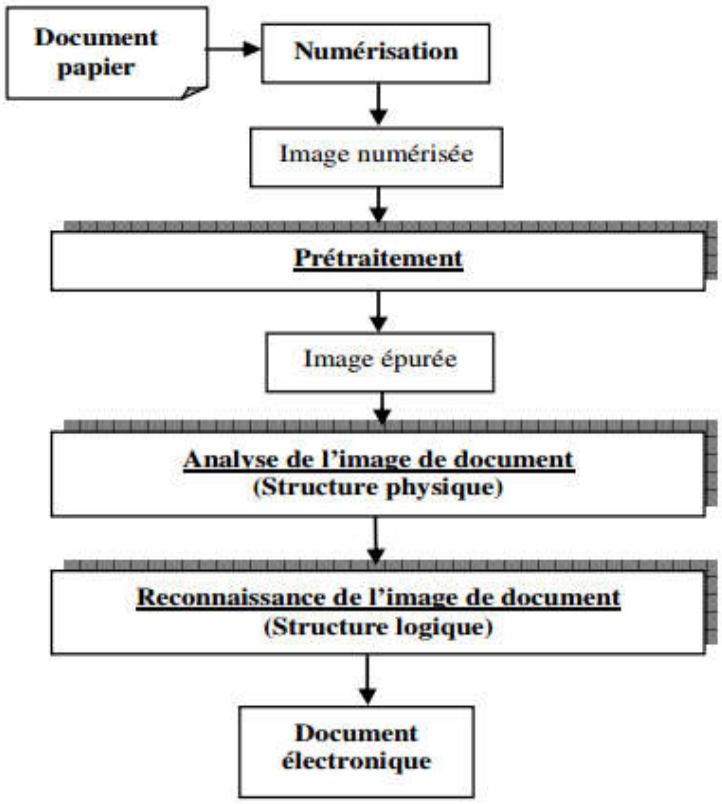


***Figure 1.4**: Schéma de reconnaissance d'un document [KET 10].*

4.2.1. Numérisation

La numérisation est le résultat de la conversion du papier sous la forme d'une image numérique. Ce processus est effectué par le biais d'un scanner ou d'une caméra [HAD 06].

4.2.2. Prétraitement

Le prétraitement est une étape essentielle dans tous les systèmes de traitement et d'analyse de documents. Son rôle est de préparer l'image du document numérisée au traitements. Les opérations de prétraitement sont relatives au redressement de l'image, à la suppression du bruit et de l'information redondante, et enfin à la sélection des zones de traitement utiles [BEL 01]. Le prétraitement peut inclure diverses opérations (binarisation, filtrage, lissage...).

4.2.3. Reconnaissance de la structure physique

L'extraction de la structure physique, appelée aussi Analyse de document est, d'après Nagy [NAG 00], « *une théorie et une pratique de reconstruction de la structure symbolique des images numériques directement produites par l'ordinateur ou simplement numérisées à partir du papier* ». C'est également « *l'analyse de la mise en page pour trouver la structure physique. Il s'agit de segmenter l'image de document en composantes homogènes et de classifier chaque zone en texte, image, graphique, etc.* » [KET 10].

Cependant, les méthodes de segmentation peuvent être regroupées en deux familles : méthodes classiques et méthodes à base de texture [KET 10]. Nous focalisons dans cette étude sur les méthodes classiques qui traitent généralement les documents imprimés à prédominance textuelle et présentant une structure simple. Elles sont souvent réparties en trois grandes approches : *approche descendante, approche ascendante et approche mixte.*

a) Approche ascendante (Bottom-up)

Les algorithmes ascendants reposent sur (le paradigme de croissance de région), partent d'une petite échelle et tentent d'agglomérer les éléments de cette échelle en éléments plus gros, à la hauteur du document entier. Ils se basent sur trois échelles principales les pixels, les composants connexes et les «correctifs», qu'est une échelle définie par l'utilisateur [ESK 17].

b) Approche descendante (Top-Down)

Les méthodes descendantes prennent pour point de départ le document dans sa globalité et essaye de le découper ou le partitionner en fragments ou blocs ou segments. Elles requièrent généralement des connaissances a priori plus ou moins précises sur la structure de document à traiter [DUO 05].

c) Approche mixte

Les approches mixtes résultent de la combinaison des méthodes descendantes et ascendantes ou de l'utilisation conjointe d'une de ces dernières avec une autre méthode [HAD 06].

Toutes ces méthodes seront abordées en détails en chapitre 2.

4.2.4. Reconnaissance de la structure logique

L'extraction de la structure logique, dite également la reconnaissance de document, consiste à retrouver structure logique du document. Cette dernière est, comme nous avons déjà dit, l'organisation de l'image de document en entités logiques. L'extraction ou la reconnaissance de la structure logique sert donc à associer des étiquettes logiques, qui expriment la sémantique du document, aux différentes entités physiques extraites lors de l'étape de l'extraction de structures physiques : Titres, chapitres, ou paragraphes, telles qu'elles ont été perçues par l'auteur indépendamment de leur mise en page [KET 10].

D'après [KET 10] et [DUO 05], il existe plusieurs approches d'extraction de la structure logique. Ces approches s'appliquent généralement à des documents à structure relativement simple. Pour les documents à structure plus complexes, des approches hybrides et des approches perceptuelles, peuvent être appliquées [KET 10].

a) Approches de type « intelligence artificielle »

Ces approches regroupent des méthodes issues de l'intelligence artificielle. Elles reposent sur la construction de règles à partir de différentes informations extraites au niveau physique pour trouver la structure logique [DUO 05].

b) Approches structurelles

Ces approches regroupent des méthodes qui s'appliquent directement sur les structures de représentation de données. Il s'agit d'algorithmes de transformation de graphes, d'arbres, etc. ou d'inférence de grammaires [DUO 05].

c) Approches probabilistes

Les méthodes probabilistes consistent à considérer que les éléments ont été générés par un ensemble de distributions de probabilité. Le but est de s'adapter, au moyen des probabilités, au manque de régularité qui est dû à la structure du document même ou engendré par des erreurs de segmentation au niveau physique du traitement du document [DUO 05].

5. Conclusion

Un document renvoie à un ensemble formé par un support et une information. Il a une valeur explicative, descriptive ou de preuve. Il présente la pensée humaine [WEB 1]. Le nombre massif de documents qui existent actuellement a obligé les chercheurs à développer des méthodes pour les analyser et les reconnaitre automatiquement, ce qui a donné naissance à *l'analyse et la reconnaissance de documents*. Ce dernier est un domaine très important dans le traitement d'image. Il englobe un ensemble de techniques informatiques avec comme but la reconstitution du contenu du document sous la forme de documents structurés.

Dans ce chapitre Nous avons présenté un aperçu sur le domaine de l'analyse et la reconnaissance de documents. Nous avons commencé le chapitre par la présentation d'une vue d'ensemble du document. Ensuite nous avons abordé la notion de structure, les types de structures, ainsi que leur relation. Dans le reste du chapitre nous avons étudié la reconnaissance de document, tout en présentant ses différentes étapes de traitement, et les différentes approches d'analyse et de reconnaissance.

Chapitre 2. Segmentation d'images de document: Etat de l'art

1. Introduction

Le présent chapitre est une suite du chapitre précédent, dans lequel nous allons exposer un état de l'art sur les principaux travaux effectués dans la littérature sur la segmentation ou l'analyse de la structure physique d'images de documents.

Le chapitre commence par quelques définitions de la reconnaissance de structure physique. Le reste du chapitre est consacré à la présentation des différentes approches et méthodes proposées, et des divers travaux effectués dans la littérature pour la reconnaissance des structures physiques. Le chapitre se termine par une conclusion.

2. Segmentation d'images de documents

Namboordiri dans [NAM 07] propose de définir la segmentation d'une image comme suit : « *la segmentation d'une image est un ensemble de sous régions mutuellement exclusives et collectivement exhaustives de l'image* ».

A partir de cette définition il définit la segmentation de document ou l'analyse de sa structure physique comme : « *Le processus d'extraction de la structure physique décompose une image de document en une hiérarchie de régions homogènes pour laquelle chaque région est segmentée de façon itérative en sous- régions d'autres types spécifiques* ».

Selon [MON 11], l'objectif principal de l'extraction de la structure physique est de déterminer les frontières des différentes régions ou blocs de l'image du document. Cette extraction a pour but de décomposer l'image en une hiérarchie de régions homogènes. Le critère d'homogénéité dans le cas de son agencement se réfère au type de région telle qu'un bloc de texte, une image, un graphique, une ligne de texte, un mot… etc.

3. Travaux existants de reconnaissance de la structure physique

En effet, plusieurs travaux ont été effectués dans la littérature pour la reconnaissance de la structure physique de documents. Cependant, comme annonça [MON 11], [HAD 06], et autres, les méthodes et techniques proposées sont classées en trois types ou approches principaux: *ascendante*, *descendante*, et *hybride*. [TRA 16] ajoute une quatrième approche des *méthodes de résolution multi-échelle*.

Dans cette partie on vient de présenter un état de l'art sur les principaux travaux accomplis.

3.1. Approche ascendante

Les techniques de cette approche de segmentation procèdent par fusion hiérarchique des entités physiques jusqu'à ce que la structure complète (racine de la hiérarchie) de l'image traitée soit obtenue. Dans ce genre de techniques, la fusion est fondée en général sur une analyse des caractéristiques locales. Dans le cas des documents simples à prédominance textuelle par exemple,

les méthodes ascendantes commencent d'abord par déterminer les caractères, puis les fusionner en mots, les mots en lignes de texte, et fusionner les lignes de texte en blocs ou en paragraphes. Ces méthodes sont généralement applicables à diverses mises en page mais sont généralement couteuses en temps et en espace [TRA 16].

Dans cette section, nous exposons les principaux travaux de segmentation ascendante effectués dans la littérature.

3.1.1. Méthodes utilisant l'algorithme de lissage RLSA

L'algorithme RLSA (*Run Lengh Smoothing Algorithm*) est l'un des premiers algorithmes de segmentation d'images de document proposés dans la littérature. Cet algorithme est dû à Wahl et al. en 1982 [WAH 82] et fondée sur un double lissage unidirectionnel de l'image à segmenter. Il consiste à noircir suivant une direction donnée, les segments de pixels blancs de longueur inférieure à un seuil donné ***c.*** La segmentation est alors obtenue en appliquant l'opérateur logique ***Et*** sur les deux images résultant respectivement d'un lissage horizontal et d'un lissage vertical. La nature des blocs isolés est intimement liée au choix des seuils. Les seuils trop faibles provoquent une sur-segmentation alors que les seuils trop élevés provoquent une sous-segmentation [AZO 95].

Cet algorithme est restreint l'application aux documents avec une structure régulière et droite. D'après Azokly [AZO 95], les principales limites de cette technique sont le choix arbitraire des seuils de lissage, sa sensibilité aux inclinaisons, et son inadaptation à segmenter des blocs graphiques formules et tableaux

La figure 2.1 présente un exemple de segmentation en utilisant l'algorithme RLSA.

(a) *(b)* *(c)* *(d)*

Figure 2.1: Segmentation RLSA, (a) Image originale, (b) Lissage horizontal, (c) lissage vertical, (d) RLSA [LEL 07].

Plusieurs auteurs ont tenté d'étendre l'algorithme RLSA afin de l'utiliser à la segmentation des documents plus complexes.

Yamashita [YAM 96] propose un algorithme de lissage RLSA avec un seuil adaptatif. De ce fait, le moindre changement dans l'espacement entre les mots et au niveau de la taille des fontes affecte peu le résultat de la segmentation.

Dans le cadre de la segmentation en lignes de documents manuscrits complexes, un RLSA flou a été proposé dans [SHI 04]. Le principe de cette version de RLSA est de construire une image en niveaux de gris à partir d'une image binaire de façon à ce que la valeur d'un pixel dans cette image correspond à la distance qui le sépare du $n^{ème}$ pixel noir dans la direction horizontale ou verticale. n étant un paramètre de l'algorithme.

Pour extraire la structure physique, Sun [SUN 06] a divisé l'algorithme de segmentation en deux exécutions de RLSA. La première utilise un contexte local pour déterminer la taille des seuils permettant d'extraire les composantes textuelles de petites tailles. La seconde est réalisée sur le résultat de lissage de la première exécution en utilisant des informations globales. Cela permet d'extraire les composantes textuelles ayant polices plus grandes.

Afin de pallier les inconvénients de l'algorithme RLSA classique pour la segmentation en lignes, [NIK 10] a proposé une version modifiée du RLSA horizontal, appelée RLSA adaptatif (ARLSA). Cette dernière permet empêcher le regroupement dans la même ligne des caractères de tailles nettement différentes. L'idée principale d'un ARLSA est de connecter deux composantes connexes si elles vérifient un ensemble de contraintes géométriques relatives à leurs dimensions et à leurs positions [GHA 16].

Dans [SAR 11], une version modifiée de RLSA appelée RLSA spiral a été proposée pour extraire du graphique dans les images de documents. A la différence d'un RLSA classique, qui permet de connecter les pixels proches en balayant l'image horizontalement puis verticalement, le RLSA spiral effectue un balayage en spiral pour connecter les pixels voisins.

L'algorithme RLSA a été également utilisé par Bouressace et Zebiri [BOU 16] dans leur projet de fin d'études master pour segmenter les pages de journaux arabes. Une alternance d'application de RLSA horizontal et vertical a permet de regrouper les mots en lignes, les lignes en blocs (entête ou paragraphes), et puis en articles.

Dans le cadre se son projet de fin d'études master aussi, Drabsia [DRA 18] a basé sur l'algorithme RLSA pour la détection du cadre entourant les relevés de bac algériens. Pour ce faire elle n'a appliqué l'algorithme RLSA que sur les parties de l'image pouvant contenir le cadre (parties haute, basse, droite, et gauche). La combinaison des résultats de RLSA sur les quatre parties de l'image a permet de former le cadre du relevé.

3.1.2. Méthodes utilisant les composantes connexes

Une autre catégorie importante, que [HAD 06] même l'a considéré comme la base de toutes autres méthodes ascendantes, est celle des méthodes utilisant les composantes connexes. L'inconvénient de

la méthode des composantes connexes est qu'elle est sensible à l'interligne, à l'espacement entre caractères et à la résolution de la numérisation [HAD 06].

Fisher [FIS 90] a combiné l'extraction des composantes connexes avec l'algorithme de lissage. Les blocs de la structure physique sont constitués à partir des composantes connexes et de leurs rectangles englobants, en se basant sur un ensemble de caractéristiques des composantes connexes. Cette approche permet d'identifier les zones textes et non textes mais reste cependant sensible à la rotation de l'image du document [HAD 06].

Saitoh [SAI 92] procède par échantillon de $n \times n$ pixels sur toute l'image, puis extrait les composantes connexes. Ces dernières sont classées en texte, bruit, table, diagramme, image bitonale, ou filet, en utilisant les attributs des blocs tels que la hauteur du bloc, la proportion hauteur/largeur, etc. Les blocs sont divisés selon les critères suivants : la distance verticale entre les lignes et la hauteur des lignes dans les blocs.

Une autre méthode de segmentation de pages utilisant les composantes connexes a été présentée par Drivas dans [AMI 01]. Cette méthode comporte un ensemble d'algorithmes. Le premier algorithme permet de déterminer l'angle de rotation, le deuxième permet la segmentation et le troisième permet l'étiquetage des blocs obtenus en texte et en image. L'algorithme de segmentation extrait les composantes connexes ensuite applique la fusion de ces derniers. L'approche de fusion repose sur la recherche des plus proches composantes connexes et le regroupement des composantes connexes ayant une même dimension.

3.1.3. Méthodes utilisant les diagrammes de Voronoi

Une méthode de segmentation de pages basée sur la surface approximée des diagrammes de Voronoi a été présentée par Kise et al. en 1998 [KIS 98]. Cette méthode est applicable sur des images de document possédant une structure irrégulière de type Manhattan (structure constituée de zones rectangulaires possédant la même orientation) [MON 11], et ayant subies une rotation. La méthode comprend les étapes suivantes: extraction des points d'échantillonnage situés sur le contour des composantes connexes, suppression du bruit, génération du diagramme de Voronoi en utilisant des points d'échantillonnage obtenus, et finalement suppression des bords Superflus de Voronoi.

Agrawal et Doermann ont amélioré l'algorithme de Voronoi original avec Voronoi ++ qui adapte les paramètres de Voronoi au contexte spatiale local [AGR 09]. Ensuite, ils ont proposé dans [AGR 10] une version floue de celui-ci (avec des contours flous) appelé CVS. Cette dernière formule 320 régions hypothétiques, puis les valide. La phase de validation est basée sur des contextes de distance et de similarité (texture).

3.1.4. Méthodes basées sur des techniques de Clustering

D'autres méthodes de segmentation de documents techniques, notamment les pages de revues scientifiques et les cartes de visite, sont basées sur la technique de Clustering.

O'gorman [OGO 93] introduit la technique "*Docstrum*", qui se base sur la combinaison de l'analyse ascendante et du *Clustering*. Le principe de cette technique est de regrouper les composantes connexes dans un voisinage proche. Après la suppression du bruit, les composantes connexes sont séparées en deux groupes. L'un regroupe les composantes connexes possédant des tailles proches de celles des caractères dominants, et l'autre regroupe les composantes ayant des tailles éloignées de la taille dominante. Ensuite, elle détermine les *k plus proches voisins* pour chaque composante connexe. Puis, les lignes du texte sont trouvées par association des *k plus proches voisins* ayant une distance et un angle similaires. Enfin, ces lignes du texte sont fusionnées pour former des blocs de texte.

Les auteurs dans [JOU 08] utilisent des caractéristiques au niveau pixel. Ils soulignent l'importance d'une approche multi-résolution pour réduire le bruit dans les techniques de *clustering* de pixels. En fonctionnant au niveau des pixels, cela permet de regrouper de nombreux types d'objets, tels que des lettrines, un type spécifique de graphique, du texte, etc.

Faure et Vincent [FAU 09] utilisent un *clustering* géométrique pour segmenter les documents historiques. L'ajout intéressant qu'ils ont est l'utilisation d'une valeur de confiance pour chaque alignement (ligne du texte) et d'un post-traitement de résolution de conflit en cas d'incohérence entre deux lignes de texte.

3.1.5. Méthodes utilisant le filtrage à base de fenêtres

Les méthodes utilisant le filtrage à base de fenêtres, reposent sur un balayage d'une fenêtre de taille quelconque sur l'image entière de document. Lebourgeois [LEB 92] utilise un filtre de 8 × 3 pixels. L'image échantillonnée est dilatée par un élément de structure horizontale pour rassembler les caractères adjacents l'un vers l'autre. Chaque composante connexe est définie par un rectangle englobant et par la moyenne des longueurs de plages de valeurs de pixels noirs. Si la composante connexe est à l'intérieur de l'intervalle, elle sera classée en une zone de texte qui seront fusionnées en blocs, sinon elle sera classée en zone *non texte* [HAD 06].

3.1.6. Méthodes basées sur la classification

La déférence entre ces méthodes et les méthodes basées sur le *Clustering* est que les algorithmes de classification nécessitent tous un apprentissage.

Baechler et Ingold dans [BAE 11] chainent ensemble trois perceptrons dynamiques multicouches (DMLP pour *Dynamic Multi-Layer Perceptrons*) à trois résolutions pour segmenter les documents historiques. Chaque DMLP utilise la sortie d'étiquette du DMLP à la résolution inférieure en plus des caractéristiques de texture à sa résolution. Chaque niveau traite uniquement une partie des étiquettes produites par le niveau inférieur.

Garg et al. présentent un algorithme dans [GAR 11] pour séparer le texte et les graphiques. Ils utilisent un SVM (*Support Vector Machine*) pour classifier les caractéristiques de Gabor et de

contours, suivis d'un *Champ aléatoire conditionnel* (CRF pour *Conditional Random Field*) pour inclure le contexte spatial local. Le CRF améliore la performance par 2 points [ESK 17].

Dans [PEN 13], les auteurs travaillent au niveau des composantes connexes et des patches. Les patchs sont détectés avec une fermeture morphologique. Ils utilisent un *Champ de Markov aléatoire* (MRF pour *Markov Random Field*) pour classifier les patches en texte imprimé, manuscrit, ou chevauché. Ensuite, ils utilisent un autre MRF pour les re-calssifier en fonction de leur contexte. Le texte chevauché est séparé avec un troisième MRF

3.2. Approche descendante

Les méthodes descendantes (*top-down*) cherchent des informations globales dans la page et les divisent en régions de plus en plus fins. La page du document est segmentée en blocs, puis les blocs en lignes de texte et les lignes de texte en mots jusqu'à arriver au niveau des composantes connexes ou au niveau pixel. Les méthodes descendantes sont généralement rapides, impliquent généralement une complexité et une efficacité linéaires lorsque les documents ont une mise en page Manhattan [TRA 16].

Dans cette partie, nous exposons les principaux travaux effectués et les différentes méthodes proposées dans la littérature en ce qui concerne les méthodes descendantes.

3.2.1. Méthodes basées sur l'analyse des profils de projection

La méthode basique d'analyse des profils de projection consiste à faire une projection orthogonale des pixels de l'image de document sur les axes des repères (abscisses et ordonnés). Ainsi, le nombre de pixels noirs est comptabilisé pour chaque ligne de l'image ce qui produit un histogramme de projections horizontales, et pour chaque colonne de l'image pour obtenir l'histogramme de projections verticales. L'analyse de ces deux histogrammes nous permet de déduire diverses informations sur la présente des lignes de découpes, bandes blanches horizontales et verticales, lignes de texte, etc. La figure 2.2 présente un exemple d'histogrammes *de projection verticale et horizontale.*

Parmi les méthodes descendantes les plus fameuses utilisant les profils de projection nous citons l'algorithme de découpage X-Y (en anglais *X-Y CUT*). En 1984, Nagy [NAG 84] a introduit cet algorithme qui est approprié aux documents imprimés qui sont bien structurés et qui ne contiennent pas beaucoup des variations (journaux, formulaires…etc.). L'hypothèse de base repose sur le fait que les éléments structurés de la page sont généralement présentés dans des blocs rectangulaires, mais aussi sur le fait que les blocs peuvent être divisés en groupes de telle sorte que les blocs qui sont adjacents l'un à l'autre, dans un groupe, ont une dimension en commun. Le document est successivement divisé en de petits blocs rectangulaires en faisant une alternance de découpages horizontaux et verticaux le long des espaces blancs. Ces espaces blancs sont trouvés en utilisant un seuil de profil de projection. Le résultat d'une telle segmentation peut être représenté dans un arbre

X-Y, dans lequel la racine correspond à la page toute entière et les feuilles représentent les blocs de la page et chaque niveau de l'arbre représente alternativement les résultats de la segmentation horizontale ou verticale [HAD 06]. La figure 2.3 illustre un exemple de découpage X-Y.

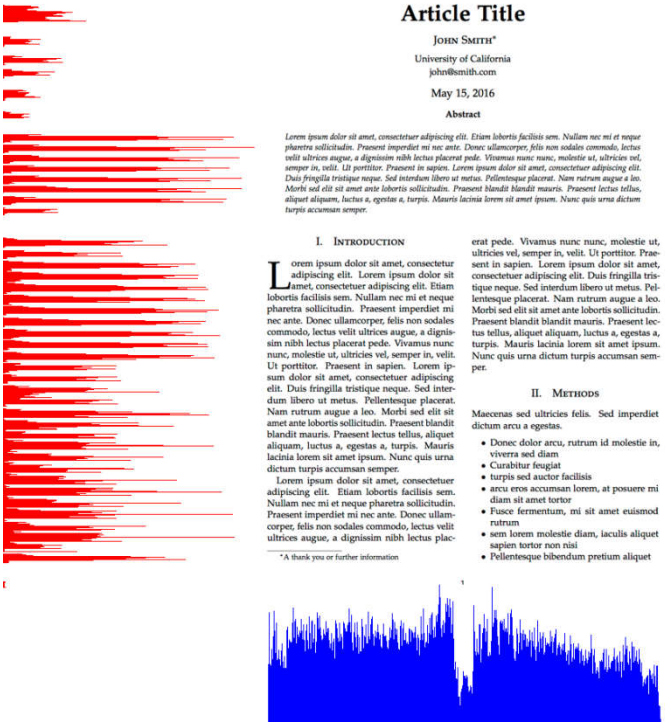


Figure 2.2 *Exemple de projection verticale (en bleu) et horizontale (en rouge)*

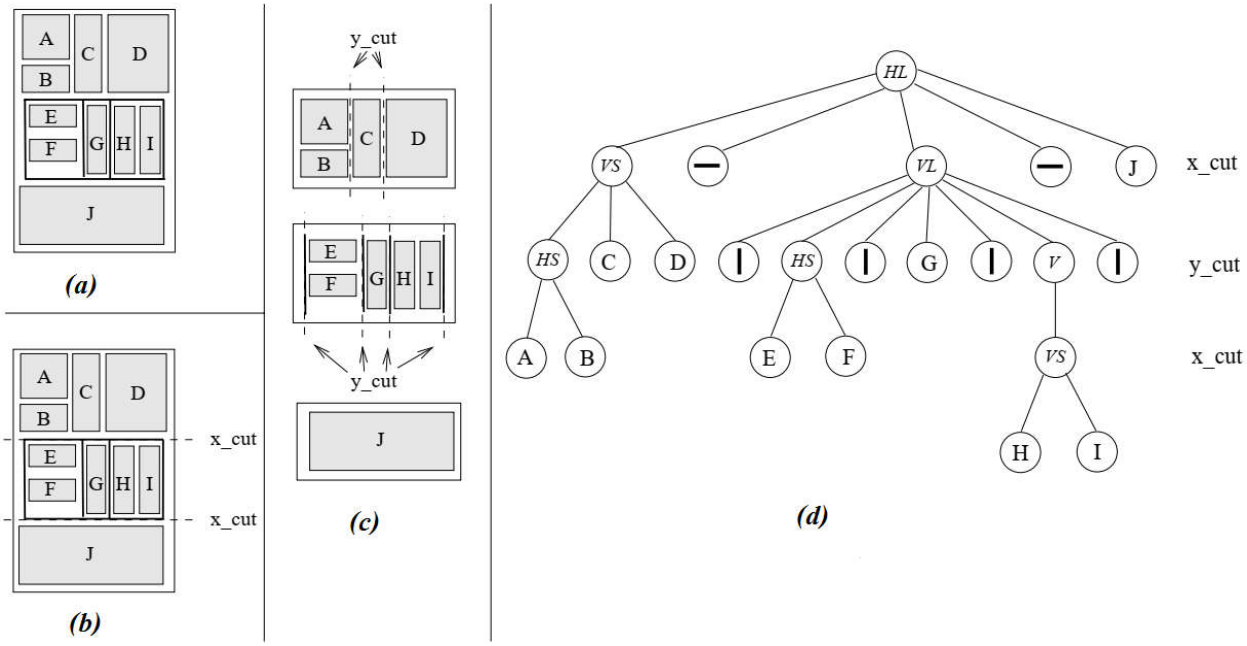


Figure 2.3. *Exemple de découpage X-Y, (a) une image de document, (b) découpage horizontal (X cut), (c) découpage vertical (Y cut), (d) l'arbre X-Y correspondant [CES 99].*

Des améliorations de l'algorithme de découpage X-Y ont été proposées. Dans [AKI 93], des règles topologiques ont été introduites pour permettre d'assouplir l'hypothèse de blocs rectangulaires en autorisant des polygones à travers la méthode de suivi de segments. Sylwester et Seth [SYL 95] proposent d'utiliser un module d'apprentissage pour rendre dynamique les seuils de division.

La technique de Ha et al. [HAL 95] sert à réaliser premièrement une capture des connexités, puis à projeter les boites englobantes des connexités au lieu de projeter les pixels.

En 2012, Ouwayed et Belaid dans [OUW 12] ont utilisé les profils de projection pour segmenter des documents multi-orientés. Ils ont fait d'abord un alignement du document avec des rectangles. Calculent ensuite le profil de projection de chaque rectangle selon plusieurs directions. Après, utilisent des heuristiques combinées avec des profils de projection locaux pour détecter les régions avec une orientation de texte non homogène et les lignes de texte.

3.2.2. Méthodes basées sur l'identification de lignes droites

Certaines méthodes essayent de segmenter le document en identifiant les lignes droites présentes dans le document.

Les auteurs dans [LOU 09] divisent les composantes connexes horizontalement en blocs en se basant sur la hauteur moyenne des caractères. Une fois ce partitionnement est effectué, ils appliquent la transformation de *Hough* sur les centres de gravité de chaque bloc pour détecter les lignes de texte.

En 2015, Wang et al. [WAN 15] tentent de reconstruire la bordure des cadres dans des bandes dessinées afin de les segmenter. Leur algorithme est capable de segmenter les cadres avec seulement deux frontières apparentes, mais est limité aux régions quadrangulaires. Ils séparent le fond puis ils utilisent un autre algorithme pour ajuster les quadrangles aux cadres candidats. Ceci est suivi d'une classification de la complexité du cadre et des heuristiques spécifiques sont utilisées pour compléter la bordure du cadre.

3.2.3. Méthodes utilisant l'analyse du fond de l'image

Les algorithmes de ce type reposent sur l'hypothèse que les entités sont délimitées par des flux de zones blanches verticales et horizontales plus grandes que les zones blanches comprises dans une entité. Ces approches cherchent à déterminer la structure de l'arrière plan correspondant aux zones blanches et non la structure du premier plan [MON 11].

Un des premiers travaux basés sur l'analyse des zones blanches est le travail de Spitz [SPI 90]. Son principe est de rechercher les flux blancs dans les deux directions verticale et horizontale, et les exploiter comme délimiteurs génériques de structures.

Antanacopoulos [ANT 94] a proposé une méthode, qui consiste en la recherche de la plus longue plage de valeurs de pixels dans le sens vertical pour noircir les zones. Cependant, il utilise des

rectangles de différentes tailles, pour couvrir le fond de l'image. L'extraction intervient en considérant les bords des rectangles coïncidant avec les bords des zones noircies.

Une autre méthode utilisant l'analyse du fond de l'image est celle de Bruel dans [BRE 02]. En fait, Bruel présente deux algorithmes pour la résolution des problèmes relatifs à l'analyse géométrique de structures de documents. Le premier est utilisé pour l'analyse des espaces blancs, ou l'analyse de structure du fond de l'image en termes de couverture rectangulaire. Le deuxième algorithme est utilisé pour la détection de lignes de texte en présence d'obstacles.

Chen et al. [CHE 13] analysent les espaces blancs pour segmenter le document en colonnes de texte. Les composantes connexes sont regroupées dans des chaînes horizontales pour créer des espaces blancs entre ces chaînes. Ensuite, ils sont regroupés verticalement pour créer des séparateurs blancs de lignes / colonnes. Cet algorithme a gagné les deux compétitions de segmentation de l'ICDAR 2013 [ESK 17].

3.2.4. Méthodes basées sur la description de structure par une grammaire

Ces algorithmes sont conçus pour un type de mise en page très spécifique et ne peuvent donc être utilisés que pour des documents avec une mise en page structurée [ESK 17].

L'utilisation de grammaires pour la segmentation de documents est due à Couasnon, qui a conçu en 2006 une méthode appelée **DMOS** (Description and Modification of Segmentation), constituée d'un nouveau langage grammatical et d'un analyseur associé [COU 06]. Le langage proposé peut décrire n'importe quelle mise en page et l'analyseur associé reconnaît cette mise en page dans une image.

En 2008, Lemaitre et al. [LEM 08] ont amélioré ce travail en ajoutant une approche multi-résolution qui le rendait suffisamment flexible pour segmenter les lettres manuscrits et identifier les lignes de texte dans les documents administratifs en français et en bengali.

Carton et al. [CAR 15] a ensuite poursuivi ce travail par une étape d'apprentissage interactive capable de créer un ensemble exhaustif de modèles pour un grand ensemble de données**.**

Shafait et al. [SHA 08] ont proposé un autre algorithme de grammaire basé sur une formulation de mise en page probabiliste. L'utilisateur définit un ensemble de découpes dont la position est définie approximativement. Ensuite, pour chaque image, un ajustement probabiliste est effectué pour obtenir les régions appropriées.

3.3. Approche hybride

Résulte de la combinaison des approches descendante et ascendante. Elle surmonte certaines faiblesses des deux approches classiques mentionnées ci-dessus. Ces algorithmes étaient principalement axés sur l'analyse des composantes connexes et des espaces blancs les séparant [TRA 16]. Dans cette approche on distingue diverses sous-classes de méthodes.

3.3.1. Segmentation basée sur une combinaison d'algorithmes

Une grande majorité d'entre elles font collaborer plusieurs techniques ascendantes et descendantes pour obtenir de meilleurs résultats.

En 1989, Wang [WAN 89] décrit une méthode à basée sur la combinaison entre l'algorithme de lissage RLSA et l'algorithme de découpage X-Y récursif pour extraire les blocs rectangulaires homogènes d'une page de journal. Les blocs sont ensuite classés en fonction des caractéristiques textuelles statistiques et des techniques de décision de l'espace.

Une méthode similaire utilisant le même principe est celle de Govindaraju et al. [GOV 90]. La seule différence est au niveau de l'obtention des blocs qui se fait en fusionnant les composantes connexes en de grandes zones.

Esposito [ESP 95] adoptait une méthode qui consiste en l'utilisation de l'algorithme de lissage RLSA avec une méthode ascendante pour classer les blocs selon leurs contenus en utilisant un arbre de décision. La classification est basée sur l'évaluation des dix caractéristiques.

Dans [CHA 99], les auteurs utilisent une méthode mixte pour la segmentation de pages de journaux numérisés ainsi que l'identification des articles, basée sur un ensemble d'algorithmes intégrés. Elle combine une technique ascendante (l'étiquetage des composantes connexes pour l'extraction de blocs), avec une technique descendante (analyse du fond de l'image).

Une autre contribution importante est celle de Barlas et al. [BAR 14] qui peut segmenter une gamme de documents extrêmement diverse et complexe en plusieurs langues. Après l'extraction des composantes connexes, une première classification des composantes textuelles et non-textuelles est effectuée à l'aide d'un Perceptron Multi Couches (PMC). La deuxième étape est la séparation des couches qui consiste à séparer les composantes textuelles en imprimées ou manuscrits. Pour ce faire, un livre de code est d'abord construit et utilisé pour entrainer un deuxième PMC. Une fois les couches sont séparées, la dernière étape est la segmentation en blocs en combinant l'algorithme RLSA et l'analyse des espaces blancs.

Un des travaux significatifs les plus récents était la méthode MHS (Multilevel Homogeneous Structure) développée par Tran et al. [TRA 15] qui a remporté la compétition de segmentation de documents complexes organisée par ICDAR (*International Conference on Document Analysis and Recognition*) en 2015. Cette méthode fonctionne en classant de manière itérative les composantes connexes en fonction de l'analyse de régions homogènes à plusieurs niveaux et d'espaces blancs.

3.3.2. Méthodes utilisant l'analyse syntaxique du document

L'analyse syntaxique qui tient ses origines dans la compilation des programmes informatiques a été approchée pour la reconnaissance de structures physiques de documents.

Le système proposé dans [NAG 92],commence par la segmentation du document en utilisant l'algorithme de découpage X-Y. Ensuite, l'analyse syntaxique est appliquée en vue d'extraire les blocs physiques. En effet, l'analyse syntaxique intervient dans le but de générer une grammaire par type de documents. Cette grammaire exprime les conventions de structure prédéfinie des publications de la revue.

Une approche similaire a été utilisée par Viswanathan dans [VIS 92]. Ainsi, les pages de revues techniques sont analysées selon une approche syntaxique afin d'identifier de manière hiérarchique leur structure spatiale. Les connaissances spécifiques à la publication sont utilisées dans la segmentation des blocs. Les informations sont méticuleusement codées sous la forme de grammaires par blocs utilisées pour décrire les relations entre les différentes classes d'entités ou blocs.

La méthode de [KRI 91] permet la combinaison de la segmentation d'une image avec l'étiquetage. Cette méthode est différente des autres méthodes d'analyse syntaxique en deux points : (1) les grammaires utilisées forment une hiérarchie, (2) la combinaison de formules syntaxiques avec la méthode de recherche du "branch and bound" [HAD 06].

3.3.3. Méthodes de segmentation par découpage et fusion

Le principe de ces méthodes est de procéder à une segmentation en petites régions et de les fusionner par la connaissance d'un modèle a priori [MON 11].

La méthode proposée dans [PAV 92] permet de distinguer entre les régions bitonales et non bitonales tout en permettant la séparation entre texte et graphiques. Elle commence par une segmentation descendante pour créer une sur-segmentation du document. Puis les segments qui sont similaires et proches sont fusionnés pour former une région. Finalement la séparation entre le texte et graphique est effectuée en utilisant le critère de densité de pixels noirs.

Dans [AZO 95], un algorithme de découpage hiérarchique basé sur l'analyse de rectangles structurants (rectangles blancs qui constituent le fond de l'image) est d'abord appliqué pour sur-segmenter l'image. Ensuite, un algorithme de fusion de composants gouverné par des règles décrivant les structures à reconnaitre est utilisé pour fusionner ces segments.

Liu [LIU 96] a développé une méthode qui repose sur la séparation en des zones non homogènes et la fusion de ces derniers en des zones homogènes. Pour le découpage de l'image en zones non homogène, c'est l'algorithme de découpage X-Y qui est utilisé. La fusion de ces zones en zones homogènes se fait à l'aide d'un algorithme ascendant qui est RLSA adaptatif, où le calcul des seuils est dynamique.

Une technique similaire à celle de [LIU 96] a été proposée par Hadjar et al. dans [HAD 01] pour la segmentation de pages de journaux. Après avoir séparé les filets horizontaux et verticaux par une méthode ascendante, l'image de la page est découpée en de petites zones en s'inspirant de

l'algorithme de découpage X-Y. Ensuite, ces petites zones sont fusionnées pour former des régions plus grandes.

Stamatopoulos et al. [STA 09] conçoivent une procédure pour améliorer les performances des algorithmes de segmentation individuels en combinant leurs résultats. La procédure est basée sur le chevauchement des régions produites par les algorithmes. Ils sont considérés comme bons au-delà de 90% de chevauchement. Toutes les régions inférieures à 90% subissent un découpage basé sur leur intersection, suivie d'une fusion partant des régions avec le chevauchement le plus élevé.

3.3.4. Méthodes de segmentation par analyse multi-résolutions

Ces méthodes permettent de segmenter une image de document à partir de différentes représentations de celle-ci. Ces représentations sont obtenues pour différents niveaux de résolution de l'image originale [MON 11].

En 1998, Cinque et al. [CINQ 98] ont proposé une méthode multi-résolutions qui repose sur l'analyse d'un ensemble de cartes de caractéristiques disponibles à différents niveaux de résolution. Dans cette méthode, l'analyse des caractéristiques de texte est basée sur des informations globales et la classification se fait par boîte englobantes.

Dans [LEE 01], les auteurs proposent une méthode basée sur l'analyse multi-résolutions conjointement avec la transformée en ondelette pour segmenter les images de document en régions homogènes maximales et les identifier en tant que textes, images, tableaux et lignes directrices. Une structure pyramidale à quatre arbres est d'abord construite pour une analyse multi-échelle et une mesure de périodicité est suggérée pour trouver un attribut périodique de régions de texte pour une segmentation de page.

[CHE 01] propose un algorithme de segmentation bayésien multi-résolutions permettant de modéliser des aspects complexes du comportement contextuel local et global. Le modèle utilise une chaîne de Markov à l'échelle pour modéliser les étiquettes de classe qui forment la segmentation, mais il complète cette structure de chaîne en incorporant des classifieurs à base des arbres pour modéliser les probabilités de transition entre des échelles adjacentes.

L'analyse multi-résolutions a également été utilisée par Shi et al. dans [SHI 05]. Une implémentation d'un algorithme de connectivité locale transforme d'abord l'image un domaine de paramètres. Dans ce dernier, une valeur de paramètre à un emplacement de pixel représente une propriété de connectivité pour ses pixels voisins de premier plan dans l'image d'origine. Ensuite, une approche descendante avec recherche linéaire révèle les régions du document à chaque niveau sous forme de bloc de texte, de lignes de texte et de graphiques.

Dans [LEM 07], une analyse à deux niveaux de résolution est effectuée pour la segmentation en lignes de texte. La première résolution permet de trouver l'orientation principale des lignes. La deuxième est utilisée pour permettre l'extraction de caractéristiques précises sur la connexité des

composantes inter et intra lignes. Les caractéristiques extraites sur ces deux niveaux sont ensuite combinées par une méthode à base de règles pour extraire les lignes.

4. Conclusion

D'un document papier vers un document électronique, et l'abaissement le coût de production des documents, ce sont les intérêts principaux des systèmes de reconnaissances et de compréhension de documents. Cette tâche est importante pour offrir une manipulation plus facile des données. En effet, les documents papiers manipulés sont tellement diverses qu'il est difficile d'avoir un système qui permet de comprendre n'importe quel document. La difficulté réside dans le fait que la reconnaissance doit prendre en considération la structure de documents et du fait que chaque type de documents possède sa propre structure. La compréhension d'un document doit donc s'intéresser à la reconnaissance de la structure du document avant de reconnaitre son texte.

Cependant, deux types de structures sont présents dans un document : physique et logique, et de nombreux travaux de recherche se sont focalisés sur leur extraction. Nous avons été intéressés dans ce chapitre à l'extraction de la structure physique de documents. L'objectif de ce chapitre était la présentation d'un état de l'art sur les principaux travaux, méthodes, et approches, proposée dans la littérature pour l'extraction de la structure physique de documents.

Chapitre 3.

Conception

1. Introduction

Après avoir étudié les déférentes techniques et méthodes de reconnaissance de la structure physique et logique proposées dans la littérature, nous avons pu avoir une idée générale sur les algorithmes et les techniques qui peuvent être exploitées pour atteindre nos objectifs.

Ce chapitre est destiné à la description de la conception ou bien la structure générale de notre application. Nous essayons d'expliquer comment nous puissions faire passer d'une image de relevé de notes de baccalauréat à un ensemble d'informations structurées exploitables représentant l'organisation du document.

D'abord, nous débutons ce chapitre par la description des caractéristiques et structures des relevés de notes du baccalauréat utilisés tout au long de ce travail. Nous présentons par la suite la démarche suivie tout en détaillant les différentes étapes inclues et les méthodes utilisées, avant de conclure.

2. Analyse physique des relevés de notes du baccalauréat

2.1. Caractéristiques des différents relevés de notes existants

Notre application est conçue pour traiter des **relevés de notes du baccalauréat Algérien**. C'est un type particulier de documents administratifs qui possède ses propres caractéristiques que ce soit au niveau du format ou du contenu.

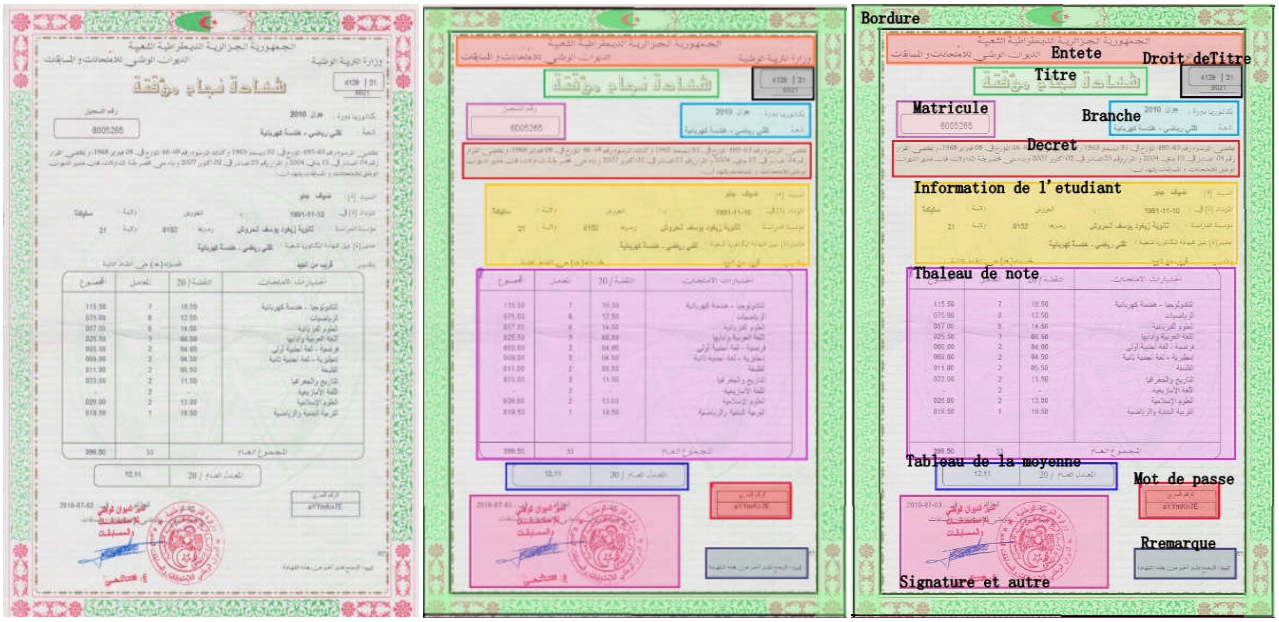
Après l'analyse physique des échantillons de relevés de notes, nous avons remarqué que le format des relevés change chaque année mais le contenu reste le même. Les variations sont à plusieurs niveaux : par exemple, au niveau de la qualité du papier (papier standard ou spécial), de la police d'écriture, de la langue dans laquelle les pièces de relevé sont écrites (arabe ou français), des couleurs de texte et d'arrière-plan, etc. [DRA 18]. Toutes ces variations compliquent le traitement des relevés à plusieurs niveaux. La figure 3.1 présente quelques exemples de relevés de différents formats.

***Figure** 3.1. Exemples de relevés de différents formats [DRA 18].*

2.2. Structures des relevés de notes

Le relevé est organisé, comme indiqué sur ***la figure 3.2***, en plusieurs blocs:

- Blocs textuels, composés d'une ou plusieurs lignes de texte, correspondant aux : *Entête, Titre, Numéro de la commission, Matricule, Branche d'étude,* et *Informations de l'étudiant*.
- Blocs non textuels, pouvant être des tableaux: *tableau de notes* et *tableau de la moyenne*, ou des graphiques: *logos*, ou *tampons et signature.*



(a) *(b)* *(c)*

Figure 3.2. Structures d'un relevé de Bac, (a) structure physique, (b) structure logique.

3. Description de l'approche proposée

Nous décrivons dans cette section l'approche proposée pour la segmentation des relevés de notes du baccalauréat algérien. L'approche proposée [KEF 19] inclue plusieurs étapes de traitement regroupées en trois modules principaux: le prétraitement, l'extraction de la structure physique, et l'extraction de la structure logique. ***La figure 3.3*** illustre un schéma récapitulatif des étapes principales inclues dans notre système. Ce schéma se compose de :

a) **Prétraitement** : vise à améliorer la qualité de l'image obtenue après la numérisation du document.

b) **Extraction de la structure physique** : regroupe un ensemble d'étapes de segmentation permettant de séparer les entités physiques composant le document.

c) **Extraction de la structure logique** : tend à étiqueter les entités physique extraites précédemment par des étiquettes logiques.

d) **Génération d'un fichier XML** : crée pour chaque relevé un fichier XML structuré représentant l'organisation logique du document.

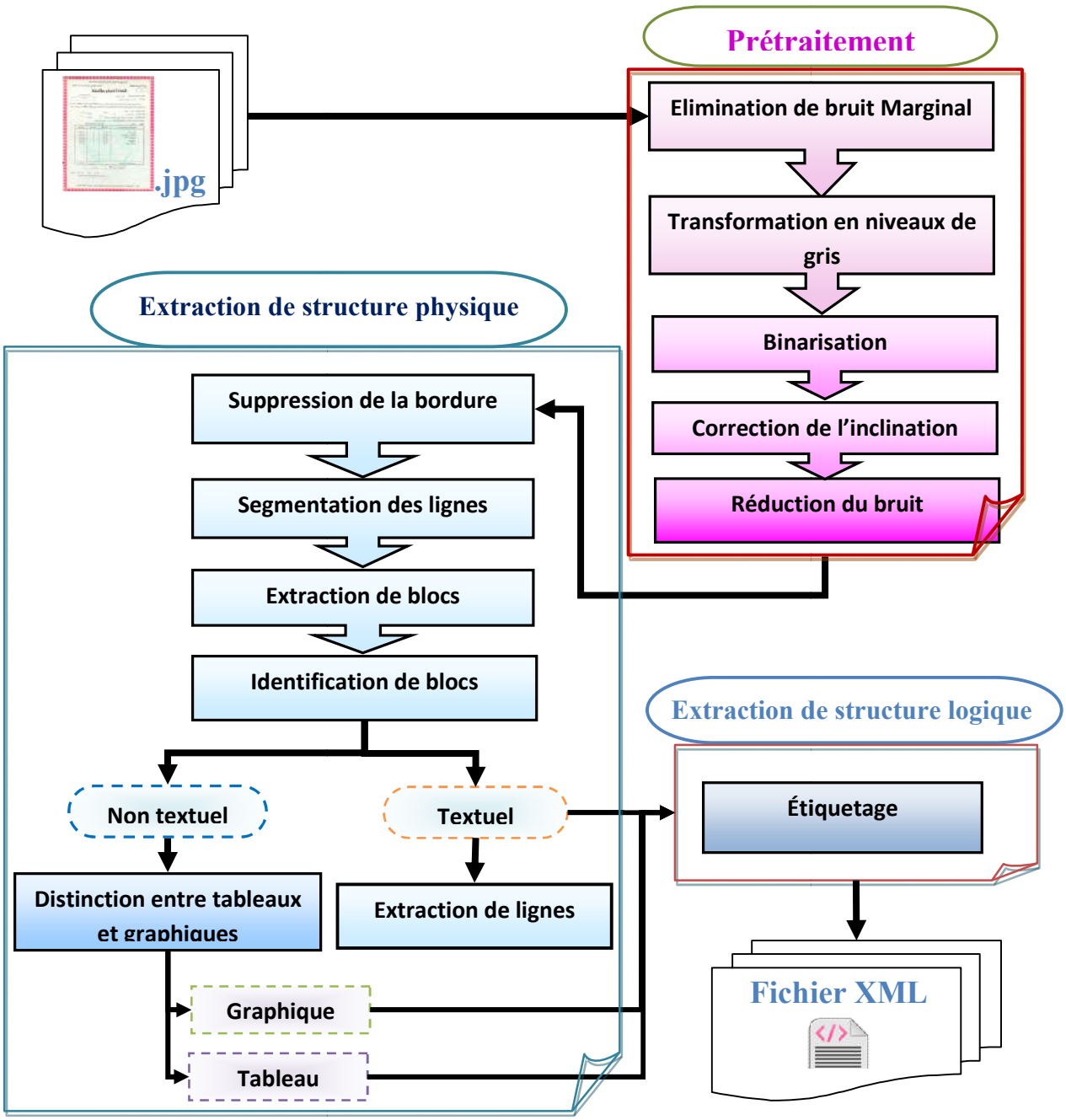


***Figure** 3.3: Architecture générale de l'approche proposée.*

3.1. Prétraitement des relevés

Le prétraitement regroupe un ensemble d'opérations visant à éliminer le bruit et les défauts liés à l'image numérisée (taches d'encre, traces d'effondrement…etc.). Il est très important pour que le bruit n'influencera pas sur les prochaines étapes de traitement. Plusieurs opérations peuvent être inclues dans le prétraitement. Les étapes qui constituent le prétraitement dans notre système sont : l'élimination du bruit marginal, la transformation en niveaux de gris, binarisation, correction de l'inclinaison, et réduction de bruit ou composantes parasites.

3.1.1. Élimination du bruit marginal

Le bruit marginal est formé de l'ensemble d'ombres qui apparaissent en noir ou en une couleur

proche du noir dans les marges verticales ou horizontales d'une image. Il en produit à cause de la perforation, l'inclinaison, la numérisation de documents épais ou des bords de pages dans les livres [DRA 18]. Cet espace sombre entourant le cadre du relevé a un impact négatif sur les processus suivants, nous devons le supprimer pour le bon fonctionnement de notre système.

Les étapes suivies pour la suppression du bruit marginal sont les mêmes utilisées dans [DRA 18]. Ces étapes sont les suivantes:

Etapes de détection et d'élimination du bruit marginal

Entrée : image originale en couleurs (***ImgOr***)
Sortie : image sans bruit marginal (***ImgSbruit***)
Début

- Détecter le bruit marginal haut, bas, droit, et gauche. Pour trouver le bruit marginal haut par exemple (respectivement bas, droit, gauche), on procède à l'étiquetage des composantes connexes[1] se trouvant dans les premières lignes de l'image originale (respectivement dernières lignes, premières colonnes, dernières colonnes) et formées des pixels ayant une couleur proche du noir (les valeurs du rouge, vert et bleu du pixel sont inférieurs à un certain seuil). Le bruit marginal sera formé de toutes les composantes connexes étiquetées.
- Afficher les composantes connexes formant le bruit marginal en bleu.
- Colorer les pixels du bruit marginal par la couleur dominante de l'image. Cette couleur est extraite en utilisant un tableau de 256^3 cases représentant chacune une couleur et stockant le nombre de pixels dans l'image ayant cette couleur. La couleur dominante correspond à la case contenant la valeur maximale du tableau.

Fin.

3.1.2. Transformation en niveaux de gris

Parce que la segmentation dans notre système fonctionne sur des images bitonales, l'image couleurs d'entrée doit d'abord être transformée en niveaux de gris avant sa binarisation car la méthode de binarisation qui va être employée n'est applicable que sur des images en niveaux de gris.

La transformation en niveaux de gris est effectuée en attribuant à chaque pixel la moyenne de ses quantités des couleurs rouge, vert, et bleu.

3.1.3. Binarisation (seuillage)

La binarisation est la technique la plus simple de segmentation d'images. A partir d'une image en niveaux de gris, cette opération produit une image ayant deux classes de pixels: pixels de fond en blanc et pixels de l'avant-plan (objets) en noir.

En effet, un grand nombre de techniques de binarisation ont été proposées dans la littérature. Dans notre système, nous avons choisi d'utiliser la méthode ***d'Otsu*** [OTS 79] qui permet d'effectuer un

[1] Une composante connexe est un ensemble de pixels connectés.

seuillage global automatique à partir de la forme de l'histogramme de l'image. L'algorithme itératif d'Otsu calcule le seuil optimal *T* qui sépare les deux classes de l'image de façon à ce que la variance interclasses soit maximale [WEB 2]. Le calcul de la variance interclasses est basé sur l'histogramme normalisé $H = [h_0 ... h_{255}]$ de l'image.

Algorithme de binarisation par la méthode d'Otsu

Entrée : Image en niveaux de gris (***ImgNvg***)
Sortie: Image binaire (***ImgBin***) avec 0 = Noir et 1 = Blanc
Début
Calculer *h* l'histogramme de niveaux de gris de *ImgNvg* ;
Calculer *h2* l'histogramme normalisé ;
Pour chaque niveau de gris *S* allant de 0 à 255 faire

$$q1(S) \leftarrow \sum_{i=0}^{S-1} h2(i); \quad q2(S) \leftarrow \sum_{i=S}^{255} h2(i);$$

$$u_1(S) \leftarrow \frac{1}{q1(S)} \sum_{i=0}^{S-1} h2(i) \times i; \; u_2(S) \leftarrow \frac{1}{q2(S)} \sum_{i=S}^{255} h2(i) \times i;$$

$$\sigma_{inter}{}^2 \leftarrow q1(S) \times q2(S) \times [\mu 1(S) - \mu 2(S)]^2;$$

Fin pour
T ← le niveau de gris dont la variance est maximale ;
Pour chaque pixel (*x*,*y*) de l'image *ImgNvg* faire
Si *ImgNvg*(*x*,*y*) < *T* alors *ImgBin*(*x*,*y*) ←0
Sinon *ImgBin*(*x*,*y*) ←1;
Fin Pour
Fin.

3.1.4. Correction de l'inclinaison

Notre application vise à extraire et à identifier les différentes informations contenues dans les relevés de notes. Ainsi, les techniques que nous allons utiliser pour la segmentation des relevés de notes sont sensibles à l'inclination, et comme certains de nos documents sont inclinés, une étape de correction de l'inclinaison est nécessaire.

Cependant, nous avons utilisé une méthode simple de correction de l'inclinaison basée sur la transformée de Radon [BRA 95]. Le choix de l'utilisation de la transformée de Radon est justifié par sa capacité à décrire l'orientation des lignes droites (qui sont présentes dans les relevées de notes et forment des tableaux), la simplicité de sa mise en œuvre et son indépendance par rapport aux opérations de paramétrages préalables [BEN 14].

a) La transformée de Radon

La transformée du Radon est un outil permettant de tracer l'histogramme de projection des pixels selon des orientations bien définies (de 0° à 180°). Elle est définie par l'équation suivante:

$$f(p,\theta) = \int_{-\infty}^{+\infty} f(p.\cos(\theta) - s.\sin(\theta), p.\sin(\theta) + s.\cos(\theta))d_s$$

Où θ est l'angle de projection, p est la coordonnée du point P sur l'hyperplan de projection des pixels et s est la coordonnée du point P selon la perpendiculaire à cet hyperplan.
Le principe de transformation de Radon est de concentrer la somme des intensités de pixels d'une ligne droite en un point de l'espace transformée. Une ligne droite dans une image est ainsi transformée en un point de forte intensité dans l'espace de Radon, comme le montre la *figure 3.4.*

Notons que la transformée de Radon n'est applicable que sur une image carrée, ne prend pas en compte les angles d'inclinaison négatifs, traite uniquement des angles de degrés entiers, et qu'elle est couteuse en temps d'exécution à cause du calcul qui se fait pour les 180 degrés.

b) Personnalisation de l'algorithme de transformée de Radon

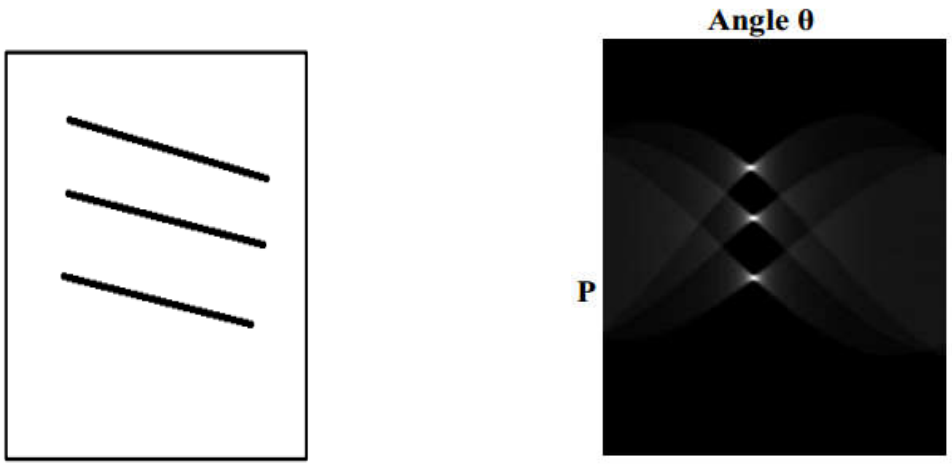
Comme la transformée de Radon n'est applicable que sur une image carrée, nous avons proposé de l'appliquer sur une portion carrée de l'image de surface égale à la *largeur de l'image* × *largeur de l'image*. De plus, d'après l'étude physique qu'on a fait, nous avons remarquée que l'inclinaison dans nos relevés peut être d'angle négatif, réel (par exemple 0.6°, -3.2°, etc.), et qu'il ne dépasse jamais ±15°. Ainsi, afin de couvrir toutes les situations possibles et en vue de diminuer le temps d'exécution, nous avons proposé de personnaliser l'algorithme de Radon de façon à ce qu'il prendra en compte les angles de degrés réels entre -15° et 15°.

Algorithme de transformée de radon personnalisé

Entrée : Image binaire (***ImgBin***) de largeur M et d'hauteur N
Sortie: Image en niveau de gris (***ImgRad***) correspondante à l'espace de Radon
Début
  $Nt \leftarrow 301$; $Np \leftarrow (\sqrt{M^2+N^2})]1$;
  ImgRad ← Nouvelle image de *Np* lignes et *Nt* colonnes
  Pour tout t de -150 à 150 faire
    $angle \leftarrow t/10$;
    Pour x allant de 0 à M faire
      Pour y allant de 0 à N faire

$$P \leftarrow \left(x - \frac{M}{2}\right)\sin(angle) + \left(y - \frac{N}{2}\right)\cos(angle);$$

        ImgRad (*P*,*t*) ← *ImgRad* (*P*,*t*)+*ImgBin(x,y)*;
      Fin pour
    Fin Pour
  Fin pour
Fin.



(a) (b)

Figure 3.4*: Transformée de Radon d'une image, (a) image contenant des lignes droite, (b) espace de Radon correspondant* [KEF 19]

c) Correction de l'inclinaison à l'aide de transformée de Radon

A partir de l'espace de Radon obtenu (l'image *ImgRad*), l'angle d'inclinaison θ du relevé peut être extrait facilement en identifiant le numéro de colonne des pixels d'intensité forte, et il ne reste plus qu'à faire pivoter l'image d'angle θ.

Etapes de correction de l'inclinaison à partir de l'espace de Radon

Entées : Image binaire (***ImgBin***) et image de l'espace de Radon (***ImgRad***)
Sortie : Image binaire après correction de l'inclinaison (***ImgBRoté***)
Début

- *angleInc* ← l'angle d'inclinaison du document calculé comme la moyenne des numéros de colonne des pixels d'intensité forte (pixels blancs) dans *ImgRad*;
- Roter *ImgBin* d'angle égale à *angleInc;*
- Colorer les pixels transparents dans *ImgBin* en blanc, car la rotation produit des pixels transparents dans l'image;
- Ne garder que la zone d'intérêt dans *ImgBin*, c'est la zone contenant les pixels noirs, car la rotation augmente la taille de l'image.

Fin.

3.1.5. Réduction du bruit

Les processus d'acquisition et de binarisation peuvent introduire du bruit dans l'image, aussi la présence des taches, des point et des composantes parasites répartis sur différentes positions dans le document, tous ça peut influencer sur les étapes traitements suivantes et diminue les performances de notre système.

Pour obvier ce problème nous avons appliqué une méthode d'élimination de bruit basée sur le filtrage des composantes connexes, extraites à partir de l'image binaire. Ce filtrage sert à éliminer

tous les pixels qui confuse l'image. Cette méthode se compose de deux étapes: étiquetage des composantes connexes et filtrage basé sur la taille des composantes connexes.

a) Etiquetage des composantes connexes

L'étiquetage des composantes connexes consiste à regrouper tous les pixels noirs voisins qui composent une unité distincte, et pour cela nous utilisons la méthode d'agrégation des pixels. Le but de cet étiquetage n'est pas de construire une liste des composantes connexes mais d'attribuer chaque pixel d'une image binaire à sa composante connexe.

Le résultat de cette étape est une image colorée dont chaque composante connexe est affichée par une couleur différente.

b) Filtrage des composantes connexes étiquetées

Un filtrage est ensuite appliqué afin de supprimer les composantes connexes de petites taille et les composantes connexes isolées car elles correspondent aux composantes parasites ou au bruit.

Algorithme du filtrage basé sur la taille des composantes connexes

```
Entrée : Image binaire après correction de l'inclinaison (ImgBRoté)
Sortie : Image sans bruit ImgNette
Début
    Etiquetage des composantes connexes de ImgBRoté;
    ImgNette ← ImgBRoté:
    Moy ← la taille moyenne de toutes les composantes connexes;
    Seuil ← Moy/3 ;
    Pour toute composante connexes CC faire
        Si la taille de CC < Seuil alors ajouter CC à la liste des composantes parasites ;
    Fin Pour
    //Chercher si ces composantes sont isolées ou pas
    Tant que la liste des composantes connexes parasites change faire
        Pour tout CCBruit de la liste des composantes parasites faire
            d1 ← distance entre CCbruit et la composante la plus proche en haut;
            d2 ← distance entre CCbruit et la composante la plus proche en bas;
            seuilDis ← (hauteur de CCbruit) * 3;
            Si Min(d1,d2) < seuilDis alors retirer CCbruit de la liste des composantes parasites;
        Fin Pour;
    Fin tant que ;
    Pour tout CCBruit de la liste des composantes parasites faire
        Pour tout pixel p de CCBruit faire
        ImgNette(p) ← blanc;
 Fin.
```

3.2. Extraction de structure physique

La structure physique du relevé s'organise, comme il est illustré dans la figure 3.2, en blocs. Il y a des blocs textuels, c'est-à-dire composés d'un ou plusieurs lignes de texte, et des blocs non textuels, pouvant être des tableaux, ou graphiques (logo, cachet et signature,…).

Afin d'extraire la structure physique des relevées de notes, nous procédons dans notre proposition une segmentation mixte. Premièrement, la bordure de la relevée est séparée de l'image en utilisant une technique ascendante, car il ne porte aucune information pertinente. Ensuite, on extrait les lignes et puis les blocs à partir de l'image du relevée sans bordure. Après, les blocs textuels sont identifiées et les autres blocs sont localisées.

3.2.1. Élimination de la bordure

Selon l'étude physique des relevés de notes que nous avons mené, nous avons constaté que, les relevés des années de 1997 à 2012 et ceux de 2015, contiennent différents formats de bordures, et qu'il existent également d'autres relevés qui ne contiennent aucune bordure (relevés des années 2013 et 2014). Les bordures peuvent être sous forme de cadre formé d'une seule unité, sous forme d'une série d'étoiles, de fleurs ou d'autres formes géométriques de différentes couleurs, etc.

Le cadre du relevé est sans importance et sa présence dans le relevé peut empêcher l'extraction des informations du document. Pour cela la séparation du cadre est une étape nécessaire.

Pour éliminer le cadre nous procédons une méthode basée sur l'algorithme RLSA et considérée comme amélioration de la méthode utilisée dans [DRA 18]. L'idée est de rassembler les pixels de la bordure ensemble en une seule unité, et puis de les séparer du relevé, la tâche qui peut être accomplie efficacement par l'algorithme RLSA. Les étapes de cette méthode sont illustrée par la figure 3.5 suivante:

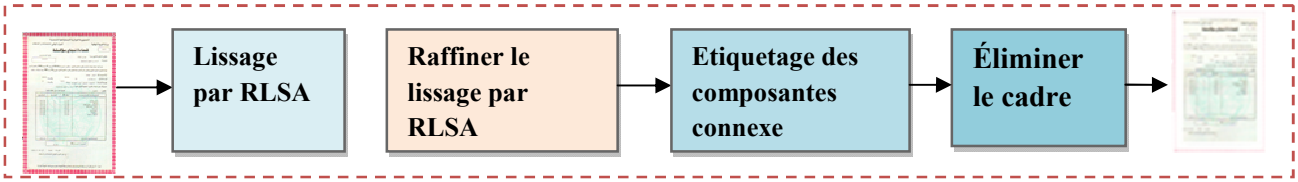


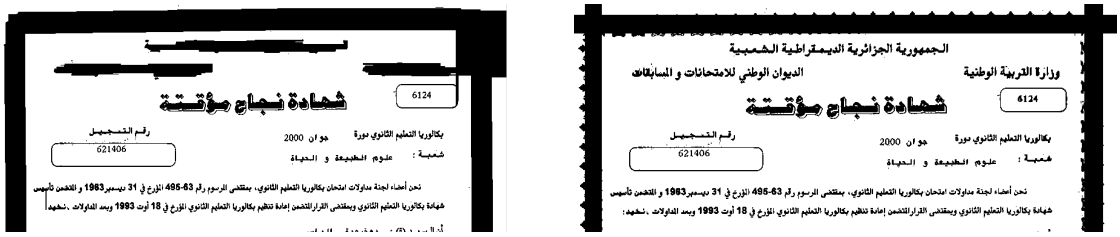
***Figure** 3.5* **:** *Schéma du processus d'élimination de la bordure*

a) Lissage par RLSA

Le but de ce lissage est de localiser les quatre côtés de la bordure sur le document. Ainsi, l'étude physique que nous avons effectuée nous permet de connaître l'emplacement approximatif des quatre côtés de la bordure. Ces derniers se défèrent légèrement d'un relevé à un autre mais ils se trouvent toujours dans les première parties (haute, basse, droite, et gauche) de l'image avec une épaisseur qui ne dépasse jamais la valeur (largeur du document /10).

Pour trouver les côtés horizontaux (haut et bas) de la bordure par exemple, on applique l'algorithme RLSA horizontalement sur les parties haute et basse de l'image avec un seuil n= (largeur de l'image

× 5%). La localisation des deux côtés verticaux (droit et gauche) s'effectue de la même manière mais en appliquant un RLSA vertical sur les parties droite et partie gauche de l'image, avec un seuil *s* = (hauteur de l'image * 5%). Les valeurs du seuil *n* ont été choisies par expérimentations de façon à ce qu'il permet de relier les composantes proches de la bordure. La Figure 3.6.a montre le résultat de cette étape :



(a) (b)

Figure 3.6. Détection de la bordure, (a) Lissage par RLSA, (b) Raffinement de lissage par RLSA

Etapes du lissage par RLSA

Entrée : image binaire bien orientée et épurée de bruit (***ImgNette***)
Sortie : Image avec bordure lissée par RLSA (***ImgBordRlsa)***
Début

- *haut* ← hauteur de l'image *10 /100;
- *larg* ← largeur de limage *10/100;
- Trouver les deux cotés horizontaux de la bordure. Pour ce faire :
  - Diviser l'image horizontalement en sous-images d'hauteur égale à *haut.* Les deux cotés horizontaux se trouvent respectivement dans la première et la dernière sous-image.
  - Appliquer RLSA horizontale sur la première et la dernière sous-image séparément en prenant *n* = 5% de la largeur de l'image.
- Trouver les deux cotés verticaux de la bordure. Pour ce faire :
  - Diviser l'image verticalement en sous-images de largeur égale à *larg.* Les deux cotés verticaux se trouvent respectivement dans la première et la dernière sous-image.
  - Appliquer RLSA verticale sur la première et la dernière sous-image séparément en prenant *n* = 5% de la hauteur de l'image.

Fin.

b) Raffiner le lissage par RLSA

L'application du RLSA permet de relier les pixels proches, des quatre côtés de la bordure, mais le résultat obtenu n'est pas exact à 100%; Certains pixels de la bordure ont été omis, et certains autres pixels n'appartenant pas à la bordure ont été considérés comme font partie de la bordure. Un raffinement est alors nécessaire afin de rassembler tous les pixels de la bordure, et uniquement ces pixels, en une seule unité. La *Figure 3.6.b* montre le résultat de cette étape. Le raffinement se fait en colorant les pixels blancs appartenant à une des cotés en noir et à blanchir les pixels noirs qui n'appartiennent pas, en se basant sur l'analyse des profils de projections horizontales et verticales.

Le calcul des profils de projections horizontales se fait à l'aide de l'algorithme suivant. Les profils de projections verticales sont obtenues de la même manière mais en procédant verticalement.

Algorithme de calcul des profils de projections horizontales

```
Entrée : Image binaire (Img) de largeur M et d'hauteur N
Sortie : Tableau (Hist) de N cases correspondant à l'histogramme de projections horizontales
Début
  //Calculer l'histogramme de profils de projection horizontales de Img
  Pour toute ligne i allant de 0 à N faire
    Hist[i] ← 0;
    Pour toute colonne j allant de 0 à M faire
      si Img (i,j) = noir alors Hist[i] ← Hist[i]+1;
    Fin Pour
  Fin Pour
Fin.
```

L'algorithme suivant explique le principe de raffinement du côté haut de l'image.

Algorithme de raffinement du lissage par RLSA

```
Entrée : Image binaire (imgBin) et Image lissée par RLSA (ImgBordRlsa) de largeur M et d'hauteur N
Sortie : Image lissée par RLSA raffinée (ImgBordRaf)
Début
  Hist ← l'histogramme de projection horizontales de ImgBordRLSA;
  Seuil ← M * 6/7 ; haut ← N *10% ;
  //Premier raffinement: enlever les pixels n'appartenant pas au côté haut
  Pour toute ligne i allant de 0 à haut faire
    Si Hist[i] <Seuil alors
      Pour toute colonne j allant de 0 à M faire ImgRlsaRaf(i,j) ← ImgBin (i,j) ;
      Fin Pour;
    Sinon ImgRlsaRaf(i,j) ← noir ;
  Fin Pour;
  /*Deuxième raffinement: chercher la ligne de début et la ligne de fin du côté haut et colorer tous les
  pixels se trouvant entre les deux en noir*/
  debutHaut ← 0; finHaut ← haut;
  Tant que debutHaut ≤ haut et Hist[debutHaut] < M faire debutHaut ← debutHaut +1 ;
  Tant que finHaut ≥ 0 et hist[finHaut] < M alors finHaut ← finHaut - 1 ;
  Pour toute ligne i allant de debutHaut à FinHaut faire
    Pour toute colonne j allant de 0 à M faire ImgRlsaRaf(i,j) ← noir ;
    Fin Pour
  Fin Pour
Fin.
```

Le raffinement des autres côtés se fait de la même manière.

c) Élimination de la bordure

Après l'application du RLSA sur les quatre cotés du cadre, tous les pixel noirs de la bordure sont devenus connectés entre eux et forment une seule unité ou bien une seule composante connexe dans le relevé. La bordure constitue la plus grande composante connexe dans le relevé. Pour l'éliminer on procède les étapes suivantes:

Algorithme d'élimination de la bordure

Entrée: Image lissée par RLSA raffinée (***ImgBordRaf***)
Sortie: Image sans bordure ***(ImgSBord)***
Début
  Etiquetage des composantes connexes de *ImgBordRaf*;
  Cadre ← la plus grande composante dans la liste des composantes connexes étiquetées;
  //Raffiner le cadre détecté
  Créer une nouvelle image ***ImgCdr*** contenant seulement *Cadre*;
  Refaire l'étiquetage des composantes connexes sur la partie de l'image se trouvant après le cadre (entre *finHaut*, *finGauche*, *debutBas*, *debutDroit*);
  /*Les composantes connexes étiquetées forment des petites portions de *Cadre* mais on ne sait pas si elles appartient réellement au cadre ou pas (voir figure 3.7.a)*/
  L ← hauteur du côté haut ou bas de *Cadre*;
  H ← largeur du côté droit ou gauche de *Cadre*;
  $SV \leftarrow L/6$; $SH \leftarrow H/3$;
  Pour toute composante *CC* de la liste des composantes connexes précédente Faire
    h ← hauteur de *CC*;
    l ← largeur de *CC*;
    Si($h > SV$ et $l/h < 4$ et *CC* appartient au côté haut ou bas) ou ($l > SH$ et $h/l < 2$ et *CC* appartient au côté droit ou gauche) alors enlever les pixels de *CC* de *Cadre*;
  Fin Pour
  //Supprimer le cadre en rendant blanc tous ses pixels.
  ImgSBord ← *ImgBordRaf*;
  Pour chaque pixel *p* de *Cadre* faire
    ImgSBord(p) ← blanc;
  Fin Pour;
Fin.

La bordure est apparue en mauve sur la figure 3.7.b.

(a) *(b)*

Figure 3.7 : *Elimination de la bordure, (a) composantes connexes qui ne font pas partie du cadre, (b) cadre en mauve avant sa suppression*

3.2.2. Première détection des lignes

L'objectif de notre travail est de segmenter les relevés de notes du baccalauréat et d'en extraire leurs différents blocs d'informations. Dans notre proposition, l'extraction des blocs repose sur la détection préalable des lignes de texte, même si cette détection n'est pas correcte à 100%. Une première étape de détection ou des lignes est alors nécessaires.

Une fois que la bordure est séparée de l'image, seuls les éléments pertinents du relevé (texte, tableaux, ...) restent en noir. Et puisque l'image est bien orientée (après l'étape de correction de l'inclinaison), les lignes de texte du relevé peuvent être extraites en procédant par lissage RLSA.

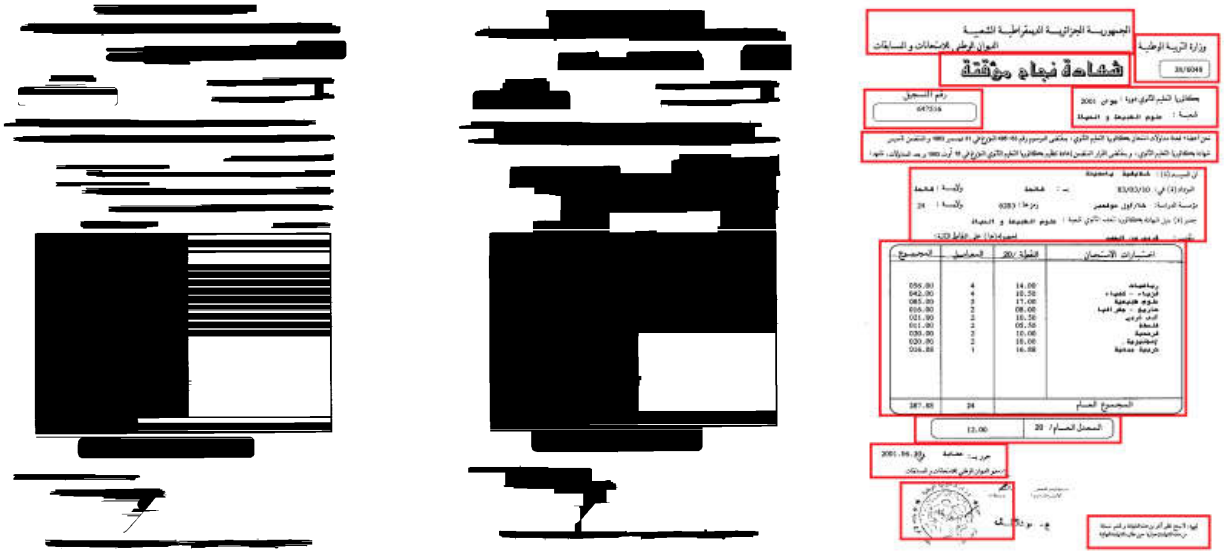
Tout d'abord nous appliquons encore une fois le filtrage des composantes connexes, utilisé dans la phase de prétraitement (section 3.1.5), pour éliminer les pixels de la bordure qui n'ont pas été éliminées, ou les composantes parasites qui n'ont pas été supprimées lors de l'étape de réduction de bruit car elle étaient proches du cadre.

a) Application de RLSA horizontal

Après l'élimination du bruit, nous appliquons un lissage RLSA horizontal, avec un seuil adaptatif, sur l'image résultante en vue d'éliminer les espaces entre les mots d'une même ligne de texte. Durant ce lissage, plusieurs valeurs de seuils sont alors utilisées sur les différentes zones de l'image afin de tenir compte de la variabilité de l'écriture (taille de police, style, ...). Ces valeurs de seuil ont été choisies par expérimentations.

Notons que le relevé contient plusieurs blocs et que deux blocs peuvent se chevaucher horizontalement. De ce fait, le seuil de RLSA horizontal doit être choisi de façon à ce qu'il permet de relier les mots d'une ligne de texte d'un bloc, et en même temps ne permet pas de relier les lignes de texte qui appartenant à deux blocs qui se chevauche horizontalement.

La ***figure 3.8.a*** présente le résultat de détection de lignes à l'aide d'un lissage RLSA Horizontal.



(a) (b) (c)

Figure 3.8 : *Détection de lignes et de blocs, (a) Lignes détectées à l'aide de RLSA horizontal, (b) blocs extraites à l'aide de RLSA vertical, (résultat final d'extraction de blocs)*

Etapes de détection des lignes de texte par RLSA horizontal

Entrée : Image binaire sans bordure ***(ImgSBord)***

Sortie : Image segmentée en lignes ***(ImgLignes)***, liste des lignes de texte détectées ***(LiLignes)***

Début

- Eliminer les composantes parasites en utilisant l'algorithme présenté dans la section 3.1.5;
- Pour toute zone ***SubImg*** de *ImgBord* faire
  - *n* ← seuil de lissage RLSA spécifique à *SubImg*;
  - Appliquer RLSA Horizontal avec un seuil *n* sur *SubImg;*
- Etiquetage des composantes connexes;
- *LiLignes* ← liste des composantes connexes étiquetées, qui correspondent aux lignes de texte;

Fin.

b) Filtrage des lignes détectées

L'étape précédente produit parfois des fausses lignes. Ces dernières correspondent soit aux composantes parasites, qui n'ont pas été supprimées lors des deux passes précédentes de réduction du bruit, ou à des signes diacritiques. Un filtrage est alors indispensable pour supprimer le bruit et pour attribuer les signes diacritiques aux lignes correspondantes.

Le filtrage se fait en cherchant les lignes de petite hauteur ou petite largeur et en essayant de les attribuer à la ligne de texte la plus proche en haut ou en bas (cas où la ligne s'agit de signes diacritiques). Si aucune ligne n'est proche, elle est supprimée.

L'algorithme suivant détaille le processus de filtrage:

Algorithme de filtrage des lignes de texte détectées

Entrée : Image segmentée en lignes (***ImgLignes)*** de largeur *M* et d'hauteur *N*, liste des lignes de texte détectées ***(LiLignes)***
Sortie : Image segmentée en lignes (***ImgLignes)***, liste des lignes de texte détectées ***(LiLignes)***
Début
seuilHaut ← *N*/140;
seuilLarg ← *M*/40;
Pour toute ligne ***L*** de *LiLignes* faire
haut ← la hauteur de la ligne *L*;
larg ← la largeur de la ligne *L*;
seuilDis ← *haut* * 6;
si *haut* < *seuilHaut* ou *larg* < *seuilLarg* alors
d1 ← distance entre *L* et la ligne la plus proche en haut;
d2 ← distance entre *L* et la ligne la plus proche en bas;
Si Min(*d1*,*d2*) < *seuilDis* alors affecter *L* à la ligne la plus proche
Sinon supprimer *L* de *LiLignes*;
Fin Pour;
Fin.

3.2.3. Extraction des blocs

Comme nous avons dit précédemment, les blocs peuvent être textuels, formés de plusieurs lignes de texte, ou non textuels. Dans les deux cas, les blocs peuvent être extraits en appliquant l'algorithme RLSA vertical sur l'image résultante du lissage RLSA précédent. Ce deuxième lissage RLSA a pour but de relier les lignes de texte d'un même bloc textuel ou de connecter les pixels proches verticalement d'un bloc non textuel.

a) Application de RLSA vertical

Théoriquement, la distance interlignes est plus petite que la distance inter-blocs successifs, et de ce fait un seul seuil de RLSA fixé avec soin permettra d'accomplir cette tâche. Malheureusement, cette hypothèse n'est pas vérifiée pratiquement pour nos relevés. C'est pourquoi que nous avons proposé d'appliquer un RLSA vertical avec un seuil adaptatif. RLSA vertical est donc appliqué sur chaque région de l'image avec une valeur de seuil propre à cette région. Les seuils ont été choisis avec soin et après plusieurs tests. Il doivent être suffisamment grands pour permettre la connexion des lignes du même bloc et en même temps doivent être insuffisant pour relier les blocs entre eux.

La figure 3.8.b illustre le résultat de l'extraction de blocs par l'application de RLSA vertical.

Algorithme de détection de blocs par RLSA vertical

Entrée : Image segmentée en lignes (***ImgLignes***), liste des lignes de texte détectées ***(LiLignes)***
Sortie : Image segmentée en blocs (***ImgBlocs***), liste des blocs détectés ***(LiBlocs)***
Début
  Pour toute zone ***SubImg*** de *ImgLignes* faire
    n ← seuil de lissage RLSA spécifique à *SubImg*;
    Appliquer RLSA vertical avec un seuil *n* sur *SubImg;*
  Fin Pour;
  Etiquetage des composantes connexes;
  LiBlocs ← liste des composantes connexes étiquetées, qui correspondent aux blocs;
  //Attribuer les lignes aux blocs correspondants
  Pour tout bloc *B* de *LiBlocs* faire
    rB ← la rectangle englobante de *B*;
    Pour tout ligne *L* de *LiLignes* faire
      rL ← la rectangle englobante de *L*;
      Si *rL* est inclue dans *rB* alors assigner *L* à B;
    Fin pour
  Fin Pour
Fin.

b) Filtrage des blocs détectés

En effet, l'étape précédente basée sur l'algorithme RLSA produit souvent des erreurs lors de l'extraction des blocs. Ces erreurs concernent notamment la fusion de deux ou plusieurs blocs à cause d'un mauvais choix du seuil de lissage, de la mise en page du document, etc. Une poste-segmentation est alors nécessaire.

Pour remédier ce problème, nous proposons de procéder un ensemble d'analyses visant à filtrer les blocs extraits, en fonction d'un ensembles de critères de position, de taille, etc. Le filtrage se fait par segmentation de certains blocs et par regroupement d'autres blocs séparés, tout en procédant de haut vers le bas.

Notons que les seuils de RLSA vertical ont été choisis de façon à ce qu'ils nous garantissent au moins l'extraction correcte des blocs se trouvant en haut du relevé (bloc d'entête, titre, matricule, ...), c'est à dire que ces blocs ne soient pas fusionnés avec d'autres blocs.

La *figure 3.8.c* présente le résultat final de l'extraction de blocs.

Etapes de filtrage des blocs extraits

Entrée : Image segmentée en blocs (***ImgBlocs***) de largeur *M* et d'hauteur *N*, liste des blocs détectés ***(LiBlocs)***

Sortie : Image segmentée en blocs (***ImgBlocs***), liste des blocs détectés ***(LiBlocs)***

Début

- Chercher les blocs de petites tailles à gauche du relevé et les relier en un seul bloc. Les blocs concernés sont ceux se trouvant dans le premier quart de l'image et ayant une largeur inférieure à la largeur de l'image/4.
- Le bloc le plus haut du relevé regroupe l'entête et le titre. Il faut les séparer. Ce bloc contient dans le cas normal trois lignes. L'entête se compose des deux premières lignes du bloc, et la troisième ligne forme le bloc titre.
- Le bloc titre se compose lui même de deux blocs: le titre à proprement parler, et un autre bloc au droit du titre. Pour les séparer, on applique d'abord un autre lissage RLSA avec un petit seuil sur la partie de l'image contenant le bloc titre. Cela permettra de relier les pixels de chacun des deux blocs mais de ne pas coller ces deux blocs. Ensuite, un étiquetage des composantes connexes se trouvant à cet emplacement permet d'extrait les deux blocs facilement.
- Les blocs au dessous du titre jusqu'au tableau de la moyenne se trouvent toujours fusionnés en un seul bloc *BB* et nécessitent une séparation. Nous allons les séparer un par un tenant profit de la constatation que les blocs réunis forment une alternance entre bloc de courte largeur (sa largeur est inférieure à la largeur de l'image/2) et bloc de longue largeur (sa largeur est supérieure à la largeur de l'image/2).
  - Pour séparer un bloc de courte largeur, il suffit de parcourir la liste des lignes de *BB* de haut en bas jusqu'à trouver une ligne longue signifiant qu'un nouveau bloc commence. Dans ce cas là, toutes les lignes précédentes sont toutes séparées de BB et affectées à un nouveaux bloc créé.
  - Pour séparer un bloc de longue largeur, il suffit de parcourir la liste des lignes de *BB* de haut en bas jusqu'à trouver une ligne courte signifiant qu'un nouveau bloc commence. Dans ce cas là, toutes les lignes précédentes sont toutes séparées de BB et affectées à un nouveaux bloc créé.
  - Un cas exceptionnel est le cas du bloc du tableau de notes. Pour le séparer, on cherche dans *BB* si un ligne de grande hauteur, existe. Si oui, cette ligne correspond au tableau de notes, et la ligne se trouvant juste au dessous est le tableau de la moyenne. Sinon, on conclue qu'un tableau complet n'existe pas car la bordure du tableau est effacée, et la détection du tableau doit se faire ligne par ligne, comme un bloc normal ayant des lignes larges.
- Le bloc *BB* se trouvant au dessous du tableau de la moyenne à gauche regroupe parfois plusieurs blocs (la date et lieu, le soussigné, et le cachet et signature, etc.), il faut les séparer :
  - Si les *n* premières lignes de *BB* sont de petit hauteur, elles correspondent aux lignes textuelles et elles forment donc un ou deux blocs à part.
  - Si n =1, un seul bloc existe, si $2 \geq n \geq 3$ alors deux blocs existent: le premier contient la première ligne, et l'autre contient les autres.
- Les blocs se trouvant au dessous du tableau de la moyenne à droite sont normalement séparés.

Fin.

3.2.4. Identification de blocs

Comme nous avons dit précédemment, les blocs peuvent être textuels, formés de plusieurs lignes de texte, ou non textuels. L'étape suivante est l'identification du type de blocs.

a) Identification des blocs textuels

Pour identifier les blocs textuels, on examine chaque bloc et teste s'il est composé de plusieurs lignes de texte ou pas. En effet, nous avons déjà extrait les lignes de texte lors de la première étape de segmentation, mais cette première segmentation peut ne pas être précise à cause de la présence de bruit collant deux lignes ensemble, par exemple. C'est pour ça qu'on propose de procéder une deuxième extraction de lignes de texte pour confirmer.

Cependant, la présence de lignes de texte dans un bloc peut être testée en se basant sur l'analyse des profils de projections. Ainsi, nous calculons l'histogramme de projections horizontales de chaque bloc extrait à partir de l'image du document binarisée (avant segmentation). La présence de plusieurs minimas globaux, présentant peu de pixels noirs, dans l'histogramme montre éventuellement des espaces interlignes pour un bloc de texte. Une ligne de texte sera entre deux minimas successifs. Pour s'assurer qu'il s'agit effectivement d'un bloc de texte, nous appliquons dans un deuxième temps l'analyse du profil de projection verticale pour chaque ligne de texte issue de la première analyse. La présence de plusieurs vallées blanches dans l'histogramme montre certainement des espaces inter mots. Evidemment, la non présence d'espaces interlignes et inter mots confirme qu'il ne s'agit pas d'un bloc textuel.

La figure 3.9 présente un exemple d'histogramme de projections horizontale (en rouge) d'un bloc textuel et d'un bloc non-textuel.

Algorithme d'identification des blocs textuels

```
Entrée : Image sans bordure (ImgSBord), liste des blocs (LiBlocs)
Sortie :  liste des blocs étiquetés par textuel ou non-textuel (LiBlocs)
Début
    Pour chaque bloc B de LiBlocs faire
        HistH ← l'histogramme de projections horizontales;
        Si HistH contient plusieurs vallées alors
            Pour chaque ligne probable L se trouvant entre deux vallées successifs faire
                HistV ← l'histogramme de projections verticales de L;
                Si HistV contient plusieurs vallées alors  B est un bloc textuel;
                Sinon  B  n'est pas un bloc textuel et sort de la boucle;
            Fin Pour
        Sinon  B  n'est pas un bloc textuel
    Fin Pour;
Fin.
```

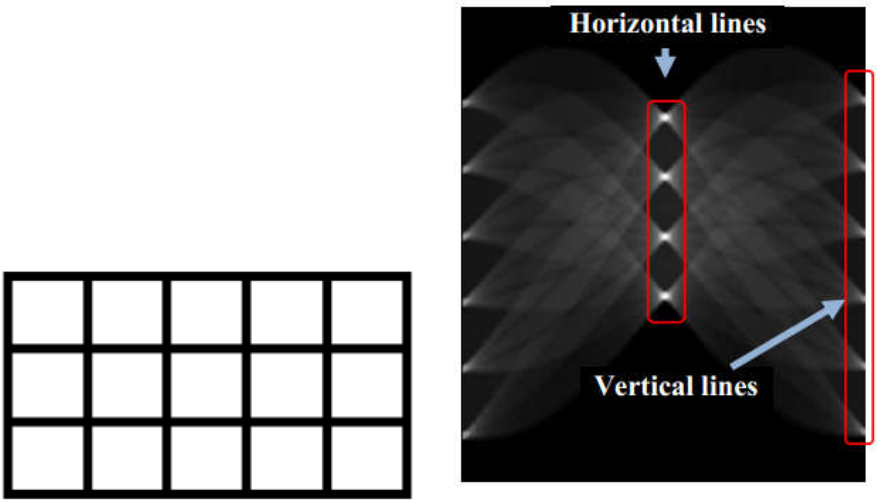
(a) (b)

Figure 3.9: *Histogramme de projections horizontales, (a) d'un bloc textuel, (b) d'un bloc non textuel*

b) Distinction entre les tableaux et les graphiques

Une fois les blocs textuels sont identifiés, il ne reste que des blocs non-textuels. Ces derniers peuvent être des tableaux ou graphique (logos, cachet et signature). Pour distinguer les tableaux des graphiques on repose de nouveau sur l'utilisation de la transformée de Radon. En effet, la transformée de Radon semble un bon choix car elle permet de détecter la présence de lignes droites. Ces derniers sont considérés comme l'élément clé permettant de discriminer la présence de tableaux. Ainsi, la transformée de Radon est appliquée sur chaque bloc non textuel extrait à partir de l'image binarisée en vue de vérifier la présence des lignes droites dans ce bloc.

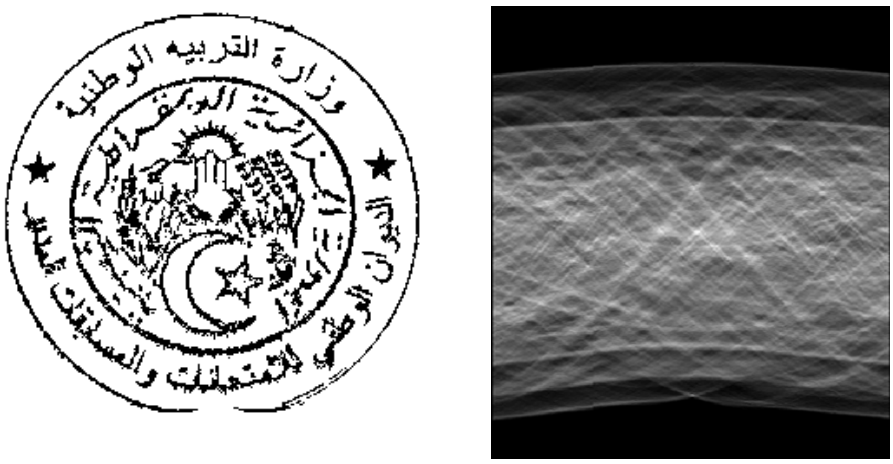
Comme nous avons avancé, la transformée de Radon retourne des pics sous forme de points ce qui signalent la présence des lignes droites. La *figure3.10* montre le résultat de la transformée de Radon sur un tableau. L'espace de Radon ainsi construit indique la présence de 10 points de forte luminosité signalant la présence de 10 lignes droites dans l'image du tableau : 4 points correspondants au lignes horizontales et 6 points correspondants au lignes verticales.



(a) *(b)*

Figure 3.10 : *Détection d'un tableau, (a) Image de tableau, (b) sa transformée de Radon* [KEF 19]

Dans le cas où le bloc non-textuel ne contient pas de lignes droites (s'agit d'un graphique), la transformée de Radon de ce bloc ne présente aucun pic sous forme de point de forte intensité (Figure 3.11).



(a) *(b)*

Figure 3.11: *Détection d'un graphique, (a) image de cachet, (b) sa transformée de Radon*

Algorithme d'identification des blocs textuels

Entrée : Image sans bordure ***(ImgSBord)***, liste des blocs ***(LiBlocs)***
Sortie : liste des blocs étiquetés par tableau ou graphqie ***(LiBlocs)***
Début
  Pour chaque bloc non-textuel *B* faire
    ImgRad ← la transformée de Radon de *B* calculée comme dans la section 3.1.4;
    Si *ImgRad* contient des pixels de forte intensité alors *B* est un tableau
    Sinon *B* est un graphique;
  Fin Pour
Fin.

3.3. Extraction de la structure logique

Cette phase a pour objectif l'étiquetage de tous les blocs physiques extraits dans la première phase du système, et la génération d'un fichier XML représentant la structure logique du relevé.

3.3.1. Etiquetage

L'étiquetage consiste à reconnaitre tous les composants du relevé de notes. Dans notre système, l'étiquetage repose sur la connaissance à priori de la position des blocs extraits. Ainsi, de haut en bas, de gauche à droite, les blocs ont les étiquettes suivantes: *Entête, Titre, Numéro de la commission, Logo, Matricule, Branche, Décret, Information de l'étudiant, Tableau de notes, Tableau de la moyenne, Date et lieu, Cachet et signature, Numéro secret,* et *Avertissement.*

3.3.2. Génération d'un fichier XML

Cette étape est très importante dans notre application car elle résume toute la structure extraite sous une forme bien organisée et structurée et nous avons choisi le format XML (eXtensible Markup Language), en français (langage de balisage étendu). XML est un langage portable, un sous

ensemble de SGML, défini par le standard ISO8879 en 1986, utilisé dans le milieu de la Gestion Electronique Documentaire. Il s'agit effectivement d'un langage permettant de mettre en forme des documents grâce à des balises, et aussi permet facilement l'échange des résultats.

Cependant, nous faisons correspondre à chaque relevé de notes, un fichier d'annotation XML correspondant (Figure 3.12). Chaque fichier d'annotation contient les informations suivantes sur l'image de document:

- Nom, hauteur, et largeur de l'image
- Nombre de blocs du relevée
- Pour chaque bloc: le type (texte, tableau, graphique), l'étiquette, et la position dans l'image (coordonnées de son rectangle englobant), le nombre de lignes.
- Pour chaque ligne, la position dans l'image (coordonnées de son rectangle englobant)

La figure 3.12 illustre un exemple de tel fichier.

```
<?xml version="1.0" encoding="UTF-8" ?>
  <IDENTIFICATION nameImage="Nom de l'image" height="Hauteur de l'image" width= "Largeur de
  l'image"   nb_Blocs = "Nombre de blocs dans le relevé" >
      < Bloc  nb= "Numéro du Bloc"  label="Etiquette  du bloc"  nb_lines="Nombre de lignes dans le bloc"
      type= "Type du Bloc" box= "Rectangle englobant du bloc" >
          <Line  nb= "Numéro de la ligne"  box= "Rectangle englobant du sous-mot"/>
          ……
          ……
      </Bloc>
  </IDENTIFICATION>
```

Figure 3.12: *Exemple d'un fichier XML représentation la structure logique d'un relevée*

4. Conclusion

Dans ce chapitre nous avons exposé les principales étapes de notre projet et les approches choisies en détails afin de montrer l'effort que nous avons réalisé pour résoudre les problèmes rencontrés dans notre travail dans le but d'atteindre les objectifs visés.

L'approche utilisé dans notre system inclut trois phases regroupant chacun un ensemble de traitements. La première phase est le prétraitement visant à préparer le relevé aux prochaines étapes par l'amélioration de sa qualité. La deuxième est la segmentation du relevé afin d'extraire les différents blocs constituant le relevé tout en identifiant leur type (textuel, tableau, graphique. La dernière phase est l'étiquetage logique des blocs physiques extraits et la représentation de cette structure logique dans un fichier XML.

Dans le prochain chapitre, nous présenterons l'implémentation de notre conception, avec un scénario complet d'utilisation du système, les expérimentations menées ainsi que les résultats obtenus.

Chapitre 4. implémentation et résultats

1. Introduction

Dans ce chapitre nous aborderons la réalisation pratique de notre système, qui est la phase la plus importante après celle de la conception décrite dans le chapitre précédent. Nous allons présenter dans ce chapitre les ressources et les outils de développement utilisés pour l'implémentation de notre application, l'architecture générale et les interfaces principales, ainsi que les tests effectués, les résultats obtenus, et la discussion de ces résultats.

2. Environnement de développement

L'expression environnement de développement veut dire tous les supports et les moyens matériels et logiciels utilisés pour l'implémentation de notre l'application.

2.1. Environnement matériel

Le matériel utilisé est constitué d'un PC dont les caractéristiques sont présentées dans le tableau suivant :

Modèle	PC portable
Processeur	Intel(R) Core (TM) i3-32bits
RAM	4.00 GO

Tableau 4.1 : *Caractéristiques du matériel utilisé.*

2.2. Environnement logiciel

Notre application a été développée en langage de programmation Java, avec l'environnement de développement NetBeans, version IDE 8.2.

- **JAVA :** est un langage de programmation orienté objet qui produit des logiciels pour plusieurs plates-formes. Lorsqu'un programmeur écrit une application Java, le code compilé (appelé bytecode) s'exécute sur la plupart des systèmes d'exploitation (OS), notamment Windows, Linux et Mac OS. Java tire une grande partie de sa syntaxe des langages de programmation C et C ++.

  Java a été développé au milieu des années 90 par James A. Gosling, un ancien informaticien de Sun Microsystems [WEB 3]. Le système java comporte plusieurs parties : un environnement, le langage, les interfaces de programmation d'application, et diverses bibliothèques de classes.

- **NetBeans :** est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License) et GPLv2.

  En plus de Java, NetBeans permet la prise en charge native de divers langages tels le C, le C++, le JavaScript, le XML, le Groovy, le PHP et le HTML. Il offre toutes les facilités d'un IDE moderne (éditeur en couleurs, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web (Voir la figure 4.1).

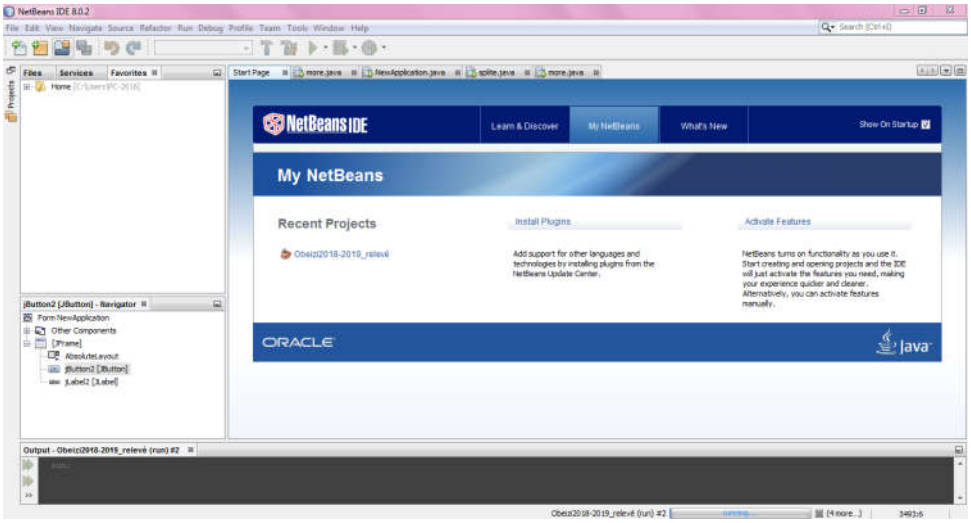


Figure 4.1: Interface graphique de NetBeans.

3. Présentation de l'application

Nous présentons dans cette partie notre application et les modules principaux de l'interface. Ainsi, notre application se compose de trois fenêtres ou interfaces. La figure ci-dessous illustre l'interface d'accueil de notre application.

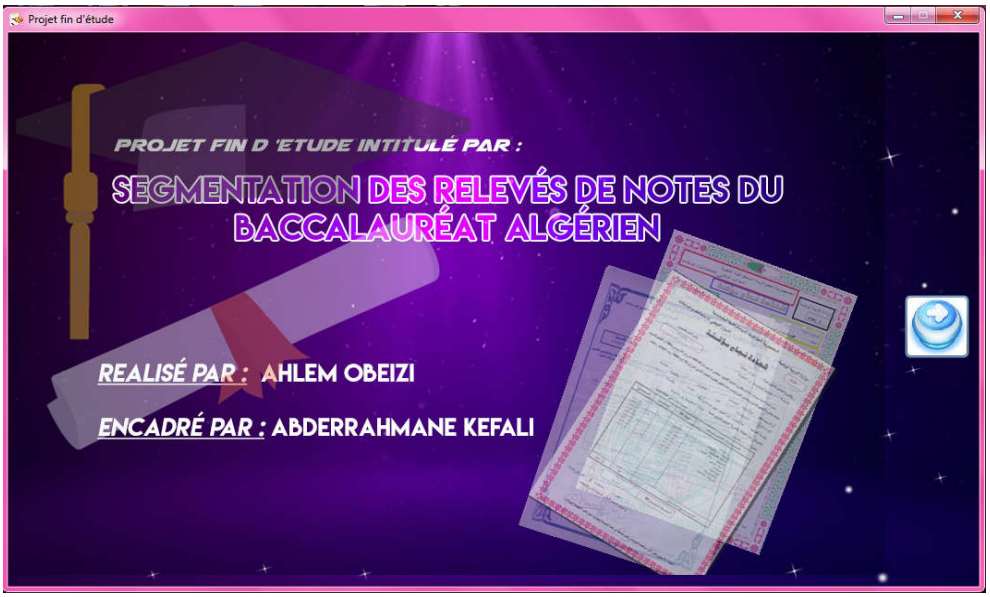


Figure 4.2 *: Interface d'accueil de notre application.*

En cliquant sur le bouton  , l'interface principale de l'application s'affiche, dans laquelle s'effectuent tous les traitements sur le relevé (***Figure* 4.3**). Elle contient: une barre de menus, plusieurs barres d'outils, une zone d'affichage de l'image traitée, et une barre d'état.

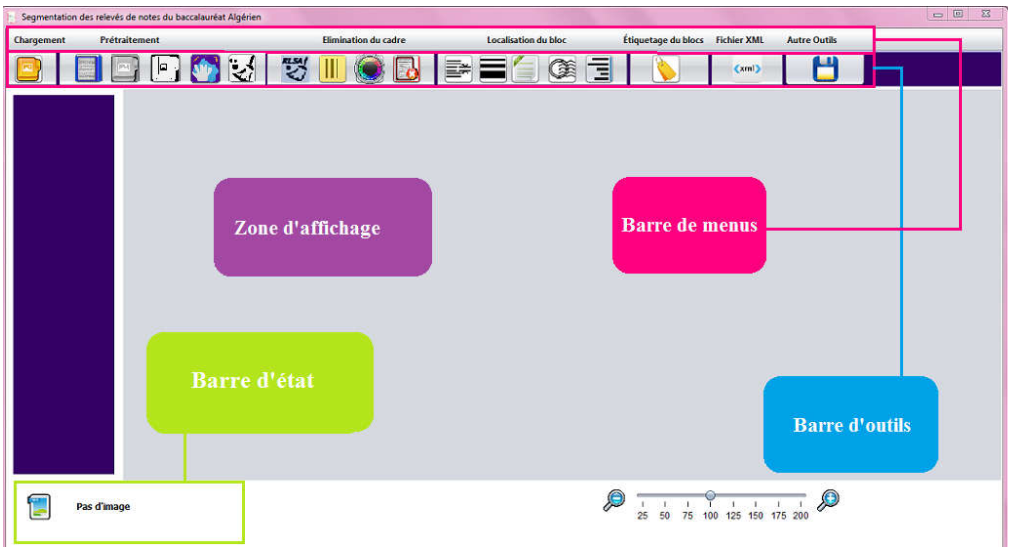


Figure 4.3 : *Interface principale de notre application.*

La troisième fenêtre est destinée à l'affichage du code XML généré, comme le montre la figure 4.4.

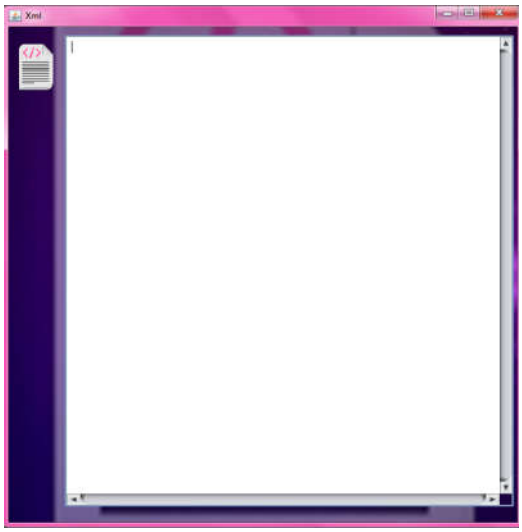


Figure 4.4 : *Interface d'affichage du fichier XML.*

Les modules principaux de notre application sont récapitulés par la figure suivante :

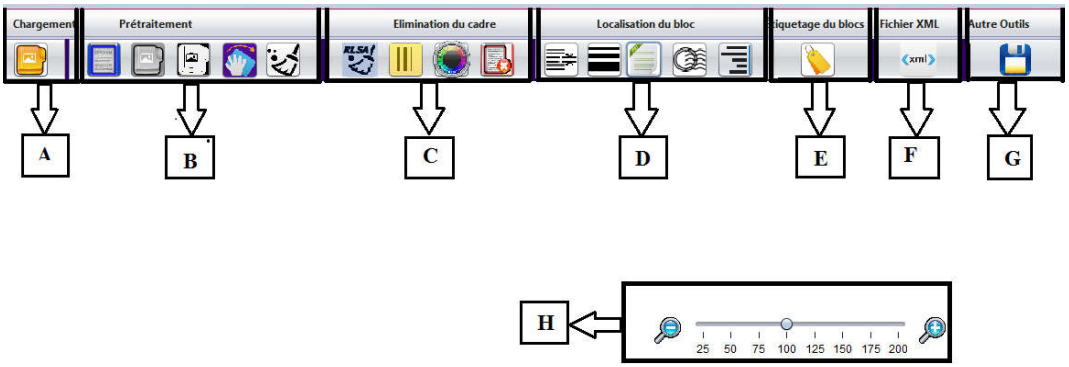


Figure 4.5*: *Les modules principaux de l'application

A. Barre d'outils *Chargement*. Il inclue un seul bouton permettant le chargement de l'image à traiter.

B. Barre d'outils *Prétraitement*. Il regroupe les boutons permettant le prétraitement : l'élimination du bruit marginal, la transformation en niveaux de gris, la binarisation, et la réduction de bruit.

C. Barre d'outils *Elimination du cadre*. Il regroupe les boutons permettant la détection et la suppression de la bordure du relevé : lissage par RLSA, Raffiner le lissage par RLSA, étiquetage des composantes connexes, et élimination de la bordure .

D. Barre d'outils *Localisation de blocs*. Regroupe les boutons permettant la localisation des blocs: extraction des lignes, extraction des blocs, identification des blocs textuels et non textuels, distinctions entre les tableaux et les graphiques, et extraction finale des lignes des blocs textuels.

E. Barre d'outils *Etiquetage*. Il contient un seul bouton permettant l'étiquetage logique des blocs.

F. Barre d'outils *Autres Outils*. Inclue un seul bouton pour la sauvegarde d'une image traitée.

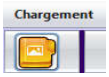
G. Barre d'outils *Fichier XML*. Contient un bouton chargé de la génération d'un fichier XML.

H. Volet de défilement *Zoome* pour contrôler les dimensions des images affichées.

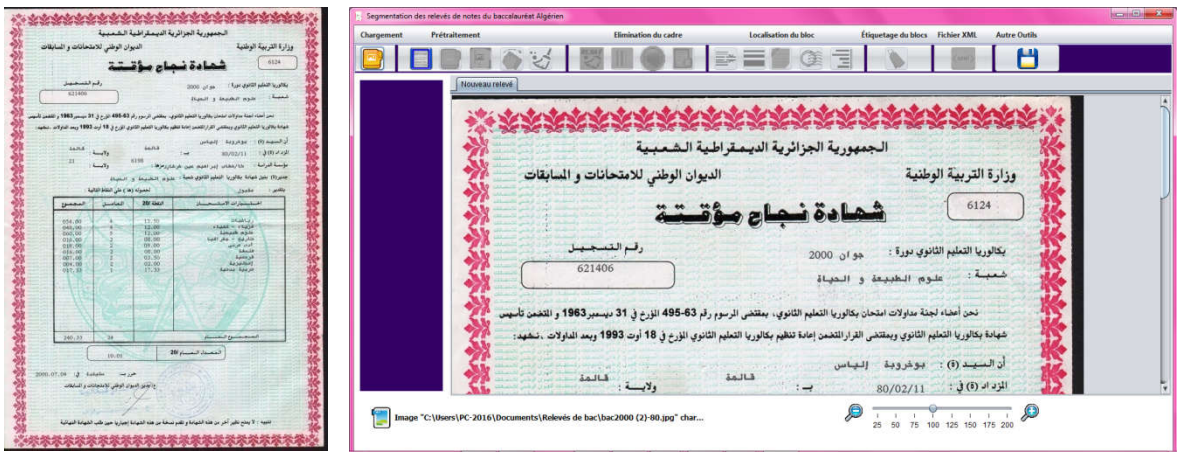
4. Scenario d'utilisation complet

Comme tout système informatique, notre système fonctionne selon un scénario spécifique. Nous déroulons dans cette section une utilisation complète, qui explique le fonctionnement de notre système. Nous allons utiliser une seule image de relevé comme exemple d'utilisation de l'application. Cette image est présentée par la ***figure 4.5.a.***

4.1. Chargement de l'image

Nous commençons par le chargement de l'image à traiter à partir du sous menu «Nouveau » du menu « Chargement » ou en cliquant directement sur le bouton 

de la barre d'outils «Chargement ». Une boite de dialogue s'affiche nous permettrons de sélectionner l'image de relevé à traiter. Cette dernière sera affichée dans la zone d'affichage, comme le montre la figure suivante :



(a) *(b)*

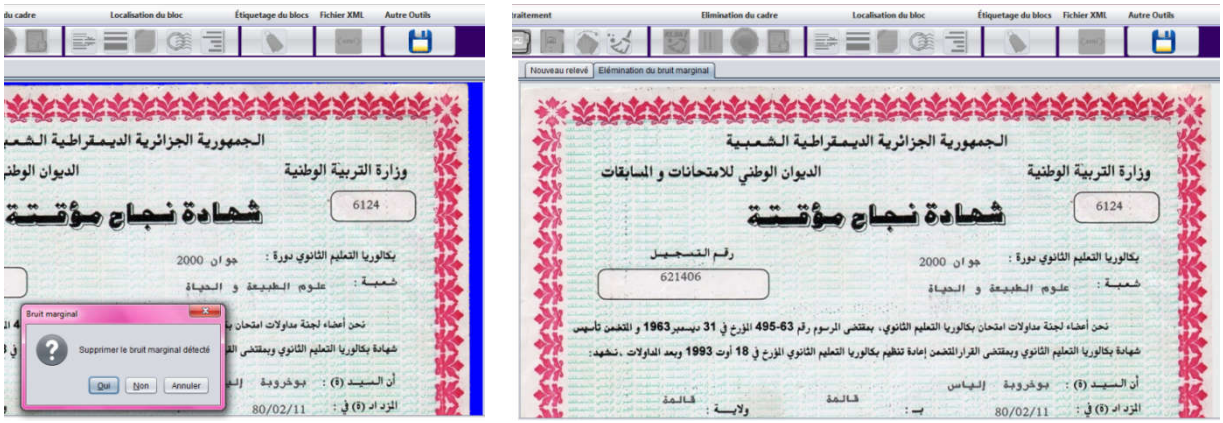
Figure 4.6 : *Chargement d'une image, (a) Exemple d'une image de test, (b) image chargée*

4.2. Prétraitement des relevés

Après le chargement de l'image, vient à l'étape de prétraitement. Les traitements de cette étape est accessible depuis la barre de menu ou utilisé la barre d'outils directement On commence par l'élimination du bruit marginal, transformation en niveau de gris, binaraisation et enfin l'élimination de bruit.

4.2.1. Détection et élimination de bruit Marginal

L'image est chargée et le premier bouton activé est « Élimination du bruit Marginal ». Cette commande permet de détecter et de supprimer le bruit marginal de l'image originale. Si le bruit existe, l'image sur laquelle le bruit coloré en bleu s'affiche dans un nouvel onglet intitulé «Détection de bruit » et une boite de dialogue sera affichée nous demandant si on veut « Supprimer le bruit marginal détecté ?» (***Figure 4.7.a***). En appuyant sur « Oui », le bruit marginal sera supprimé de l'image originale, et l'image résultante sera affichée dans le même onglet mais le nom de l'onglet va être remplacé par « Elimination de bruit marginal » (***Figure 4.7.b***).



(a) *(b)*

Figure 4.7 : *Elimination du bruit marginal, (a) détection, (b) suppression.*

4.2.2. Transformation en niveaux de gris

En cliquant sur le bouton « Niveaux de gris » de la barre d'outils «Prétraitement», qui sera activé après le résultat précédent, l'image du relevé est transformée en niveaux de gris et affichée dans la zone d'affichage de l'image traitée, comme dans la figure suivante :

Figure 4.8 : *Transformation en niveau de gris*

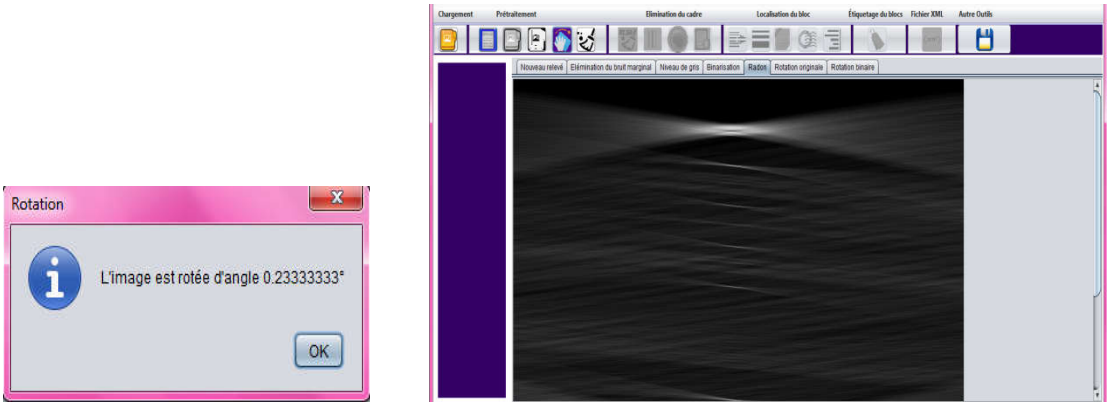
4.2.3. Binarisation

Après d'avoir eu le résultat du bouton précédent, le bouton « Binarisation » sera activé. En cliquant sur lui, l'image binarisée est affichée dans un nouvel onglet comme le montre la figure 4.9.

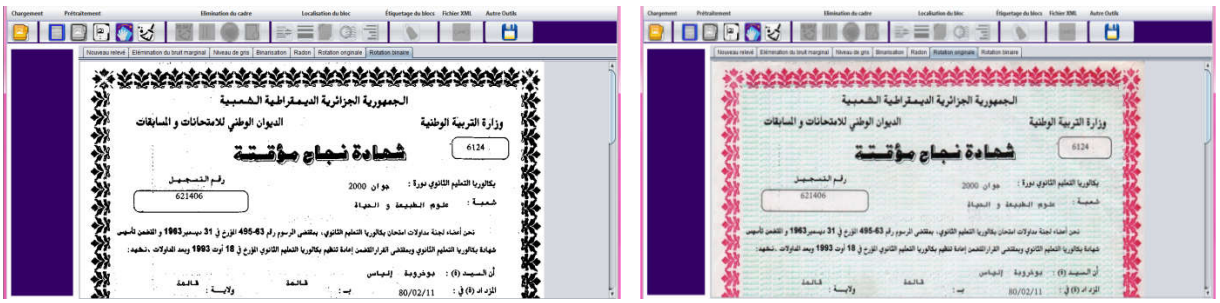
Figure 4.9 : *Résultat de la Binarisation.*

4.2.4. Correction de l'inclinaison

Après l'activation du bouton « Correction de l'inclinaison », deux cas peuvent se produire. Si l'image est bien orienté, un message indiquant que « l'image est bien orientée » sera affiché, sinon un autre message affichant l'angle d'inclinaison est apparait et le résultat s'affiche dans trois nouveaux onglets ; le premier onglet contient l'image de l'espace de radon, le deuxième contient l'image binaire bien rotée, et le dernier affiche l'image originale bien rotée (*Figure 4.10)*.



(a) *(b)*



(c) *(d)*

Figure 4.10: *Affichage du résultat de la correction de l'inclinaison, (a) message informatif, (b) espace de Radon, (c) Image binaire bien rotée, (d) Image originale bien rotée*

4.2.5. Réduction du bruit

Lorsqu'on clique sur « Réduire le bruit », l'image résultante sera affichée dans un nouvel onglet (Voir la ***Figure 4.11***).

Figure 4.11: *Image épurée de bruit (le bruit est en rouge)*

4.3. Élimination de la bordure

La tâche de l'élimination de la bordure regroupe plusieurs traitements. Ces traitements sont accessibles depuis le menu «Élimination du cadre » ou à partir de la barre d'outils «Élimination du cadre ». On Commence par Lissage par RLSA.

4.3.1. Lissage par RLSA

Après l'activation du bouton « Lissage par RLSA», une image binaire contenant la bordure du relevé coloré en noire sera affichée dans un nouvel onglet (***Voir la figure 4.12)***

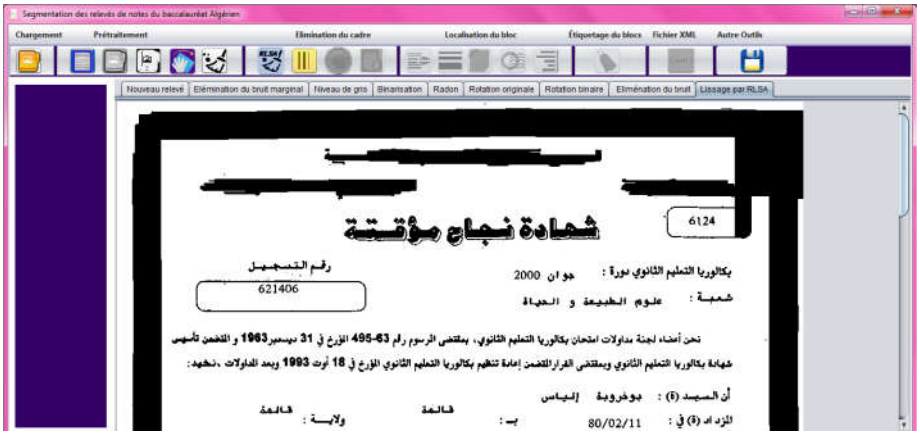


Figure 4.12 : *Image lissée par RLSA*

4.3.2. Raffiner le lissage par RLSA

En cliquant sur le bouton «Raffiner le lissage par RLSA», s'il existe une bordure (existence de ses quatre cotés) , le résultat s'affiche dans un nouvel onglet sinon un message va être affiché indiquant que « l'image est sans cadre ». Voir la figure suivante :

Figure 4.13 : *Raffinement du lissage par RLSA*

4.3.3. Étiquetage des composantes connexe

Dans la figure suivante, chaque composante connexe est étiquetée par une couleur distincte.

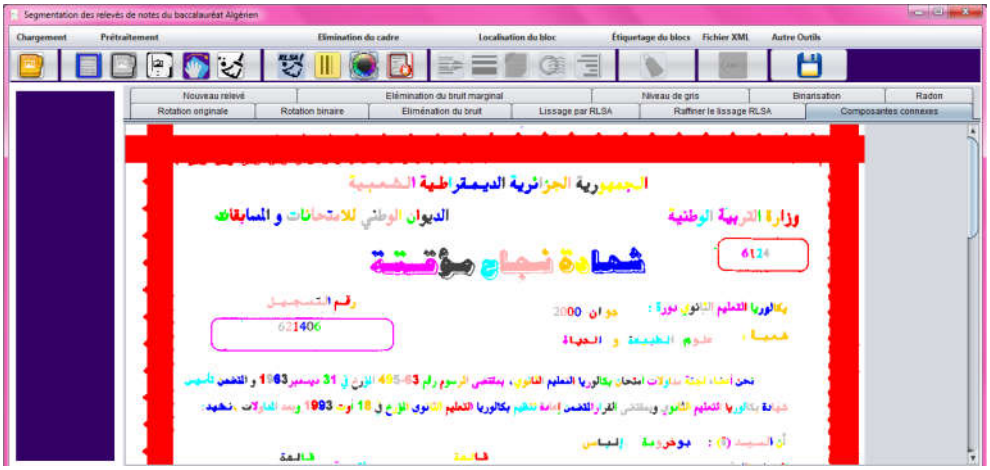


Figure 4.14 : *Étiquetage des composantes connexes*

4.3.4. Élimination du cadre

En cliquant sur le bouton « Élimination du cadre », la bordure est affichée en mauve sur le relevé binaire (***Figure 4.15)***.

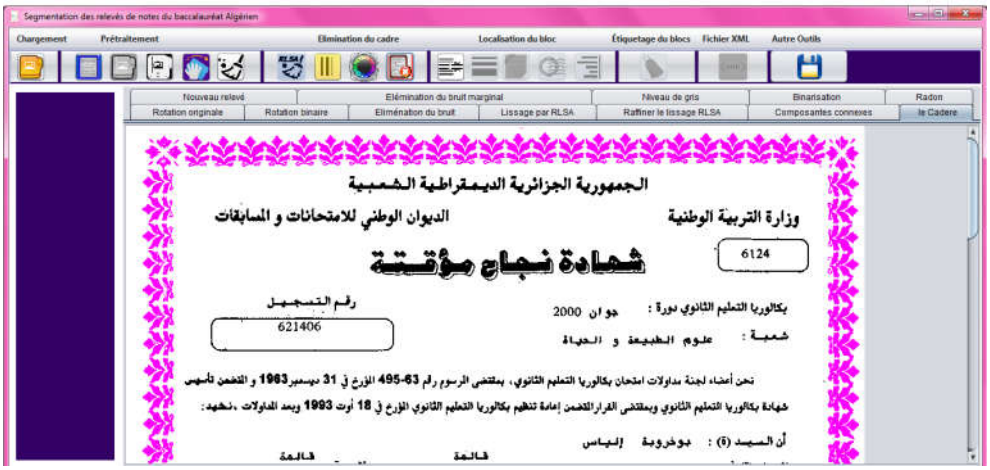


Figure 4.15 : *Cadre du relevé bien détecté*

4.4. Localisation des blocs

Après l'élimination du cadre on passe a l'étape le plus implorante dans notre system c'est l'extraction du bloc. Cette phase inclue plusieurs traitements accessibles depuis le menu « Localisation du bloc » ou depuis la barre d'outils.

4.4.1. Détection des lignes

En cliquant sur le bouton « Segmentation des lignes », l'image résultante sera affichée dans nouvel onglet (***Figure 4.16***).

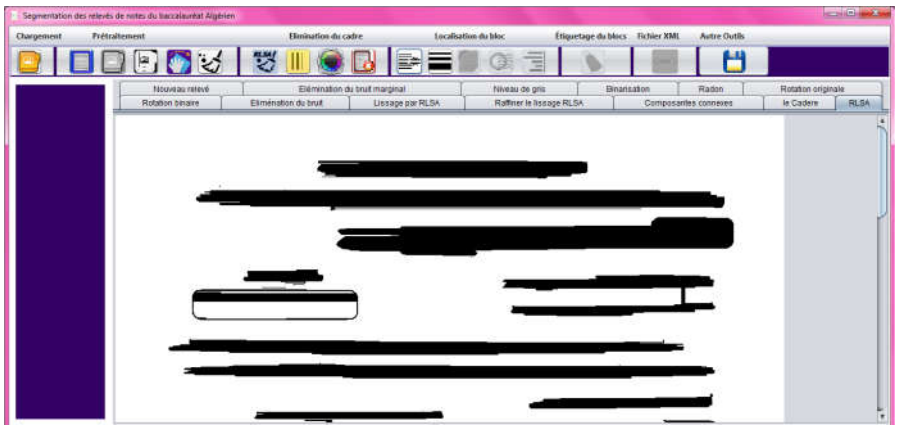
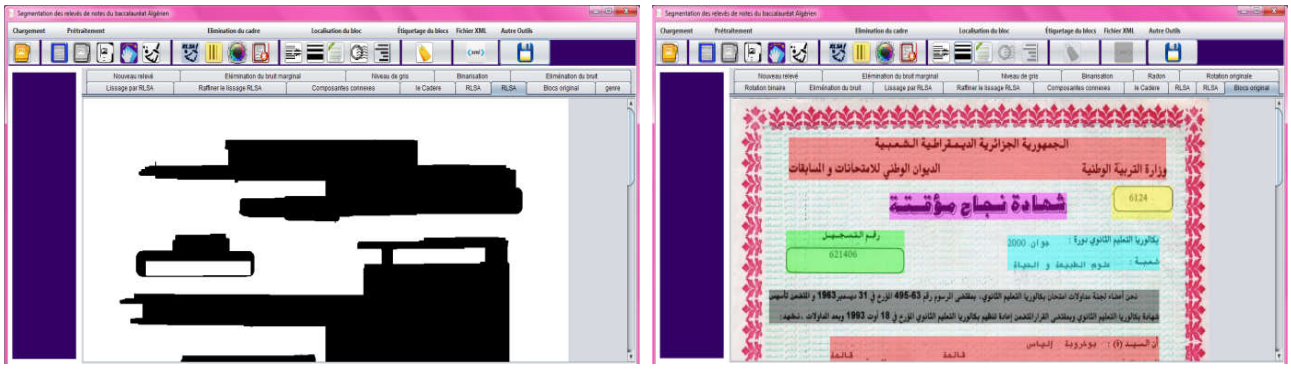


Figure 4. 16: *Détection des lignes par RLSA*

4.4.2. Extraction des blocs

En cliquant sur le bouton « Extraction de blocs », deux nouveaux onglets affichant chacun une image sont apparus. Le premier affiche l'image lissée par RLSA vertical, et l'autre affiche l'image originale sur laquelle les blocs sont colorés (voir la ***figure 4.17***).



(a) *(b)*

Figure 4.17 : *Extraction des blocs, (a) Lissage par RLSA vertical, (b) blocs extraits.*

4.4.3. Identification des blocs textuels et non textuels

Après l'extraction des blocs, nous identifions les blocs textuels des blocs non textuels, en cliquant sur le bouton « Identification des blocs » et l'image résultante sera affichée voir ***figure 4.18.***

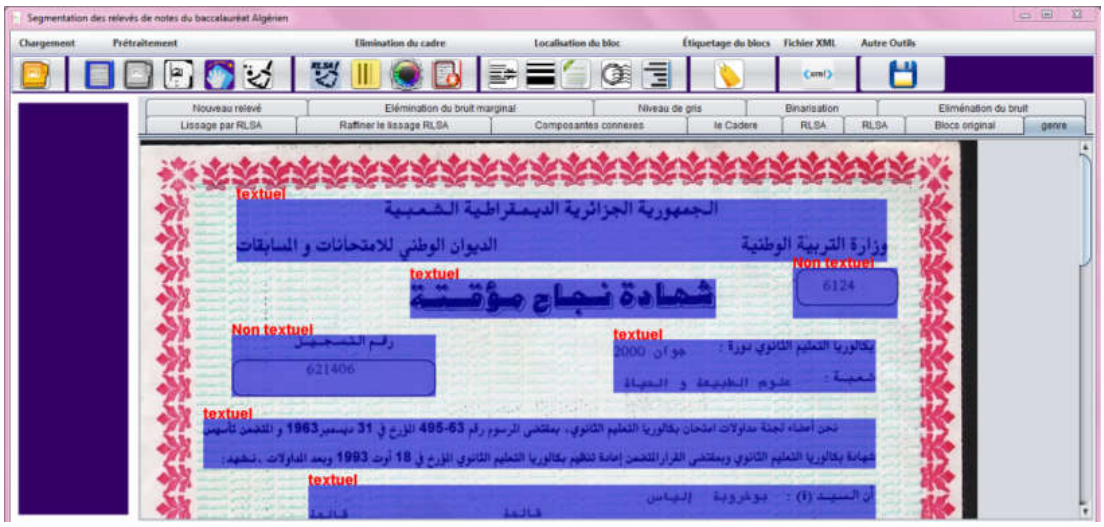


Figure 4.18: *Identification des blocs textuels et non textuels*

4.5. Etiquetage logique des blocs

Cette commande permet l'attribution d'une étiquète à chaque bloc du relevé, en cliquant sur le bouton correspondant, et l'image résultante sera affichée (voir ***figure 4.19***)

Figure 4.19 : *Etiquetage logique des blocs*

4.6. Génération d'un fichier XML.

Après l'étiquetage des blocs nous générons un fichier XML résumant la structure de notre relevé. Cela se fait en cliquant sur le bouton « Génération XML » et une autre fenêtre sera affichée comme le montre la figure suivante :

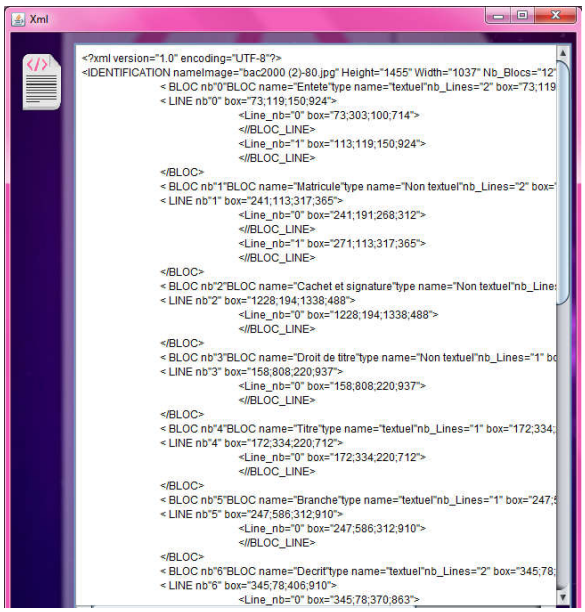


***Figure 4.20**: Fichier XML généré*

5. Expérimentations et résultats

Dans cette section nous présentons les expérimentations menées pour l'évaluation des performance de notre approche proposée.

5.1. Corpus de documents utilisé

Comme nous avons dit précédemment, nous nous intéressons dans le présent travail aux relevés de notes de baccalauréat Algérien. Notre corpus de test utilisé tout au long de notre travail est composé de 40 images de relevés depuis bac « 1997 » jusqu'au bac « 2015». Les documents de notre corpus de test sont de différentes structures et formats rendant leur traitement et analyse difficiles. Des exemples de relevés de notre corpus de test sont présentés par la figure 3.1.

5.2. Résultats et discussions

Le système proposé a été appliqué sur les 40 images de relevés de notes du corpus de test en vue d'évaluer ses performances.

Cependant, nous avons évalué notre système à deux niveaux: au niveau de l'extraction des blocs, et aux niveau de l'étiquetage des blocs extraits.

5.2.1. Evaluation de l'extraction des blocs

Les performances de notre système au niveau de l'extraction des blocs sont mesurées en termes de Rappel, Précision, et F-Mesure.

Notons ***VP***, ***FP***, et ***FN***, le nombre de vrai positifs, faux positifs, et faux négatifs respectivement.

- Un vrai positif est le résultat où le système extrait correctement un bloc.
- Un faux positif est le résultat où le système détecte un bloc qui n'existe pas réellement.
- Un faut négatif est le résultat où le système échoue à détecter un bloc existant.

F-Measure a été introduit la première fois par Chinchor in [CHI 92]. A cause de sa simplicité, *F-measure* est considéré comme l'une des mesures les plus utilisées pour l'évaluation quantitative de la segmentation. *F-measure* est donné par :

$$FM = \frac{2 \times Rappel \times Precision}{Rappel + Precesion}$$

Avec $Rappel = \frac{VP}{VP+FN}$ et $Precision = \frac{VP}{VP+FP}$

F-Measure atteint sa meilleure valeur à **1** (précision et rappel parfaite) et le pire à **0**.

Les résultats d'extraction des blocs pour les 40 images de tests ainsi que les résultats moyens sont récapitulés dans le tableau suivant:

Relevé	Nb de blocs	VP	FP	FN	Rappel	Précision	FM	Relevé	Nb de blocs	VP	FP	FN	Rappel	Précision	FM
1997	11	9	0	2	0,82	1	0,90	**2013**	13	12	0	1	0,92	1	0,96
1998	11	10	0	1	0,91	1	0,95	**2013**	13	12	0	1	0,92	1	0,96
2000	12	12	0	0	1,00	1	1,00	**2013**	13	11	0	2	0,85	1	0,92
2000	12	12	0	0	1,00	1	1,00	**2013**	13	12	0	1	0,92	1	0,96
2001	12	12	0	0	1,00	1	1,00	**2013**	13	11	0	2	0,85	1	0,92
2001	13	12	1	0	1,00	0,92	0,96	**2014**	14	13	0	1	0,93	1	0,96
2001	13	13	0	0	1,00	1	1,00	**2014**	14	14	0	0	1,00	1	1,00
2002	13	11	0	2	0,85	1	0,92	**2014**	14	14	0	0	1,00	1	1,00
2005	13	11	0	2	0,85	1	0,92	**2014**	14	13	0	1	0,93	1	0,96
2007	13	13	0	0	1,00	1	1,00	**2014**	14	14	0	0	1,00	1	1,00
2008	13	12	0	1	0,92	1	0,96	**2014**	13	11	0	2	0,85	1	0,92
2009	14	13	0	1	0,93	1	0,96	**2014**	13	11	0	2	0,85	1	0,92
2010	14	12	0	2	0,86	1	0,92	**2015**	14	13	0	1	0,93	1	0,96
2011	14	12	0	2	0,86	1	0,92	**2015**	14	13	0	1	0,93	1	0,96
2012	14	13	0	1	0,93	1	0,96	**2015**	14	13	0	1	0,93	1	0,96
2013	13	11	0	2	0,85	1	0,92	**2015**	14	11	0	**3**	0,79	1	0,88
2013	13	12	0	1	0,92	1	0,96	**2015**	15	13	0	2	0,87	1	0,93
2013	13	13	0	0	1,00	1	1,00	**2015**	15	11	0	**4**	0,73	1	0,85
2013	13	12	0	1	0,92	1	0,96	**2015**	15	13	0	2	0,87	1	0,93
2013	13	12	0	1	0,92	1	0,96	**2015**	15	13	0	2	0,87	1	0,93
Moyenne													**0,91**	**1,00**	**0,95**

Tableau 4.2. *les résultats de détection des blocs pour toutes les images du corpus de test*

A partir du tableau 4.2 nous avons remarqué que notre système a montré des bonnes performances au niveau de l'extraction de blocs en termes de rappel, précision, et F-Measure. Ainsi, le système à réussi à extraire 91% des blocs existants dans les 40 images de relevés utilisés dans les tests avec une précision parfaite de 100%, et montre ainsi un compromis élevé entre le rappel et la précision (F-Measure = 95%). L'analyse des résultats individuels nous montre que 9 relevés ont produit une

valeur de F-Measure = 1, c'est à dire que la structure physique de 9 relevés parmi 40 a été parfaitement reconnait. De plus, 15 relevés ont une valeur de F-Measure supérieure à 95%.

Finalement, la valeur un peu basse de F-Measure avec certains relevés est causée par des défauts dans les relevés originaux et non pas des lacunes dans notre système.

5.2.2. Evaluation de l'étiquetage des blocs

Sur les blocs extraits correctement de toutes les images de test, lors de la première évaluation, nous appliquons une deuxième évaluation affin d'estimer la capacité de notre système à attribuer correctement des étiquettes ou des rôles aux blocs détectés. En effet, chaque relevé contient plusieurs blocs mais on ne trouve pas deux blocs ayant la même étiquette.

Cependant, les étiquettes assignés par notre système ont été comparées aux étiquettes réelles et les résultats de comparaison sont récapitulés dans la matrice de confusion suivante:

Nombre d'étiquettes assignées par notre système

Nombre d'étiquettes réelles	E	T	B	DT	M	D	IE	TN	TM	DL	S	CS	A	N	Somme
E	40	0	0	0	0	0	0	0	0	0	0	0	0	0	**40**
T	0	40	0	0	0	0	0	0	0	0	0	0	0	0	**40**
B	0	0	40	0	0	0	0	0	0	0	0	0	0	0	**40**
DT	0	0	0	40	0	0	0	0	0	0	0	0	0	0	**40**
M	0	0	0	0	40	0	0	0	0	0	0	0	0	0	**40**
D	0	0	0	0	0	40	0	0	0	0	0	0	0	0	**40**
IE	0	0	0	0	0	0	40	0	0	0	0	0	0	0	**40**
TN	0	0	0	0	0	0	0	40	0	0	0	0	0	0	**40**
TM	0	0	0	0	0	0	0	0	36	0	0	0	0	0	**36**
DL	0	0	0	0	0	0	0	0	0	29	0	0	0	0	**29**
S	0	0	0	0	0	0	0	0	0	0	32	0	0	0	**32**
CS	0	0	0	0	0	0	0	0	0	0	0	40	0	0	**40**
A	0	0	0	0	0	0	0	0	0	0	0	0	32	2	**34**
N	0	0	0	0	0	0	0	0	0	0	0	0	0	25	**25**

E= Entête **M**= Matricule **B**= Branche **DT**= Droit de titre **TN**= Tableau de notes
T= Titre **D**= Décret **S**= Soussigné **DL**= Date et lieu **CS**= Cachet et signature
IE= Informations de l'étudiant **TM**= Tableau de la moyenne **A**= Avertissement **N**= Numéro secret

Tableau 4.3 : *Matrice de confusion présentant les étiquetage assignées par notre système versus les étiquettes réelles.*

A partir de la matrice de confusion précédente, nous avons calculé la précision d'attribution de chacune des étiquettes et les résultats obtenus sont résumés dans le tableau 4.4 suivant.

$$précision\ d'attribution\ de\ l'étiquette\ i = \frac{\text{nb de fois où l'étiquette est assigné correctement}}{nb\ réel\ de\ de\ cet\ étiquette}$$

Etiquette	Précision	Etiquette	Précision
Entete	100%	Tableau de notes	100%
Titre	100%	Tableau de la moyenne	100%
Matricule	100%	Date et lieu	100%
Droit du titre	100%	Soussigné	100%
Branche	100%	Cachet et signature	100%
Décrit	100%	Numéro secret	100%
Informations de l'étudiant	100%	Avertissement	94,11%
		Précision moyenne	**99,58%**

Tableau 4.4 : *Précision d'attribution des étiquettes*

A partir de la matrice de confusion précédente et du tableau 4.4, nous pouvons remarquons que la précision moyenne d'attribution des étiquette atteinte par notre système est de 99,58%, ce qui une valeur très élevée montrant la capacité d'étiquetage logique de notre système.

Ce résultat très satisfaisant est dû à la technique solide d'étiquetage procédé par notre système qui repose sur la connaissance à priori de la position de chaque type de blocs. Cependant, toutes les étiquettes ont été parfaitement attribuées sauf l'avertissement. Ce dernier se confond parfois avec le numéro secret lorsqu' il soit très proche de lui, ou étiqueté en tant que partie du bloc du cachet et signature pour la même raison.

6. Conclusion

Dans ce dernier chapitre nous avons présenté notre travail, de manière globale et détaillée. Ainsi nous avons présenté les différents outils utilisés au développement de l'application. Le scénario de l'utilisation de notre système, les différents résultats obtenus pendant l'évaluation de notre application. Notre application à sa première version comme toute autre application à besoin des améliorations. Ces améliorations restent comme perspectives de notre travail

Conclusion générale et perspectives

Conclusion générale

Les innovations technologiques y compris l'ordinateur et l'informatique engendrent une quantité de documents de plus en plus complexes. Cette grade masse de documents a obligé l'être humain à chercher des moyens pour les exploiter facilement et plus efficacement. Des nouveaux champs de recherches, notamment l'analyse et la reconnaissance de document, et l'archivage électronique ont été nés. Chacun de ces domaines intègre une multitude de pistes de recherche dont la segmentation ou l'extraction de structure physique des documents constitue l'une des plus importants.

Dans ce mémoire, nous avons proposé une approche de segmentation d'un type particulier de documents, à savoir les relevés de notes du baccalauréat algérien. Ces derniers portent une importance considérable dans le dossier de l'étudiant. L'objectif était de développer le noyau d'un système de numérisation, d'analyse, de reconnaissance, et de recherche des archives au sein des universités algériens.

L'approche proposée se situe dans la classe des méthodes hybrides et elle regroupe plusieurs étapes de traitements réunies en deux modules. Un premier module intégrant divers prétraitements, à savoir l'élimination de bruit marginal, binarisation, correction de l'inclinaison, et réduction du bruit, visant à améliorer la qualité des images d'entrée et de les préparer aux étapes suivantes. Le deuxième module vise à l'extraction de la structure physique des relevés en appliquant plusieurs étapes. Premièrement la bordure du relevé est enlevée car elle n'apporte aucune information pertinente. Ensuite, une première segmentation en lignes est effectuée basée sur l'algorithme RLSA. Les lignes sont ensuite réunies en blocs par application de RLSA de nouveau et puis par analyse des profils de projections. Les blocs textuels sont ensuite identifiés et leurs lignes sont extraites. Les blocs non textuels sont séparés en tableaux ou graphiques à l'aide de la transformée de Radon.

Plusieurs tests ont été menés sur une base d'images locale afin d'évaluer les performances du système développé et les résultats obtenus sont encourageants. Ces résultats montrent la fiabilité et la robustesse du système développé et confirment l'efficacité de l'approche de segmentation proposée et la validité des choix pris durant la conception du système.

Notons que le présent travail fait l'objet d'une communication [KEF 19], et l'article a été sélectionné parmi les meilleurs papiers et suggéré pour publication dans une revue internationale (*The International Journal of Informatics and Applied Mathematics*).

Perspectives

Plusieurs perspectives peuvent être envisagées:

- Essayer de réduire le temps d'exécution en révisant certains étapes et en utilisant des techniques de traitement plus légères.
- Généralisation du système proposé pour qu'il soit applicable aux relevés plus complexes (anciens relevés de baccalauréat où les informations sont en manuscrit)

- Etendre l'application développée à la reconnaissance de la structure logique complète des relevés de notes et à identification des informations extraites (localisation de la matricule, identification de la branche d'étude, extraction du nom et prénom, etc.)
- Développement d'un module de reconnaissance de chiffres et de mots et l'intégrer dans notre système.
- Conception d'un réel système d'indexation et de recherche des relevés de baccalauréat. La recherche peut être par image exemple ou par requête textuelle correspondante à un mot ou à une phrase.

Références

[AGR 09] Agrawal, M., & Doermann, D. (2009, July). Voronoi++: A dynamic page segmentation approach based on voronoi and docstrum features. In 2009 10th International Conference on Document Analysis and Recognition (pp. 1011-1015). IEEE.

[AGR 10] Agrawal, M., & Doermann, D. (2010, June). Context-aware and content-based dynamic Voronoi page segmentation. In Proceedings of the 9th IAPR International Workshop on Document Analysis Systems (pp. 73-80). ACM.

[AKI 93] Akindele, O. T., & Belaid, A. (1993, October). Page segmentation by segment tracing. In Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR'93) (pp. 341-344). IEEE.

[AMI 01] Amin, A., & Shiu, R. (2001). Page segmentation and classification utilizing bottom-up approach. International Journal of Image and Graphics, 1(02), 345-361.

[ANT 94] Antonacopoulos, A., & Ritchings, R. T. (1994, October). Flexible page segmentation using the background. In Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3-Conference C: Signal Processing (Cat. No. 94CH3440-5) (Vol. 2, pp. 339-344). IEEE.

[AZO 95] Azokly, A. S. (1995). Une approche uniforme pour la reconnaissance de la structure physique de documents composites fondée sur l'analyse des espaces (Doctoral dissertation, Université de Fribourg).

[BAR 14] Barlas, P., Adam, S., Chatelain, C., & Paquet, T. (2014, April). A typed and handwritten text block segmentation system for heterogeneous and complex documents. In 2014 11th IAPR International Workshop on Document Analysis Systems (pp. 46-50). IEEE.

[BAE 11] Baechler, M., & Ingold, R. (2011, September). Multi resolution layout analysis of medieval manuscripts using dynamic mlp. In 2011 International Conference on Document Analysis and Recognition (pp. 1185-1189). IEEE.

[BEL 01] Belaïd, A. (2001). Reconnaissance automatique de l'écriture et du document. Campus scientifique, Vandoeuvre-Lès-nancy.

[BEN 14] Ben Salah, A. (2014). Maîtrise de la qualité des transcriptions numériques dans les projets de numérisation de masse (Doctoral dissertation).

[BOU 16] Bouressace,H, Zebiri,A.(Juin 2016). Reconnaissance de la structure logique des pages du journal , Mémoire de Master, Université 8 Mai 1945-Guelma.

[BRA 95] Bracewell, R. N. (1995). Two Dimensional Imaging. Englewood Cli s, NJ: Pretence Hall.

[BRE 02] Breuel, T. M. (2002, August). Two geometric algorithms for layout analysis. In International workshop on document analysis systems (pp. 188-199). Springer, Berlin, Heidelberg.

[CAR 15] Carton, C., Lemaitre, A., & Coüasnon, B. (2015, August). Automatic and interactive rule inference without ground truth. In 2015 13th International Conference on Document Analysis and Recognition (ICDAR) (pp. 696-700). IEEE.

[CES 99] Cesarini, F., Gori, M., Marinai, S., & Soda, G. (1999, September). Structured document segmentation and representation by the modified XY tree. In Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318) (pp. 563-566). IEEE.

[CHA 99] Gatos, B., Mantzaris, S. L., Chandrinos, K. V., Tsigris, A., & Perantonis, S. J. (1999, September). Integrated algorithms for newspaper page decomposition and article tracking. In Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318) (pp. 559-562). IEEE..

[CHE 13] Chen, K., Yin, F., & Liu, C. L. (2013, August). Hybrid page segmentation with efficient whitespace rectangles extraction and grouping. In 2013 12th International Conference on Document Analysis and Recognition (pp. 958-962). IEEE.

[CHI 92] Chinchor, N. (1992). MUC-4 Evaluation Metrics. In Proceedings of the Fourth Message Understanding Conference, pp. 22–29.

[CINQ 98] Cinque, L., Lombardi, L., & Manzini, G. (1998). A multiresolution approach for page segmentation. Pattern Recognition Letters, 19(2), 217-225.

[COU 06] Coüasnon, B. (2006). DMOS, a generic document recognition method: Application to table structure analysis in a general and in a specific way. International Journal of Document Analysis and Recognition (IJDAR), 8(2-3), 111-122.

[DRA 18] Drabsia,S. (2018, Juin). Localisation du tableau de notes dans les relevés de notes du BAC, Mémoire de Master, Département d'Informatique, Université 8 Mai 1945-Guelma, Algérie.

[DUO 05] Duong, J. (2005). Étude des documents imprimés: Approche statistique et contribution méthodologique. INSA de Lyon.

[ESK 17] Eskenazi, S., Gomez-Krämer, P., & Ogier, J. M. (2017). A comprehensive survey of mostly textual document segmentation algorithms since 2008. Pattern Recognition, 64, 1-14.

[ESP 95] Esposito, F., Malerba, D., & Semeraro, G. (1995, August). A knowledge-based approach to the layout analysis. In Proceedings of 3rd International Conference on Document Analysis and Recognition (Vol. 1, pp. 466-471). IEEE.

[FAU 09] Faure, C., & Vincent, N. (2009, January). Simultaneous detection of vertical and horizontal text lines based on perceptual organization. In Document Recognition and Retrieval XVI (Vol. 7247, p. 72470M). International Society for Optics and Photonics.

[FIS 90] Fisher, J. L., Hinds, S. C., & D'Amato, D. P. (1990, June). A rule-based system for document image segmentation. In [1990] Proceedings. 10th International Conference on Pattern Recognition (Vol. 1, pp. 567-572). IEEE.

[GAR 11] Garg, R., Hassan, E., Chaudhury, S., & Gopal, M. (2011, September). A crf based scheme for overlapping multi-colored text graphics separation. In 2011 International Conference on Document Analysis and Recognition (pp. 1215-1219). IEEE.

[GHA 16] Ghanmi, N. (2016). Segmentation d'images de documents manuscrits composites: application aux documents de chimie (Doctoral dissertation).

[GOV 90] Govindaraju, V., Lam, S. W., Niyogi, D., Sher, D. B., Srihari, R., Srihari, S. N., & Wang, D. (1989, December). Newspaper image understanding. In International Conference on Knowledge Based Computer Systems (pp. 375-384). Springer, Berlin, Heidelberg.

[HAD 01] Hadjar, K., Hitz, O., & Ingold, R. (2001, September). Newspaper page decomposition using a split and merge approach. In Proceedings of Sixth International Conference on Document Analysis and Recognition (pp. 1186-1189). IEEE.

[HAD 06] Hadjar, K. (2006). Une étude de l'évolutivité des modèles pour la reconnaissance de documents arabes dans un contexte interactif (Doctoral dissertation, Université de Fribourg).

[HAL 95] Ha, J., Haralick, R. M., & Phillips, I. T. (1995, August). Document page decomposition by the bounding-box project. In Proceedings of 3rd International Conference on Document Analysis and Recognition (Vol. 2, pp. 1119-1122). IEEE.

[JOU 07] Journet, N., Mullot, R., Eglin, V., & Ramel, J. Y. (2006, September). Analyse d'images de documents anciens: Catégorisation de contenus par approche texture.

[KEF 19] Kefali, A., Obeizi, A., and Ferkous, C. , (Juin 2019) Segmentation of Algerian baccalaureate transcripts ,In19th IAM Conference, Guelma – Algeria.

[KET 10] Ketata, D., & Khemakhem, M. (2010, March). Un survol sur l'Analyse et la Reconnaissance de Documents: Imprimé, Ancien et Manuscrit.

[LEB 92] Lebourgeois, F., Bublinski, Z., & Emptoz, H. (1992, August). A fast and efficient method for extracting text paragraphs and graphics from unconstrained documents. In Proceedings., 11th IAPR International Conference on Pattern Recognition. Vol. II. Conference B: Pattern Recognition Methodology and Systems (pp. 272-276). IEEE.

[LEE 01] Lee, S. W., & Ryu, D. S. (2001). Parameter-free geometric document layout analysis. IEEE Transactions on pattern analysis and machine intelligence, 23(11), 1240-1256.

[LEL 07] LELORE, T. (2007). Segmentation d'image.

[LEM 07] Lemaitre, A., Camillerapp, J., & Couasnon, B. (2007, September). Contribution of multiresolution description for archive document structure recognition. In Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) (Vol. 1, pp. 247-251). IEEE.

[LEM 08] Lemaitre, A., Camillerapp, J., & Coüasnon, B. (2008). Multiresolution cooperation makes easier document structure recognition. International Journal of Document Analysis and Recognition (IJDAR), 11(2), 97-109.

[LIU 96] Liu, J., Tang, Y. Y., He, Q., & Suen, C. Y. (1996, August). Adaptive document segmentation and geometric relation labeling: Algorithms and experimental results. In Proceedings of 13th International Conference on Pattern Recognition (Vol. 3, pp. 763-767). IEEE.

[LOU 09] Louloudis, G., Gatos, B., Pratikakis, I., & Halatsis, C. (2009). Text line and word segmentation of handwritten documents. Pattern Recognition, 42(12), 3169-3183.

[MIC 00] MICHARD. (2000). Finding groups in data .Eyrolles.

[MON 11] Montreuil, F. (2011). Extraction de structures de documents par champs aléatoires conditionnels: application aux traitements des courriers manuscrits (Doctoral dissertation).

[NAG 92] Nagy, G., Seth, S., & Viswanathan, M. (1992). A prototype document image analysis system for technical journals. Computer, 25(7), 10-22.

[NAM 07] Namboodiri, A. M., & Jain, A. K. (2007). Document structure and layout analysis. In Digital Document Processing (pp. 29-48). Springer, London.

[NAG 84] Nagy, G., & Seth, S. C. (1984). Hierarchical representation of optically scanned documents.

[NAG 00] Nagy, G. (2000). Twenty years of document image analysis in PAMI. IEEE Transactions on Pattern Analysis & Machine Intelligence, (1), 38-62.

[NIK 10] Nikolaou, N., Makridis, M., Gatos, B., Stamatopoulos, N., & Papamarkos, N. (2010). Segmentation of historical machine-printed documents using adaptive run length smoothing and skeleton segmentation paths. Image and Vision Computing, 28(4), 590-604.

[OGO 93] O'Gorman, L. (1993). The document spectrum for page layout analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(11), 1162-1173.

[OTS 79] Otsu, N. (1979). A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics, 9(1), 62-66.

[OUW 12] Ouwayed, N., & Belaïd, A. (2012). A general approach for multi-oriented text line extraction of handwritten documents. International Journal on Document Analysis and Recognition (IJDAR), 15(4), 297-314.

[PAV 92] Pavlidis, T, & Jiangying Z. (1992). Page segmentation and classification. CVGIP: Graphical models and image processing 54.6,484-496.

[PEN 13] Peng, X., Setlur, S., Govindaraju, V., & Sitaram, R. (2013). Handwritten text separation from annotated machine printed documents using Markov Random Fields. International Journal on Document Analysis and Recognition (IJDAR), 16(1), 1-16.

[ROB 01] Robadey, L. (2001). 2 (CREM): Une méthode de reconnaissance structurelle de documents complexes basée sur des patterns bidimensionnels (Doctoral dissertation).

[SAI 92] Saitoh, T., & Pavlidis, T. (1992, September). Page segmentation without rectangle assumption. In Proceedings., 11th IAPR International Conference on Pattern Recognition. Vol. II. Conference B: Pattern Recognition Methodology and Systems (pp. 277-280). IEEE.

[SAR 11] Sarkar, R., Moulik, S., Das, N., Basu, S., Nasipuri, M., & Kundu, M. (2011, November). Suppression of non-text components in handwritten document images. In 2011 International Conference on Image Information Processing (pp. 1-7). IEEE.

[SHA 08] Shafait, F., Van Beusekom, J., Keysers, D., & Breuel, T. M. (2008, September). Structural mixtures for statistical layout analysis. In 2008 The Eighth IAPR International Workshop on Document Analysis Systems (pp. 415-422). IEEE.

[SHI 04] Shi, Z., & Govindaraju, V. (2004, January). Line separation for complex document images using fuzzy runlength. In First International Workshop on Document Image Analysis for Libraries, 2004. Proceedings. (pp. 306-312). IEEE.

[SHI 05] Shi, Z., & Govindaraju, V. (2005, August). Multi-scale techniques for document page segmentation. In Eighth International Conference on Document Analysis and Recognition (ICDAR'05) (pp. 1020-1024). IEEE.

[SPI 90] Spitz, A. L. (1990). Recognition processing for multilingual documents. Proc. of International Conference on Electronic Publishing, Document Manipulation and Typography, Gaithersburg, Maryland. (pp. 193-205).

[STA 09] Stamatopoulos, N., Gatos, B., & Perantonis, S. J. (2009). A method for combining complementary techniques for document image segmentation. Pattern Recognition, 42(12), 3158-3168.

[SUN 06] Sun, H. M. (2006). Enhanced constrained run-length algorithm for complex layout document processing. International Journal of Applied Science and Engineering, 4(3), 297-309.

[SYL 95] Sylwester, D., & Seth, S. (1995, August). A trainable, single-pass algorithm for column segmentation. In Proceedings of 3rd International Conference on Document Analysis and Recognition (Vol. 2, pp. 615-618). IEEE.

[TRA 15] Tran, T. A., Na, I. S., & Kim, S. H. (2015, January). Hybrid page segmentation using multilevel homogeneity structure. In Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication (p. 78). ACM.

[TRA 16] Tran, T. A., Na, I. S., & Kim, S. H. (2016). Page segmentation using minimum homogeneity algorithm and adaptive mathematical morphology. International Journal on Document Analysis and Recognition (IJDAR), 19(3), 191-209.

[TRU 05] Trupin, É. (2005). 01-La reconnaissance d'images de documents: Un panorama.

[VIS 92] Viswanathan, M. (1992). Analysis of scanned documents—A syntactic approach. In Structured Document Image Analysis (pp. 115-136). Springer, Berlin, Heidelberg.

[WAH 82] Wahl, F. M., Wong, K. Y., & Casey, R. G. (1982). Block segmentation and text extraction in mixed text/image documents. Computer graphics and image processing, 20(4), 375-390.

[WAN 15] Wang, Y., Zhou, Y., & Tang, Z. (2015, August). Comic frame extraction via line segments combination. In 2015 13th International Conference on Document Analysis and Recognition (ICDAR) (pp. 856-860). IEEE.

[YAM 96] Yamashita, A, & al. (1996). A document recognition system and its applications. IBM journal of research and development 40.3, 341-352.

[WEB 1] https://fr.wikipedia.org/wiki/Document#Typologie_documentaire, **Consulté le 13/02/2019**

[WEB 2] https://sites.google.com/site/lizantchristopher/services/binarisation-1, **Consulté le 28/06/2019**

[WEB 3]https://www.techopedia.com/definition/3927/java**, Consulté le 08/07/2019**

[WEB 4] http://dictionnaire.sensagent.leparisien.fr/matrice%20de%20confusion/fr-fr/ **Consulté le 10/07/2019**