Ecole Polytechnique de l'Université de Tours
Département informatique
64 avenue Jean Portalis
37200 TOURS

07/10/2004

RAPPORT DE STAGE 2004

Segmentation et analyse structurelle interactives de documents imprimés anciens

~ AGORA 2.0 ~

JY RAMEL Laboratoire Informatique Polytech tours 64 avenue Jean Portalis 37200 TOURS		Stéphane LERICHE EPU DI 3ème année / DEA 2004

1 INTRODUCTION 1

2 CONTEXTE 2

3 CARACTERISTIQUES DES OUVRAGES ANCIENS 3

3.1 Techniques de réalisation 3

3.2 Techniques d'acquisition 4

3.3 Caractérisation 4

4 EVALUATION DES METHODES EXISTANTES 5

4.1 Méthodes ascendantes 5
4.1.1 Filtrages morphologiques et différentiels 5
4.1.2 Méthodes basées sur les composantes connexes 6

4.2 Méthodes descendantes 7
4.2.1 Projections horizontales et verticales 7
4.2.2 Méthodes à base de pavages 7

4.3 Méthodes mixtes 8
4.3.1 Méthode DMOS 8

4.4 Bilan 9

5 VUE D'ENSEMBLE DU PROJET 10

6 NUMERISATION ET RESTAURATION 11

7 EXTRACTION DES COMPOSANTES CONNEXES 12

7.1 Décompression 13

7.2 Conversion en 256 niveaux de gris 13

7.3 Binarisation 13

7.4 Détection des contours 13

7.5 Opérateurs morphologiques 14

8 SEGMENTATION PAR FUSION DES COMPOSANTES CONNEXES 15

8.1 Observations et propositions 15
8.1.1 Observation 1 15
8.1.2 Problème1 15

8.1.3 Proposition1 16
8.1.4 Observation2 16
8.1.5 proposition2 16

8.2 Principe retenu 16

8.3 Traitements effectués sur les composantes connexes 18
8.3.1 La structure zone 18
8.3.2 Etude de la taille des composantes connexes 19

8.4 Création de la carte des niveaux de gris 21

8.5 La fusion 22

9 ROBUSTESSE DE LA FUSION A LA ROTATION 24

10 SEGMENTATION TEXTE / IMAGE 26

10.1 Problème posé 26

10.2 Principe de la méthode 27

10.3 Illustration de la méthode 28

10.4 conclusion 30

11 LA CLASSIFICATION SUPERVISEE 31

11.1 Introduction 31

11.2 Les différentes classes 32

11.3 Expertise sur les documents 33

11.4 Classification selon la position géographique 34

11.5 Classification selon les relations de voisinage 35
11.5.1 Titre principal 35
11.5.2 Numéro de page 35
11.5.3 Réclame 36
11.5.4 Marges 36
11.5.5 Lettrines 36

11.6 Classification selon la forme et la texture des zones 37

11.7 Stratégie de classification 37

11.8 De la stratégie vers le scénario… 38

12 MISE EN ŒUVRE DES SCENARIOS 39

12.1 Présentation 39

12.2 Les règles de classification 39
12.2.1 Classification géographique 40
12.2.2 Classification par relation de voisinage 41
12.2.3 Classification en fonction de la forme et les caractéristiques intrinsèques 43

12.3 Les règles de fusion 45

12.4 Création et destruction de types 45
12.4.1 Création d'un nouveau type de zone 45
12.4.2 Destruction d'un certain type de zone 46

13 EXEMPLE DE RESULTATS ET TESTS 47

13.1 Exemple de scénario 47

13.2 Exemple de résultat 49

13.3 Campagne de tests 57

14 CONCLUSION ET EVOLUTIONS 58

15 BIBLIOGRAPHIE 59

16 ANNEXE A : MANUEL PROGRAMMEUR 60

16.1 Diagramme UML 60

16.2 CElement 61

16.3 CZone 61

16.4 CListeCC 62

16.5 CImage 63

16.6 CAtelier 63

16.7 CConcept 64

16.8 CTypeClasse 64

16.9 Cscenario 64

16.10 Cregle 64

17 ANNEXE B : MANUEL D'UTILISATION 66

17.1 Traitement d'une image 66
17.1.1 Ouvrir une image 66

17.1.2 Traitements prédéfinis 67

17.2 Traitement par lot 71

17.3 création des scénarios 72
17.3.1 menu « classification » 72
17.3.2 menu « scenario » 77

17.4 autres traitements 77
17.4.1 menu « Traitements » 77
17.4.2 menu « carte de ndg » 80

1 Introduction

Une collaboration avec le CESR de Tours nous donne l'opportunité de travailler sur la numérisation et l'indexation d'ouvrages anciens (datant de la Renaissance).

En effet, une des problématiques actuelle qui préoccupe une grande majorité de bibliothèques est la mise à disposition de leurs ouvrages au public le plus large possible, notamment par le biais d'Internet. De plus, certains ouvrages ne peuvent pas être manipulés par le grand public à cause de leur état d'ancienneté.

On parle de conservation et valorisation du patrimoine.

Le CESR (centre d'études supérieures de la renaissance) de Tours à été créé en 1956 et était alors rattaché à l'université de Poitiers. C'est en 1970 qu'il devient une UFR au sein de l'université de tours. Le CESR possède un fonds d'ouvrages anciens estimé à quelques 3000 ouvrages. Ces derniers couvrent la période de la renaissance.

Ce fonds est constitué d'une grande variété d'ouvrages : philosophiques, juridiques, scientifiques, littéraires…

Ce type de projet concerne directement les équipes de recherche en reconnaissance des formes et plus encore celles qui travaillent sur l'analyse de documents. Notre travail de recherche concerne l'extraction de la structure physique et logique de ces documents à partir de la version numérisée de chacune des pages afin d'automatiser leur conversion au format XML.

Ces recherches s'inscrivent dans le cadre de l'ACI Madonne (ACI Masse de données), labellisée et financée par le ministère de la recherche (septembre 2003 – septembre 2006) mettant en collaboration de nombreux laboratoires de recherche français.

C'est donc après avoir répertorié les caractéristiques de ces ouvrages, puis évalué les techniques existantes que nous présenterons la technique que nous avons développé et les résultats obtenus.

2 Contexte

Ce travail s'inscrit dans la continuité d'un PFE réalisé par Gaëlle LEROUX puis d'un stage effectué par Christophe GALLANT. Il convient de dresser un bilan des travaux réalisés afin d'exploiter et tirer les conséquences de ce qui a été mis en oeuvre. L'étape préliminaire de ce projet a donc consisté à comprendre, tester et analyser l'application baptisée AGORA délivrée au CESR l'année dernière.

Cette version proposait de travailler sur les composantes connexes et d'en déduire les informations suivantes :

- Extraction des images par analyse de la taille des composantes connexes.
- Extraction des marges et blocs de textes par noircissement des composantes connexes puis analyse des histogrammes horizontaux et verticaux.

Après un certain nombre de tests, nous tirons les conséquences suivantes :

- Les composantes connexes constitueront notre base de travail
- L'extraction des images fonctionne dans la plupart des cas et ne fera l'objet que de modifications mineures
- La technique de détection des marges et des blocs de textes sera abandonnée car elle est mise en échec dans un très grand nombre de cas, nécessite un trop grand nombre de paramètres et la connaissance préalable de la structure de la page.

Une partie non négligeable du projet a été dédiée à la refonte du code à disposition pour

- lui donner une forme plus orientée objet de façon à faciliter la suite de la programmation, l'exploitation des méthodes et à améliorer la « réutilisabilité » du code
- Répondre aux exigences de programmation basées sur l'architecture document/vue de Visual C++
- Utiliser les structures et objets fournis par Visual C++ de façon à disposer de fonctions puissantes et éprouvées.

Le développement de ce projet étant effectué en collaboration avec le CESR, il a fallut gérer l'évolution de deux versions du programme :

- Une version dédiée au développement qui a servi à élaborer et tester les nouvelles méthodes mises en oeuvre
- Une version livrée régulièrement au CESR qui comprenait l'implémentation des techniques validées ainsi que l'adaptation du programme à leurs exigences spécifiques.

3 Caractéristiques des ouvrages anciens

3.1 Techniques de réalisation

Depuis sa création, le CESR a constitué un fonds d'ouvrages anciens qui compte actuellement environ 3000 ouvrages, couvrant la période s'étendant du milieu du XIVe siècle au début du XVIIe siècle. Les premiers ouvrages remontent aux débuts de l'imprimerie. Les polices utilisées, la présentation des pages et l'utilisation de l'espace étaient alors très proches de celles des ouvrages manuscrits. Les ouvrages plus récents ont profité des progrès technologiques. De plus, le fonds du CESR provient de toute l'Europe : France, Allemagne, Italie, Suisse, Hollande. Les langues des ouvrages sont le plus souvent le latin ou le français, ce qui amène un facteur supplémentaire de variabilité pour les ouvrages. La typographie des caractères des manuscrits médiévaux a une variabilité de forme importante. Lorsque la lisibilité du manuscrit n'est pas suffisante, le travail de paléographe (la transcription) qui se fait au crayon, est nécessaire. Des exemples d'images d'ouvrages anciens sont présentés figure 2.

Au niveau de la mise en page, les contraintes techniques imposaient également une présentation particulière. Les écrits ont généralement, pour un ouvrage donné, la même mise en page et les mêmes structures bien que la variabilité soit beaucoup plus grande que dans les ouvrages actuels. On y retrouve un corps de texte qui prend la majorité de la page, des marges ou des annotations de chaque côté du texte. La page peut aussi contenir des zones graphiques de différentes tailles et des lettrines. Concernant le texte, on retrouve des structures connues comme les titres et sous-titres, les paragraphes, les numéros de page…

Le style employé peut différer alternant un style normal justifié ou aligné à gauche. Une autre particularité provient des faibles séparations existant entre les différents blocs de texte (notes en marge et le corps du texte). Enfin, sur certaines pages, les règles de mise en page classiques ne sont pas respectées : par exemple une illustration peut déborder sur les marges (figure 2). Dans les ouvrages de la Renaissance, les illustrations ont généralement été imprimées à l'aide de plaques de bois ou de métal, gravées avec l'image à reproduire et encrées. Elles sont généralement de structure rectangulaire et incluses dans un rectangle blanc qui peut être entouré de texte.

Figure 1 : exemples de pages d'ouvrages

3.2 Techniques d'acquisition

D'autres difficultés sont dues aux procédés de numérisation employés. Elles concernent les défauts d'éclairage dans la reliure, la courbure des lignes de texte, l'inclinaison des pages, et l'élimination des taches. De nombreux travaux de recherche ont été menés pour corriger ces défauts [Debora]. Des solutions commerciales existent et sont même considérées comme satisfaisantes (Book Restorer) même si des problèmes subsistent. Par exemple, la correction des défauts de courbure peut entraîner une dégradation sur les frontières des blocs de texte et des images. Nous avons choisi de ne pas aborder cette problématique dans ce travail.

3.3 Caractérisation

Ces quelques remarques nous permettent de dresser une liste de caractéristiques dont il est indispensable de tenir compte lors de la conception d'algorithmes d'extraction de la structure physique de documents anciens. Voici celles qui nous ont paru les plus importantes :

- Mise en page complexe qui peut présenter plusieurs colonnes utilisant des tailles de corps et d'interligne différentes
- Faible espacement entre les lignes provoquant des contacts entre caractères.
- Faible espacement entre blocs de texte
- Présence de notes en marges imprimées ou manuscrites
- Présence d'indicateurs de repérage : numéros de lignes, de pages, réclames, …
- Utilisation de fontes particulières
- Usage fréquent d'ornements (zones non textuelles) tels que les bandeaux, lettrines, enluminures, …
- Disposition fluctuante des illustrations graphiques et des légendes associées
- Positionnement anarchique du texte par rapport aux illustrations
- Images fortement dégradées même après restauration

La typographie et la technologie de l'imprimerie ont depuis fait énormément de progrès et les ouvrages actuels répondent à des normes bien différentes en matière de présentation.

Les logiciels conçus pour reconnaître les documents actuels, notamment les logiciels d'OCR commerciaux (Omnipage, easy reader…), s'avèrent donc bien souvent médiocres sur les ouvrages de la Renaissance. (Voir PFE 2002-2003 Gaëlle Leroux)

4 Evaluation des méthodes existantes

Les méthodes d'extraction de structures peuvent être classées en 3 grandes catégories : les méthodes ascendantes, descendantes et mixtes [Belaid97]. Avant de décrire notre système, nous présentons les résultats d'une campagne d'expérimentation qui nous a permis de valider ou non les différentes approches sur ce type de document.

4.1 Méthodes ascendantes

Pour être classée parmi les méthodes ascendantes, une méthode doit travailler à partir des pixels. On retrouve donc dans cette catégorie, les méthodes se basant sur des techniques de filtrages morphologiques ou différentiels et sur l'étude des composantes connexes de l'image.

4.1.1 Filtrages morphologiques et différentiels

Ce type de méthodes a été beaucoup mis en œuvre et testé dans le cadre du projet Débora pour fusionner les caractères en mots puis en lignes et pour l'élimination du bruit [Debora]. La figure 3 illustre les résultats pouvant être obtenus à l'aide de ce type de méthodes. Le classique algorithme RLSA peut également être utilisé, il a un fonctionnement similaire et fournit des résultats identiques à ceux des méthodes morphologiques.

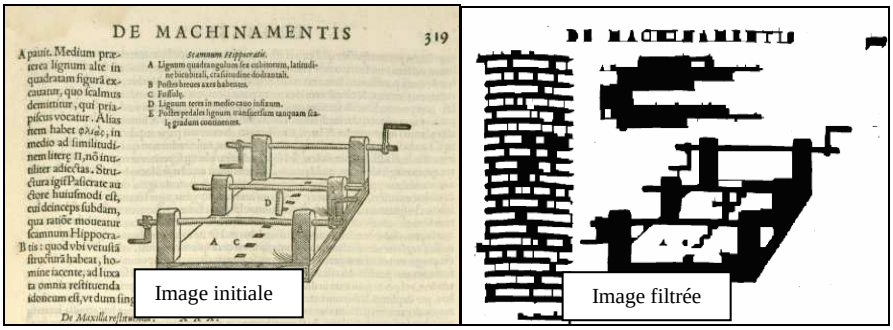


Figure 2 : filtrage morphologique sur image binarisée

Il est également possible de travailler directement sur les images en niveaux de gris par filtrage différentiel. L'idée est d'utiliser des filtres permettant d'agglomérer les variations d'intensité périodiquement produites par les contours des caractères puis de rechercher des alignements horizontaux pour les lignes de textes.

Les problèmes que soulève ce type de méthodes viennent des nombreux paramètres nécessaires et difficiles à régler ainsi que des temps de calcul souvent prohibitifs. De plus, par définition, ces méthodes ne permettent pas d'utiliser de connaissances a priori sur les documents à traiter.

4.1.2 Méthodes basées sur les composantes connexes

Cette approche consiste à considérer chaque page comme un ensemble de composantes connexes.

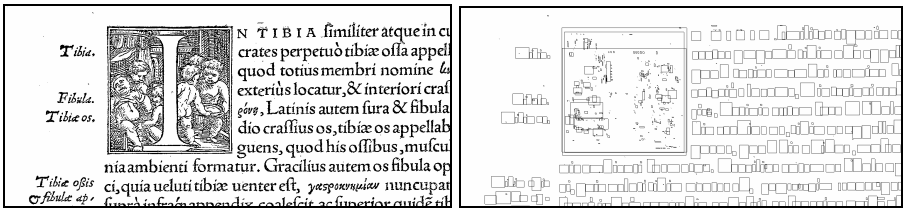


Figure 3 : composantes connexes et segmentation

Comme on peut le constater sur la figure 4, la taille, la proximité, et la position relative des composantes connexes peuvent être utilisées pour extraire la structure physique d'une page. Les rectangles circonscrits aux composantes connexes se chevauchent fréquemment dans les graphiques et rarement dans les textes. Ainsi, les zones graphiques correspondent aux composantes connexes de cet ensemble de rectangles circonscrits dont les dimensions (largeur ou hauteur) dépassent un seuil fixé.

Un bloc de texte peut aussi être vu comme un ensemble de petites composantes connexes « proches ». Deux caractères sont dits voisins si la distance les séparant est inférieure à un espace maximal. O'Gorman a proposé d'utiliser uniquement le voisinage entre composantes pour localiser les zones de texte dans une image [OGorman93]. Dans notre adaptation, pour chaque composante, on recherche dans les quatre directions principales d'autres composantes connexes dans le voisinage défini. Ensuite, un code déterminant l'existence ou non de voisins dans chaque direction peut être associé à chaque composante. Une composante n'ayant pas de voisin dans au moins une direction est une composante de contour d'un bloc. Un chaînage circulaire fermé traduit alors la frontière d'un bloc de texte. (figure 5).

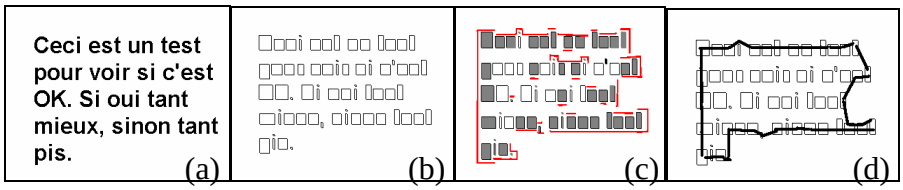


Figure 4 : (a) I*mage initiale, (b) composantes connexes, (c) en gris : composantes de contour, en rouge : côtés sans voisin, (d) chaînage circulaire obtenu*

Comme les méthodes à base de filtrage, ces méthodes sont peu adaptées aux ouvrages anciens à cause de la proximité entre blocs de texte dans les manuscrits anciens. Pour ce qui concerne la localisation des zones graphiques, les seuils doivent correspondre à une taille légèrement supérieure au plus grand caractère contenu dans l'ouvrage. Des études statistiques peuvent permettre d'automatiser la sélection de ces seuils. De plus, ces méthodes supportent difficilement les variabilités dans les fontes de caractères et dans la mise en page des illustrations susceptibles d'apparaître fréquemment dans les manuscrits anciens. La mauvaise qualité des images (bruits, taches, …) pose aussi certains problèmes.

4.2 Méthodes descendantes

Nous venons de voir qu'il était difficile de séparer les blocs de texte d'une page par des critères locaux. Les approches descendantes tentent de localiser les espaces entre blocs de manière plus globale. En effet, deux blocs de texte sont nécessairement séparés par un espace blanc de surface importante, que ce soit dans le sens horizontal ou vertical.

4.2.1 Projections horizontales et verticales

La recherche de ces séparations blanches doit être faite horizontalement et verticalement. Une localisation des séparations horizontales (espaces entre lignes et entre paragraphes), précédant une recherche des séparations verticales sur chacun de ces blocs horizontaux augmente la robustesse de l'analyse.

La localisation des séparations blanches se fait généralement par analyse de la forme de l'histogramme des projections des pixels noirs sur les lignes et les colonnes de l'image. On peut considérer une différence importante entre deux valeurs successives dans l'histogramme comme une délimitation entre deux blocs de texte. Le problème est l'estimation des différences significatives dans l'histogramme. En général, on utilise des connaissances a priori sur le document en cours d'analyse (nombre de colonnes, marges, …) pour localiser plus facilement les séparations. Sur les documents anciens, les tests ont montré qu'il était préférable de noircir l'ensemble des rectangles circonscrits aux composantes connexes de l'image avant de calculer l'histogramme pour avoir de meilleurs résultats.

Un autre problème lié à cette technique concerne la différenciation entre les minima locaux et les minima globaux dans l'histogramme. De plus certaines séparations correspondent à des minima non significatifs (cas des images mal redressées). Il est donc difficile d'utiliser ces méthodes lorsque la mise en page est complexe ou lorsque les pages sont mal redressées.

4.2.2 Méthodes à base de pavages

Pour traiter les mises en page plus complexes, nous avons choisi de localiser les zones blanches en adaptant l'algorithme de split & merge pour qu'il fournisse les zones homogènes blanches dans une image (figure 6). Cette méthode peut être comparée à d'autres techniques à base de pavage plus ou moins évoluées comme la méthode RXY-C [Nagy84] [Akindele93]. Dans tous les cas, c'est l'analyse du voisinage (réalisé avec l'aide de graphes ou de quadtree) de chaque région blanche découverte via le pavage qui permet ensuite de localiser et caractériser les blocs contenus dans la page plus ou moins finement (figure 6).

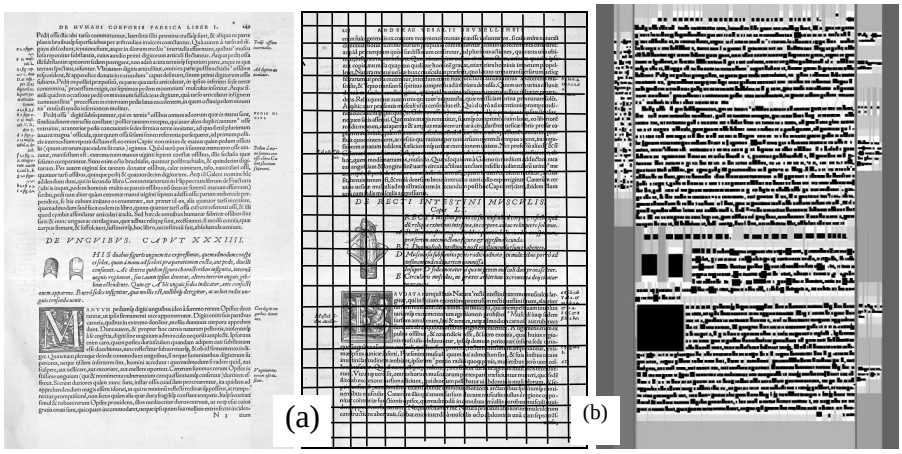


Figure 5 : Pavage RXY-C (a) et Split & Merge (b) pour localiser les zones blanches

Ces méthodes descendantes sont moins sensibles au bruit que les méthodes ascendantes. Lorsque les images sont bien redressées, elles résolvent en partie le problème de proximité entre blocs et entre caractères. Néanmoins, elles fonctionnent mal sur les documents ayant une mise en page fluctuante ou non rudimentaire puisqu'elles nécessitent l'utilisation de connaissance a priori sur le document (nombre de colonnes, de marges, …). Elles ne sont donc pas adaptées au traitement des documents anciens.

4.3 Méthodes mixtes

4.3.1 Méthode DMOS

Cette méthode permet de générer automatiquement un système de reconnaissance de documents structurés [Couasnon].

Elle utilise pour l'aspect descendant :

- Une grammaire pour décrire la structure type du document à analyser
- Un compilateur qui va utiliser cette grammaire pour générer un langage de description du document
- Un module d'analyse

L'aspect ascendant :

- Un module dit de vision précoce qui va permettre :
    - L'application de méthodes de binarisation adaptatives
    - L'extraction des segments rectilignes du document en s'appuyant sur le filtrage de Kalman
    - L'utilisation d'un classifieur qui par apprentissage est capable de reconnaître les symboles

La technique est implémentée dans le logiciel FormuRead qui permet d'extraire automatiquement la structure de formulaires d'incorporation militaire du XIXe siècle. Malgré leur dégradation, 60 000 formulaires de recrutement militaire du XIX^{ème} siècle ont été testés avec succès (99.6%).

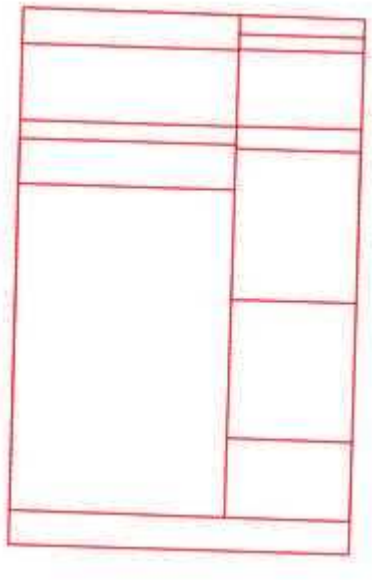


Figure 6 : formulaire avec retombes **Figure 7 : structure correspondante**

4.4 Bilan

Les tests précédents ont mis en évidence les limites des méthodes traditionnelles pour traiter des documents anciens ainsi que les raisons des échecs. Sur cette base, nous avons mis au point une nouvelle méthode exploitant à la fois les atouts des méthodes descendantes et des méthodes ascendantes puisqu'elle se base sur la construction d'une carte des frontières entre blocs présents dans la page (approche descendante) et sur la carte des composantes connexes contenues dans l'image (approche ascendante). Nous proposons ensuite d'utiliser simultanément les informations fournies par ces deux cartes car cela permet de résoudre bon nombre des difficultés que nous avons mentionnées dans notre caractérisation des documents anciens.

5 Vue d'ensemble du projet

On peut diviser le projet en trois grandes étapes :

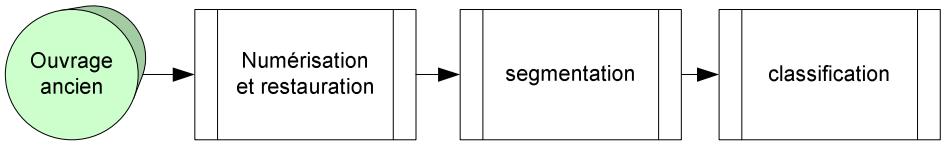


Figure 8 : principales étapes de la chaîne de traitement

Ø Numérisation et restauration

Cette étape consiste en la conversion de l'ouvrage papier en un ouvrage numérique. Les images obtenues subissent alors des traitements de corrections pour mettre à disposition les images les plus propres et les moins déformées possible.

Ø Segmentation

Après avoir extrait les composantes connexes, le but de cette étape est de les fusionner afin de former les plus grandes zones possibles. Une zone est donc un ensemble de composantes connexes ayant la même classe présumée (on ne la connaît pas encore).

Ø Classification

Représente la dernière étape du traitement et consiste à donner une étiquette à chaque zone issue de la segmentation.

La première étape est réalisée par le CESR. Les deux autres font l'objet du logiciel dont le fonctionnement est décrit ici.
Ces différentes étapes sont détaillées dans les paragraphes suivants.

6 Numérisation et restauration

La première étape consiste à numériser chaque page des ouvrages disponibles et à appliquer les techniques habituellement utilisées dans la restauration d'images (correction géométrique).

Cette étape est assurée par Sébastien BUSSON (CESR) qui nous fournit un échantillon des images scannées en haute résolution (300ppp*300ppp) des pages de certains ouvrages.
La manipulation d'ouvrages anciens nécessitant le plus grand soin et notamment le moins de contacts possible, le scanner utilisé est du type suivant :

Figure 9 : exemple de scanner utilisé pour éviter tout contact avec l'ouvrage

Quant aux traitements de restauration, c'est le logiciel « BookRestorer » de I2S qui est utilisé. Celui-ci va permettre, entre autre, de corriger l'erreur de courbure et d'inclinaison des images.

Exemple :

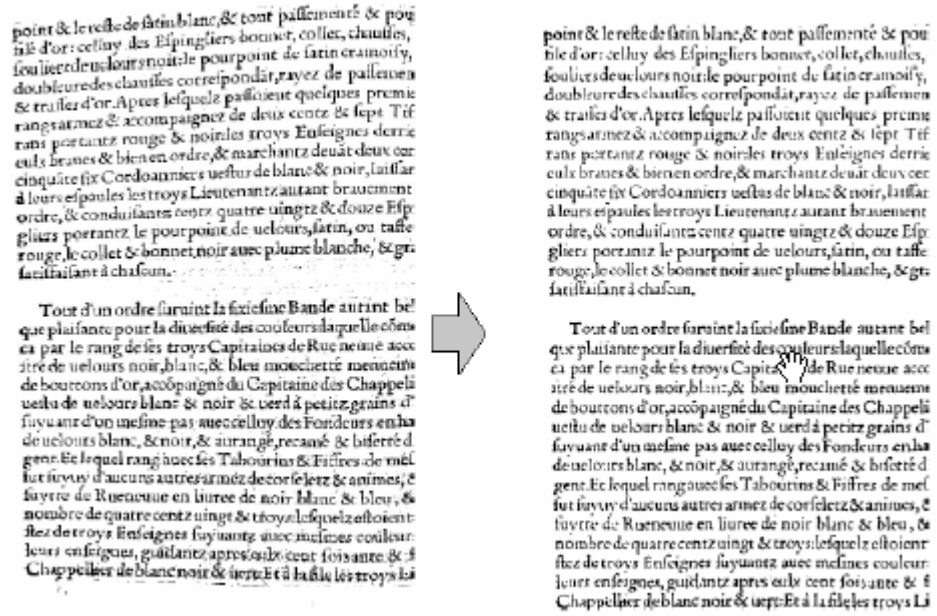


Figure 10 : exemple de page après la correction de la courbure

7 Extraction des composantes connexes

Nous disposons donc d'une image qui est en général au format « jpeg » ou « tiff » et qui peut être en couleur. Ce format correspond à un format compressé qu'il est nécessaire de décoder pour pouvoir effectuer un quelconque traitement. Pour cela, la bibliothèque « paintlib » est utilisée, ce choix étant le fruit du travail du précèdent stagiaire. Avec cette bibliothèque, nous allons donc pouvoir effectuer un certain nombre de traitements de base. Ceux-ci consistent essentiellement en :

- Ø La conversion des différents formats compressés en un format « bmp »
- Ø La conversion de l'image d'origine en 256 niveaux de gris.
- Ø La sauvegarde d'images dans le format souhaité (dans la version de paintlib utilisée, la sauvegarde est uniquement possible en 32bits)

Ci dessous les différents traitements appliqués à l'image d'origine pour obtenir les composantes connexes :

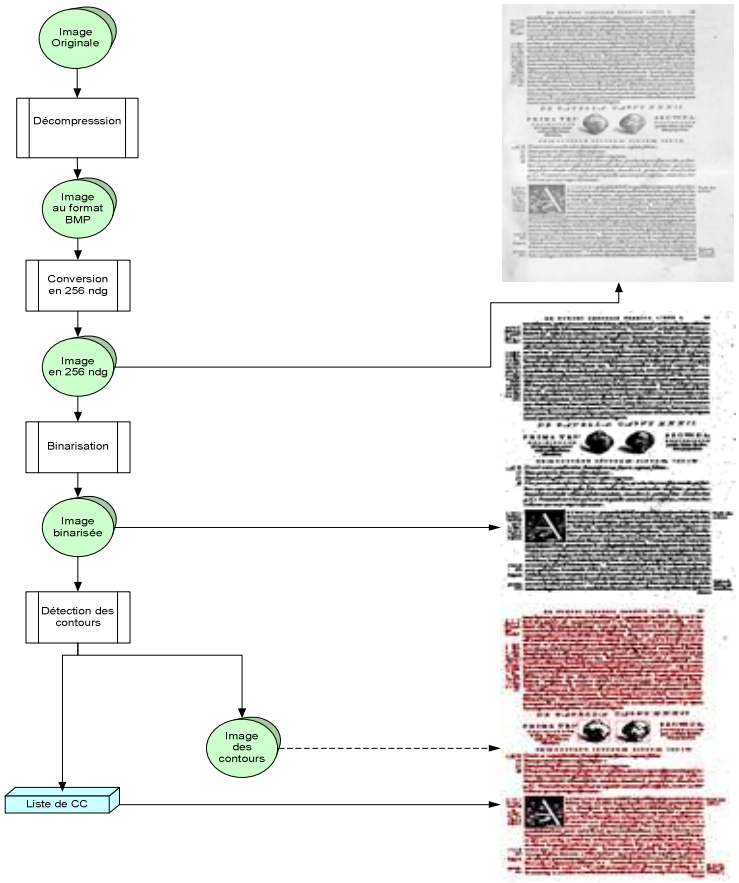


Figure 11 : diagramme fonctionnel illustré de l'étape d'extraction des composantes connexes

7.1 Décompression

Nous disposons d'une image stockée dans un format compressé. Pour pouvoir effectuer des traitements au niveau des pixels de l'image, il faut donc effectuer une décompression de l'image afin de disposer d'une image au format « bmp ». Cette fonction est assurée par une routine de la bibliothèque paintib.

Fonction : ***Charger*** de la classe CImage
Entrée : image couleur dans un format compressé (jpeg, tiff …)
Sortie : image couleur au format « bmp »

7.2 Conversion en 256 niveaux de gris

Les traitements que nous appliquerons à l'image nécessitent de disposer d'une image dont chaque pixel est représenté par une valeur de niveau de gris (le traitement suivant consiste à binariser l'image). Les images dont nous disposons étant en couleur, il faut donc effectuer un filtrage pour assurer la conversion en 256 niveaux de gris. Cette fonction étant intégrée à la bibliothèque paintlib, elle sera utilisée pour effectuer cette tâche.

Fonction : ***ConvertirNdg*** de la classe CImage
Entrée : image couleur au format « bmp »
Sortie : image en 256 niveaux de gris au format « bmp ».

7.3 Binarisation

L'objectif convoité à l'issue de cette première étape est l'obtention d'une liste de composantes connexes. Il faut donc appliquer un algorithme capable de réaliser la détection des contours. Ce type de traitement doit être effectué sur une image binaire. Cette opération est réalisée par un simple seuillage des valeurs de niveau de gris de chaque pixel de l'image. Le choix de la valeur du seuil est pour l'instant laissé à l'utilisateur. Cette valeur est cependant à choisir avec soin car elle détermine l'efficacité de l'algorithme de détection des contours.
Il a été ajouté la possibilité d'effectuer la binarisation de façon automatique. C'est l'algorithme de Niblack qui est chargé de réaliser un seuillage adaptatif de l'image à traiter. Cet algorithme étant relativement long, il proposera une valeur à appliquer en seuillage de base.

Fonction : ***Binariser*** et ***Binariser_Niblack*** de la classe CImage
Entrée : image en 256 niveaux de gris au format « bmp ».
Sortie : image binaire au format « bmp ».

7.4 Détection des contours

Nous ne reviendrons pas ici sur la description de l'algorithme de détection des composantes connexes. En effet, celui-ci était intégré dans la version 1.0 d'agora. L'implémentation est le fruit du travail de Christophe GALANT qui a codé en C++ l'algorithme de détection de contours et d'extraction de composantes connexes déjà mis en œuvre par M. RAMEL.

L'algorithme s'étant avéré rapide et efficace, il n'a pas fait l'objet de modifications. Seul la structure des résultats obtenus a été adaptée pour disposer d'une structure de données orientée objet qui nous facilitera les traitements ultérieurs.

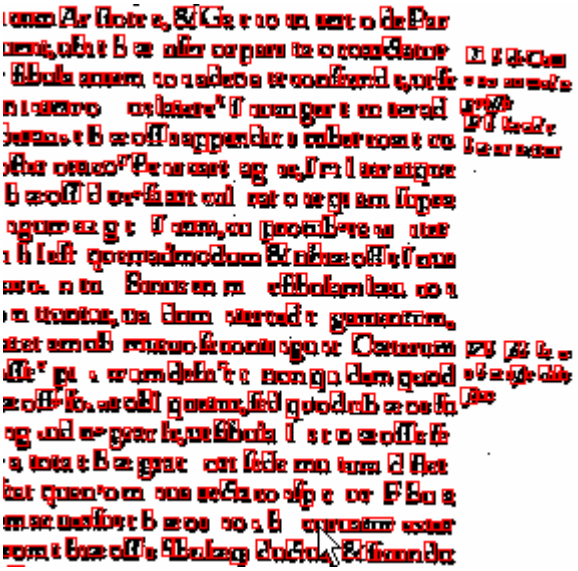


Figure 12 : en rouge, les rectangles englobant les composantes connexes détectées

Fonction : ***DetectionContours***
Entrée : image binaire au format « bmp »
Sortie : image binaire représentant les contours au format « bmp »
Liste de composantes connexes

7.5 Opérateurs morphologiques

Les choix qui avaient été fait dans la version précédente au niveau de l'élimination du bruit présent sur l'image étaient les suivants :

1) Entre l'étape de binarisation et celle d'extraction des composantes connexes, application d'une ouverture morphologique.

2) Puis, après détection des composantes connexes, élimination de celles ayant une taille inférieure à un seuil.

L'étape de traitement consistant à appliquer une ouverture morphologique à l'image a été supprimée et ce pour les deux raisons suivantes:

- Ø Le traitement est redondant par rapport à celui effectué en 2)
- Ø L'application de l'ouverture morphologique a des conséquences sur les résultats de la détection des contours. En effet, sur les illustrations dont le tracé est relativement fin, l'application d'un tel opérateur les fait disparaître et donc dégrade l'image juste avant la détection des contours.

Néanmoins, le code permettant d'appliquer les opérations morphologiques de base à été conservé, de même que la possibilité de les appliquer via les menus du logiciel.
Les fonctions concernées se trouvent dans la classe CImage et se nomment ***Eroder, Dilater, Fermeture, Ouverture.***

8 Segmentation par fusion des composantes connexes

8.1 Observations et propositions

8.1.1 Observation 1

Les composantes connexes sont capables d'isoler quasiment chacune des lettres d'un texte.
Les composantes connexes représentant les lettres d'un mot sont extrêmement proches. Les composantes connexes représentant la fin d'un mot et le début du mot suivant sont proches.
Il est donc possible de former un regroupement de composantes connexes qui représente une ligne de texte.

Méthodes habituellement utilisées : RLSA, opérateurs morphologiques

8.1.2 Problème1

Dans les documents anciens, la mise en forme n'est pas irréprochable.
Par exemple, la dernière lettre d'une ligne peut être plus proche du texte de la marge que de la lettre qui la précède sur la ligne.

Figure 13 : illustration de la proximité entre la marge et le texte

On peut constater quelquefois que l'alignement du texte n'est pas strictement vertical :

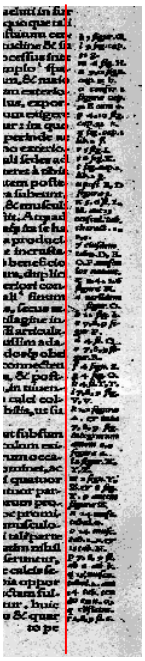


Figure 14 : illustration de la structure non rectiligne d'un document

8.1.3 Proposition1

La fusion des composantes connexes représentant une ligne de texte peut se faire en réalisant la fusion des composantes les plus proches horizontalement. Cependant pour palier au premier problème cité précédemment, il nous faut une information supplémentaire qui nous autorise ou non à fusionner les composantes connexes.

8.1.4 Observation2

L'image étant binarisée, le fond de la page est représenté par des pixels blancs.
Un grand nombre de pixels blancs sont alignés verticalement

- dans la zone séparant un texte de sa marge
- dans la zone séparant deux colonnes

Un grand nombre de pixels blancs sont alignés horizontalement

- dans la zone séparant un titre du texte
- dans la zone séparant deux paragraphes

Le nombre de pixels blancs que l'on est capable d'aligner horizontalement est faible

- entre deux lettres d'un mot
- entre deux mots d'une phrase

Le nombre de pixels blancs que l'on est capable d'aligner verticalement est faible

- entre deux lignes d'un même paragraphe

8.1.5 proposition2

Par conséquent, nous pouvons dire que le nombre de pixels blancs alignés horizontalement additionné du nombre de pixels blancs alignés verticalement est un bon critère permettant d'autoriser ou non la fusion de composantes connexes.
Pour chaque pixel appartenant au fond de la page à étudier, nous calculons donc ce paramètre de façon à former une carte de niveaux de gris. Cette carte est à 256 niveaux. Pour améliorer l'analyse visuelle de la carte, nous complémentons à 255 les valeurs de façon à obtenir un pixel sombre pour une valeur élevée et un pixel clair pour une valeur faible.
Lorsqu'on cherchera à fusionner deux composantes connexes proches, on cherchera à savoir :

- Si dans la zone reliant ces deux composantes, on trouve un pixel de niveaux de gris faible. Dans ce cas, la fusion ne devra pas avoir lieu
- Sinon, la fusion est possible

Remarque : nous faisons l'hypothèse que le texte est écrit horizontalement

8.2 Principe retenu

Le problème revient à affecter à tout pixel qui n'est contenu dans aucune composante connexe (le fond de la page), une valeur de niveau de gris. Ces valeurs représentent les barrières entre composantes. Moins cette valeur sera élevée et plus la probabilité que la fusion soit justifiée est grande. Considérons deux composantes connexes potentiellement fusionnables, la valeur des pixels rencontrés sur le chemin qui les joint doit être grande.
Ci-après la description des différentes étapes appliquées aux composantes connexes afin de segmenter la page.

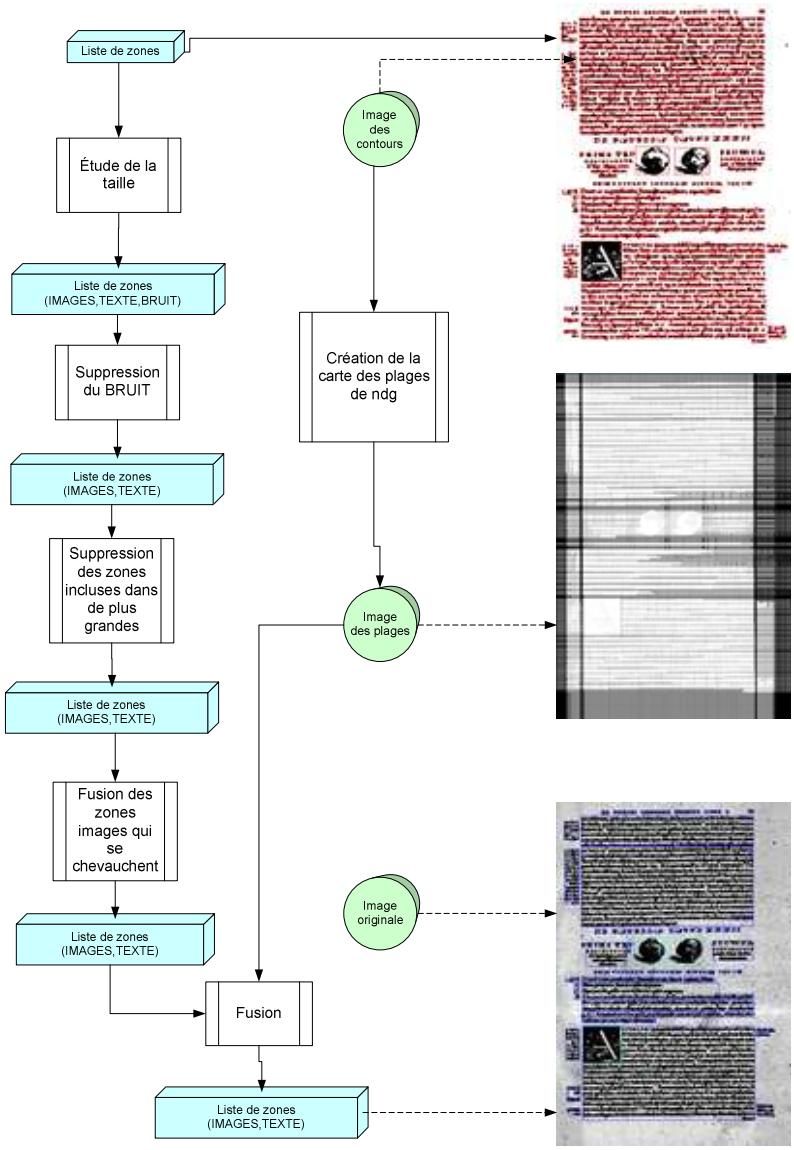


Figure 15 : diagramme fonctionnel illustré de l'étape de segmentation

8.3 Traitements effectués sur les composantes connexes

8.3.1 La structure zone

Nous disposons à cette étape d'une liste de composantes connexes :

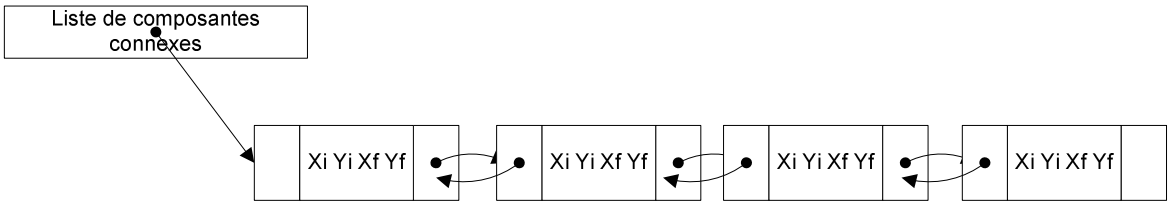


Figure 16 : structure utilisée pour représenter une liste de composantes connexes

Xi : abscisse du point haut gauche du rectangle englobant la composante connexe
Yi : ordonnée du point haut gauche du rectangle englobant la composante connexe
Xf : abscisse du point bas droite du rectangle englobant la composante connexe
Yf : ordonnée du point bas droite du rectangle englobant la composante connexe

Au niveau de l'implémentation, une composante connexe est modélisée par un objet de la classe CElement. Celle-ci contient deux attributs, le rectangle englobant le contour détecté et son centre de gravité :

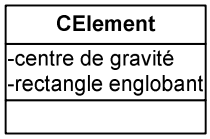


Figure 17 : structure utilisée pour représenter une composante connexe

La première étape est de constituer une structure de données adaptée à la finalité de la technique, c'est-à-dire être capable de réaliser la classification. Pour cela nous utiliserons une structure appelée zone qui est un objet capable de contenir une liste de composantes connexes. Le document sera donc décrit comme une liste de zones.

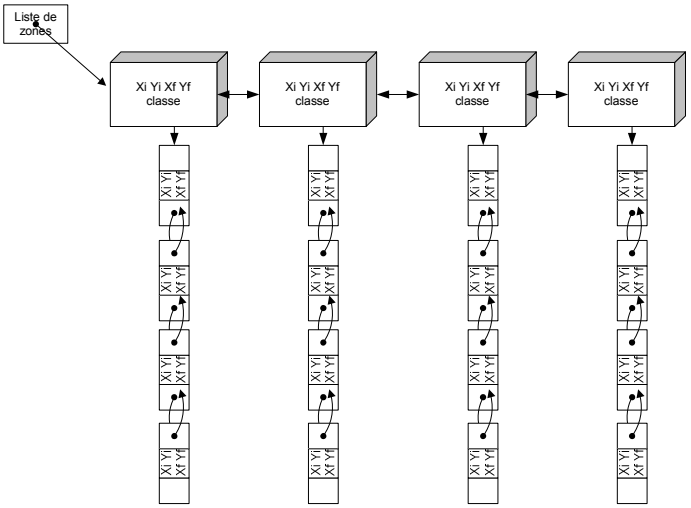


Figure 18 : structure utilisée pour représenter une liste de zones

Chaque zone a un champ destiné à indiquer à quelle classe appartient cette zone. De plus, elle contient les coordonnées du rectangle englobant l'ensemble des composantes connexes contenues dans celle-ci.

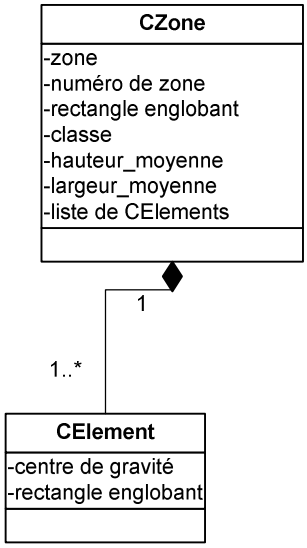


Figure 19 : échantillon du diagramme UML représentant le lien entre une zone et un élément

Fonction : **DetectionContours** de la classe CListeZones
Entrée : image binaire
Sortie : liste de zones

8.3.2 Etude de la taille des composantes connexes

Comme stipulé dans le paragraphe « contexte », nous avons fait le choix de conserver la technique utilisée initialement pour détecter les images. Cette technique est basée sur la taille des composantes connexes. Elle part du constat qu'une composante connexe représentant une image (illustration, lettrine..) est beaucoup plus grande qu'une composante connexe représentant du texte.

L'utilisateur définit la taille minimale MIN d'une grande composante et la taille maximale MAX d'une petite composante.
Nous considérons donc que :

- Ø Le bruit est une composante qui est inférieure à MIN
- Ø Les composantes représentant les lettres d'un texte sont des composantes qui se situent entre MIN et MAX
- Ø Les images sont les composantes dont la taille est supérieure à MAX

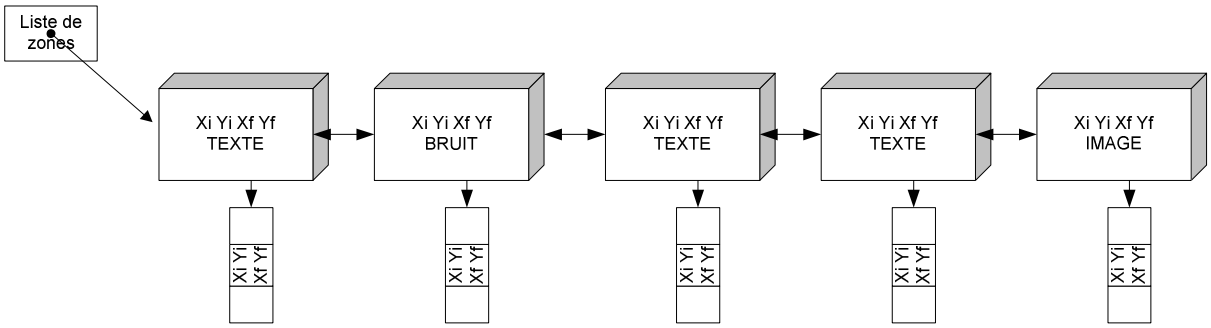


Figure 20 : exemple d'une liste de zones et leur classe potentielle

Fonction : **AjouterZone** de la classe CListeZones
Entrée : coordonnées de la composante connexe détectée
Sortie : une nouvelle zone est ajoutée à la liste de zones et on lui ajoute un premier élément en se servant des coordonnées passées en paramètre

8.4 Création de la carte des niveaux de gris

Soit I1, l'image des contours obtenue à la fin de l'étape d'extraction des composantes connexes. C'est donc une image binaire.
Soit I2, une deuxième image qui représente ce qu'on appellera la carte des niveaux de gris.

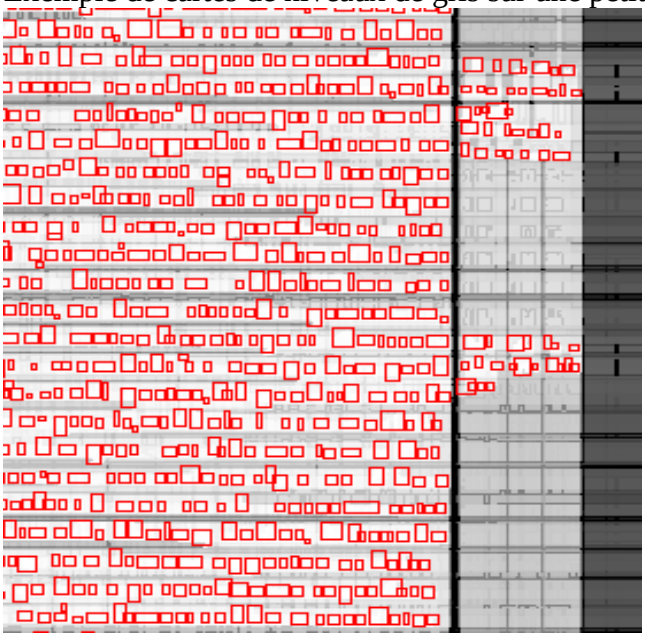
La détermination de la valeur d'un pixel de l'image I2 est effectuée de la façon suivante :

Pour un pixel donné, si il est blanc alors on se posera les questions suivantes :

- Ø Combien de pixels blancs est-on capable d'aligner horizontalement en partant de ce pixel (à droite et à gauche).
- Ø Combien de pixels blancs est-on capable d'aligner verticalement en partant de ce pixel (en haut et en bas).

Nous obtenons donc deux valeurs qui seront additionnées pour donner la valeur au pixel correspondant dans l'image I2.
Une fois la mesure effectuée pour chaque pixel, la carte sera ensuite normalisée. Ici, nous prendrons comme valeur maximale 255.
Cette image I2 sera utilisée pour déterminer si deux composantes connexes peuvent ou non fusionner.

Exemple de cartes de niveaux de gris sur une petite portion d'image :



La signification que l'on veut donner à la valeur de niveau de gris est la suivante :

- Ø Plus la zone est claire et plus la probabilité que la fusion de deux composantes connexes qui se trouvent de part et d'autre soit justifiée est grande.
- Ø Inversement une zone foncée est une zone qui ne doit pas autoriser la fusion

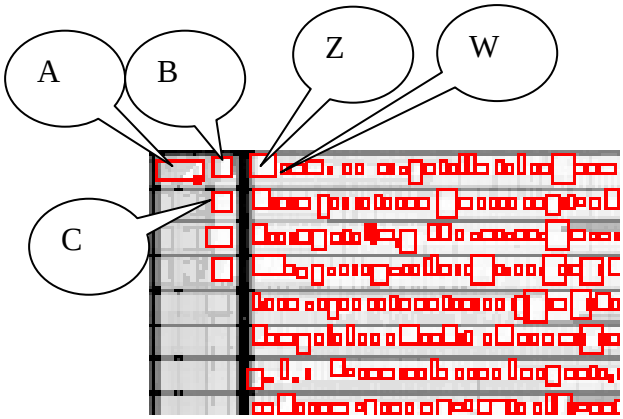
Fonction : **Plage** de la classe CAtelier
Entrée : Image des contours
Sortie : image de la carte des niveaux de gris

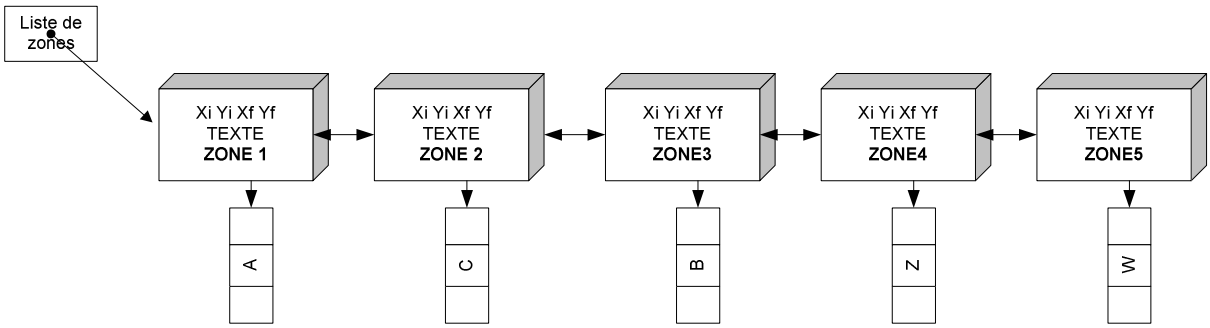
8.5 La fusion

Nous distinguons ici deux types de fusion : la fusion horizontale et la fusion verticale. En effet, la démarche consiste tout d'abord à fusionner les caractères pour former les lignes puis les lignes pour former les paragraphes. Ces deux opérations font donc l'objet d'un paramétrage distinct pour pouvoir atteindre une segmentation optimale.

Avant d'appliquer la fusion nous disposons d'une liste de zones. Une zone est une structure qui est capable de regrouper un ensemble de composantes connexes ayant un caractère identique. Une zone contient donc elle-même une liste de composantes connexes.
A l'initialisation, nous avons autant de zones que de composantes connexes.

Exemple :





Nous allons donc considérer chaque composante connexe et déterminer son voisin le plus proche dans la direction voulue n'appartenant pas à la même zone que lui-même.

Nous devons déterminer si oui ou non ces deux composantes doivent fusionner. Pour cela, nous allons nous servir de la carte de niveaux de gris créée précédemment. Sur cette carte, plus un pixel est sombre et plus deux composantes de part et d'autre de celui-ci seront difficiles à fusionner.

On veut prendre en compte une valeur qui :

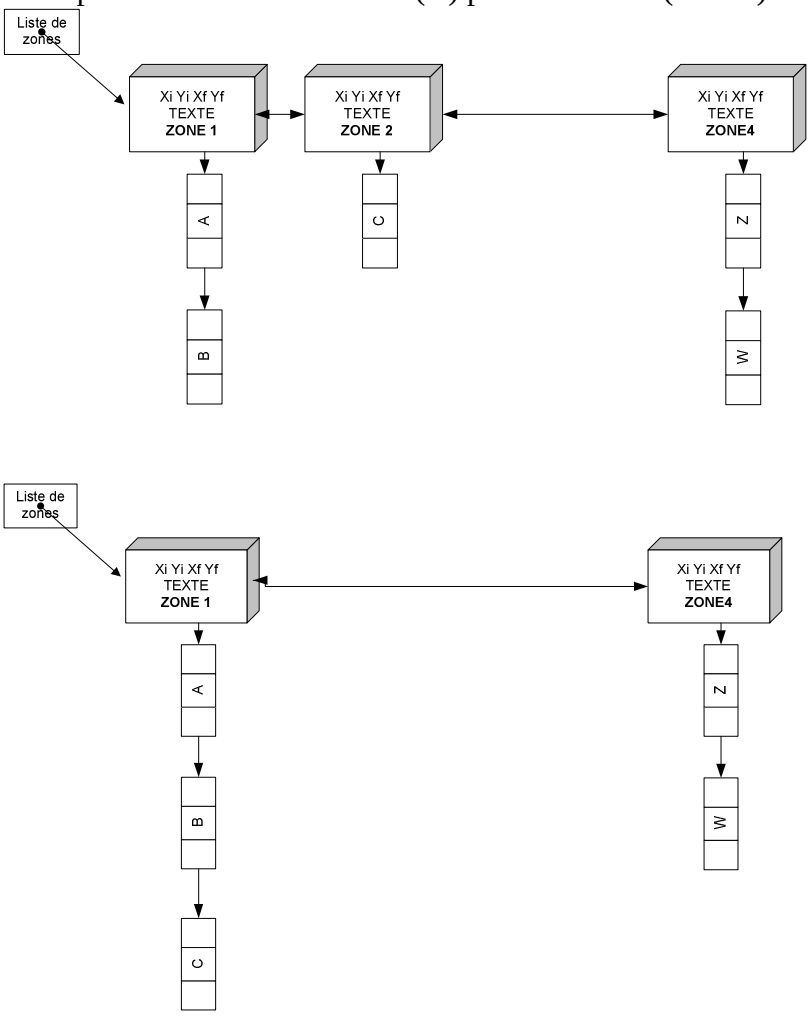
- Augmente avec la distance
- Augmente plus la zone est foncée donc plus le niveau de gris est faible

Dans la mesure de distance entre deux composantes, nous prendrons donc en compte la distance euclidienne pondérée par le complément à 255 de la valeur du pixel le plus sombre rencontré.

Plus les composantes sont éloignées et plus la probabilité que la fusion soit justifiée décroît. Notons ***d*** la distance entre les centres de gravité des deux composantes connexes considérées.
Sur le chemin joignant le centre de gravité des deux composantes, plus la valeur de la carte des niveaux de gris rencontrée est grande et plus la probabilité que la fusion soit justifiée décroît. Notons ***ndg*** la valeur de niveau de gris la plus faible rencontrée.

On va donc tester si :
d * (255 – ***ndg***) est inférieur à un seuil.
Si c'est le cas, on fusionne alors les deux composantes connexes.

Exemple de la fusion de B avec (A) puis de C avec (A et B):



Fonction : **Fusion** de la classe CAtelier
Entrée : liste de zones
Sortie : liste des zones fusionnées

9 Robustesse de la fusion à la rotation

La robustesse de l'algorithme de fusion par rapport à la rotation a été testée sur deux types de documents :

- Les documents anciens
- Les documents actuels

Le protocole de test a donc consisté à effectuer une rotation du document par pas de 1° dans le sens horaire jusqu'a l'échec de la fusion.
Voici les résultats obtenus sur deux exemples représentatifs de chaque catégorie :

Figure 4 : aucune rotation => OK

Figure 6 : 2° : ECHEC sur le titre principal

Figure 5 : 1° => OK

Figure 7 : 3° ECHEC sur le titre secondaire

Figure 21 : test de robustesse à la rotation sur un document ancien

Figure 12 : aucune rotation OK

Figure 13 : 5° => OK

Figure 14 : 6° => OK

Figure 15 : 7° => ECHEC paragraphe

Figure 16 : 8° => ECHEC paragraphe

Figure 17 : 9° => ECHEC paragraphe

Figure 18 : 10° => ECHEC paragraphe

Figure 19 : 12° => ECHEC paragraphe

Figure 20 : 13° => ECHEC colonnes

Figure 22 : test de robustesse à la rotation sur un document actuel

On aboutit donc la plupart du temps à un échec sur les documents anciens pour une rotation d'environ 2°, ce qui est relativement faible.
Au contraire, sur les documents actuels, les résultats sont corrects jusqu'à 6° voir beaucoup plus selon la structure du document.

Puisque nous traitons les documents anciens, nous pouvons conclure que l'algorithme de fusion n'est pas suffisamment robuste par rapport à la rotation. Nous devrons donc considérer l' hypothèse que les document ont été correctement redressés lors du processus de restauration comme réalisée.

10 Segmentation texte / image

10.1 Problème posé

A l'issue de l'extraction des composantes connexes, un étiquetage de base est effectué en fonction de la taille de celles-ci. Après cette étape, nous disposons donc de composantes connexes étiquetées « TEXTE » ou bien « IMAGE ».
Visualisons le résultat sur un exemple :

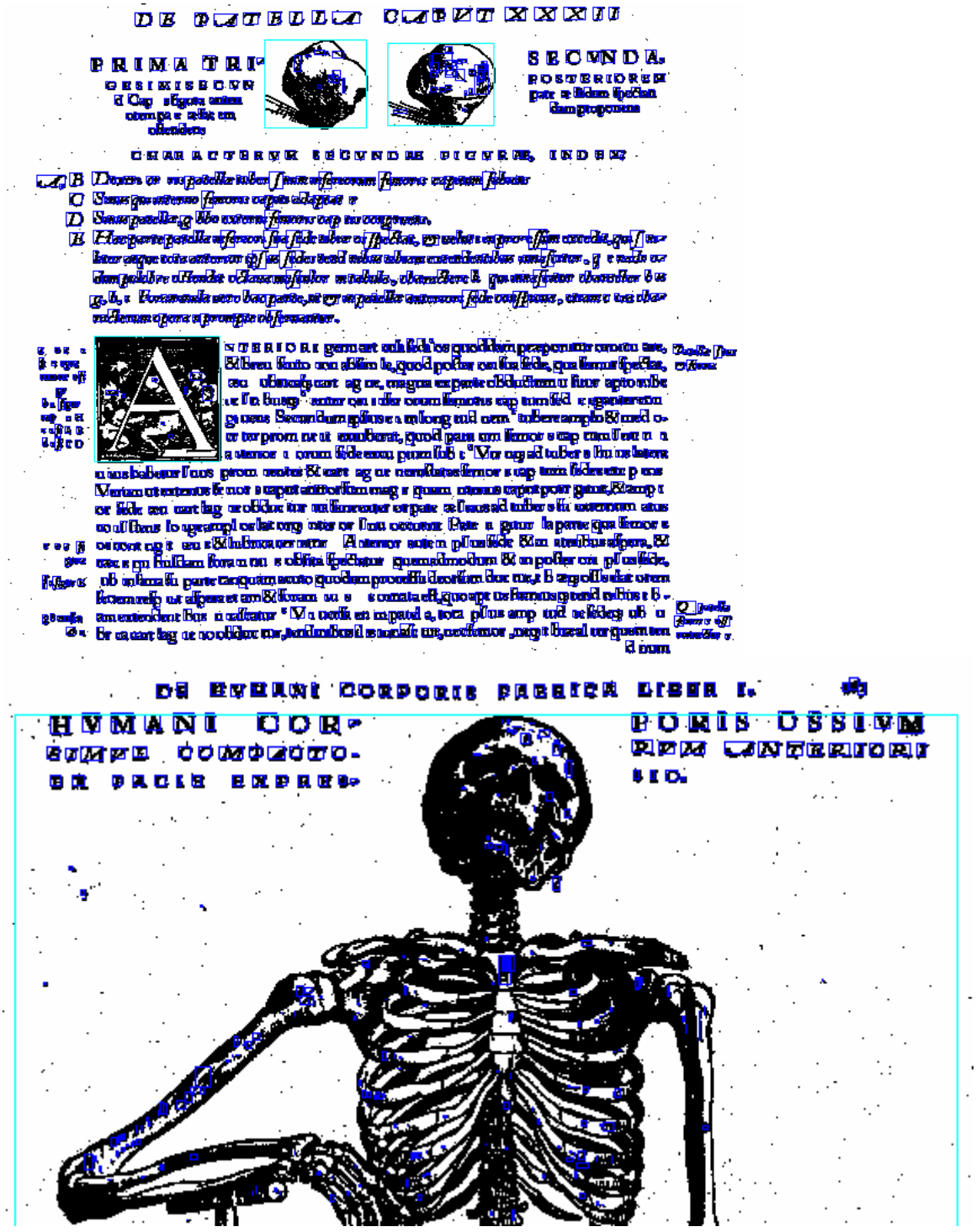
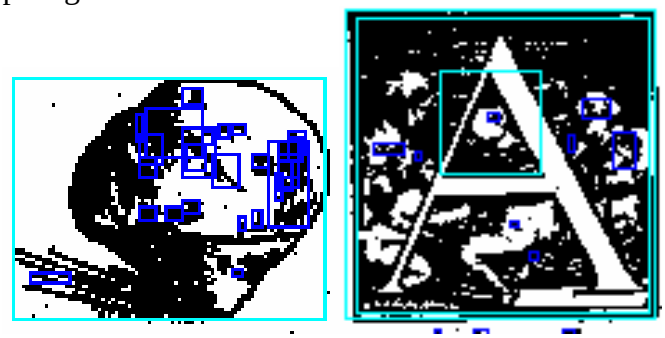


Figure 23 : visualisation des zones considérées comme de type TEXTE (bleu foncé) dans les zones de type IMAGE (bleu turquoise)

L'extraction des composantes connexes, basée sur un algorithme de suivi de contours, a donc généré de petites composantes qui correspondent à de petites formes à l'intérieur d'un contour plus grand.



10.2 Principe de la méthode

Dans les précédentes versions, pour résoudre ce type de situation, les composantes de type TEXTE incluses dans des composantes de type IMAGE étaient supprimées. Cependant, si sur l'exemple des lettrines, cela n'a aucune conséquence, il en est tout autre concernant les cas suivants :

- Ø les pages d'ouvrage représentant par exemple un squelette accompagné de légendes textuelles : impossibilité d'extraire cette légende en tant que zone de type TEXTE.
- Ø Les pages d'ouvrage représentant un texte totalement encadré : le cadre étant reconnu comme image, tout le texte contenu dans celui-ci est donc inaccessible

Par définition, les zones de type TEXTE sont petites et les zones de type IMAGE sont grandes. Ce constat simpliste a pourtant la conséquence suivante : le fait de blanchir les zones de type TEXTE a des répercutions extrêmement limitées puisque la dégradation du contour principal n'est pas ou peu affecté.

L'application de l'algorithme des plages donne alors une information importante. En effet, cet algorithme est en mesure de déterminer les régions de la page où l'on est capable d'aligner un grand nombre de pixels blancs. Le but est d'arriver à détecter les zones qui se trouvent à l'intérieur du contour principal de celles qui se trouvent à l'extérieur. Or, il se trouve qu'à l'intérieur de ce contour, on n'est jamais capable d'aligner énormément de pixels blancs, en tout cas moins qu'à l'extérieur de ce contour (puisque toutes les zones TEXTE sont blanchies).

Une binarisation est alors appliquée sur cette carte des plages (le seuil est actuellement fixé de façon empirique). L'intérieur du contour réel contient alors une majorité de pixels noirs tandis ce que le reste contient une majorité de pixels blancs.

Il suffit alors de calculer pour chaque zone de type TEXTEI la densité de pixels blancs contenus dans son rectangle englobant. Pour les zones qui dépassent une certaine valeur de densité, cela signifie qu'elles ne sont pas superposées au contour réel, ces zones gardent donc le type TEXTEI. Elles correspondent au texte inclus dans le rectangle englobant le contour de type IMAGE détecté mais n'appartenant pas au contour réel. Les autres zones sont alors supprimées puisqu'elles correspondent à des sous contours du contour réel et non pas à du texte.

Les étapes sont les suivantes:

- Ø Les zones de type IMAGE possédant une intersection non vide avec une autre zone de type IMAGE sont fusionnées
- Ø Les zones de type TEXTE possédant une intersection non vide avec une zone de type IMAGE devient de type TEXTEI (sous entendu : zone de type texte incluse dans une zone de type image)
- Ø Les zones de type TEXTE et TEXTEI sont blanchies
- Ø On applique l'algorithme des plages
- Ø On binarise le résultat obtenu
- Ø On calcul pour chaque zone de type TEXTEI la densité de pixels blancs contenus dans le rectangle englobant de celle-ci.
    - o On élimine celles qui ont une densité forte (ces zones sont contenues dans le contour principal, ce n'est donc pas réellement du texte)
    - o On garde celles qui ont un faible densité (ces zones sont donc à l'extérieur du contour principal, c'est donc effectivement du texte)

10.3 Illustration de la méthode

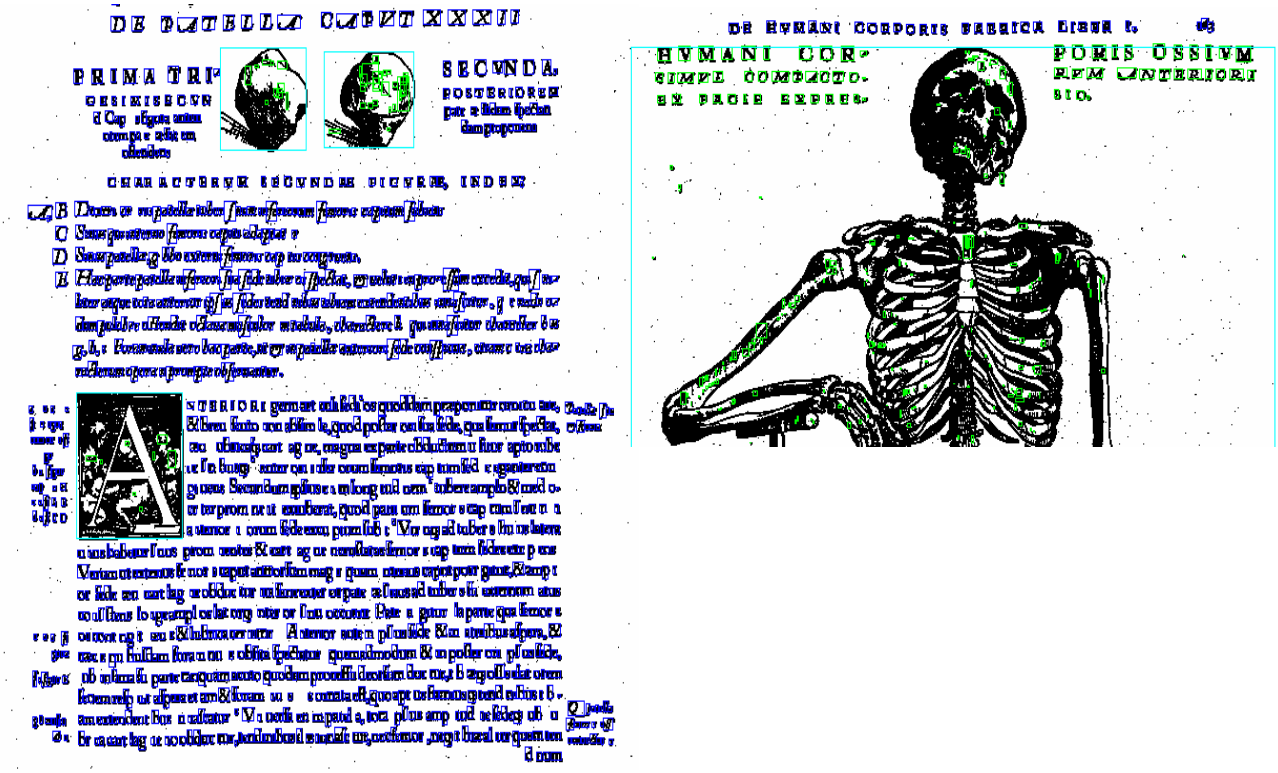


Figure 24 : les zones de type TEXTE incluses dans des zones de type IMAGE deviennent de type TEXTEI (en vert)

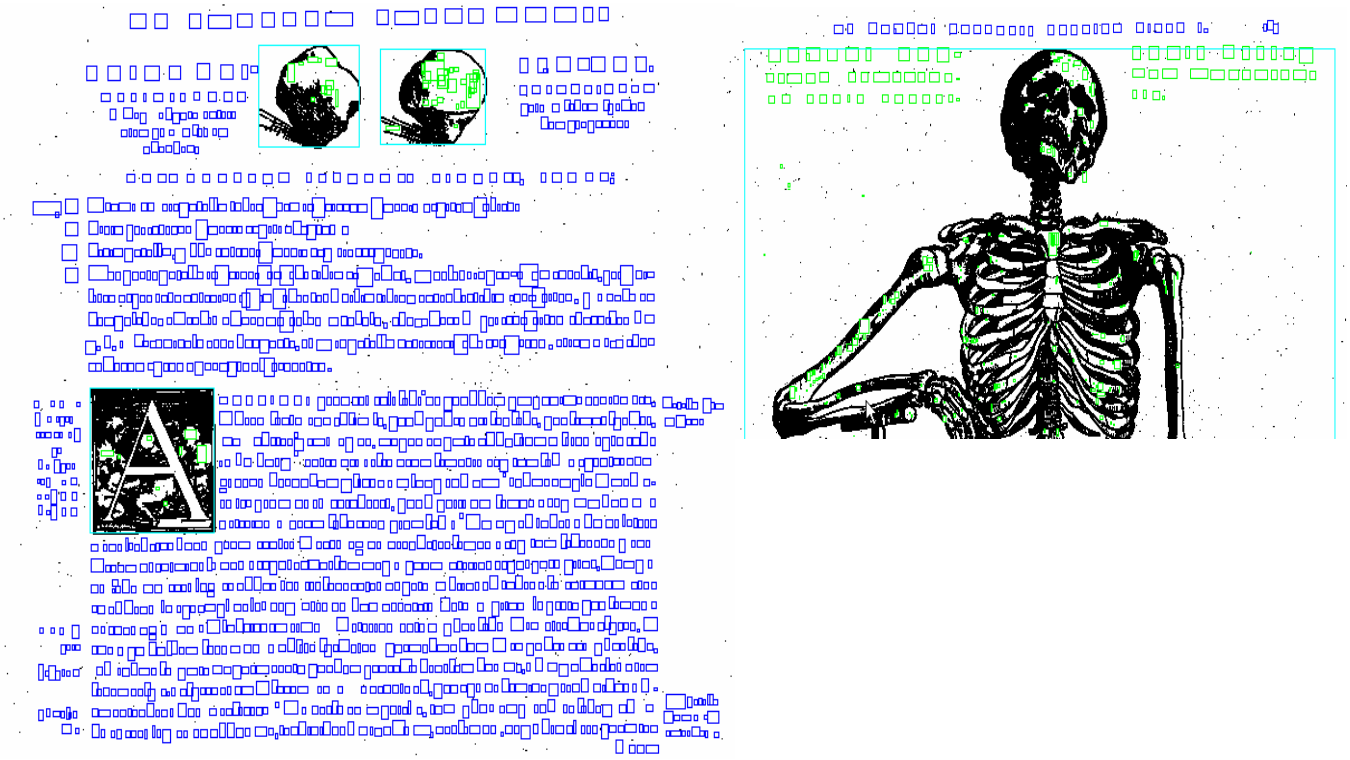


Figure 25 : coloriage des zones de type TEXTE et TEXTEI en BLANC

Figure 26 : application de la méthode des plages

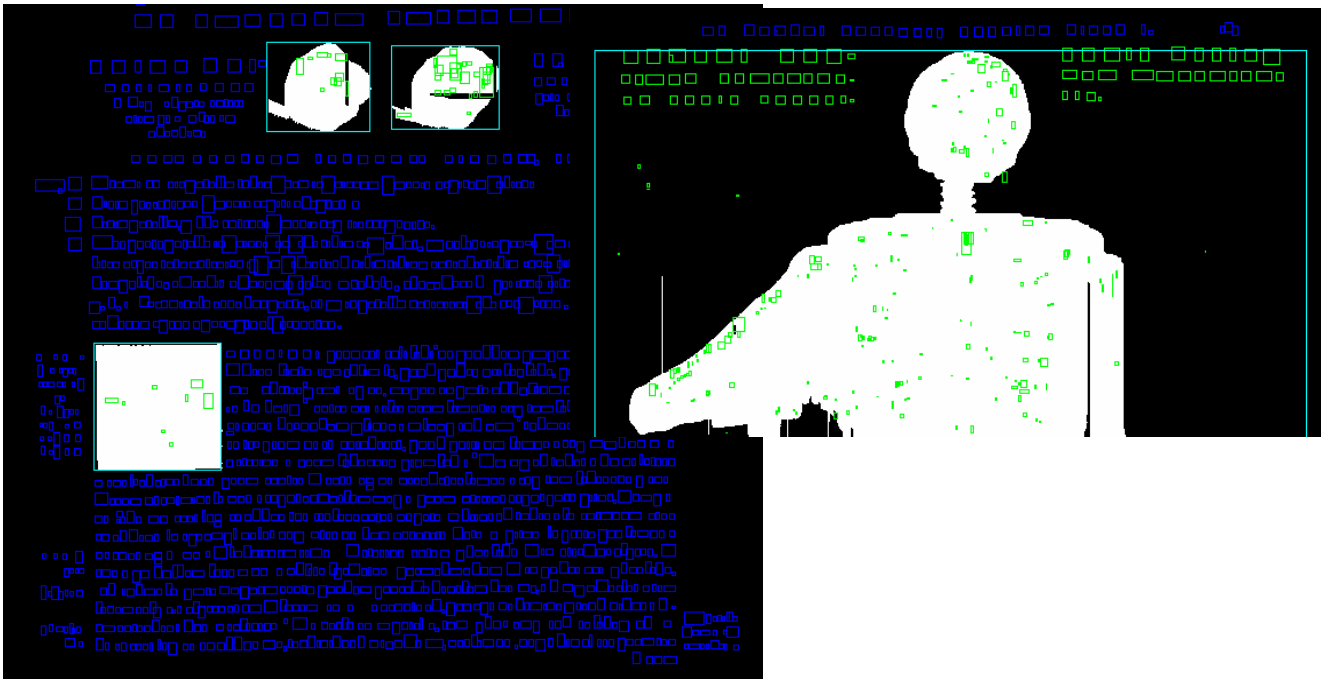


Figure 27 : binarisation

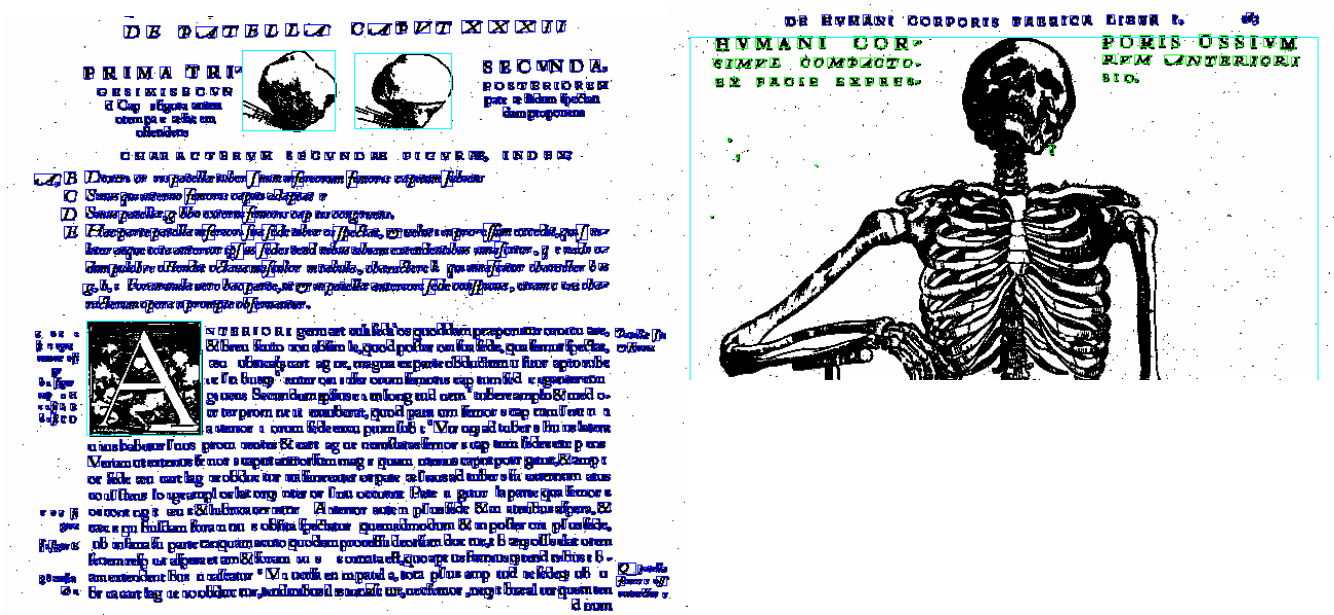


Figure 28 : élimination des zones de type TEXTEI qui possèdent une forte densité de pixels blancs

10.4 conclusion

Cette technique ne peut dans l'état suffire à segmenter de façon parfaite les zones textuelles des zones images. Cependant, elle permet d'éliminer une grande majorité de zones non textuelles. Il apparaît nécessaire de combiner cette technique avec d'autres…

De plus elle nécessite le réglage de deux paramètres : le seuil de binarisation et le seuil de densité de pixels blancs.

11 La classification supervisée

11.1 Introduction

En ce qui concerne le problème de la classification des zones détectées pendant l'étape de segmentation, nous pouvons dégager trois axes selon lesquels il est possible d'orienter les recherches :

- L'étude de la **position géographique** des zones identifiées
- L'étude des **relations de voisinage** entre zones identifiées
- L'étude statistique de la **forme et de la texture** des zones identifiées

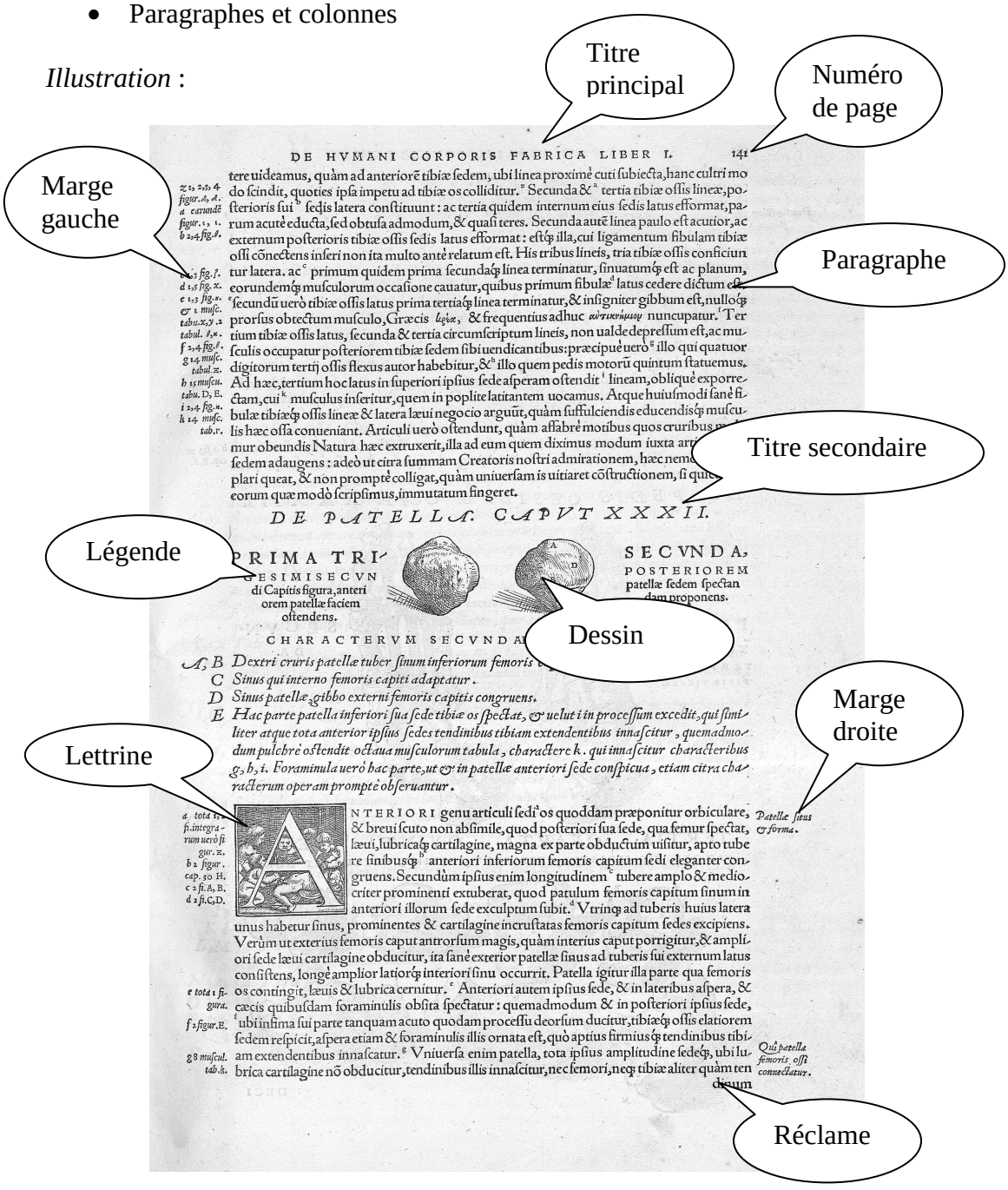
Après une expertise des documents mis à notre disposition, nous allons extraire un ensemble de règles de mise en page que nous ferons valider par des experts du domaine.
Ensuite, nous étudierons dans quelle mesure nous pouvons croiser cette connaissance avec les trois axes définis ci avant pour en déduire une stratégie de classification.

11.2 Les différentes classes

Avant tout, nous allons définir les différentes classes que nous souhaitons identifier :

- Titre principale
- Titres secondaires
- Marge gauche/droite
- Numéro de page gauche/droite
- Réclame
- Lettrine
- Images
- légendes
- Paragraphes et colonnes

Illustration :



11.3 Expertise sur les documents

Dans cette partie, nous allons extraire un certain nombre d'observations de l'étude d'un échantillon de pages appartenant à un ouvrage. Cette suite d'observations va nous servir de base pour l'édification de règles de mise en page. Nous classerons ensuite ces observations en fonction de leur niveau de pertinence par rapport aux trois axes cités précédemment.

Etude du Vésale

- *Titre*

Le titre principal est présent sur toutes les pages. Il est situé en haut de la page et est centré.

- *numéros*

Au même niveau que le titre, on trouve la numérotation des pages. Ces numéros sont situés soit à droite, soit à gauche, soit les deux.
Ces numéros sont toujours situés entre les marges lorsqu'elles existent.

- *Marges*

Les marges ne sont pas systématiquement présentes. Lorsqu'elles le sont, elles sont à droite ou à gauche, ou les deux à la fois. Ces marges comportent la plupart du temps des annotations d'une police légèrement plus petite que du texte normal et bien plus petite qu'un titre. L'écriture est souvent plus dense (les lettres sont plus rapprochées). On a pu observer dans quelques rares cas que des images étaient présentes dans ces zones de marge. En tout cas, les zones représentant la marge contiennent à 99% du texte. On peut remarquer que ces zones sont dédiées à ces annotations et qu'en aucun cas il est possible de trouver un autre élément qu'une marge dans cette zone. Ces marges, quand elles sont présentes, sont donc les zones situées le plus à gauche ou le plus à droite du document. En bas et en haut d'une zone identifiée comme marge, on ne peut trouver que : une marge ou rien. (Excepté le cas rare ou se trouve une image dans la marge)
Une marge n'a d'existence que si elle est en vis-à-vis avec une zone autre qu'une marge. Pour une marge gauche, on est forcé de trouver à sa droite un élément identifié comme du texte, une image … Idem pour la marge droite. En revanche, il est impossible de trouver des éléments comme le titre principal, un numéro de page ou une réclame.

Autrement dit, les marges étant identifiées, cela implique une restriction au niveau de l'espace de recherche du titre principal, du numéro de page et de la réclame. En effet, pour les deux premiers, ils se trouvent systématiquement entre le haut de la page et le bloc le plus haut identifié comme marge. Par le même principe, la réclame se trouve entre le bloc le plus bas identifié comme marge et le bas de la page.

- *Lettrines*

On remarque que les lettrines sont systématiquement de forme carrée. La présence d'une lettrine implique le début d'un paragraphe. Immédiatement à gauche d'une lettrine on peut trouver la marge gauche ou rien. A droite et en bas, on trouve du texte. Pour ce qui est de l'élément situé en haut de la lettrine, il peut être variable. On peut cependant remarquer qu'à 75-80%, on y trouve une zone de type titre secondaire, ce qui peut représenter un indice important pour la détermination de ces derniers.

11.4 Classification selon la position géographique

	haut	bas	gauche	droite	centré		
Titre principal	*				*		
Titre secondaire					*		
Marge gauche			*				
Marge droite				*			
Numéro de page gauche	*		*				
Numéro de page droite	*			*			
réclame		*		*			
lettrine			*				

Il semble possible d'édifier un certain nombre de règles à partir de la position géographique des zones identifiées. Le but ici n'est pas d'imposer des règles extrêmement contraignantes car la mise en page peut être variable (ex : il est abusif de dire que le centre de gravité d'une marge gauche se trouve entre le pixel d'abscisse 205 et 213).
Cependant, on peut dire raisonnablement que la position du centre de gravité d'une marge est très certainement située dans le premier tiers de la largeur de la page.
Si cette approche se révélait par la suite trop spécialisée (capacité de généralisation), on pourrait modéliser cette connaissance par un degré d'appartenance (sachant que le centre de gravité de la zone se situe dans le premier tiers de la largeur de la page, on peut dire qu'il a une probabilité de 0.8 d'être une marge gauche, 0.4 d'être un titre, 0.0 d'être une marge droite).

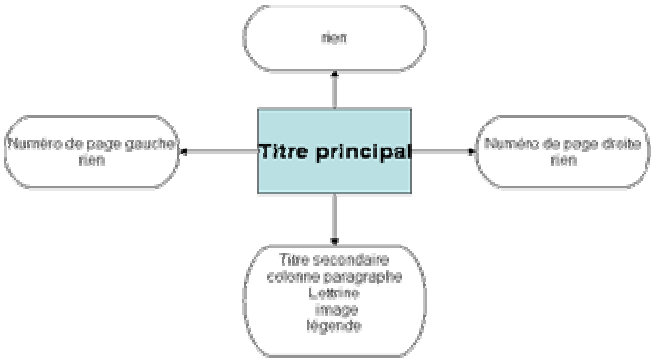
Nous prendrons donc en compte la position géographique du centre de gravité des différentes zones étant donné qu'elle constitue le premier indice (naturel) dans la classification.

11.5 Classification selon les relations de voisinage

Nous pouvons identifier certaines règles au niveau des relations de voisinage pour les classes suivantes :

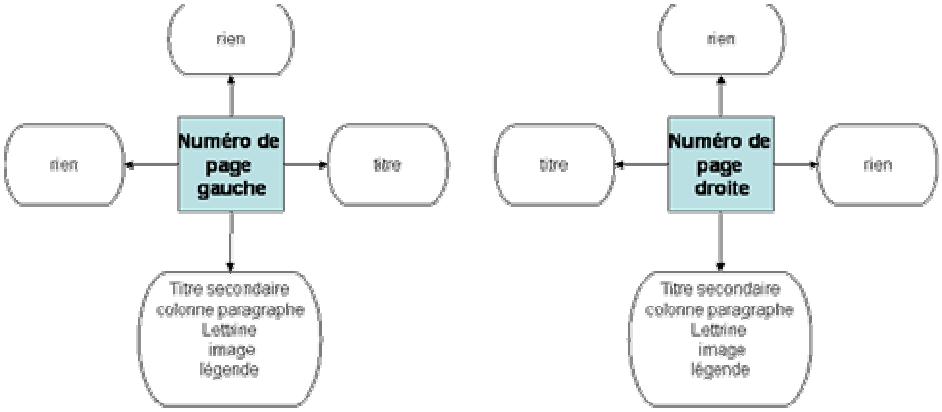
- Titre principale
- Numéro de page gauche/droite
- Réclame
- Marge gauche/droite
- Lettrine

11.5.1 Titre principal



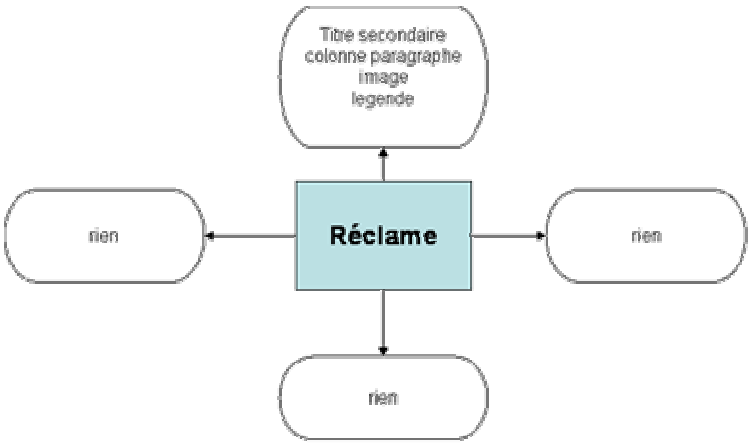
Exemple de règles de voisinage pouvant être utilisées pour reconnaître une zone de type « titre principal ».

11.5.2 Numéro de page



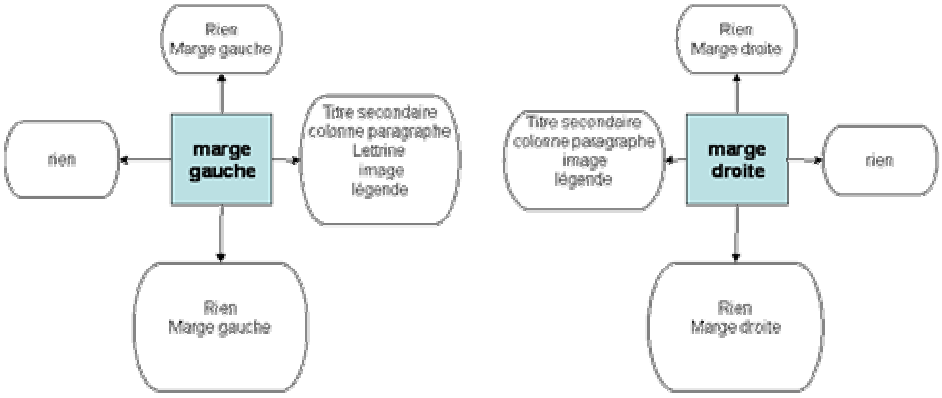
Exemple de règles de voisinage pouvant être utilisées pour reconnaître une zone de type «numéro de page gauche » ou « numéro de page droite ».

11.5.3 Réclame



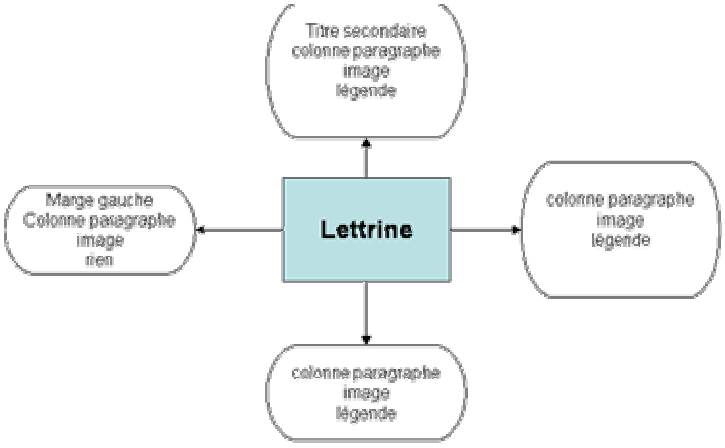
Exemple de règles de voisinage pouvant être utilisées pour reconnaître une zone de type «réclame».

11.5.4 Marges



Exemple de règles de voisinage pouvant être utilisées pour reconnaître une zone de type « marge gauche » ou « marge droite ».

11.5.5 Lettrines



Exemple de règles de voisinage pouvant être utilisées pour reconnaître une zone de type « lettrine ».

11.6 Classification selon la forme et la texture des zones

Dans cette partie, nous nous intéresserons aux aspects suivants :

- L'étude statistique des caractéristiques d'une composante connexe (taille, densité, …)
- Ratio hauteur/largeur
- L'étude de la similarité du contenu des composantes connexes (constitution d'un dictionnaire local)

11.7 Stratégie de classification

Première étape

Utilisation de la position géographique du centre de gravité de chaque zone identifiée par l'algorithme de segmentation.

A cette étape, nous ne prenons aucune décision définitive quant à l'appartenance d'une zone à une classe. Cependant, nous allons étiqueter chaque zone comme appartenant potentiellement à telle ou telle classe.

Deuxième étape

Utilisation du voisinage des zones.

Nous allons parcourir les différentes zones et déterminer pour chacune d'elles la zone la plus proche dans les quatre directions habituelles.
Nous allons ensuite déterminer si le schéma de voisinage correspond à un schéma connu (voir paragraphe sur les relations de voisinage).
A cette étape, nous prenons des décisions définitives quant à l'appartenance d'une zone à telle classe. Nous ne pourrons pas à cette étape affecter une classe à toutes les zones mais les décisions prises seront « sûres » (pas d'éléments mal classés).

Troisième étape

Utilisation du contenu et de la forme des éléments contenus dans chaque zone.

Cette étape va nous permettre de prendre une décision sur la classe d'une zone qui peut potentiellement appartenir à plusieurs classes.

Identification des marges -> facilite l'identification du titre principal, des numéros de pages, de la réclame.

Identification des marges, titre principal, numéros de page, réclame -> facilite la classification car on a restreint :

- La zone de recherche
- Le nombre de classes potentiellement présentes dans cette zone.

11.8 De la stratégie vers le scénario…

Les stratégies de classification ayant donné de bons résultats, certaines limitations sont cependant très vite apparues. En effet, de part leur mise en page hétérogène, les stratégies doivent être adaptées à chaque ouvrage traité. De plus, nous ne sommes pas en mesure de dresser une liste exhaustive du nombre de classes à découvrir. Ce type de travaux est en effet un domaine qui fait l'objet d'un certain nombre de recherches.

Seul l'expert a donc la connaissance des techniques d'analyse à appliquer sur les différents ouvrages. C'est donc lui seul qui est en mesure d'élaborer ses propres stratégies. Dès lors, le but est de lui fournir un outil qui soit suffisamment puissant et flexible.

12 Mise en œuvre des scénarios

12.1 Présentation

Un scénario correspond à un ensemble de traitements élémentaires dont l'exécution est soumise à une chronologie donnée.
Un traitement élémentaire correspond à l'application d'une règle définie par l'utilisateur. Ces règles peuvent être de deux types :

- Ø Règle de classification
- Ø Règle de fusion

Par l'intermédiaire de ces règles, l'utilisateur va pouvoir effectuer un certain nombre d'actions dont :

- Ø La création d'un nouveau type, définit par un nom, une couleur et qui va servir à étiqueter les zones
- Ø La suppression de zones d'un type particulier
- Ø La classification de zones en fonction de leur position géographique
- Ø La classification de zones en fonction de leur voisinage
- Ø La classification de zones en fonction de leur forme et leurs caractéristiques intrinsèques
- Ø La fusion horizontale d'un certain type de zones
- Ø La fusion verticale d'un certain type de zones

En définissant ainsi les « bonnes » règles et en les plaçant dans le « bon » ordre, l'utilisateur à donc un moyen extrêmement puissant d'extraire les régions désirées.

Les règles sont exécutées dans l'ordre où elles ont été créées. Il est cependant possible de déplacer une règle ou de la supprimer. Le chargement et la sauvegarde des scénarios est également possible. De cette façon, on est en mesure d'effectuer un traitement par lot en ayant choisit le scénario adéquat.

12.2 Les règles de classification

Toutes les règles de classification respectent le protocole suivant :

- Ø L'utilisateur précise le type des zones concernées par le test à réaliser : TYPE_DEPART
- Ø puis précise le type dans lequel transformer la zone si le test réussi : TYPE_SI_REUSSI
- Ø et le type si la zone ne respecte pas les conditions de test : TYPE_SI_ECHEC

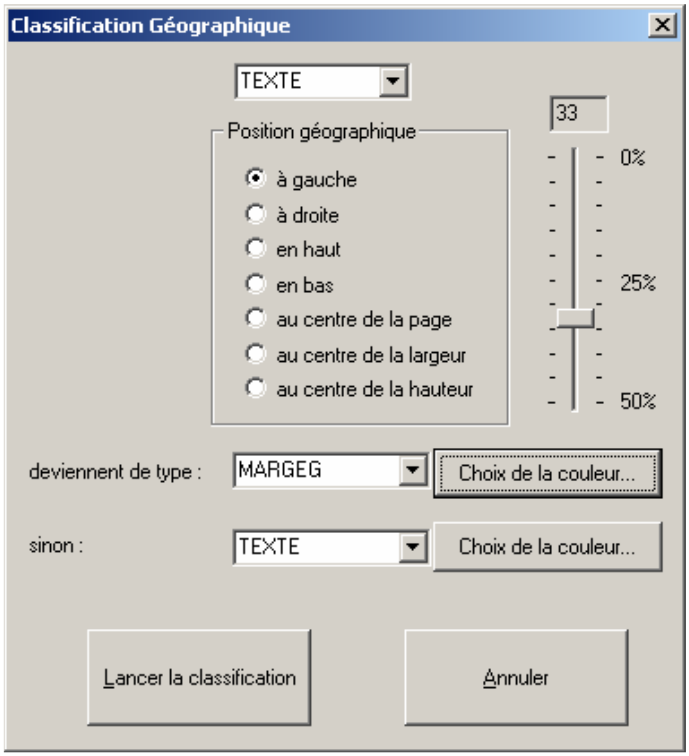
Le choix des types SI_REUSSI et SI_ECHEC se fait de deux façons :

- Ø On souhaite affecter à la zone un type déjà existant, il suffit alors de le choisir dans la liste proposée.
- Ø On souhaite créer un nouveau type de zone, il suffit alors d'affecter un nouveau nom à la zone d'édition puis de choisir la couleur qui sera affectée à ce nouveau type.

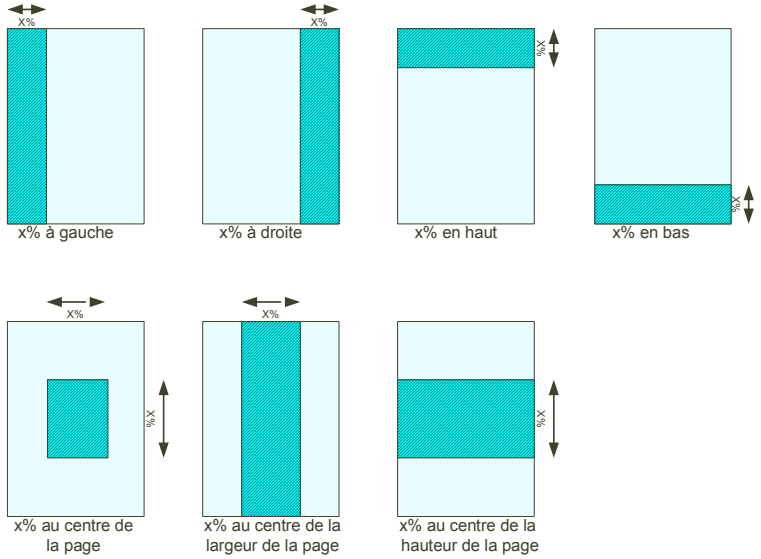
```
POUR toutes les zones FAIRE :
    SI le type de la zone = TYPE_DEPART
    ALORS
        SI la zone satisfait les conditions du test
        ALORS
            La zone devient de type TYPE_SI_REUSSI
        SINON
            La zone devient de type TYPE_SI_ECHEC
        FSI
    FSI
FPOUR
```

12.2.1 Classification géographique

On donne à l'utilisateur la possibilité d'utiliser la position géographique du centre de gravité d'une zone détectée.



Par l'intermédiaire de cette boite de dialogue, il est possible d'identifier des zones présentes dans les régions de la page définies comme ci-dessous :



12.2.2 Classification par relation de voisinage

Dans le chapitre précédent, on a vu l'utilité de disposer de ce type de classification. Elle permet en effet de se baser uniquement sur le type des voisins situés dans les directions principales pour identifier une zone particulière.

L'utilisateur définit cinq ensembles. Nous les appellerons comme ceci :
Ed : ensemble des voisins susceptibles d'être placés à la **droite** de la zone considérée.
Eg : ensemble des voisins susceptibles d'être placés à la **gauche** de la zone considérée.
Eh : ensemble des voisins susceptibles d'être placés en **haut** de la zone considérée.
Eb : ensemble des voisins susceptibles d'être placés en **bas** de la zone considérée.
Ei : ensemble des zones susceptibles d'**inclure** la zone considérée.

Il est possible de laisser vide un ou plusieurs de ces ensembles. Si un ensemble est vide alors le test qui lui correspond ne sera pas pris en compte. Si on souhaite préciser que la zone peut ne posséder aucun voisin dans telle direction il faut alors sélectionner dans la liste l'élément « RIEN ».

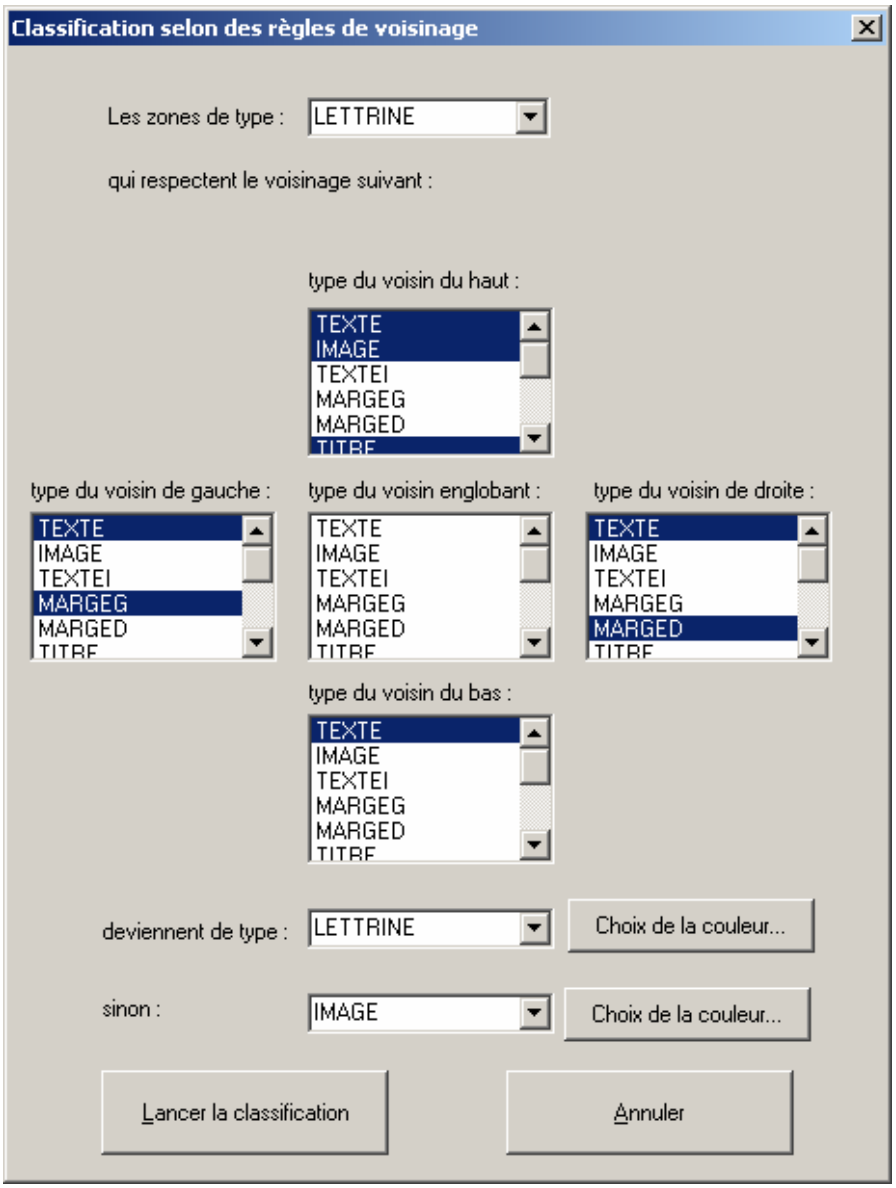
L'utilisateur choisit pour chaque ensemble un ou plusieurs éléments dans la liste proposée. S'il choisit plusieurs éléments, cela revient à appliquer un OU logique entre ces différents éléments. Par contre, tous les ensembles laissés non vides, devront satisfaire chacun des tests correspondant, c'est donc un ET logique qui unit les différents tests.

Exemple :

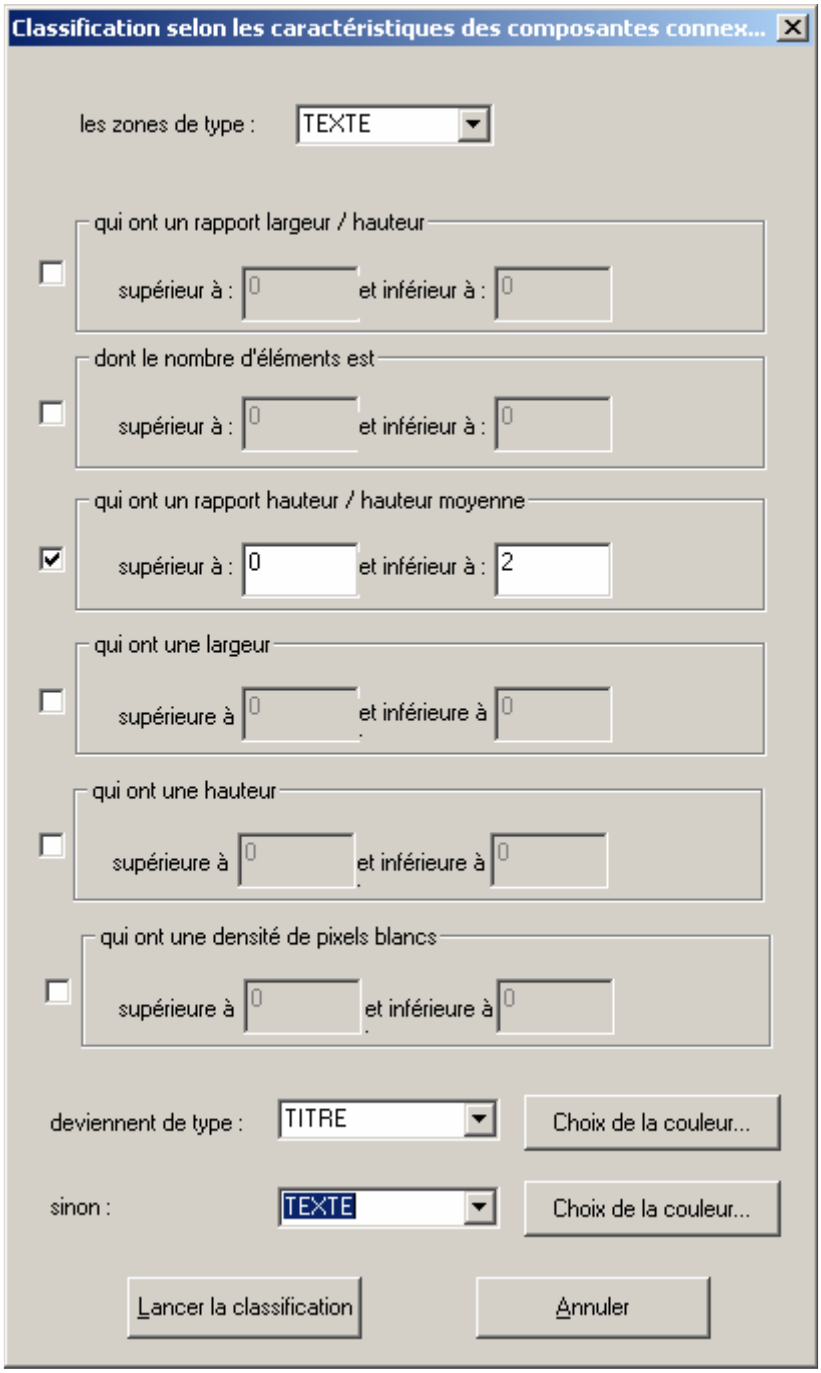
Ed={A,B}, Eg={A,C}, Eh={}, Eb={F}, Ei={}

La zone de type TYPE_*DEPART* devient de type TYPE_SI_REUSSI si le type de son voisin de droite est A OU B, et que le type de son voisin de gauche est A ou C, et que le type de son voisin du bas est F.
Si elle ne satisfait pas ces conditions, elle devient de type TYPE_SI_ECHEC.

Voici l'interface qui est utilisée pour édifier ce type de règle :


Classification selon des règles de voisinage
Les zones de type :
LETTRINE
qui respectent le voisinage suivant :
type du voisin du haut :
TEXTE
IMAGE
TEXTEI
MARGEG
MARGED
TITRE
type du voisin de gauche :
type du voisin englobant :
type du voisin de droite :
type du voisin du bas :
deviennent de type :
LETTRINE
Choix de la couleur...
sinon :
IMAGE
Choix de la couleur...
Lancer la classification
Annuler

12.2.3 Classification en fonction de la forme et les caractéristiques intrinsèques



- Ø rapport hauteur/largeur :

En général, utilisé pour détecter un certain type de zone image. La lettrine, par exemple est une zone de type image de forme carrée donc on va pouvoir préciser un rapport hauteur/largeur proche de 1.
De même , ce qui est appelé « bandeau » est une zone de type image qui à un rapport hauteur/largeur relativement constant dans un ouvrage, en général il est trois fois plus large que haut.

- Nombre d’éléments

Le nombre d’éléments décrit le nombre de composantes connexes contenues dans une zone. Ce test peut s’avérer utile pour détecter les zones ne contenant qu’un seul élément. Selon l’étape à laquelle est placée cette règle, il est probable qu’un tel type de zone corresponde à du bruit ou une tache sur la page.

- Rapport hauteur/hauteur moyenne

Ici, on considère la hauteur de la zone et la hauteur moyenne de l’ensemble des éléments contenus dans cette zone. Ce test peut être très utile pour détecter une zone qui ne forme qu’une seul ligne. En effet, en précisant que l’on veut toutes les zones de type TEXTE qui ont un rapport hauteur/hauteur moyenne proche de 1 alors on va pouvoir extraire tous les titres du document.

- Largeur et hauteur

Utilisation de la taille de la zone

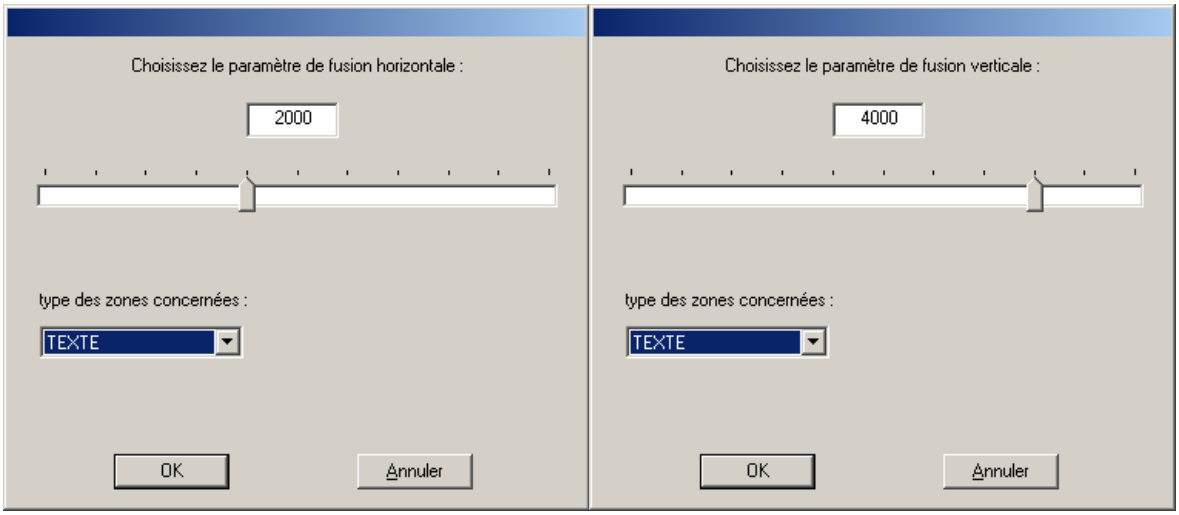
- Densité de pixels blancs

Prévue pour être utilisée sur une image binarisée, cette caractéristique permet de déceler les zones qui sont composées d’une forte densité de pixels blancs (ou noirs).

12.3 Les règles de fusion

On a vu comment segmenter la page en zones de type texte et image puis comment par fusion on formait un ensemble de zones. Les paramètres de fusion sont donc la clé d'une bonne segmentation. Il est parfois nécessaire de régler ces paramètres de façon restrictive pour ne pas effectuer de « mauvaises » fusions. L'exemple type est le cas des pages qui possèdent du texte et des marges extrêmement proches de celui-ci. Pour pouvoir traiter ce cas sans fusionner les marges avec le texte, il est obligatoire de paramétrer de façon restrictive la fusion horizontale. La première conséquence de ce type de paramétrage est la sur-segmentation résultante des zones représentant les titres.

Pour résoudre ce problème, il a donc été ajouté la possibilité de rajouter aux scénarios des règles de fusion. En effet, si on reprend l'exemple précédent, une fois les marges classifiées, il est alors possible de ré appliquer une fusion avec des valeurs plus larges qui vont dont autoriser la fusion de composantes plus éloignées. Sachant qu'on a désormais étiqueté les marges, en appliquant cette deuxième fusion sur des zones de type texte, il est alors possible de former les zones représentant les titres.



12.4 Création et destruction de types

12.4.1 Création d'un nouveau type de zone

La création d'un nouveau type est effectuée de manière implicite. L'utilisateur choisit le nouveau type de destination, c'est-à-dire qu'il effectue un choix au niveau des zones d'éditions correspondant aux labels « deviennent de type : » ou « Sinon : ». S'il choisit des éléments qui sont présents dans la liste alors c'est qu'il souhaite utiliser des types déjà existants. Cependant, s'il indique dans la zone d'édition un nouveau nom de zone (non présent dans la liste), il crée alors un nouveau type auquel il pourra préciser la couleur qui lui est associée (par défaut, c'est le noir).

La création d'un nouveau type à deux conséquences :

- Ø Les zones étiquetées avec ce nouveau type seront affichées à l'écran en utilisant la couleur précisée.
- Ø A la sauvegarde des zones détectées, un nouveau répertoire du nom du nouveau type sera créé et les zones concernées y seront placées.

12.4.2 Destruction d'un certain type de zone

La suppression va permettre d'éliminer l'ensemble des zones de tel type. Elles n'auront donc plus d'existence en tant que zone et ne seront ni affichées, ni sauvegardées. On pourra préciser une couleur de remplissage, laquelle sera utilisée sur l'image de travail et ceci afin d'appliquer des traitements spécifiques. Il est possible de ne pas supprimer de la liste les zones en décochant la case à cocher, dans ce cas les zones seront simplement coloriées en fonction du choix effectué.
Voici l'interface utilisée pour détruire et/ou colorier un type de zone :

13 Exemple de résultats et tests

13.1 Exemple de scénario

- Ø Effectuer une binarisation avec un seuil de : 150 (image courante modifiée)
- Ø détection des contours et etiquettage en fonction de leur taille 5 5 100 120 (image courante modifiée)
- Ø les zones de type BRUIT sont supprimées et sont coloriées en BLANC (image courante modifiée)
- Ø les zones de type : IMAGE ayant une intersection non vide sont fusionnées
- Ø création du type : TEXTEI
- Ø le type TEXTE devient de type TEXTEI si
- Ø son voisin de gauche est dans {} et son voisin de droite est dans {} et son voisin du haut est dans {} et son voisin du bas est dans {} et est inclus dans {IMAGE TEXTEI }

- Ø Copier l'image courante
- Ø les zones de type TEXTE sont coloriées en BLANC (image courante modifiée)
- Ø les zones de type TEXTEI sont coloriées en BLANC (image courante modifiée)
- Ø application de l'algorithme des plages (image courante modifiée)
- Ø Effectuer une binarisation avec un seuil de : 210 (image courante modifiée)
- Ø création du type : BRUIT
- Ø le type TEXTEI devient de type BRUIT si la densité de pixels blancs est comprise entre 0.10 et 1.00 sinon il devient de type TEXTE
- Ø Coller dans l'image courante (image courante modifiée)
- Ø les zones de type BRUIT sont supprimées et ne sont pas coloriées
- Ø application de l'algorithme des plages (image courante modifiée)
- Ø Fusion des zones de type TEXTE dans la direction horizontale avec un seuil de 2000
- Ø Fusion des zones de type TEXTE dans la direction verticale avec un seuil de 4000
- Ø création du type : MARGEG
- Ø le type TEXTE devient de type MARGEG s'il se trouve à GAUCHE (25%) sinon il devient de type TEXTE
- Ø création du type : MARGED
- Ø le type TEXTE devient de type MARGED s'il se trouve à DROITE (25%) sinon il devient de type TEXTE
- Ø le type MARGEG devient de type MARGEG si
- Ø son voisin de gauche est dans {MARGEG RIEN } et son voisin de droite est dans {} et son voisin du haut est dans {MARGEG RIEN } et son voisin du bas est dans {MARGEG RIEN } et est inclus dans {}
- Ø le type MARGED devient de type MARGED si
- Ø son voisin de gauche est dans {} et son voisin de droite est dans {MARGED RIEN } et son voisin du haut est dans {MARGED RIEN } et son voisin du bas est dans {MARGED RIEN } et est inclus dans {}
- Ø Fusion des zones de type MARGEG dans la direction verticale avec un seuil de 1000000
- Ø Fusion des zones de type MARGED dans la direction verticale avec un seuil de 1000000
- Ø Fusion des zones de type TEXTE dans la direction horizontale avec un seuil de 4000
- Ø Fusion des zones de type TEXTE dans la direction verticale avec un seuil de 4000
- Ø création du type : BRUIT

- Ø le type TEXTE devient de type BRUIT si le nombre d'éléments est compris entre 1 et 1 sinon il devient de type TEXTE
- Ø les zones de type BRUIT sont supprimées et ne sont pas coloriées
- Ø création du type : TITRE
- Ø le type TEXTE devient de type TITRE si le rapport hauteur moyenne/hauteur est compris entre 1.00 et 2.00 sinon il devient de type TEXTE
- Ø création du type : TITREPRINCIPAL
- Ø le type TITRE devient de type TITREPRINCIPAL si
- Ø son voisin de gauche est dans {} et son voisin de droite est dans {} et son voisin du haut est dans {RIEN } et son voisin du bas est dans {} et est inclus dans {}
- Ø création du type : NUMPAGEG
- Ø le type TITREPRINCIPAL devient de type NUMPAGEG s'il se trouve à GAUCHE (25%) sinon il devient de type TITREPRINCIPAL
- Ø création du type : NUMPAGED
- Ø le type TITREPRINCIPAL devient de type NUMPAGED s'il se trouve à DROITE (25%) sinon il devient de type TITREPRINCIPAL
- Ø Fusion des zones de type TITREPRINCIPAL dans la direction horizontale avec un seuil de 6000
- Ø Fusion des zones de type TITRE dans la direction horizontale avec un seuil de 6000
- Ø le type TITRE devient de type TITRE s'il se trouve au CENTRE de la largeur de la page (25%) sinon il devient de type TEXTE
- Ø création du type : LETTRINE
- Ø le type IMAGE devient de type LETTRINE si
- Ø son voisin de gauche est dans {} et son voisin de droite est dans {} et son voisin du haut est dans {} et son voisin du bas est dans {} et est inclus dans {TEXTE }
- Ø le type IMAGE devient de type LETTRINE si
- Ø son voisin de gauche est dans {MARGEG RIEN } et son voisin de droite est dans {TEXTE } et son voisin du haut est dans {} et son voisin du bas est dans {TEXTE RIEN } et est inclus dans {}
- Ø le type LETTRINE devient de type LETTRINE si le rapport largeur/hauteur est compris entre 0.90 et 1.10 sinon il devient de type IMAGE
- Ø Sauvegarde de l'ensemble des zones en utilisant un rapport de mappage de : 500 et un delta de decoupage de : 0
- Ø génération de la page html

13.2 Exemple de résultat

Afin d'illustrer l'ensemble des techniques utilisées dans ce projet dans un cas concret, voici le résultat étape par étape des traitements qui constituent une analyse complète d'une page d'un ouvrage. Le nom de l'étape est précisé en légende.

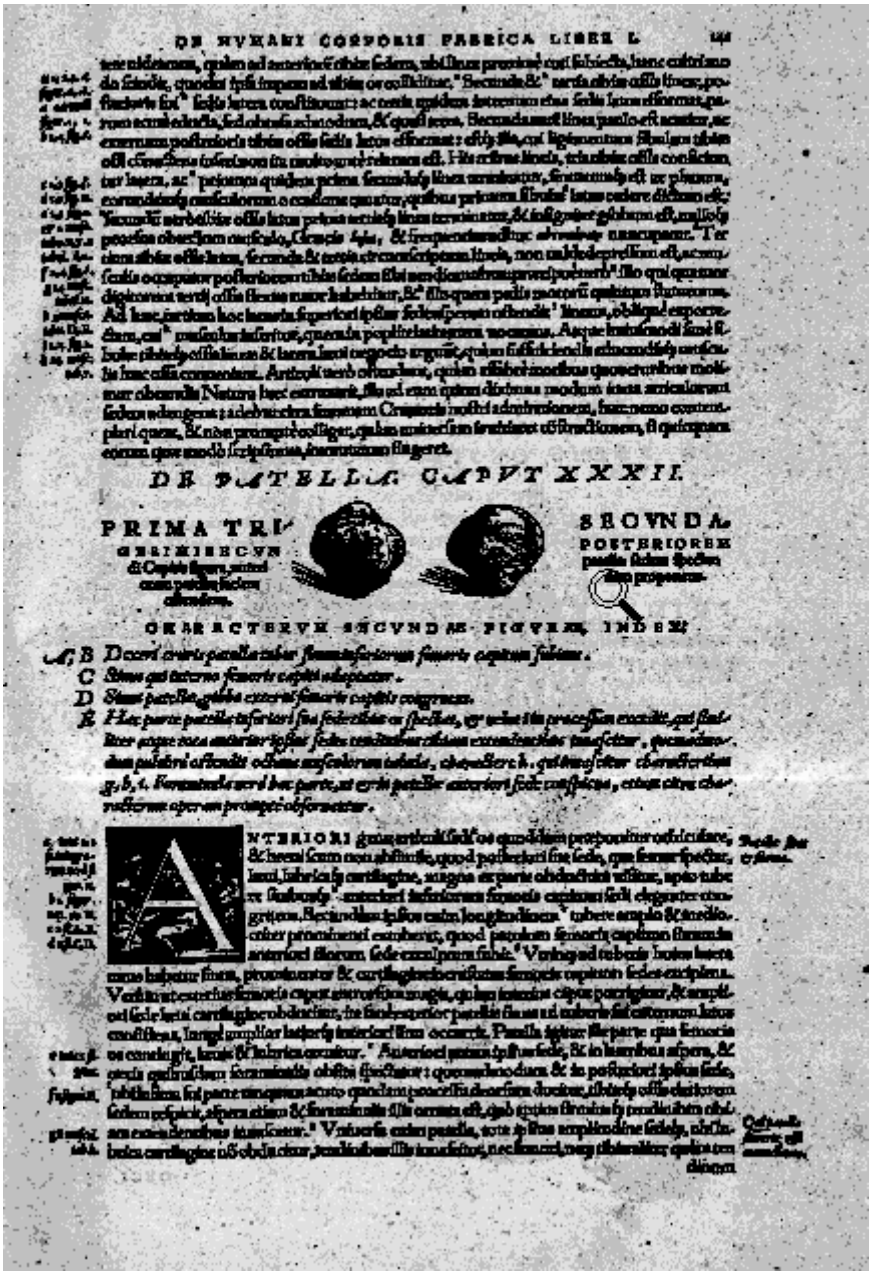


Figure 29 : image d'origine

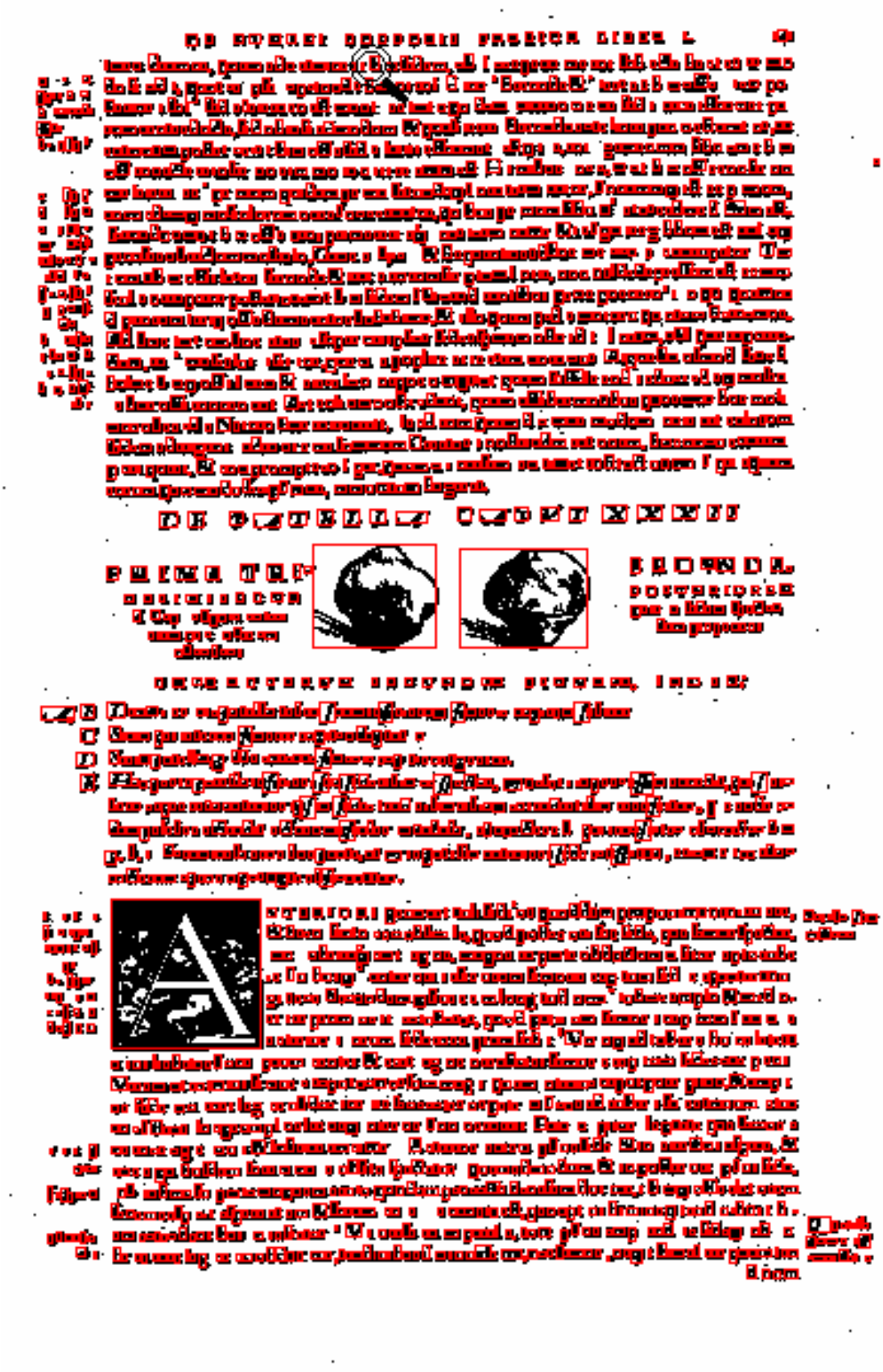


Figure 30 : image après extraction des composantes connexes

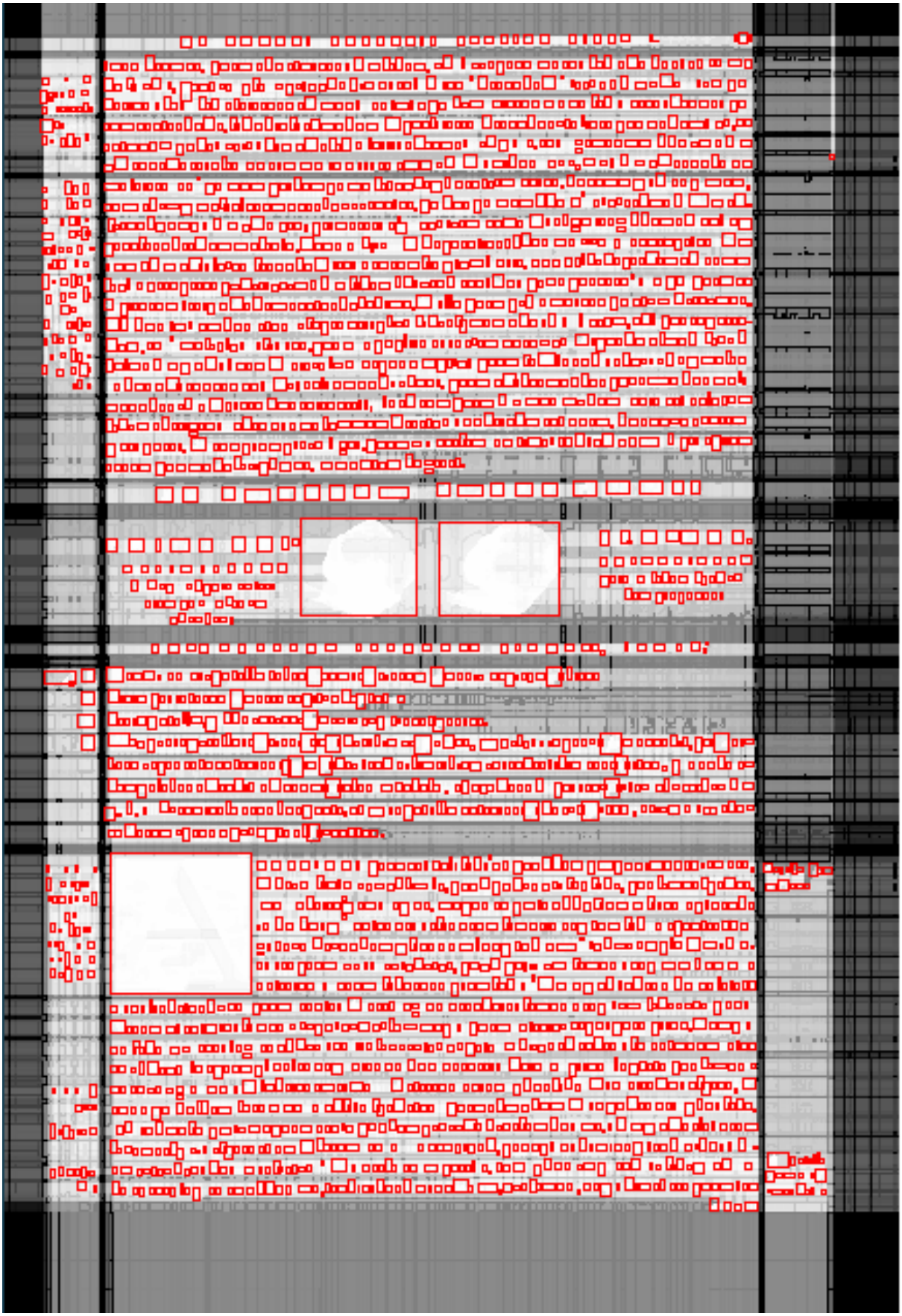


Figure 31 : image de la carte des niveaux de gris

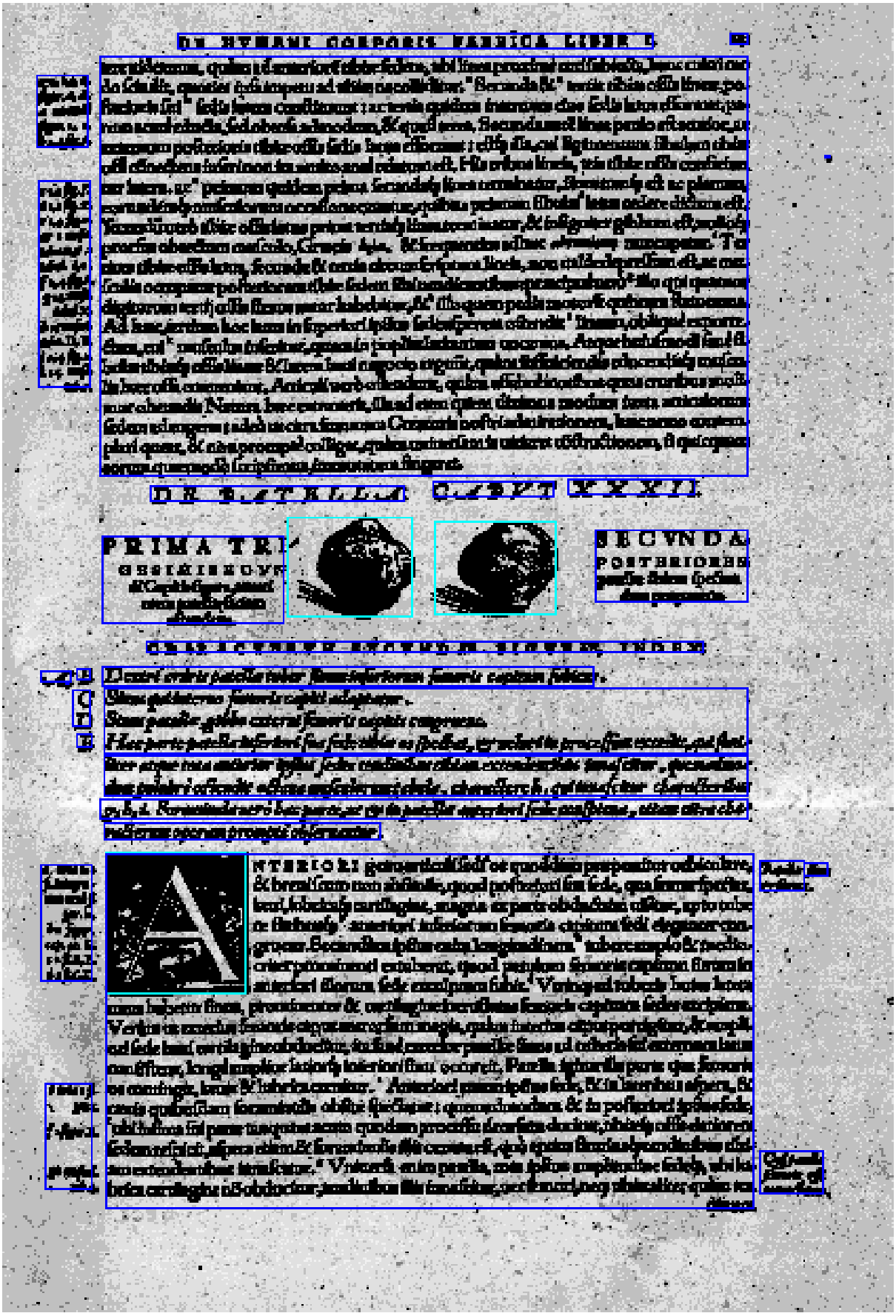


Figure 32 : image après l'étape de segmentation, en bleu foncé : zone de type texte et en bleu turquoise : zone de type image

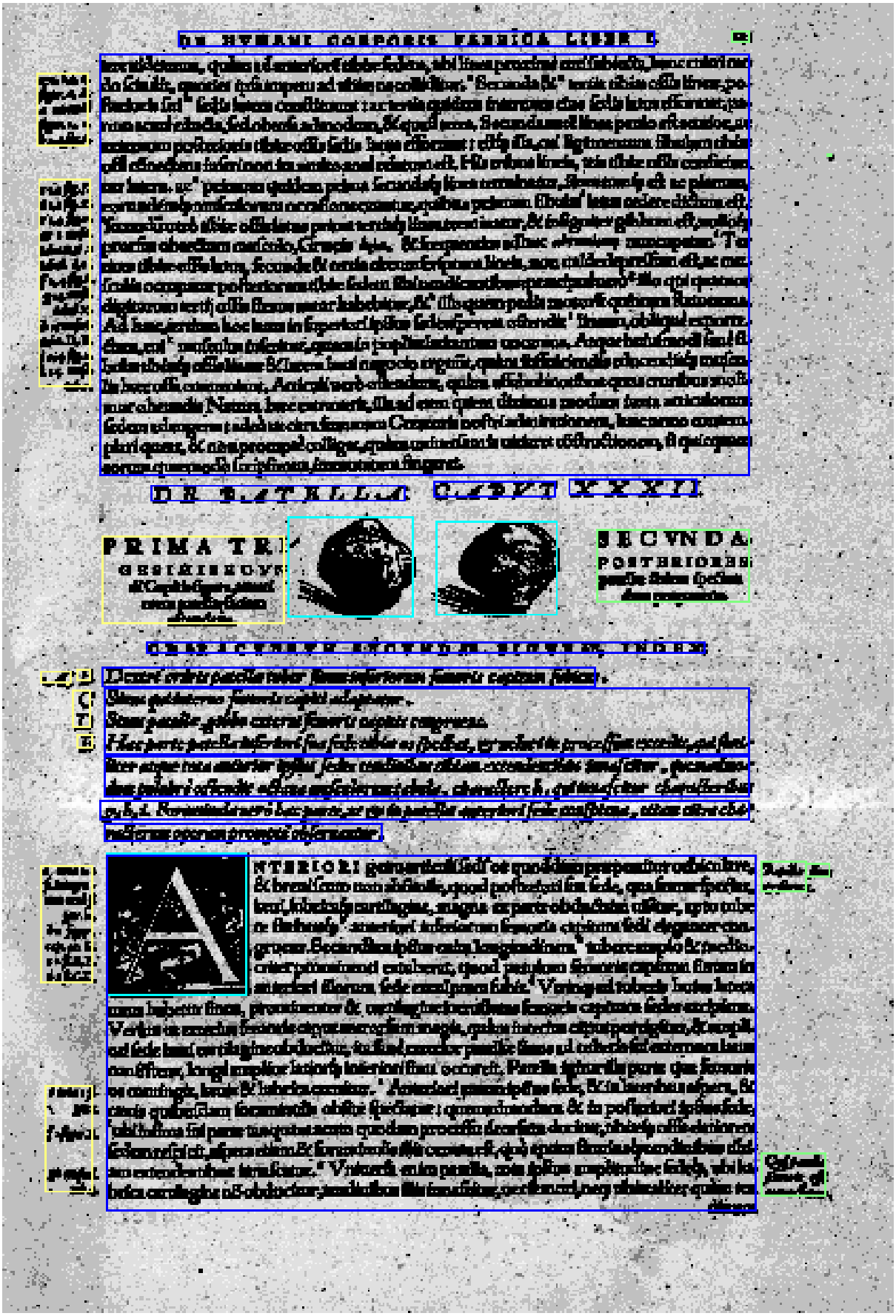


Figure 33 : image après classification selon la position géographique, en jaune : zone potentiellement marge gauche et en vert : zone potentiellement marge droite

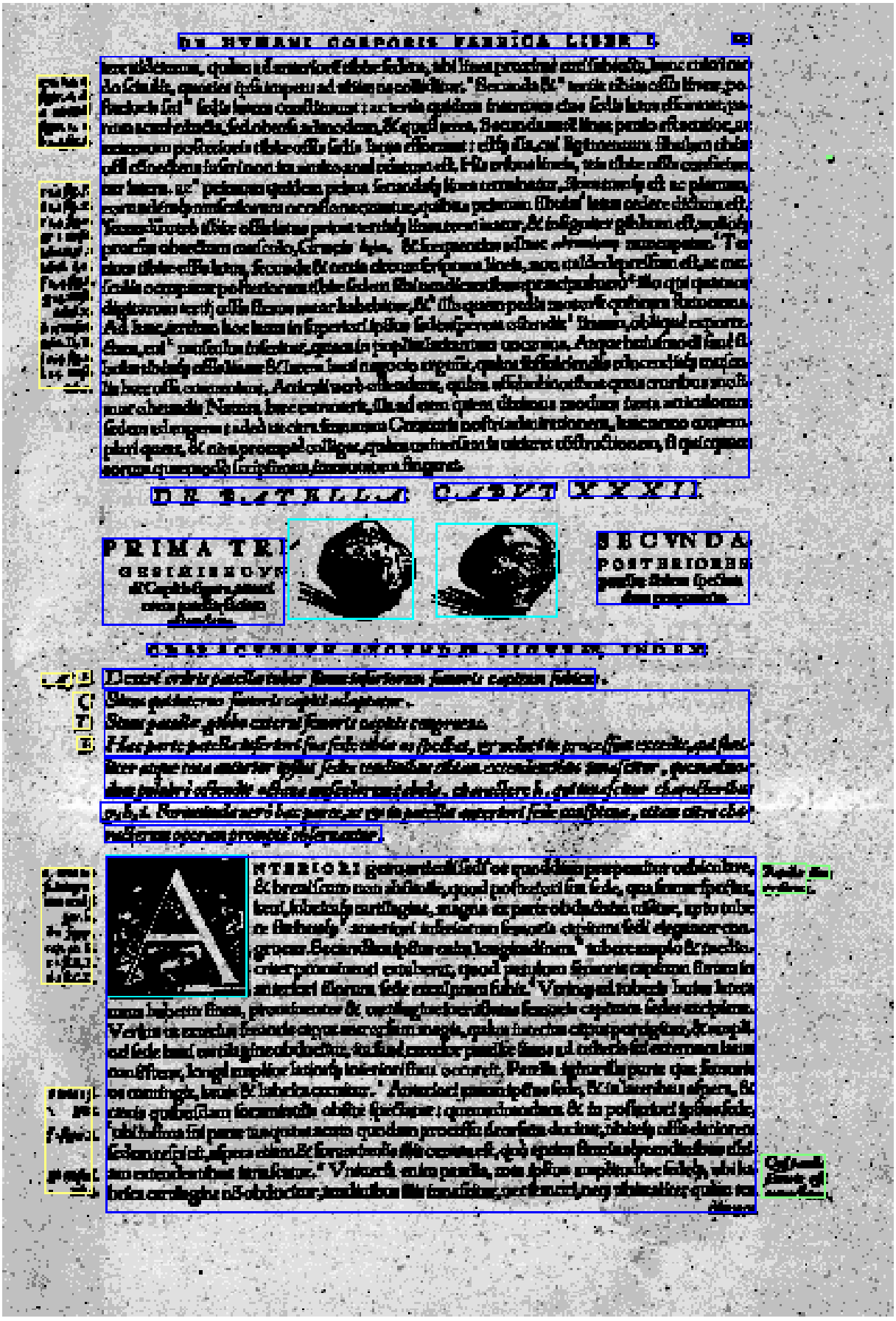


Figure 34 : image après la classification par relation de voisinage -> les marges (jaune et vert)

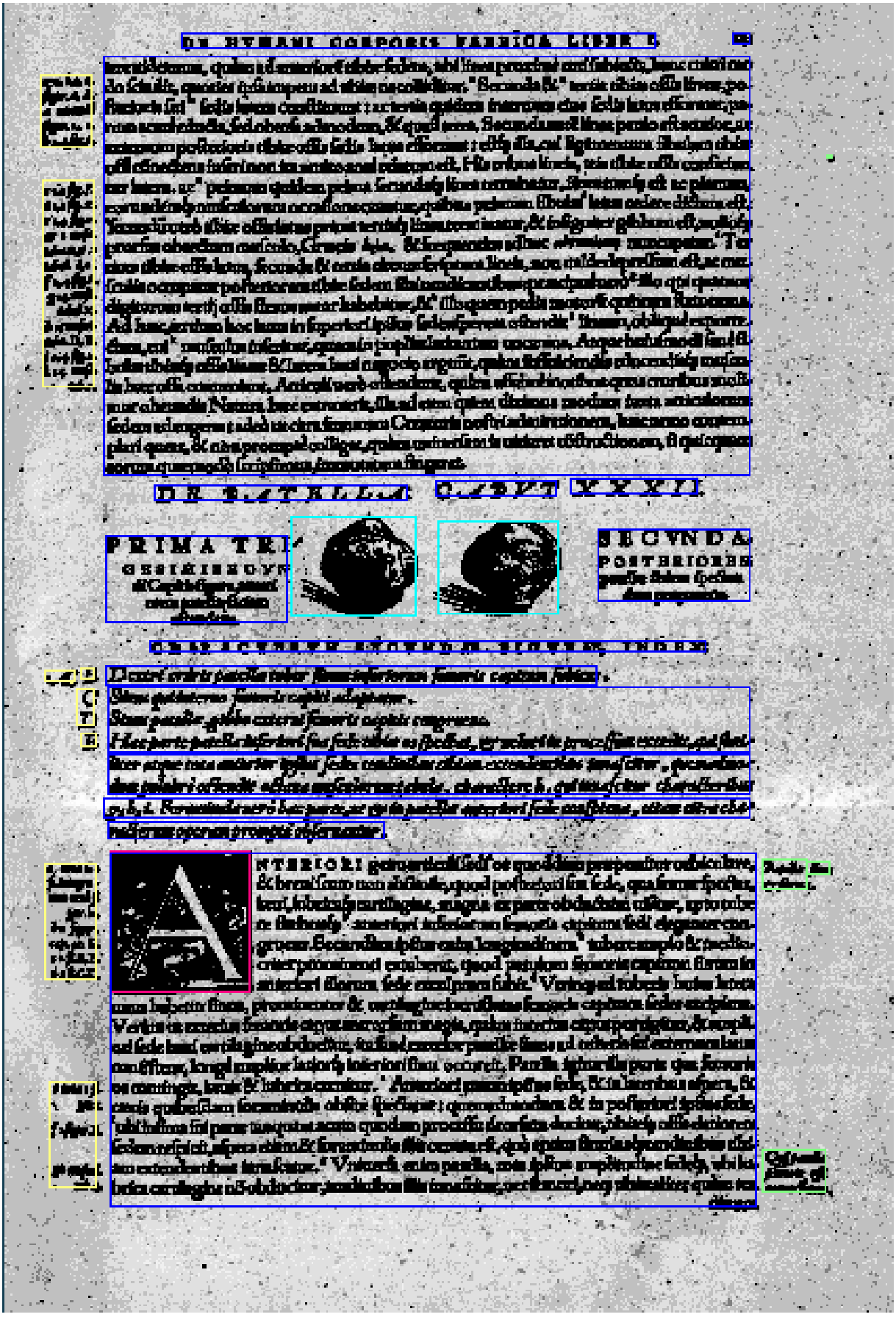


Figure 35 : image après la classification par relation de voisinage -> les lettrines (rose)

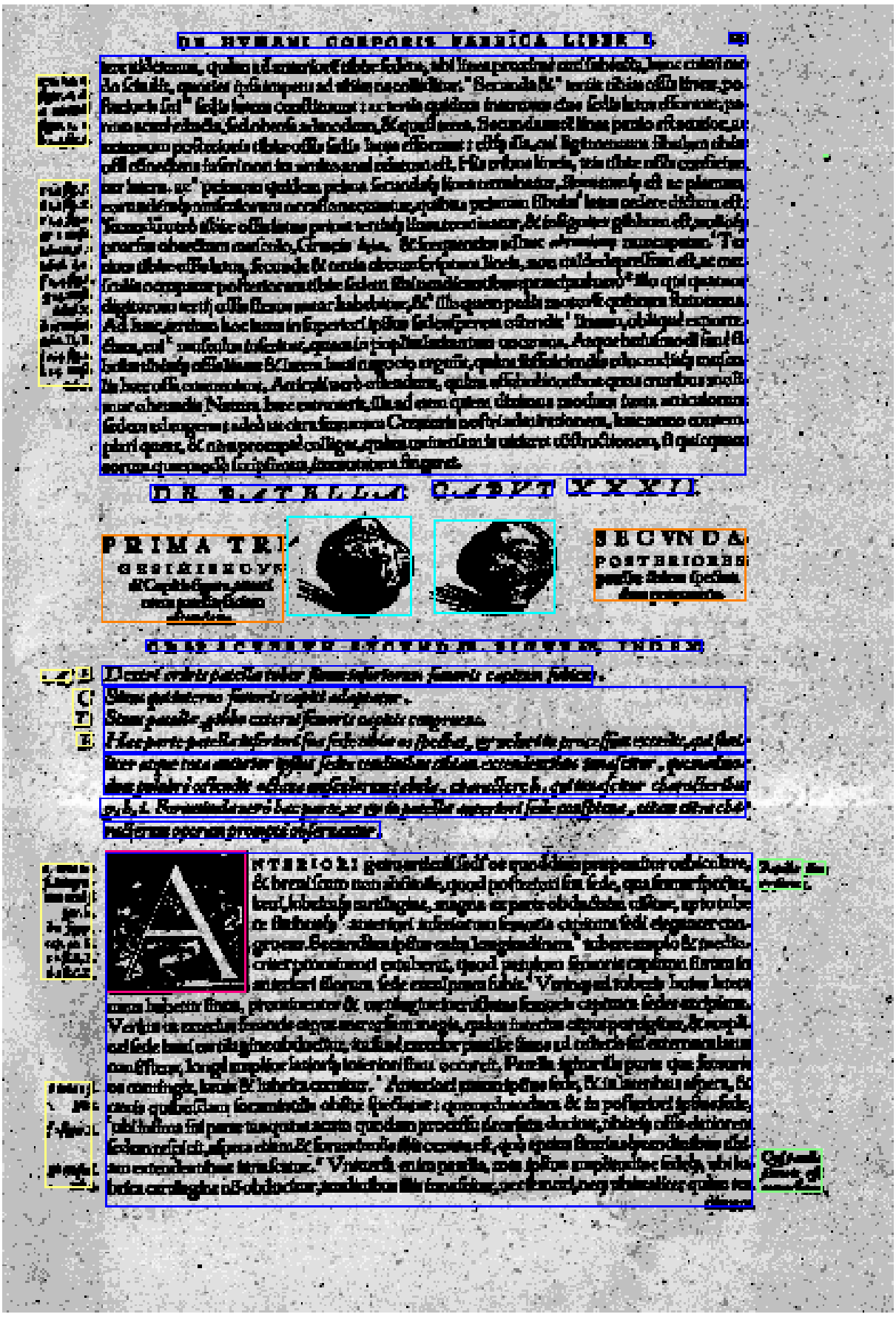
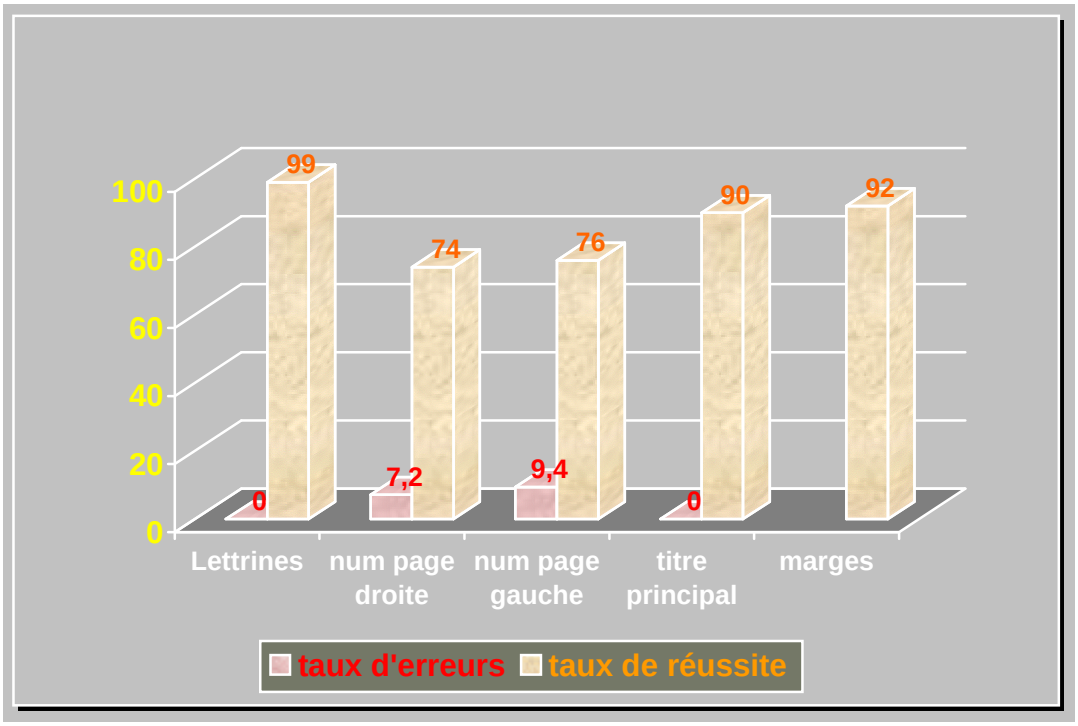


Figure 36 : image après la classification par relation de voisinage -> les légendes (orange)

13.3 Campagne de tests

Pour effectuer la campagne de tests, nous avons choisi d'utiliser l'ouvrage nommé « le vésale » qui est, de l'avis des spécialistes, un des ouvrages les plus complexes du point de vue de la mise en page. La machine utilisée possède 768Mo de mémoire vive et un processeurs cadencé à 2.5 Ghz. L'échantillon, composé de 250 pages d'ouvrage, a nécessité (a titre indicatif) environ 9h de calculs. Le scénario appliqué à cet échantillon est celui décrit au paragraphe 13.1.



Au moment de l'impression de ce rapport, 6 types de classes ont pu être testé : la classe « marge gauche », « marge droite », « lettrine», num page gauche », « num page droite » et « titre principal ».
La vérification de la classe affectée à chaque zone est effectuée de manière visuelle sur chaque page analysée, ce qui représente un travail relativement fastidieux (250 images).

A la vue des résultats, nous pouvons dire que :

- Ø 92% de reconnaissance pour les marges n'est pas suffisant. Il faudra envisager d'améliorer ces résultats lorsque les problèmes auront été identifiés (problème de segmentation ou problème de règles de classification)
- Ø 99% des lettrines ont été bien classées ce qui est très satisfaisant. De plus le taux de fausses détections est nul.
- Ø 90% de reconnaissance sur le titre est un bon résultat
- Ø les numéros de page obtiennent un taux de reconnaissance moins élevé et ceci peut s'expliquer de plusieurs façons :
  - o les règles d'identification sont basés sur des relations de voisinage utilisant les résultas de classe titre (10% non reconnus)
  - o les règles ne sont pas suffisamment précises

  le scénario est donc à revoir pour cette classe.

14 Conclusion et évolutions

Après avoir étudié les spécificités des ouvrages imprimés anciens, nous avons élaboré un outil conçu pour extraire automatiquement la structure physique et les différents objets (image, texte, lettrine, marges…) susceptibles d'apparaître dans chacune des pages numérisées.

Cet outil extrait les composantes connexes d'une page puis les fusionne en utilisant deux types d'information : la distance et la valeur de niveau de gris la plus faible rencontrée sur le chemin reliant les deux composantes connexes. Cette méthode donne de bien meilleurs résultats que celles testées jusqu'à maintenant dans le cadre de ce projet.
Cependant, il reste quelques cas dans lesquels cette méthode est mise en échec. Il est donc possible d'améliorer les résultats en explorant les pistes encore disponibles. Il est à noter que les tests sur lesquels nous basons cette conclusion concernent l'ouvrage « le vésale » qui, selon les experts du domaine, est un des ouvrages les plus complexes en matière de mise en page, ce qui nous permet d'être relativement optimistes quant à la future utilisation de cet outil.
En ce qui concerne la classification des blocs détectés au niveau de la segmentation, les résultats sont extrêmement satisfaisants. L'outil actuel est un système extrêmement flexible qui permet à l'utilisateur de définir ses propres classes et règles d'extraction. La bonne utilisation des règles et la création des scénarios demande nécessairement une petite période de prise en main. Une version a été fournie à plusieurs experts paléographes et il sera intéressant d'analyser les retours et impressions de ces derniers.

Les points qui seront probablement soumis à des évolutions ultérieures sont donc les suivants :

- Amélioration de la séparation texte/image
  - Texte inclus dans une illustration
  - Texte inclus dans un cadre
- Apprentissage des paramètres

15 Bibliographie

[Akindele93] O.T. Akindele, A. Belaid. Page Segmentation by Segment Tracing. In Proc. of the 2nd International Conference on Document Analysis and Recognition, p341--344, 1993.

[Baird92] H Baird. Background structure in document images. In Advances in Structural and Syntactic Pattern Recognition, ed H. Bunke. p253-269. 1992.

[Belaid97] A. Belaïd, Conception automatisée de modèles de page en vue de leur utilisation en reconnaissance de documents, Workshop on Electronic Page Models (LAMPE'97). 1997.

[Hadjar02] K. Hadjar, O. Hitz, L. Robadey, R. Ingold. Configuration REcognition Model for Complex Reverse Engineering Methods: 2(CREM). Proceedings of the 5th International Workshop on Document Analysis Systems. p469-479 2002

[Nagy84] G. Nagy and S. Seth. Hierarchical representation of optically scanned documents. In 7th International Conference on Pattern Recognition (ICPR), p347-349, 1984..

[Ogorman93] L O'Gorman. The Document Spectrum for Page Layout Analysis In IEEE Trans. On PAMI.15(11). p1162-1173. 1993

[Debora] DEBORA project – deliverable 5.1 / July 1999
disponible sur : http://debora.enssib.fr/DEL.5.1.PDF consulté le 01/10/2003

[Couasnon] Bertrand Coüasnon, Jean Camillerapp, DMOS, une méthode générique de reconnaissance de documents : évaluation sur 60 000 formulaires du XIXe siècle**,** *in Actes du Colloque International Francophone sur l'Écrit et le Document (CIFED'02),* Hammamet, Octobre 2002.

16 ANNEXE A : Manuel programmeur

Le logiciel qui a été utilisé pour coder cette application est le logiciel Microsoft Visual C++ dans sa version 6. Dans cette partie sont présentées les informations utiles à la reprise du développement de cette application. Pour obtenir des informations plus techniques, veuillez vous référer aux commentaires du code source de l'application.

16.1 Diagramme UML

Ici sont présentées les principales classes qui composent l'application. Le but ici est de présenter les différents liens qui unissent les différentes classes. Ce diagramme n'est donc pas exhaustif et ne sont visibles que les attributs des classes par soucis de clarté. Chaque classe est décrite plus précisément dans les parties suivantes.

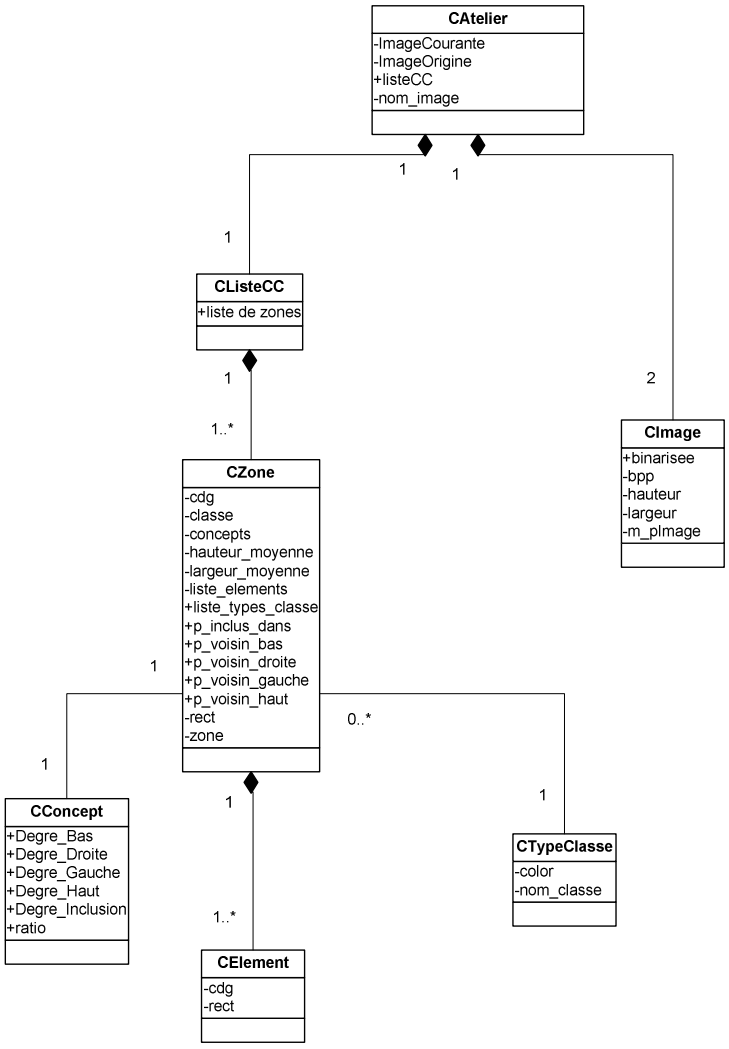


Figure 37 : diagramme UML simplifié

16.2 CElement

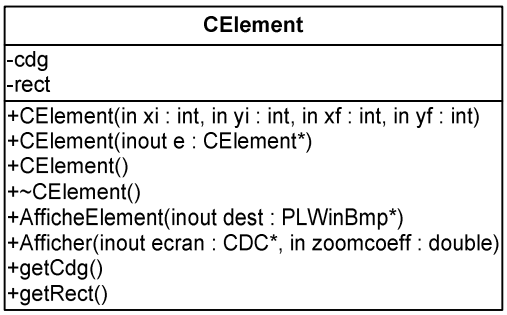


Figure 38 : la classe CElement

Cette classe est destinée à décrire une composante connexe et plus particulièrement le rectangle englobant d'une composante détectée.
Elle est donc décrite par un rectangle qui est de type CRect et qui nous permet d'accéder aux propriétés du rectangle comme par exemple ses coordonnées, sa largeur, sa hauteur …

Rect : attribut de type CRect décrivant un rectangle
Cdg : double représentant le centre de gravité du rectangle rect

16.3 CZone

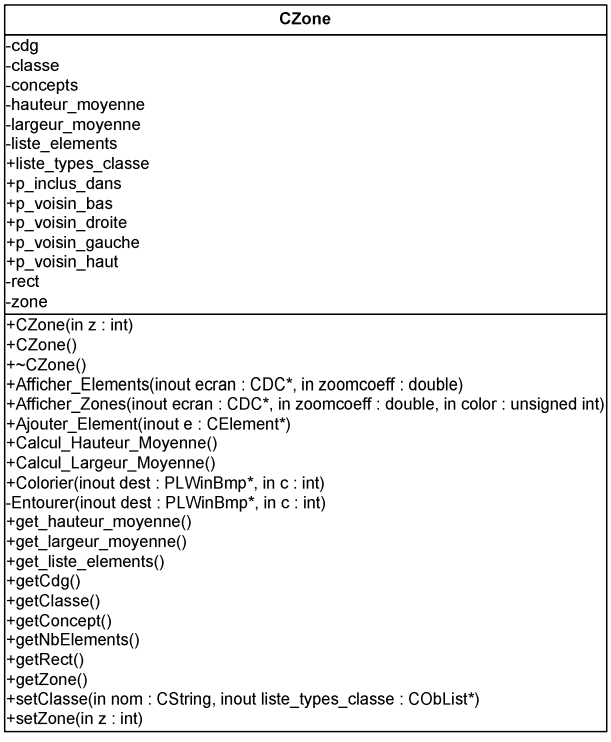


Figure 39 : la classe CZone

CZone est une classe destinée à regrouper un ensemble de CElements. Elle va nous permettre de manipuler la liste des CElements contenus dans cette zone. Elle nous permet d'obtenir un certain nombre de statistiques sur les éléments contenus (hauteur moyenne, largeur moyenne…). De plus, elle possède un pointeur sur les zones voisines (gauche,droite,haut, bas).

16.4 CListeCC

CListeCC
+liste +m_pCC +m_pVecteur
+CListeCC() +~CListeCC() +Affectation_Geographique(inout Image : CImage*) +Affiche_CC(inout ecran : CDC*, in zoomcoeff : double, in k : int) +Affiche_Zone(inout ecran : CDC*, in zoomcoeff : double) +Cherche_Voisins(inout Image : CImage*) +ChercherPoint(inout point : CPoint*, inout ecran : CDC*, in zoomcoeff : double) +Classification(inout Image : CImage*) +Classification_Geographique(in classe_source : CString, in pos : char, in classe_dest : CString) +Classification_Voisinage() +Conversion_CC_liste(in petitx : int, in petity : int, in grandx : int, in grandy : int) +DecoupageImages(inout image_origine : CImage*, in rapport : double) +DetectionContours(inout P1 : CImage*, inout P2 : CImage*) +Fusion(inout img : unsigned char* *, in x : int, in y : int, in u : int, in v : int, in seuil : int, in distance : double) +Generer_HTML() +MessageNbZone() +Ppv_horizontal(inout dest : CImage*, in seuil : int) +Ppv_horizontal_CDG(inout dest : CImage*, in seuil : int) +Ppv_vertical(inout dest : CImage*, in seuil : int) +Ppv_vertical_CDG(inout dest : CImage*, in seuil : int) +Renommer_CC(inout remplacer : CZone*, inout par : CZone*) +Supprime_inclusion_Gde_CC() +Supprime_Intersection_Image() +Supprime_Pte_CC(inout image : CImage*)

Figure 40 : la classe CListeCC

CListeCC est une classe qui va nous permettre de manipuler une liste de CZones et notamment d'appliquer les fonctions de fusion et de recherche du plus proche voisin.

16.5 CImage

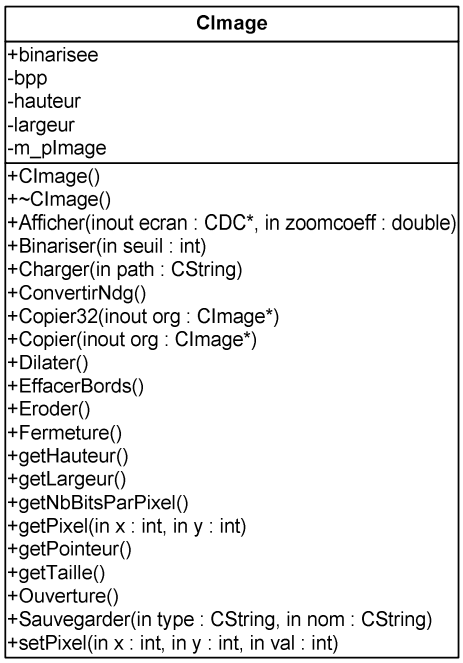


Figure 41 : la classe CImage

Cette classe regroupe toutes les actions qu'il est possible d'effectuer sur une image.

16.6 CAtelier

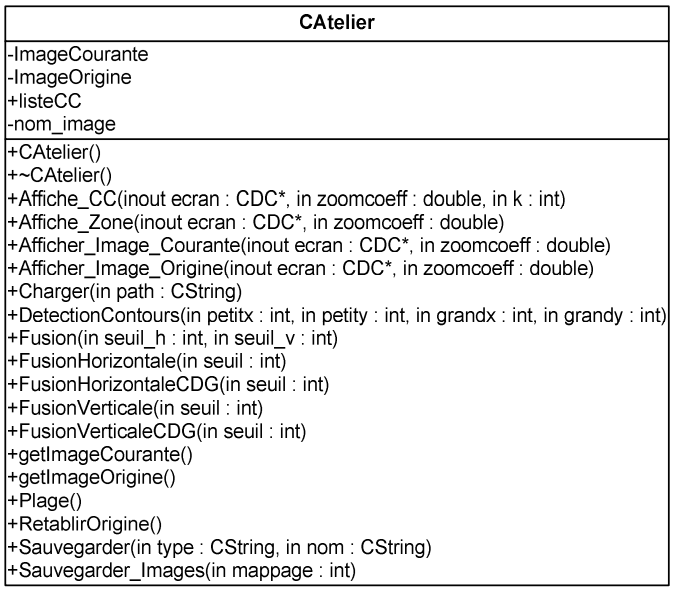


Figure 42 : la classe CAtelier

La classe CAtelier regroupe l'ensemble des initialisations, créations d'objets et fonctions principales. Le code contenu dans cette classe est typiquement le code que l'on aurait placé dans la classe CDoc du projet. Ce choix est justifié par le fait que :
Il suffit d'instancier un objet de la classe CAtelier pour avoir accès aux fonctions principales, ainsi si le projet devait servir dans une autre application (MDI, SDI ou autre) la procédure serait extrêmement simplifiée. (Copie des .h et .cpp , initialisation d'un objet CAtelier)

16.7 CConcept

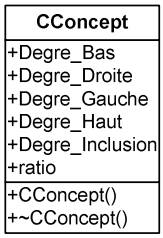


Figure 43 : la classe CConcept

Cette classe constitue un regroupement de caractéristiques propres à chaque zone.

16.8 CTypeClasse

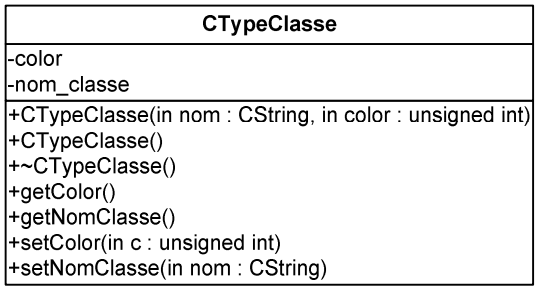


Figure 44 : la classe CTypeClasse

Cette classe est utile pour créer une nouvelle description des zones. Elle sert donc à créer un nouveau type en lui affectant un nom et une couleur de représentation.

16.9 Cscenario

Cette classe est prévue pour gérer une liste d'objet de type Cregle. Un objet « scenario » contient donc une liste d'objets « règles ». Cette classe permet la gestion d'un scenario et permet notamment d'ajouter une règle, d'effacer une règle, de déplacer une règle au sein du scénario, de sauvegarder et de charger le scénario.

16.10 Cregle

C'est la classe mère de tous les autres types de règles. Elle possède les attributs communs à tout type de règle et des fonctions virtuelles qu'il est donc obligatoire d'implémenter dans les classes filles. Pour scénariser un traitement, il suffit de créer une nouvelle classe qui hérite de CRegle. Ensuite, il faut :

1. rajouter les attributs caractéristiques de la règle et définir un constructeur surchargé.
2. redéfinir la fonction **executer** avec le code à exécuter lors de l'appel de la règle

3. redéfinir la fonction **getDescription** avec une description précise du traitement effectué, celle ci s'affichera dans la fenêtre de scénario
4. redéfinir la fonction **serialize** avec les attributs à sauvegarder
5. mettre un identifiant unique dans l'attribut **balise** de chaque constructeur
6. et enfin dans la fonction charger de la classe scénario, compléter le code en fonction du nom de la balise choisit en étape 5.

- CRegleCG

Permet la gestion des règles de classification géographique.

- CRegleCV

Permet la gestion des règles de classification selon les relations de voisinage.

- CRegleCC

Permet la gestion des règles de classification selon les caractéristiques des zones.

- CRegleBin

Permet de scénariser les traitements de binarisation et de binarisation automatique.

- CRegleContour

Permet de scénariser les traitements de détection de composantes connexes et d'étude de leur taille.

- CRegleFusion

Permet de scénariser les traitements de fusion des composantes connexes dans le sens horizontal ou vertical

- CRegleFusion

Permet de scénariser les traitements de fusion des composantes connexes dans le sens horizontal ou vertical

- CRegleIntersection

Permet de scénariser les traitements de fusion des zones d'un certain type qui ont une intersection non vide.

- CRegleMorph

Permet de scénariser les traitements dits morphologiques tels que : ouverture, fermeture, érosion, dilatation et inversion.

- CReglePlage

Permet de scénariser l'application de l'algorithme des plages.

- CReglePost

Permet de scénariser les post-traitements tels que la sauvegarde des différentes zones détectées ainsi que la génération e la page HTML.

- CRegleSR

Permet de scénariser les traitements de sauvegarde et de restauration de l'image courante.

- CRegleType

Permet de scénariser les traitements de création d'un nouveau type de zone.

- CRegleSuppType

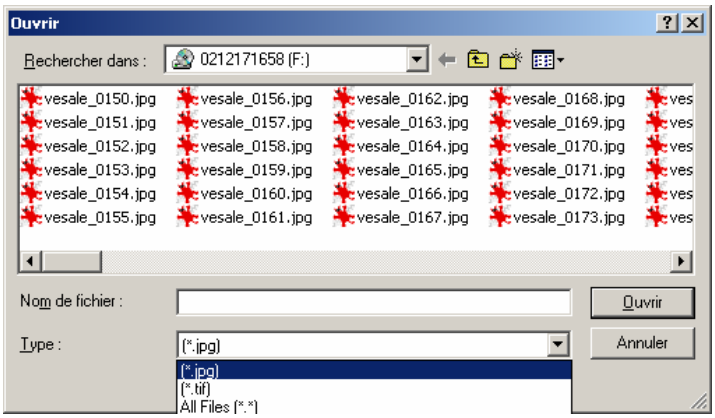
Permet de scénariser les traitements de suppression et/ou coloriage d'un type de zone.

17 ANNEXE B : Manuel d'utilisation

17.1 Traitement d'une image

17.1.1 Ouvrir une image

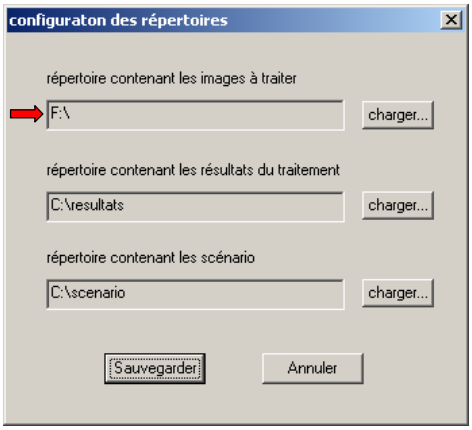
Pour effectuer des traitements sur une seule image à la fois, cliquez sur le menu « fichier » puis « ouvrir ». Celui-ci vous offre la possibilité d'ouvrir des images au format « jpeg » et « tiff ». Cependant, vous pouvez décider d'ouvrir tout autre type d'image, en mettant le filtre sur « all files » et en sélectionnant votre image.



Il est important de connaître les actions effectuées par cette commande :

- L'image sélectionnée est convertie au format « bmp »
- Elle est ensuite convertie en une image constituée de 256 niveaux de gris.

Remarque : le répertoire par défaut proposé dans la boite de dialogue « ouvrir » est le répertoire « images originales » qui se situe dans le répertoire d'exécution du programme. Vous pouvez changer ce répertoire de façon à ce que ce soit celui-ci qui soit exploré lorsque vous effectuez la commande « ouvrir ». Pour cela, allez dans le menu « configuration » puis « répertoires »



Puis changez le répertoire qui se trouve dans la zone d'édition correspondant au « répertoire contenant les images à traiter » en cliquant sur le bouton « charger… ».
Cliquez sur le bouton « sauvegarder » pour que les modifications prennent effet.

17.1.2 Traitements prédéfinis

La zone suivante qui se situe à gauche de la fenêtre de l'application correspond à des raccourcis regroupant les actions les plus fréquemment utilisées. Il est à noter que les actions effectuées prennent en compte les paramètres renseignés dans la boite de dialogue accessible par le menu « configuration » puis « paramètres » :

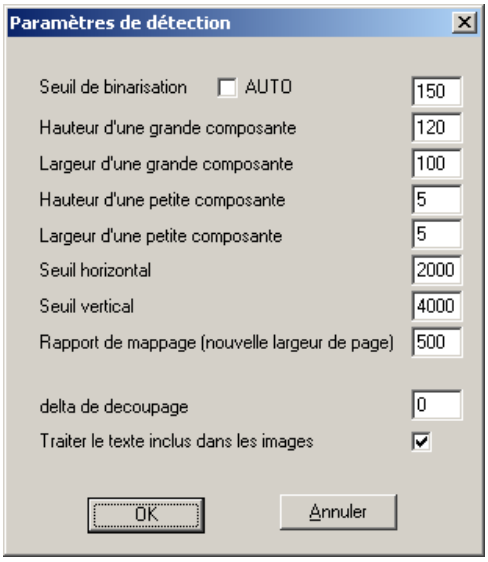


Figure 45 : fenêtre de configuration des paramètres

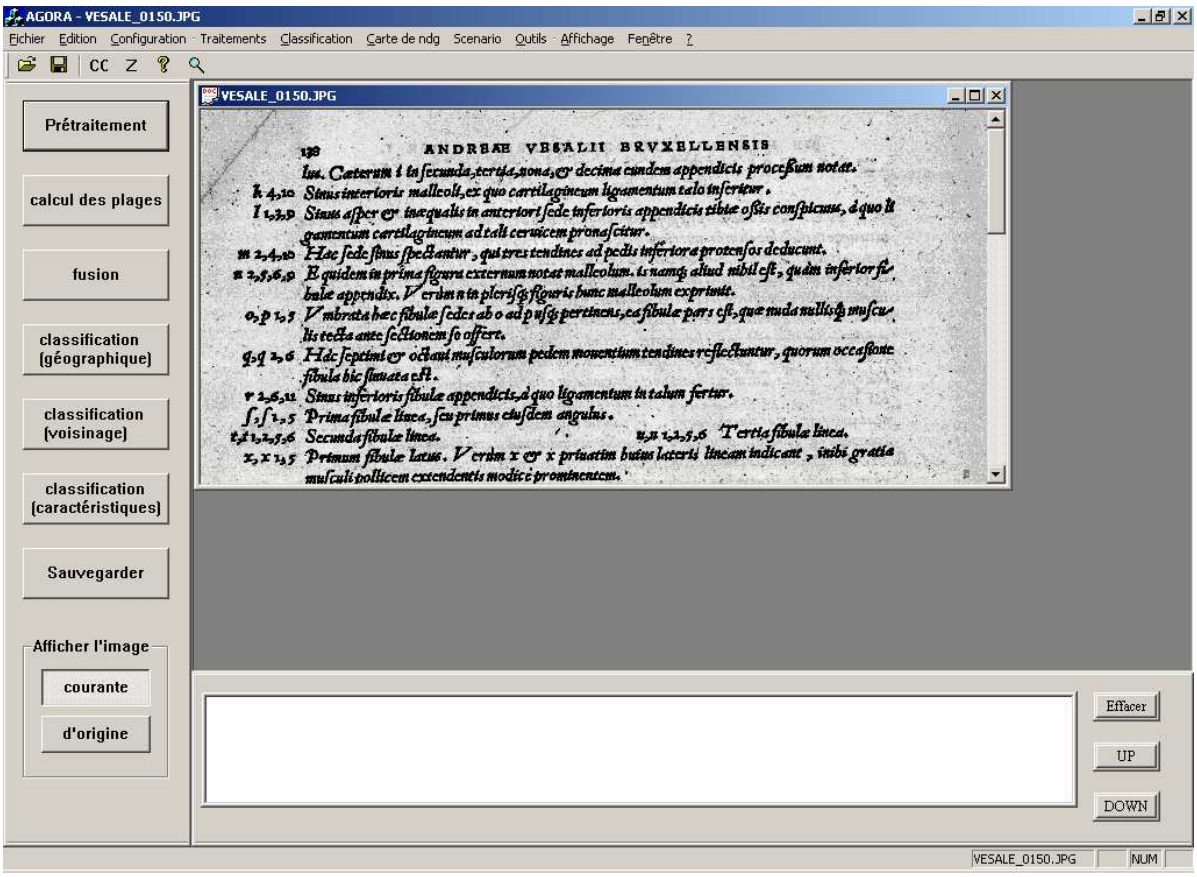


Figure 46 : la barre d'outil placée sur la gauche permet d'accéder rapidement aux traitements prédéfinis

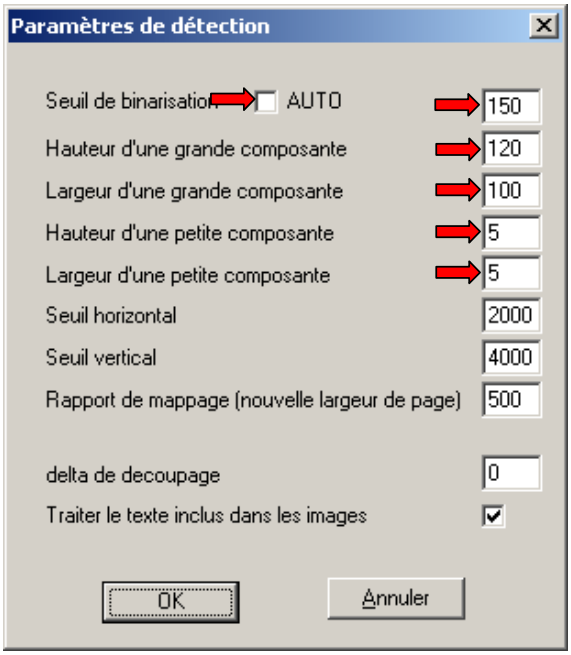
17.1.2.1 Prétraitement

Après avoir ouvert une image, c'est le premier traitement qui doit être effectué. Celui ci regroupe les actions suivantes :

- L'image est binarisée en fonction du « seuil de binarisation » ou bien de façon automatique si la case AUTO est cochée
- Détection des composantes connexes
- Affectation du label « IMAGE », « BRUIT », « TEXTE » en fonction de la taille des composantes connexes définie par les paramètres « hauteur d'une grande composante », « largeur d'une grande composante », « hauteur d'une petite composante » et « largeur d'une petite composante ».
- Les zones de type « BRUIT » sont supprimées
- Les zones de type « IMAGE » ayant une intersection non vide sont fusionnées
- Les zones de type « TEXTE » qui sont incluses dans des zones de type « IMAGE » sont renommées « TEXTEI »

Remarque : si la case « traiter le texte inclus dans les images » n'est pas cochée, les zones de type « TEXTEI » sont supprimées.

Paramètres utilisés pendant cette étape :



17.1.2.2 Calcul des plages

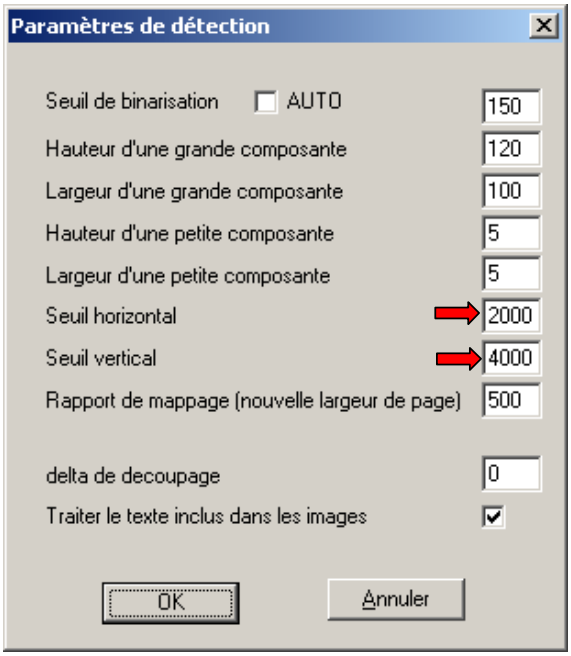
Effectue le calcul des plages de niveaux de gris. Cette étape doit suivre l'étape de « prétraitement » ou alors être appliquée sur une image binarisée. L'image résultante est créée dans le but d'être utilisée lors du processus de fusion.

17.1.2.3 Fusion

Cette étape regroupe les deux opérations suivantes (dans l'ordre d'exécution) :

- Ø La fusion horizontale qui utilise le paramètre « seuil horizontal »
- Ø La fusion verticale qui utilise le paramètre « seuil vertical »

Paramètres utilisés pendant cette étape :



L'exécution de cette étape n'est légitime que si l'on dispose de l'image des plages et des composantes connexes donc après les étapes « prétraitements » et « calcul des plages ».
En effet, cette étape a pour rôle de réaliser la fusion des composantes connexes en fonction de la distance entre celles-ci et de la valeur du niveau de gris des pixels se trouvant sur le chemin les reliant.

17.1.2.4 Classification

Cette opération réalise l'étiquetage des zones identifiées grâce à l'opération de fusion. Ici, on doit faire appel aux différentes règles de classification proposées :

- Ø Géographique
- Ø Voisinage
- Ø Caractéristiques

Pour une description de chaque règle voir chapitre 17.3

17.1.2.5 Sauvegarder

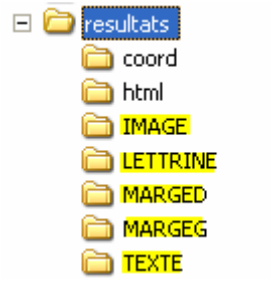
En utilisant le bouton « sauvegarder » de la barre d'outil placée à gauche, la sauvegarde s'effectue en utilisant l'image d'origine.
Vous pouvez cependant préciser l'image qui sera utilisée pour cette opération en choisissant dans le menu « traitements », « sauvegarder les zones détectées » soit

- Ø en utilisant l'image d'origine
- Ø en utilisant l'image courante
- Ø en utilisant l'image temporaire

Cette opération effectue les actions suivantes pour chaque zone identifiée :

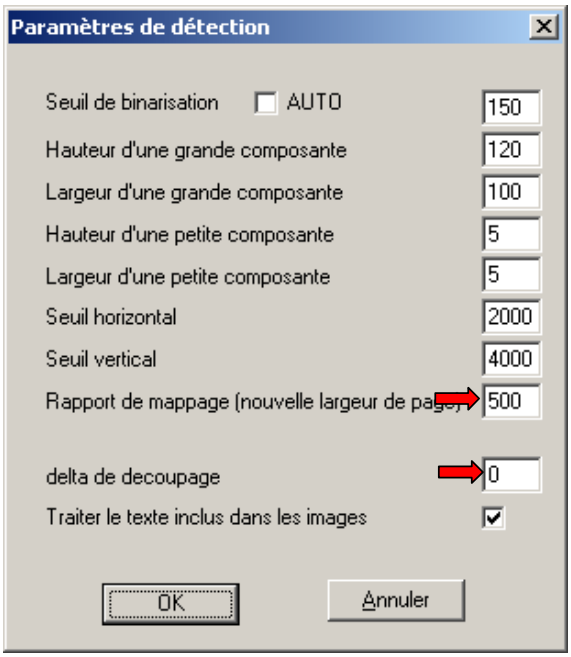
- Ø Création d'un répertoire du nom de classe identifiée dans le répertoire « resultats »

Ex :

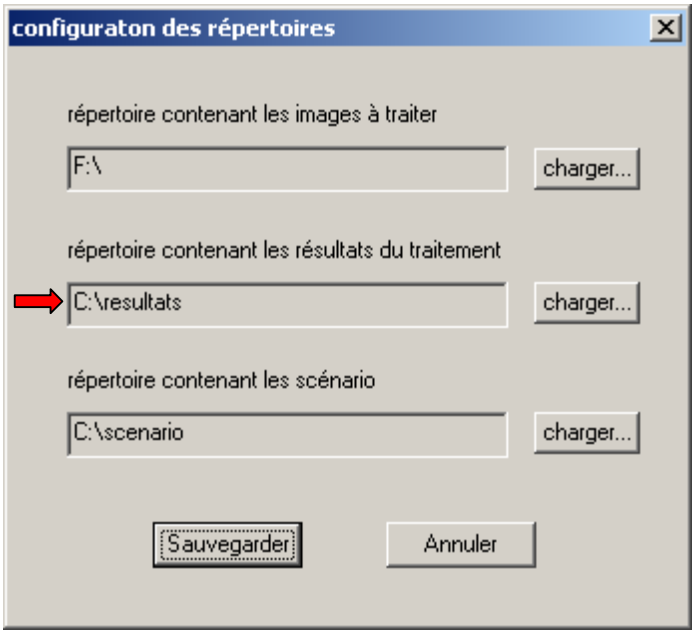


- Ø Sauvegarde de la région de l'image originale correspondant au rectangle englobant de la zone sous le nom « nom original+compteur+nom de la classe »
- Ø Inscription dans les fichiers de coordonnées «../resultats/coord/ »
  - o du nom de l'image créée à l'étape précédente
  - o ses coordonnées dans l'image originale
  - o sa classe

Les noms de fichiers de coordonnées sont « nom originale + _xy.txt » et « nom originale + _xy_map.txt ».
Le premier indiquera les coordonnées exactes de l'image d'origine. Le second indiquera les coordonnées mappées en fonction du paramètre mappage (correspond à la nouvelle largeur de page en pixels)



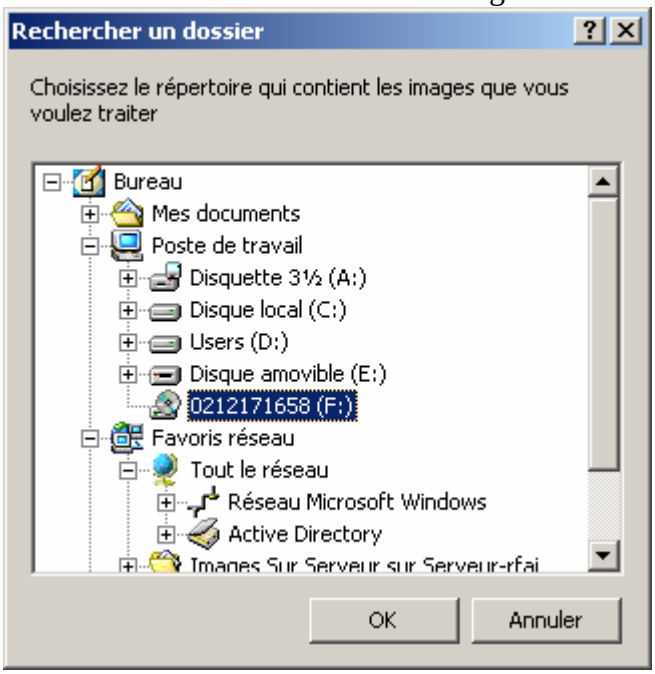
Remarque : le répertoire par défaut utilisé pour la sauvegarde des zones se situe dans le répertoire d'exécution du programme et se nomme « resultats ». Vous pouvez changer ce répertoire de façon à ce que ce soit celui-ci qui soit utilisé lorsque vous effectuez l'action « sauvegarder ». Pour cela, allez dans le menu « configuration » puis « répertoires »



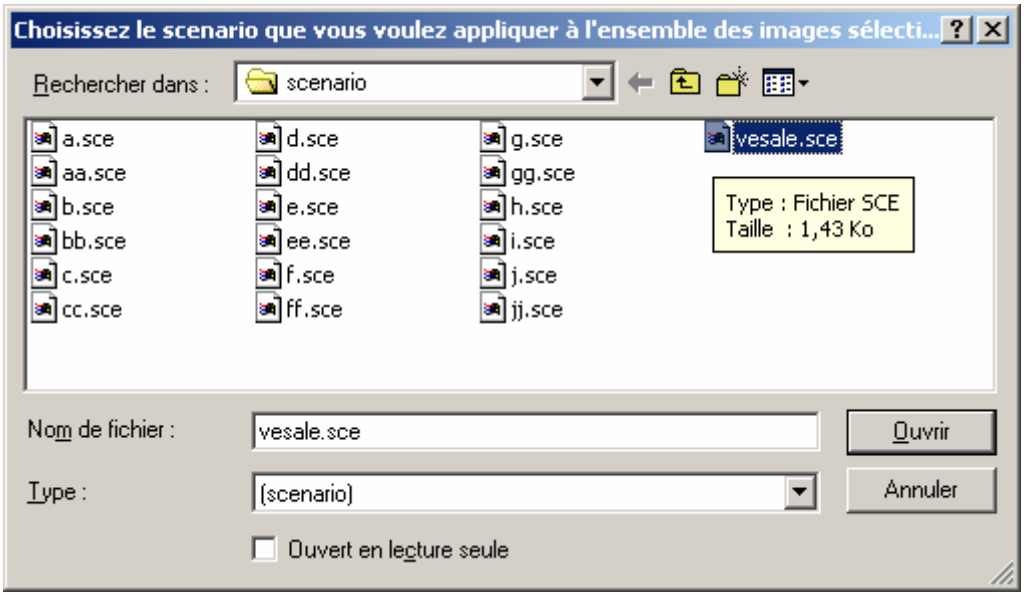
Puis changez le répertoire qui se trouve dans la zone d'édition correspondant au « répertoire contenant les résultats du traitement » en cliquant sur le bouton « charger… ».
Cliquez sur le bouton « sauvegarder » pour que les modifications prennent effet.

17.2 Traitement par lot

Ce traitement est accessible après le lancement de l'application et avant l'ouverture d'une image en le sélectionnant dans le menu « outils » puis « traitement par lots ».
Vous devrez alors choisir le répertoire qui contient les images sur lesquelles vous voulez effectuer le traitement. Toutes les images se trouvant dans ce répertoire seront alors traitées.



vous devrez également choisir le scénario que vous désirez appliquer sur ces images :



Les actions effectuées sont exactement celles décrites dans le chapitre précédent et dans le même ordre, à savoir :

- Ø Prétraitement
- Ø Calcul des plages
- Ø Fusion
- Ø Classification
- Ø sauvegarde

17.3 création des scénarios

les traitement dits « scénarisables » sont les traitement que l'on va pouvoir ajouter à un scénario. Ils pourront donc être sauvegardés, chargés, déplacés…Ces traitements sont repérés par un étoile (*) dans les différents menus.

17.3.1 menu « classification »

Toutes les règles de classification respectent le protocole suivant :

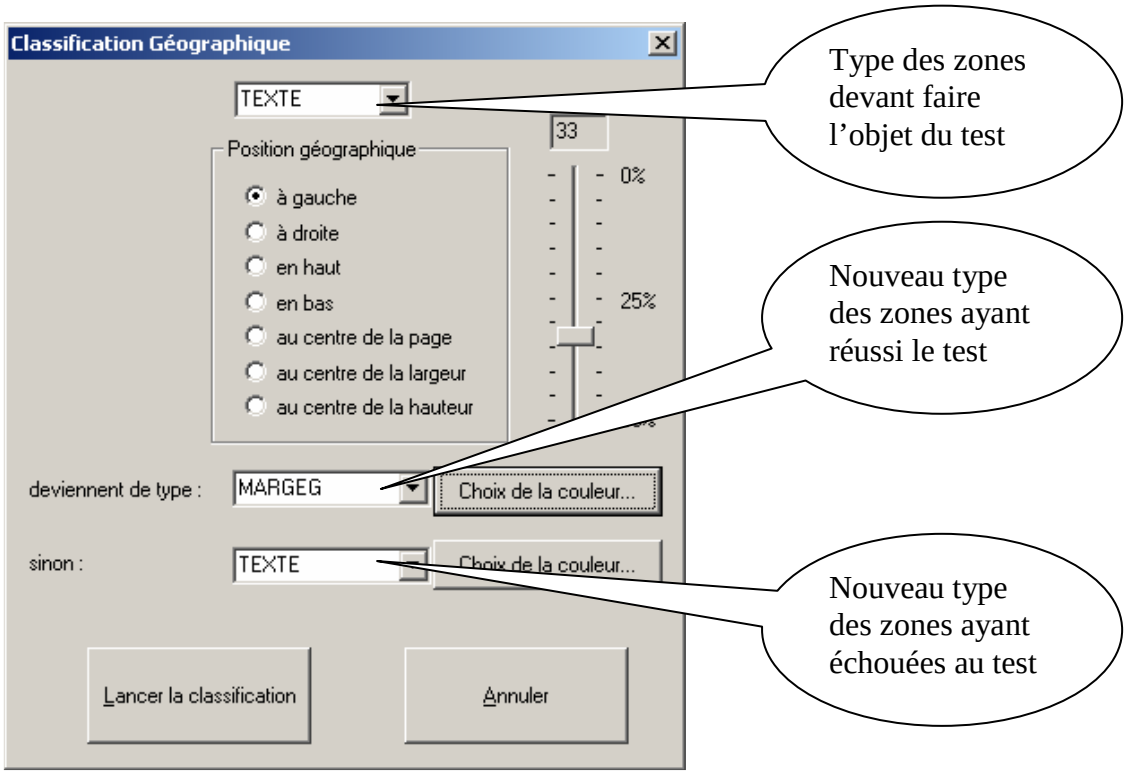
- Ø L'utilisateur précise le type des zones concernées par le test à réaliser : TYPE_DEPART
- Ø puis précise le type dans lequel transformer la zone si le test réussi : TYPE_SI_REUSSI
- Ø et le type si la zone ne respecte pas les conditions de test : TYPE_SI_ECHEC

Le choix des types SI_REUSSI et SI_ECHEC se fait de deux façons :

- Ø On souhaite affecter à la zone un type déjà existant, il suffit alors de le choisir dans la liste proposée.
- Ø On souhaite créer un nouveau type de zone, il suffit alors d'affecter un nouveau nom à la zone d'édition puis de choisir la couleur qui sera affectée à ce nouveau type.

CLASSIFICATION GEOGRAPHIQUE

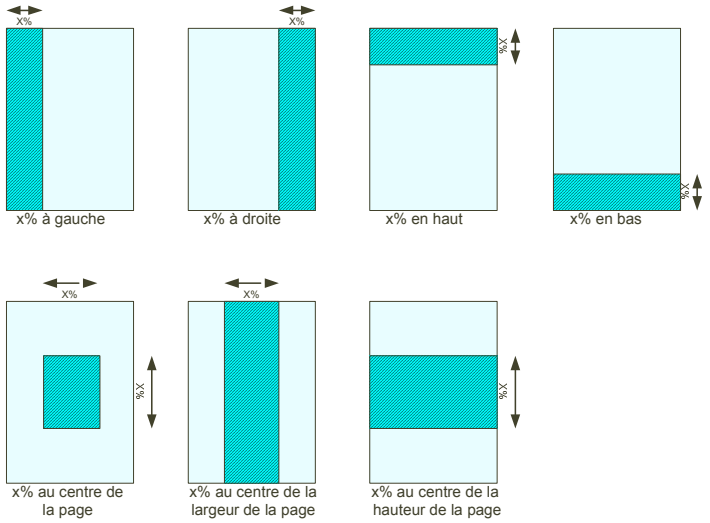
La boite de dialogue suivante permet d'exécuter et d'ajouter au scénario une règle de classification selon la position géographique du centre de gravité du type de zone considéré.



Après avoir renseigné les différents types concernés, il faut choisir la région de la page concernée par le test ainsi que le pourcentage que représente la région par rapport à la page.

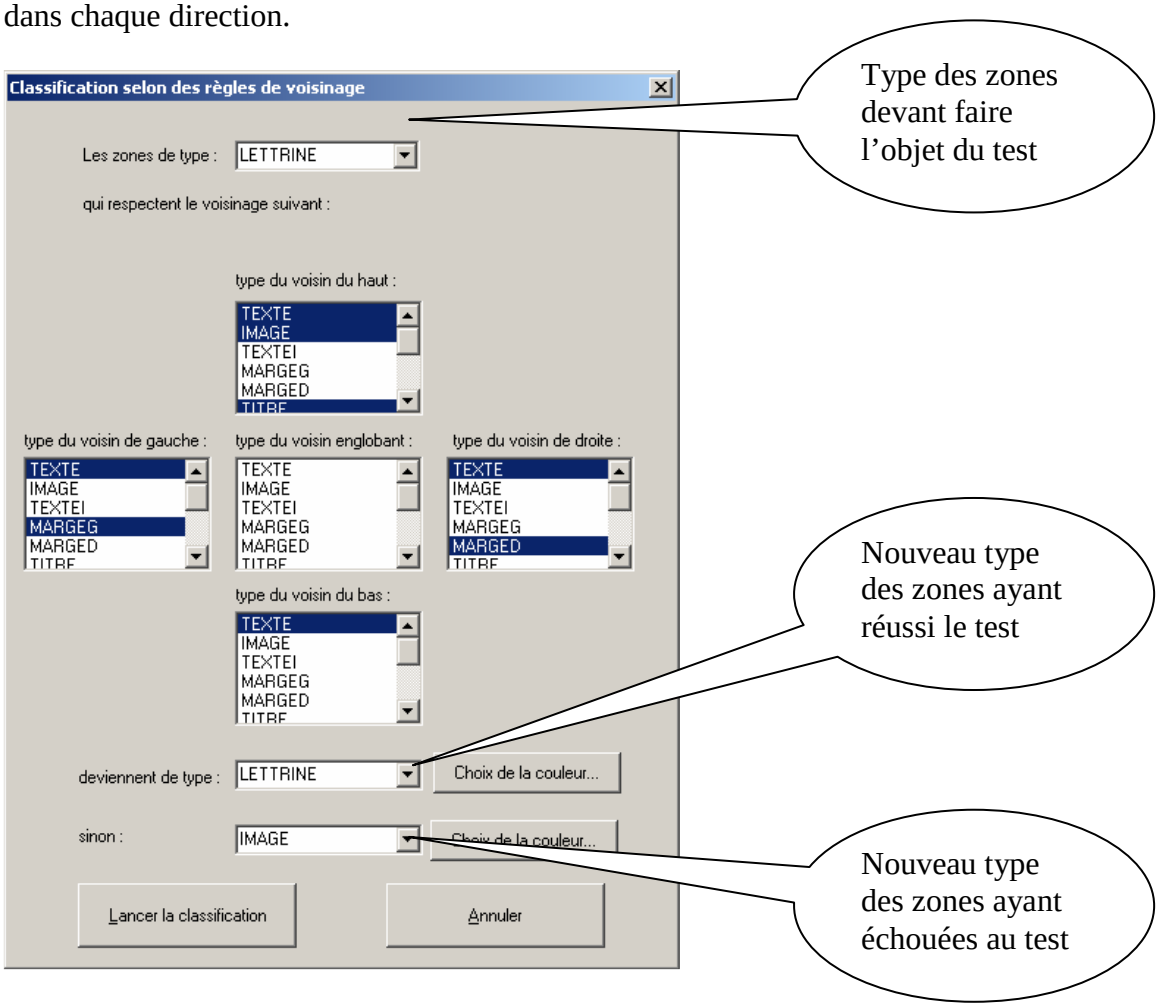
Sur l'exemple ci-dessus, la règle crée est la suivante :

Toutes les zones de type TEXTE ayant leur centre de gravité dans la premier tiers gauche de la largeur de la page seront renommées MARGEG sinon elles seront renommées TEXTE.
Voici les différentes régions concernées par les tests :

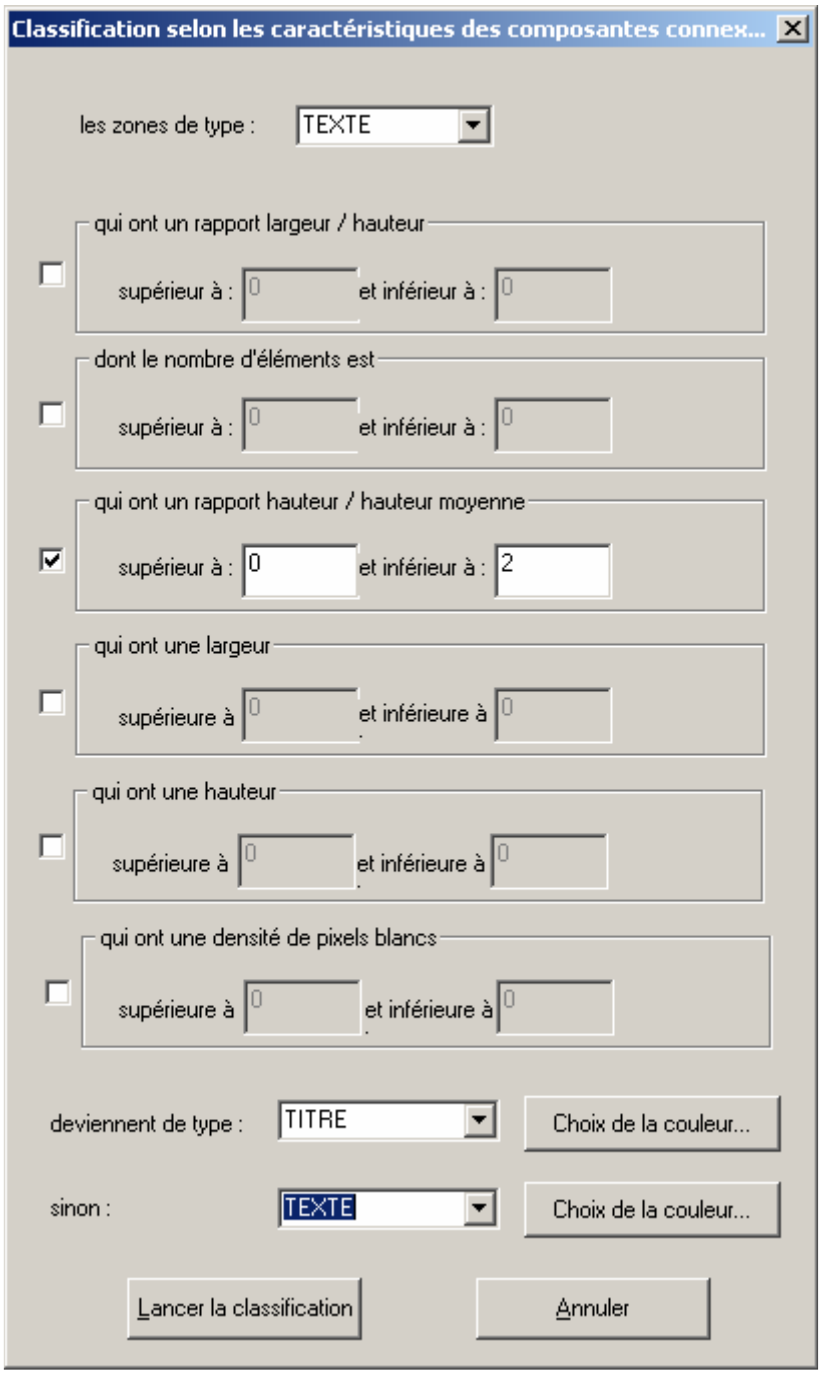


CLASSIFICATION PAR RELATIONS DE VOISINAGE

Nous avons donc la possibilité d'étudier l'appartenance ou non d'une zone à une classe en fonction de son voisinage. Pour cela, il est possible de préciser la nature des voisins potentiels dans chaque direction.



CLASSIFICATION SELON LES CARACTERISTIQUES DES COMPOSANTES CONNEXES



Ø rapport hauteur/largeur :

En général, utilisé pour détecter un certain type de zone image. La lettrine, par exemple est une zone de type image de forme carrée donc on va pouvoir préciser un rapport hauteur/largeur proche de 1.
De même , ce qui est appelé « bandeau » est une zone de type image qui à un rapport hauteur/largeur relativement constant dans un ouvrage, en général il est trois fois plus large que haut.

Ø Nombre d'éléments

Le nombre d'éléments décrit le nombre de composantes connexes contenues dans une zone. Ce test peut s'avérer utile pour détecter les zones ne contenant qu'un seul élément. Selon l'étape à laquelle est placée cette règle, il est probable qu'un tel type de zone corresponde à du bruit ou une tache sur la page.

Ø Rapport hauteur/hauteur moyenne

Ici, on considère la hauteur de la zone et la hauteur moyenne de l'ensemble des éléments contenus dans cette zone. Ce test peut être très utile pour détecter une zone qui ne forme qu'une seul ligne. En effet, en précisant que l'on veut toutes les zones de type TEXTE qui ont un rapport hauteur/hauteur moyenne proche de 1 alors on va pouvoir extraire tous les titres du document.

Ø Largeur et hauteur

Utilisation de la taille de la zone

Ø Densité de pixels blancs

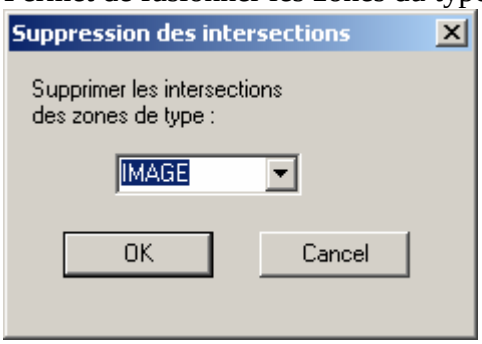
Prévue pour être utilisée sur une image binarisée, cette caractéristique permet de déceler les zones qui sont composées d'une forte densité de pixels blancs (ou noirs).

SUPPRESSION ET/OU COLORIAGE D'UN TYPE DE ZONE

La suppression va permettre d'éliminer l'ensemble des zones de tel type. Elles n'auront donc plus d'existence en tant que zone et ne seront ni affichées, ni sauvegardées. On pourra préciser une couleur de remplissage, laquelle sera utilisée sur l'image de travail et ceci afin d'appliquer des traitements spécifiques. Il est possible de ne pas supprimer de la liste les zones en décochant la case à cocher, dans ce cas les zones seront simplement coloriées en fonction du choix effectué.
Voici l'interface utilisée pour détruire et/ou colorier un type de zone :

SUPPRESSION DE L'INTERSECTION D'UN TYPE DE ZONE

Permet de fusionner les zones du type sélectionné qui ont une intersection non vide.



17.3.2 menu « scenario »

- Ø **SAUVEGARDER :** sauvegarde du scénario courant
- Ø ***CHARGER* :** le scénario courant est remplacé par celui que l'on souhaite charger
- Ø ***AJOUTER* :** le scénario sélectionné est ajouté à la suite du scénario courant.
- Ø ***EXECUTER* :** exécution du scénario courant

17.4 autres traitements

17.4.1 menu « Traitements »

BINARISATION

Cette fonction effectue la binarisation de l'image en cours. Elle offre la possibilité de pré visualiser le résultat dans une fenêtre d'aperçu. Il y a deux façons d'utiliser la pré-visualisation :

- en cliquant directement sur l'élément « binarisation »: l'aperçu est alors réalisé sur la page entière.
- En sélectionnant une région sur l'image puis en cliquant sur l'élément « binarisation »: l'aperçu est alors effectué sur cette région

Pour sélectionner une région, il faut cliquer avec le bouton droite de la souris pour sélectionner le premier coin du rectangle, puis le laisser maintenu en déplaçant la souris jusqu'au coin opposé par rapport à la diagonale du rectangle souhaité. Le rectangle apparaît alors avec une texture particulière.

Une fois dans la fenêtre de pré-visualisation, choisissez la valeur du seuil de binarisation à appliquer en déplaçant le curseur ou en indiquant directement une valeur dans la zone

d'édition. Si le résultat vous convient, appuyez sur « OK » pour effectuer le traitement sur l'image de travail. Sinon appuyez sur « Annuler » pour revenir à l'image de travail sans avoir appliqué le traitement.

BINARISATION SELON NIBLACK

Cette fonction offre la possibilité d'effectuer une binarisation automatique. A l'issue de celle-ci une valeur de seuil est proposée. Cette valeur devrait être une bonne valeur à soumettre à l'algorithme de binarisation classique pour les autres images du même ouvrage.
Notons que l 'application de cette fonction a une durée sensiblement supérieure à l'application de la binarisation classique.

INVERSER

Appliquée à une image en niveaux de gris ou binaire, cette fonction complémente à la valeur maximale de niveau de gris l'image de travail. Sur une image binaire, l'opération est donc une simple inversion des pixels blancs en pixels noirs et réciproquement.

PRETRAITEMENT

Voir chapitre sur les traitements prédéfinis. Fonction identique à celle accessible par le biais de la barre d'outil de lancement rapide.

MORPHOLOGIE

erosion
dilatation
ouverture
fermeture

COMPOSANTES CONNEXES

calculer les composantes connexes
afficher les composantes connexes

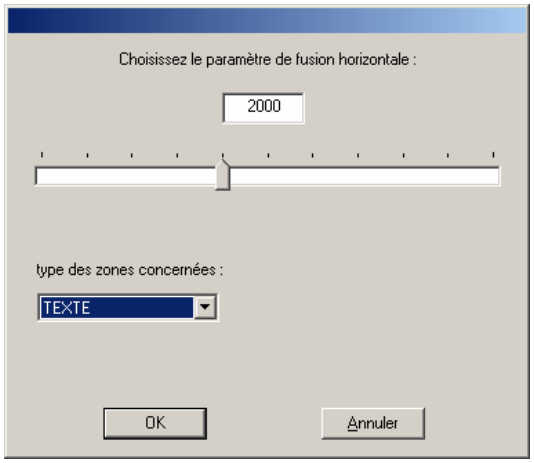
FUSION

effectuer la fusion

Fonction identique à celle accessible par le biais de la barre d'outil de lancement rapide. Effectue la fusion des zones de type « TEXTE » dans la direction horizontale puis verticale en utilisant les paramètres indiqués dans la boite de dialogue « configuration »

effectuer la fusion horizontale

Réalise la fusion dans la direction horizontale d'un type de zones que l'on aura pris le soin de sélectionner dans la boite de dialogue suivante :

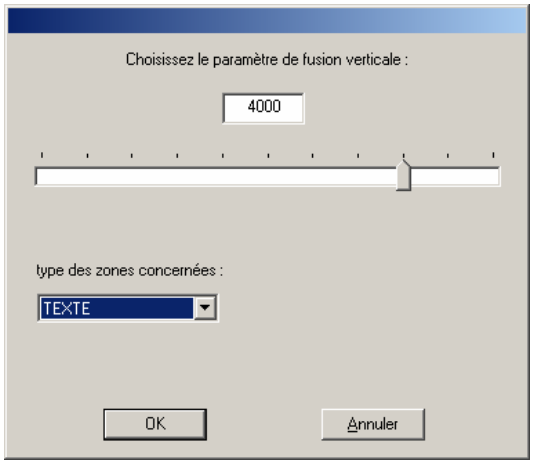


effectuer la fusion horizontale (version CDG)

Idem précédente sauf que la distance n'est plus prise en compte entre les deux centres de gravité de chaque élément de chaque zone mais entre les centres de gravité de chaque zone. (expérimental)

effectuer la fusion verticale

Réalise la fusion dans la direction horizontale d'un type de zones que l'on aura pris le soin de sélectionner dans la boite de dialogue suivante :



effectuer la fusion verticale (version CDG)

Idem précédente sauf que la distance n'est plus prise en compte entre les deux centres de gravité de chaque élément de chaque zone mais entre les centres de gravité de chaque zone. (expérimental)

SAUVEGARDER LES ZONES DETECTEES

Voir chapitre sur les traitements prédéfinis. Fonction identique à celle accessible par le biais de la barre d'outil de lancement rapide.

CLASSIFICATION

voir chapitre sur les scenarios.

GENERER HTML
cette fonction génère une page au format HTML dont la consultation est utile pour voir le découpage effectué sur le document.
Les pages HTML sont sauvegardée dans le répertoire « html » du répertoire « résultats ». Il est possible d'indiquer la largeur de la page générée dans les paramètres de configuration.

17.4.2 menu « carte de ndg »

HORIZONTALE
construit la carte des niveaux de gris uniquement dans la direction horizontale

VERTICALE
construit la carte des niveaux de gris uniquement dans la direction verticale

HORIZONTALE ET VERTICALE
construit la carte des niveaux de gris. Fonction identique à celle accessible par le biais de la barre d'outil de lancement rapide

Résumé :

Après avoir étudié les spécificités des ouvrages imprimés anciens, nous avons élaboré un outil conçu pour extraire automatiquement la structure physique et les différents objets (image, texte, lettrine, marges…) susceptibles d'apparaître dans chacune des pages numérisées.

Mots clés :

Structure - extraction - ouvrage - composante connexe – segmentation - classification - ancien - renaissance

Summary :

After having studied specificities of the old printed books, we worked out a tool designed to automatically extract the physical structure and the various objects (image, text, reference letter, margins...) likely to appear in each digitized page.

Keywords :

Structure - extraction - work - related components - segmentation - classification - old - rebirth

JY RAMEL Laboratoire Informatique Polytech tours 64 avenue Jean Portalis 37200 TOURS		Stéphane LERICHE Promotion 2004 2004