Segmentation non supervisée d'images de document en paragraphes

Remi Vieux, Jean-Philippe Domenger

Segmentation non supervisée d'images de document en paragraphes

Rémi Vieux, Jean-Philippe Domenger

LaBRI - Université de Bordeaux - CNRS UMR 5800 - 341 Cours de la libération, 33400 Talence

RÉSUMÉ. *Dans cet article, nous présentons une méthode de segmentation des images de documents dont la finalité est le découpage des images en paragraphes. Nous proposons une chaîne complète, basée sur l'application récursive de différents traitements et algorithmes de segmentation connus comme X-Y cutNagy and Seth et l'analyse des rectangles blancs maximauxBreuel. L'ensemble de la chaîne de traitement, ainsi que le paramétrage de chacun des algorithmes est guidé par des statistiques calculées sur l'image ou les blocs en cours de traitement. Ainsi, nous nous affranchissons totalement de l'étape de réglage des paramètres, que ce soit par une estimation manuelle ou automatique sur une base de validation. Enfin, nous proposons un système original et facilement extensible pour la détection de différent types de mise en page de paragraphes. Nous extrayons donc non seulement la segmentation physique du document en paragraphes, mais aussi des informations sur la mise en page de chacun des paragraphes extraits. Nous illustrons les performances de notre algorithme sur la base de donnée complexe de documents historiques utilisée lors de la compétition ICDAR2011.*

ABSTRACT. *In this article, we present an algorithm for document image segmentation into text paragraphs. We built a complete processing workflow upon basic image processing tools and well know segmentation algorithms, namely X-Y cutNagy and Seth and maximal white-space rectangle extraction Breuel. The complete workflow and the parametrization of the different algorithm is guided by the analysis of the image or image region content. No parameters are required, either as user input or as estimated on a subset of images. Finally, we propose a novel paragraph detection algorithm based on the recognition of different types of paragraph layouts. We illustrate the performances of the algorithm on the ICDAR2011 Historical Document Layout Analysis dataset.*

MOTS-CLÉS : *Image de Documents, Segmentation Automatique, Reconnaissance, Documents Historiques*

KEYWORDS: *Document Image, Parameter-free Segmentation, Recognition, Historical Documents*

1. Introduction

La segmentation est un sujet bien étudié dans la littérature du traitement d'image en général Haralick and Shapiro; Freixenet, Munoz and et. al., et du traitement des images de documents en particulier Nagy; Mao, Rosenfeld and Kanungo. L'analyse logique d'un document, c'est à dire la représentation du document sous une forme structurée, avec des informations de haut niveau (par exemple la labelisation des différentes parties en entités connues : titres, illustrations, numéros de page, ...) est préconditionnée par les résultats de la segmentation Trupin. Plusieurs travaux font état d'excellentes performances dans la classification de zones de documents Wang, Phillips and Haralick; Keysers, Shafait and Breuel, en se basant sur les caractéristiques extraites sur chaque zone, segmentées manuellement. Cependant, ces caractéristiques ne peuvent être extraites de manière robuste que si les frontières entre les zones sont bien déterminées, c'est à dire après une bonne segmentation.

Shafait *et. al.* Shafait, Keysers and Breuel ont comparé les performances des algorithmes de segmentation les plus connus de la littérature, à savoir X-Y Cut Nagy and Seth, RLSA Wong, Casey and Wahl, Whitespace Analysis Baird, Constrained Text-Line DetectionBreuel, Docstrum O'Gorman et Voronoi Kise, Sato and Iwata. Bien que différents dans leur approche du problème de la segmentation, tous ces algorithmes nécessitent le réglage de paramètres conditionnant les performances de la segmentation. Cependant, ces travaux ne proposent pas de stratégie pour l'estimation de ces paramètres. Dans leur étude, Shafait *et. al.* ont utilisé une sous partie de la base d'image University of Washington III Guyon, Haralick, J. and Phillips pour affiner les réglages, et testé les performances sur le reste de la base. Cette stratégie reste pertinente dans le cas du traitement d'une base d'image relativement homogène comme UW III. Dans le cas où l'ensemble d'images à traiter varie considérablement, ne serait ce que dans les conditions d'acquisition de l'image numérique (différentes résolutions de scan), un seul réglage de paramètres ne saurait être satisfaisant. Cette constatation est aussi valide au sein d'une même image, où les blocs de texte peuvent êtres imprimés avec des fontes et des tailles de police très différentes. Ainsi, il nous semble qu'une stratégie intelligente pour l'estimation des paramètres des algorithmes de segmentation doit se fonder sur l'analyse du document en cours de segmentation lui même, voir même s'adapter localement à la zone en cours de segmentation.

La segmentation reste un problème mal posé : dans les images de documents, le niveau de granularité souhaité dépend de l'application visée. Par exemple, la reconnaissance de caractère se basera en premier sur l'extraction des caractères. Mais un algorithme de reconnaissance performant utilisera aussi des dictionnaires pour corriger les erreurs ou les ambiguïtés sur les caractères isolés. Il est donc nécessaire d'isoler le mot en entier, voir la phrase ou le paragraphe si l'on souhaite tenir compte d'un contexte sémantique. D'autre tâches visant à une description logique du document se basent sur les paragraphes de texte comme unité physique de base : la détermination de l'ordre de lecture, l'extraction de l'organisation hiérarchique du document en chapitres, section et paragraphes Trupin. La segmentation en paragraphes de texte fut la granularité adoptée dans le cadre de la compétition de la conférence ICDAR Antona-

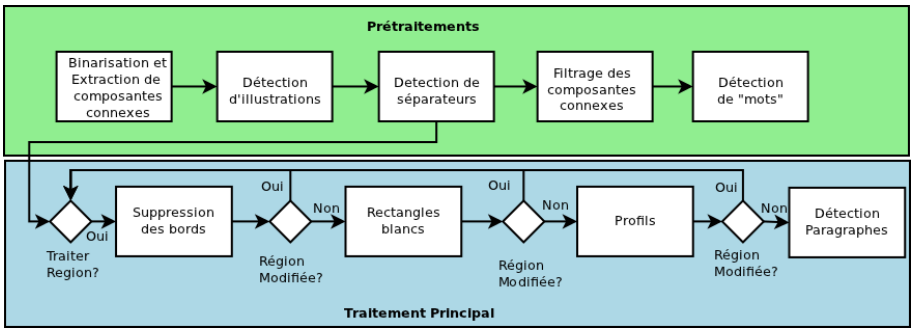

Figure 1. *Vue d'ensemble de notre algorithme de segmentation*

copoulos, Pletschader, Bridson and Papadopoulos. C'est aussi l'objectif principal de l'algorithme de segmentation que nous proposons.

2. Approche proposée et contributions

Dans ce travail, nous proposons trois contributions à la tâche de segmentation en paragraphe. La première, que nous illustrons dans la figure 1, est la définition de la chaîne de traitement permettant de réaliser cet objectif. Les deux blocs colorés de la figure 1 montrent les deux principales étapes de l'algorithme, à savoir un ensemble de pré-traitements sur l'image globale, puis le traitement principal appliqué récursivement sur des zones de l'image de plus en plus fines, comme le montrent les boucles dans le traitement principal. Ainsi, notre algorithme effectue un découpage *descendant* de l'image, en s'attachant à des zones de plus en plus petites, jusqu'à l'étape finale de détection des paragraphes. Cependant, le paramétrage des traitements dans chacune des étapes se fait à l'aide de l'analyse d'informations de bas niveau. Nous proposons donc une approche hybride du découpage de l'image, qui est descendante dans le déroulement de l'algorithme, mais se base sur des informations ascendantes.

Dans la boucle de traitement principal, nous nous appuyons les algorithmes de Whitespace Analysis (Rectangles Blancs) et X-Y cut (Profils). L'objet de notre seconde contribution consiste à présenter une version modifiée de ces algorithmes, en rectifiant leur objectif principal et surtout en proposant un réglage automatique de leurs paramètres, par l'analyse du contenu de la zone en cours de segmentation. Ainsi, l'ensemble de la chaîne de traitement est pilotée par ce réglage automatique de paramètres. Aucun paramètre de notre algorithme ne doit être explicitement ajusté par l'utilisateur pour traiter une image.

Enfin, notre troisième contribution va nous permettre d'obtenir la granularité finale de notre segmentation, c'est à dire le découpage en paragraphes. Ici, nous proposons un système non pas basé sur une approche segmentation comme dans les étapes pré-

cédentes, mais sur une approche de reconnaissance. Segmenter pour reconnaitre, ou reconnaitre pour segmenter reste une question ouverte dans la recherche. Nous optons dans ce cas pour la deuxième solution.

Nous présentons les détails de notre algorithme en adoptant le plan suivant : dans la section 3, nous détaillons l'ensemble des pré-traitements correspondant au premier bloc de la figure 1. Nous décrivons la chaîne de traitement principal dans la section 4, où nous aborderons la modification et le paramétrage des algorithmes existants, ainsi que notre module de détection de paragraphes. Dans la section 5, nous évaluerons les performances de notre algorithme sur la base d'image utilisée dans la compétition ICDAR 2011 Historical Document Layout Analysis Competition, avant de conclure dans la section 6.

3. Pré-traitements

L'ensemble des pré-traitements que nous proposons ici à pour but de faciliter l'étape d'estimation automatique des paramètres à appliquer dans les algorithmes que nous présenterons dans la section suivante. D'autres pré-traitements, comme le redressement ou l'amélioration de la qualité de l'image avant binarisation sont envisageables en utilisant des méthodes connues de la littérature. Nous ne nous sommes cependant pas attaché a développer ces méthodes à l'heure actuelle, notre but étant de fournir une chaîne complète de segmentation. Nous illustrons les principales étapes de l'algorithme sur un exemple dans la figure 3.

3.1. *Binarisation*

Pour la binarisation, nous utilisons simplement la méthode de binarisation globale d'Ostu Otsu. Cet algorithme a prouvé a maintes reprises son efficacité et cadre parfaitement avec notre approche *sans paramètres*. Nous noterons I_b l'image binaire, où $I_b(x,y) = 0$ si le pixel (x,y) est un pixel du fond, $I_b(x,y) = 1$ sinon.

3.2. *Détection d'illustrations*

Nous proposons une détection précoce des illustrations par classification SVM de différentes zones de l'image. Nous publierons prochainement les détails de cette approche. Les zones contenant potentiellement des illustrations sont segmentées de la manière suivante : On considère l'ensemble $\mathcal{CC}$ des composantes connexes des pixels de l'encre calculé à partir de l'image binaire I_b. Nous calculons l'enveloppe convexe de chaque composante connexe notée $ch_i, \forall cc_i \in \mathcal{CC}$. Soit l'image binaire I'_b définie par $I'(x,y) = 1 \text{si}(x,y) \in ch_i \forall i, I'(x,y) = 0$ sinon. Les composantes connexes des pixels non nuls de I'_b forment un ensemble de régions disjointes, noté $\mathcal{CC}'$, qui correspondent à l'union des enveloppes convexes des composantes connexes de I_b. Nous soumettons au classifieur les zones $cc'_j \in \mathcal{CC}'$ dans lesquelles on trouve plusieurs composantes

connexes de I_b. En effet les composantes connexes des lettres du texte sont la plupart du temps bien séparées et leurs enveloppes convexes ne se chevauchent pas. Au contraire, les composantes connexes des illustrations sont souvent imbriquées les unes dans les autres, et leurs enveloppes convexes se chevauchent fortement. Ainsi, nous segmentons rapidement un ensemble de régions dans l'image, que nous soumettons au classifieur afin de déterminer si ces régions sont vraiment des illustrations. Ce zonage permet une pré-segmentation rapide des parties de l'image correspondants aux illustrations.

3.3. *Filtrage des composantes connexes*

Dans cette étape, nous filtrons les *petites* composantes connexes, qui sont dues aux nombreux artefacts affectant la qualité de la numérisation (transparence, tâches, grain du papier, ...). Nous estimons un seuil global sur la taille minimale des composantes connexes par l'équation 1.

$$T_{\text{taille}} = 2 * \sqrt{\sigma_{\mathbf{a}}} \quad [1]$$

$\sigma_{\mathbf{a}}$ est l'écart type de l'aire des composantes connexes. La détection précoce des illustrations, qui sont souvent constituées de composantes connexes de grand surface, nous permet de ne pas tenir compte de ces composantes lors de l'évaluation de T_{taille}. Dans le cas contraire, le seuil aurait tendance à être surévalué selon nos hypothèses, et des composantes connexes pertinentes pourraient être effacées.

3.4. *Détection des mots*

Dans cette partie, nous groupons les composantes connexes en entités plus larges, que nous appelons (par abus de langage) *mots*. Nous montrons dans la figure 3(B) à gauche l'histogramme des distances horizontales entre les composantes connexes. Cet histogramme possède 2 pics distincts : le premier pic correspond à la distance inter lettre dans le document, tandis que le second, plus étalé, donne une estimation de la distance inter mots. Nous détectons automatiquement la vallée entre les 2 pics, et groupons les composantes connexes qui ont une distance inférieure au seuil ainsi détecté. La figure 3(B) à droite montre les composantes avant et après leur regroupement, sur une partie zoomé du document de la figure 3 (A). Dans cet exemple, nous voyons que le seuil détecté permet bien d'extraire les mots. Cependant, certains signes de ponctuation peuvent être agrégés au *mot*, et le seuil détecté globalement ne permet pas forcément d'extraire tous les mots de manière correcte (espacement inter lettres différents pour les titres, par exemple). C'est pourquoi nous considérons comme abus de langage le fait de parler de *mots* dans ce cas. Nous utiliserons ces *mots* pour l'estimation des paramètres dans les algorithmes des étapes suivantes.

3.5. *Détection des séparateurs noirs*

Nous appelons *séparateurs noirs* les traits, à l'encre, figurant sur le document dans le but de séparer différentes zones de texte (voir figure 3 (A) gauche, en rouge). Une composante connexe *cc* est considérée comme un séparateur si elle satisfait l'équation 2 :

$$\frac{\max(\mathbf{w}(cc),\mathbf{h}(cc))}{\min(\mathbf{w}(cc),\mathbf{h}(cc))} \geq T_{\text{sep}} \qquad [2]$$

w et **h** représentent respectivement la largeur et la hauteur de cc, T_{sep} est un seuil fixé à 10. Cette approche nous permet de détecter les traits horizontaux ou verticaux. Le fait de fixer une valeur seuil n'est pas en contradiction avec notre objectif de ne pas utiliser des paramétrages par l'utilisateur. En effet, l'équation 2 calcule l'élongation de la composante connexe, qui est une quantité indépendante de la résolution d'image. Nous l'avons fixé à 10 pour être sûr de ne pas considérer comme séparateur des composantes fortement allongées comme la lettre l. Une fois les séparateurs noirs détectés, nous faisons grandir au maximum la zone contenant le séparateur jusqu'à ce qu'elle touche d'autres composantes connexes ou les bords de l'image. Ainsi, comme illustré dans la figure 3 (A) droite, nous parvenons à isoler 3 régions dans l'image : la région supérieure, qui contient un en-tête avec le numéro de section un titre et un numéro de page, la région du milieu, avec les 3 paragraphes de texte, et la région inférieure, qui contient la note de bas de page. On note qu'on crée un trou dans la région du milieu, en étendant le tiret près du mot *Campbell*, qui a été détecté comme un séparateur.

4. Traitement principal

Comme illustré sur la figure 1, le traitement principal consiste en l'application récursive sur des zones de l'image de 3 traitements élémentaires, à savoir la suppression des bords, la recherche des rectangles blancs maximaux et l'examen des profils de projection. Une zone de l'image passée par ces 3 blocs sans avoir été modifiée est alors soumise au dernier module de détection de paragraphes. L'ensemble du traitement est basé sur une notion algorithmique simple : une pile de région est initialisée avec l'image originale entière, ou l'ensemble des régions obtenues après l'extension des séparateurs noirs(cf figure 3 (A)). La première région est dépilée. Si la région contient des composantes connexes, alors elle entre dans la succession de traitements. Sinon, il s'agit d'une région de fond. Si l'un des 3 traitements définis ci-dessus modifie la région, celle-ci ainsi que l'ensemble des régions éventuellement créées sont insérées dans la pile. L'algorithme continue de dérouler ainsi jusqu'à ce que la pile soit vide. Nous allons maintenant examiner chacun des traitements en détail.

4.1. *Suppression des bords*

Ce premier traitement consiste simplement à réduire la zone à traiter à la boite englobante des composantes connexes de la région considérée, et ainsi supprimer des parties de fond.

4.2. *Recherche des rectangles blancs maximaux*

La recherche des rectangles blancs maximaux est un algorithme classique pour l'analyse de l'agencement physique des documents numérisés. Ce problème a été formalisé par Breuel de la manière suivante Breuel : Soit un ensemble de rectangles $\mathcal{R} = \{r_1, \ldots, r_n\}$ contenus dans le plan. Cet ensemble est borné par un rectangle englobant r_b. Soit une fonction d'évaluation des rectangles $Q : \mathbb{R}^4 \rightarrow \mathbb{R}$ satisfaisant l'équation suivante :

$$r \subseteq r' \Rightarrow Q(r) \leq Q(r') \forall r, r' \quad [3]$$

Le rectangle blanc *maximal*, $\hat{r}$ est le rectangle qui maximise la fonction d'évaluation $Q(R)$ parmi l'ensemble des rectangles inclus dans r_b qui ne chevauchent aucun des *obstacles* $\mathcal{R}$:

$$\hat{r} = \hat{r}(\mathcal{R}, r_b, Q) = \arg\max_{r \in U} Q(r) \quad \text{where } U = \{r \subseteq r_b | \forall r_o \in \mathcal{R}, r \cap r_o = \emptyset\} \quad [4]$$

Dans le problème de l'analyse de la mise en page de documents, on considère $\mathcal{R}$ comme l'ensemble des boites englobantes des composantes connexes de l'encre, r_b les frontières de l'image numérisée, et Q l'aire des rectangles. Breuel a introduit un algorithme efficace pour la recherche des rectangles maximaux dans le cas de rectangles non orientés Breuel (c'est à dire dans les cotés sont parallèles aux bords de l'image), puis dans le cas des rectangles orientés Breuel. Les rectangles blancs maximaux aident à l'analyse de la mise en page du document en mettant en évidence, par exemple, la séparation du texte en colonnes. On peut jouer sur la fonction d'évaluation Q pour favoriser certains types de rectangles en fonction de leur aspect (les rectangles très allongés verticalement ont plus de chance de séparer des colonnes).

Un des problèmes dans la recherche des rectangles blancs maximaux est de savoir à quel moment s'arrêter. Traditionnellement, on cherche un nombre fixe de rectangles, en se basant sur une connaissance a priori du document à traiter. Trouver la couverture complète de l'image par des rectangles blancs maximaux aboutit, au final, a couvrir l'ensemble du fond de l'image (en excluant les trous à l'intérieur des composantes connexes). Nous considérons ici la recherche des rectangles blancs maximaux non pas comme un outil d'analyse de la mise en page directement, mais comme un moyen efficace pour trouver des sous parties de la région qui ne contiennent aucune composante connexe, et ainsi pouvoir les exclure du domaine de la région.

La couverture par les rectangles blancs maximaux, dessinés en bleu sur la figure 3(C) à droite, nous permet à la fois d'ajuster plus finement les contours de la région au

texte, mais aussi de couper la région originale en 3 blocs disjoints, qui vont entrer dans la pile des régions à traiter. La difficulté dans cette tâche consiste donc à faire la distinction entre des rectangles pertinents, c'est à dire qui entrent bien dans l'objectif fixé, des rectangles non pertinents, c'est à dire ceux qui viendraient couper *artificiellement* une zone en deux, par exemple en passant entre deux lignes d'un même paragraphe. L'approche que nous proposons consiste à estimer la longueur et largeur minimale d'un rectangle *pertinent* respectivement notées T_{width} et T_{height}. Nous utilisons l'algorithme de Breuel Breuel pour détecter les rectangles non orientés. Cet algorithme renvoie une liste de rectangles ordonnées selon la fonctionnelle $Q(r) = \mathbf{w}(r) \times \mathbf{h}(r)$. Nous arrêtons donc l'algorithme lorsqu'un rectangle renvoyé a une aire inférieur à $T_{\text{width}} \times T_{\text{height}}$. Nous estimons T_{width} et T_{height} à partir des observations calculées sur les espacements des composantes connexes et des mots appartenant à la région en cours de traitement :

$$\begin{array}{rcl} T_{\text{width}} & = & \arg\max H(d^h(w_i, w_j)) + \bar{\mathbf{w}}_{cc} \\ T_{\text{height}} & = & \arg\max H(d^v(cc_i, cc_j)) + \bar{\mathbf{h}}_{cc} \end{array} \quad [5]$$

On note $d^h(w_i, w_j)$ la distance horizontale séparant 2 mots distincts w_i, w_j de la région. On note de même, $d^h(cc_i, cc_j)$ les distances verticales entre composantes connexes. $H(x)$ est l'histogramme des observations considérées et $\arg\max H(x)$ est donc la valeur pour laquelle la fréquence observée dans l'histogramme est maximale. $\bar{\mathbf{w}}$ et $\bar{\mathbf{h}}$ sont respectivement la largeur et la hauteur moyenne des composantes connexes du bloc. Le choix de ces seuils est illustré dans la figure 3(C) gauche. L'histogramme supérieur montre la distribution des espacements horizontaux entre les mots du bloc. L'espacement inter mots n'est pas une mesure stable, dans la mesure où il varie beaucoup pour des problèmes évident de mise en page. Cependant, on estime sur l'histogramme l'espacement le plus fréquent, auquel on ajoute la largeur moyenne des composantes connexes, afin d'éviter de faire passer des rectangles entre des mots d'une même ligne. L'histogramme inférieur montre la distribution des espacements verticaux entre les composantes connexes à l'intérieur du bloc. Cet histogramme est beaucoup plus caractéristique, avec les 3 pics que l'on voit apparaître en début d'histogramme. Le premier pic correspond à des distances très faibles, par exemple entre des lettres basses d'une ligne et des lettres hautes de la suivante, ou entre les accents et la lettre accentuée. Le deuxième pic correspond à des distances un peu plus importante, entre des composantes allongées et des composantes *normales*. Le troisième pic correspond aux distances entre composantes connexes *normales*. Nous conjecturons que ce troisième cas est le plus fréquent, et donc que le troisième pic est celui de plus haute fréquence dans l'histogramme. Cette quantité correspond donc à une estimation de l'espace interligne à l'intérieur du bloc. Afin de ne pas faire passer des rectangles blancs entre les lignes, nous ajoutons à cette quantité la hauteur moyenne des composantes du bloc.

4.3. *Profils de projection*

L'examen des profils de projection va nous permettre de séparer des blocs de texte qui n'auraient pu être séparés lors des étapes précédentes. La technique de découpage

des images par les profils de projection a été proposée par Nagy Nagy and Seth en tant que méthode de segmentation descendante. Les profils de projection sur l'axe horizontal (respectivement sur l'axe vertical) sont obtenus par l'intégrale de l'image binaire I_b projetée sur l'axe horizontal(respectivement vertical) :

$$\begin{array}{l} P_h(i) = \iint_{i=0,j=0}^{i<width,j<height} I_b(i,j) \\ P_v(j) = \iint_{i=0,j=0}^{i<width,j<height} I_b(i,j) \end{array} \quad [6]$$

Dans le cas d'une image bien orientée, avec des lignes relativement espacée, on peut voir sur le profil vertical un ensemble de pics correspondant aux lignes de texte. Pour un document multi-colonne, une plage de blanc doit apparaître au niveau de l'espace inter-colonnes sur le profil de projection horizontal. L'approche de Nagy consiste alors à seuiller le profil, et à couper en deux les plages de blanc des profils seuillés qui ont une taille supérieure à un paramètre fixé. Le principal problème de cette approche est sa sensibilité à l'orientation du document, et son incapacité à traiter des mises en pages complexes, dans lesquelles les différents blocs de texte ne sont pas parfaitement alignés. On estime que dans le cours de notre algorithme, la suppression des bords et la recherche des rectangles blancs maximaux nous permet déjà d'isoler des zones de texte avec une mise en page simplifiée. L'avantage de la méthode des profils par rapport aux rectangles blancs maximaux est que l'algorithme est plus tolérant vis à vis des éventuelles petites tâches dues à une binarisation incorrecte. Dans le cas des rectangles blancs, ces tâches sont obligatoirement considérées comme des obstacles. Avec les profils, l'influence de ces tâches peut être occultée dès l'étape de seuillage. Contrairement à l'algorithme de Nagy, nous ne coupons pas directement les *grandes* plages de blanc dans les profils, mais nous appliquons des opérateurs morphologique d'ouverture et de fermeture sur les profils, comme illustré sur la figure 3(D). Le profil de projection du bloc central de la figure 3(C) droite sur l'axe vertical est montré dans la figure 3(D) à gauche. Dans la figure 3(D) à droite, nous montrons les différentes étapes de traitement sur ce profil : en rouge, le seuillage, en vert, l'ouverture et en bleu, la fermeture du profil. L'ouverture nous permet de conserver les projections correspondant aux lignes, plus larges que la taille de l'élément structurant, tout en éliminant les éventuelles parties non pertinentes de profil qui auraient toutefois passé l'étape de seuillage. La fermeture avec un élément structurant de taille légèrement supérieure à l'espace interligne permet de réunir les lignes appartenant à un même paragraphe, tout en séparant celles qui sont trop éloignées. Les deux premières lignes du bloc sont effectivement séparées du reste dans la figure 3(C).

La taille des éléments structurants utilisés pour l'ouverture et la fermeture sur le profil vertical et sur le profil horizontal, respectivement notés $T_v^\gamma, T_v^\phi, T_h^\gamma, T_h^\phi$ sont estimés par :

$$\begin{array}{rcl} T_v^\gamma & = & \frac{1}{2} \times \bar{\mathbf{h}}_{cc} \\ T_v^\phi & = & 1.1 \times \arg\max H(d^v(cc_i, cc_j)) \\ T_h^\gamma & = & \frac{1}{2} \times \bar{\mathbf{w}}_{cc} \\ T_h^\phi & = & \arg\max H(d^h(w_i, w_j)) \end{array} \quad [7]$$

4.4. *Détection de paragraphes*

Lorsqu'une zone est soumise au module de détection de paragraphe, on estime qu'il s'agit d'une zone relativement uniforme, contenant un ou plusieurs paragraphes de texte, sur une colonne. Ce module analyse le contenu du texte à l'intérieur de la région afin de déterminer la présence des différents paragraphes de texte le cas échéant. Pour cela, nous extrayons tout d'abord les lignes de texte à l'intérieur du bloc, puis nous détectons certains types de paragraphes à partir de critères sur les alignements de lignes. Nous allons détailler ces deux étapes.

4.4.1. *Détection de lignes*

Nous proposons ici une approche originale basée sur l'algorithme de détermination des classes d'équivalence Press, Teukolsky, Vetterling and Flannery. Cet algorithme générique permet de grouper un ensemble de données en sous ensembles, ou clusters. On le paramètre à l'aide d'un prédicat $P(i,j)$ qui indique si deux objets i et j doivent être rangés dans le même cluster. $P(i,j) = 1$ si et seulement si i et j sont dans la même classe, $= 0$ si non ou indéterminé. Ici les objets à grouper sont les composantes connexes à l'intérieur du bloc. Les groupes recherchés sont les lignes de texte. Deux composantes connexes sont dans une même ligne si elles satisfont le prédicat suivant :

$$P(cc_i, cc_j) = 1 \iff \begin{array}{ll} \frac{\max(h(cc_i),h(cc_j))}{\min(h(cc_i),h(cc_j))} < T_{ratio1} & \wedge \\ \frac{\min(h(cc_i),h(cc_j))}{v_{overlap}(cc_i,cc_j)} < T_{ratio2} & \wedge \\ d^h(cc_i, cc_j) < T_{dist} & \end{array} \quad [8]$$

$v_{overlap}$ est la taille du recouvrement sur l'axe vertical des composantes connexes. Le prédicat est décomposé en 3 conditions. La première s'assure que les composantes connexes ont un aspect ratio similaire en hauteur, pour prévenir d'intégrer dans une même ligne une composante connexe qui pourrait s'étaler sur plusieurs lignes (à cause d'une tâche, d'une binarisation incorrecte ou de la proximité de deux lettres hautes et basses sur deux lignes). La deuxième condition s'assure que les composantes connexes soient bien alignés, c'est à dire que leur recouvrement vertical doit être significatif. La troisième condition s'assure que l'on ne va pas comparer le recouvrement de composantes trop éloignées l'une de l'autre. En effet, si le document n'est pas droit, deux composantes éloignées peuvent se chevaucher selon leur projections verticales, sans appartenir à la même ligne. Nous rappelons que nous n'avons pas corrigé les défauts d'orientation. Les paramètres T_{ratio1} et T_{ratio2} sont fixés. Là encore, il s'agit de ratio relatifs, indépendant de la résolution de l'image. Le paramètre T_{dist} est quant à lui fixé à $6 \times \bar{\mathbf{w}}_{cc}$

L'algorithme de détermination des classes d'équivalence ainsi défini nous donne une forte sur-segmentation en morceaux de lignes : par exemple, les accents ou signes de ponctuation se groupent entre eux, mais dans des groupes séparés des lettres car ne respectant pas les contraintes de taille relative. Nous fusionnons les différents morceaux qui se recouvrent ou s'alignent afin d'extraire l'ensemble de la ligne.

Type de paragraphe	Prédicat (condition = 1)
Justifié	$\begin{array}{c} \lvert i-j\rvert = 1 \wedge \\ \lvert x_{\min}(i) - x_{\min}(j)\rvert < T_{\text{align}} \wedge \\ \lvert x_{\max}(i) - x_{\max}(j)\rvert < T_{\text{align}} \end{array}$
Alterné	$\begin{array}{c} j - i = 1 \wedge \\ !\text{aligné a gauche}(i,j) \wedge \\ (\text{aligné a gauche}(i,i-2) \wedge \text{aligné a gauche}(j,j-2)) \vee \\ (\text{aligné a gauche}(i,i+2) \wedge \text{aligné a gauche}(j,j+2)) \vee \\ (\text{aligné a gauche}(i,i-2) \wedge \text{aligné a gauche}(j,j-2)) \end{array}$
Aligné à gauche	$\begin{array}{c} \lvert i-j\rvert = 1 \wedge \\ \lvert x_{\min}(i) - x_{\min}(j)\rvert < T_{\text{align}} \end{array}$
Aligné à droite	$\begin{array}{c} \lvert i-j\rvert = 1 \wedge \\ \lvert x_{\max}(i) - x_{\max}(j)\rvert < T_{\text{align}} \end{array}$
Centré	$\begin{array}{c} \lvert i-j\rvert = 1 \wedge \\ \lvert \frac{\lvert x_{\min}(i) - x_{\max}(i)\rvert}{2} - \frac{\lvert x_{\min}(i) - x_{\max}(j)\rvert}{2} \rvert < T_{\text{align}} \end{array}$

Tableau 1. *Prédicats pour la détection de paragraphes*

4.4.2. *Détection de paragraphe*

La détection de paragraphe est aussi basée sur l'algorithme de détermination des classes d'équivalence. Cependant, nous nous plaçons ici à une échelle supérieure dans notre analyse, où les éléments à grouper sont les lignes de texte, et les classes d'équivalence correspondent aux paragraphes. La stratégie à appliquer consiste à définir des prédicats en fonction de chaque type de paragraphe recherché. Nous définissons ensuite une hiérarchie dans les types de paragraphe, et nous effectuons la recherche en suivant cette hiérarchie. Si un ensemble de ligne est détecté comme appartenant à un paragraphe, ces lignes sont supprimées de l'ensemble d'objets considéré avant d'être analysé par le détecteur suivant dans la hiérarchie. L'ensemble des lignes d'un bloc sont numérotées suivant leur ordre de la plus haute vers la plus basse. Dans l'ordre, nous recherchons des paragraphes *justifiés*, *alternés*, *alignes à gauche*, *alignés à droite* et *centré*. Les prédicats pour chacun des types de paragraphe sont illustrés dans la table 1.

$x_{\min}$ et $x_{\max}$ sont respectivement les coordonnées du début et de la fin de la ligne. La sensibilité des détecteurs est fonction du seuil T_{align}, qui est l'intervalle de tolérance pour lequel nous considérons que deux lignes successives sont alignées. Ce seuil varie en fonction du contenu du bloc dans lequel nous effectuons la recherche. Ainsi, nous fixons $T_{\text{align}} = \bar{w}_{cc}$, soit la largeur moyenne d'une composante connexe dans le bloc.

Le type de paragraphe et la hiérarchie que nous avons défini entre les différents module de détection de paragraphe est liée au contenu du jeu de données sur lequel nous évaluerons les performances, ce qui peut être vue comme une contradiction par rapport à notre objectif de s'affranchir du réglage *en dur* des paramètres de l'algorithme. Cependant, l'approche générale de la détection de paragraphe est totalement

modulaire : pour traiter des documents dans des langues où la lecture se fait de droite à gauche, on pourra par exemple choisir de détecter les paragraphes alignés à droite avant ceux alignés a gauche. D'autre part, on peut facilement étendre la détection à d'autres types de paragraphes, en choisissant des prédicats appropriés.

Ainsi, après la détection, nous sommes non seulement capables de segmenter les différents paragraphes, mais nous possédons aussi des informations sur l'agencement et le type de chaque paragraphe. Dans la figure 3, nous savons par exemple que nous avons un paragraphe de type *alterné* de 10 lignes en haut, et deux paragraphes de type *alignés a gauche*. Nous avons aussi un certains nombre de lignes isolées, intercalées entre ces paragraphes. Ces informations pourront se révéler très utiles, pour l'analyse logique du contenu du document, mais aussi pour corriger les erreurs de segmentation dans une étape de post-traitement qui reste à élaborer.

La limite principale de cette approche est que les prédicats que nous définissons se basent sur les coordonnées absolues des extrémités des lignes. Hors, certains éléments typographiques, comme les lettrines ou les illustrations, peuvent entraîner un décalage des lignes adjacentes qui sera détecté comme un changement de paragraphe.

5. Expérimentations

Nous évaluons notre algorithme sur une base d'image de documents historiques proposée lors de la compétition *ICDAR 2011 Historical Document Layout Analysis Competition*. Cette base contient 100 images de résolutions et qualité très hétérogènes. Ainsi, les plus petites images font environ 500x800 pixels, jusqu'à 5000x6000 pour les plus grandes. Certaines de ces images sont déjà binarisées. La majorité des images sont en couleurs ou en niveau de gris. Aucun autre pré-traitement ne semble avoir été effectué sur ces images (redressement, filtrage, ...). La vérité terrain n'ayant pas été donnée par les organisateurs d'ICDAR 2011, nous avons manuellement annoté l'ensemble des images en paragraphe de texte, par leurs boites englobantes[1]. Nous avons aussi extrait les lettrines et différentes illustrations le cas échéant. Ce travail a été fait par une personne n'ayant pas de connaissances en traitement d'images et a fortiori n'étant pas impliqué dans le développement de ces travaux.

Nous évaluons la qualité de la segmentation en calculant le F-Score Rijsbergen, au niveau pixel, entre les résultats de segmentation automatique et notre vérité terrain. Le calcul du F-Score se fait par rapport au zones de texte seulement. Les résultats sont présentés dans la table 2. Nous obtenons un score de 0.931 en terme de F-Score global, c'est à dire en accumulant les scores de pixels bien et mal classés sur l'ensemble des pixels de la base. Le F-Score moyen, c'est à dire la moyenne des F-Scores individuels obtenus sur chaque image est de 0.924. Cela confirme les bonnes performances de notre algorithme non seulement sur la base entière, mais aussi sur chacune des images. Nous rapellons que nous avons segmenté la base de manière entièrement automatique, les paramètres étant estimés comme indiqué dans les sections précédentes.

1. Vérité terrain disponible auprès des auteurs

F-score(global)	0.931
F-score(average)	0.924

Tableau 2. *Résultats de segmentation sur la base ICDAR 2011 Historical Document Layout Analysis Competition*

Nous montrons quelques exemples de segmentation dans la figure 2. Notez que nous n'avons pas nettoyé les frontières des régions, en considérant par exemple la boite englobante de chaque région, ou en ajustant les frontières aux composantes connexes. Les principaux défauts de notre algorithme sont mis en évidence sur la figure 2 au centre. Le texte dans la marge n'a pas été bien séparé du reste, et la détection de paragraphe considère donc des lignes qui ne respectent pas la mise en page justifiée de l'ensemble du paragraphe, ce qui explique les coupures au milieu. Dans l'image de droite, le dernier mot du texte du bas de page est séparé du texte. Ceci est due à une mauvaise estimation des paramètres, en particulier sur des blocs comme celui-ci, avec peu de composantes connexes, sur lesquels les statistiques estimées sont moins robustes.

6. Conclusion

Dans cet article, nous avons proposé un algorithme complet de segmentation d'image de documents en paragraphes. Notre contribution se base sur des algorithmes bien connus de la littérature, comme la recherche de rectangles blanc maximaux et l'examen des profils de projection. L'originalité de notre contribution consiste en la définition d'une chaîne de traitement, que nous appliquons de manière récursive sur l'image et les régions, jusqu'à l'obtention de la granularité finale de segmentation. Le pilotage de la chaîne de traitement, ainsi que le paramétrage des algorithmes, est entièrement défini par l'analyse du contenu de l'image et/ou des régions en cours de traitement. Ceci permet à la fois un paramétrage fin des algorithmes à l'intérieur de chaque bloc, mais aussi de s'affranchir totalement de l'aide d'un opérateur pour le paramétrage. Nous avons proposé un système de détection des lignes de texte, ainsi que des paragraphes, en nous appuyant sur l'algorithme de détermination des classes d'équivalence. Notre approche permet de définir un cadre simple et évolutif pour la détection de différents types de paragraphes, grâce à la définition de prédicats sur les classes d'équivalence. Nous avons proposé des détecteurs correspondants à plusieurs types courants de mise en page de paragraphes. Ainsi, notre approche permet non seulement de détecter les frontières des paragraphes de texte, mais aussi de les classer en fonction de leurs types, ce qui sera utile pour des tâches de plus haut niveau dans l'analyse du contenu du document.

Nous avons montré que nous pouvons segmenter entièrement une base de données complexe telle que celle utilisée pour la compétition ICDAR 2011 Historical Document Layout Analysis, sans qu'aucun paramètre de segmentation ne soit estimé, soit

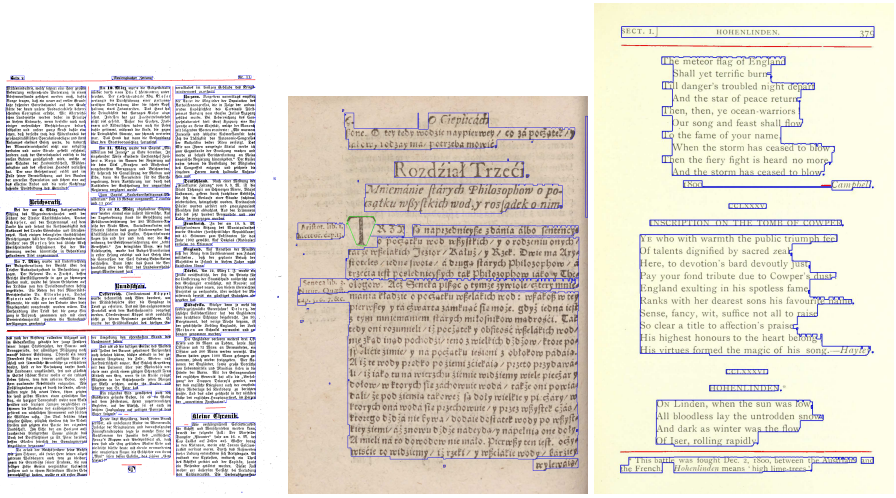

Figure 2. *Exemple de résultats de segmentation.*

par un utilisateur, soit de manière automatique sur une base de validation. Les résultats de segmentation, en terme de F-Score, sont tout a fait encourageant.

Les pistes d'amélioration de la chaîne de traitement sont nombreuses. Tout d'abord, nous pouvons bien entendu améliorer la performance de chacun des modules séparément. En particulier, certains pré-traitements comme le redressement de l'image, au niveau global ou au niveau de chaque bloc, et le filtrage en vue d'une amélioration de la binarisation apporteront certainement un gain sur les performances globales de la chaîne. La détection de séparateurs, à l'heure actuelle, ne permet de considérer que des lignes horizontales ou verticales. La détection de lignes dans n'importe quelle direction, mais aussi d'autres types de séparateurs tels que les encadrements et les bordures, qui est un sujet bien étudié dans la littérature, nous permettra de mieux isoler les différentes parties de texte. Nous prévoyons aussi d'étendre la capacité de l'algorithme à traiter de nombreux types de documents, comme des magazines en couleurs aux mises en pages complexes.

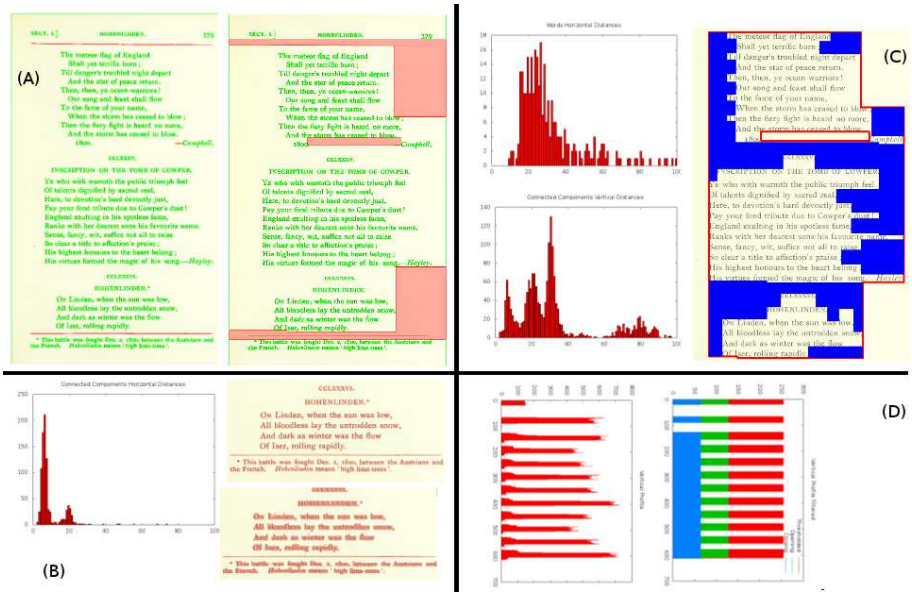

Figure 3. *Illustration des principales étapes de l'algorithme de segmentation. (A) Détection et extension des séparateurs noirs. (B) Groupement des CC en* mots. *(C) Rectangles blancs maximaux. (D) Profils de projection.*

References

Antonacopoulos A., Pletschader S., Bridson D., Papadopoulos C., « ICDAR 2009 Page Segmentation Competition », *ICDAR'09*, 2009.

Baird H., « Background structure in document images. », *Document Image Analysis*, 1994.

Breuel T., « Two Geometric Algorithms for Layout Analysis », *DAS'2002*, 2002.

Breuel T., « High Performance Document Layout Analysis », *Symposium on Document Image Understanding Technology*, 2003.

Freixenet J., Munoz X., et. al. D. R., « Yet Another Survey on Image Segmentation : Region and Boundary Information Integration », *ECCV'02*, 2002.

Guyon I., Haralick R., J. J. H. J., Phillips I., *Handbook of character recognition and document image analysis*, World Scientific, chapter Data sets for OCR and document image understanding research, p. 779-799, 1997.

Haralick R., Shapiro L., « Image segmentation techniques », *Computer Vision, Graphics, and Image Processing*, vol. 29, p. 100-132, 1985.

Keysers D., Shafait F., Breuel T., « Document image zone classification - a simple high-performance approach », *2nd Int. Conf. on Computer Vision Theory and Applications*, 2007.

Kise K., Sato A., Iwata M., « Segmentation of page images using the area Voronoi diagram », *Computer Vision and Image Understanding*, vol. 70, p. 370-382, 1998.

Mao S., Rosenfeld A., Kanungo T., « Document structure analysis algorithms : a literature survey », *Proc. SPIE Electronic Imaging*, 2003.

Nagy G., « Twenty Years of Document Image Analysis in PAMI », *IEEE PAMI*, vol. 22, p. 38-62, 2000.

Nagy G., Seth S., « Hierarchical representation of optically scanned documents », *ICPR'84*, 1984.

O'Gorman L., « The document spectrum for page layout analysis », *IEEE PAMI*, vol. 15, p. 1162-1173, 1993.

Otsu N., « A Threshold Selection Method from Gray-Level Histograms », *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, p. 62-66, 1979.

Press W., Teukolsky S., Vetterling W., Flannery B., *Numerical Recipes in C*, Cambridge University Press, chapter Sorting - Determination of equivalency classes, p. 345-346, 1992.

Rijsbergen C., *Information Retrieval*, Butterworth-Heinemann, 1979.

Shafait F., Keysers D., Breuel T., « Performance Evaluation and Benchmarking of Six Page Segmentation Algorithms », *IEEE PAMI*, vol. 30, p. 941-954, 2008.

Trupin E., « La reconnaissance d'images de documents : Un Panorama », *Traitement Du Signal*, vol. 22, p. 159-189, 2005.

Wang Y., Phillips I., Haralick R., « Document zone content classification and its performance evaluation », *Pattern Recognition*, vol. 39, p. 57-73, 2006.

Wong K., Casey R., Wahl F., « Document Analysis System », *IBM Journal of Research and Development*, vol. 26, p. 647-656, 1982.

ANNEXE POUR LE SERVICE FABRICATION

A FOURNIR PAR LES AUTEURS AVEC UN EXEMPLAIRE PAPIER
DE LEUR ARTICLE ET LE COPYRIGHT SIGNE PAR COURRIER
LE FICHIER PDF CORRESPONDANT SERA ENVOYE PAR E-MAIL

1. ARTICLE POUR LA REVUE :
   L'objet. Volume 8 – n°2/2005

2. AUTEURS :
   Rémi Vieux, Jean-Philippe Domenger

3. TITRE DE L'ARTICLE :
   Segmentation non supervisée d'images de document en paragraphes

4. TITRE ABRÉGÉ POUR LE HAUT DE PAGE MOINS DE 40 SIGNES :
   Segmentation en paragraphes

5. DATE DE CETTE VERSION :
   19 mars 2012

6. COORDONNÉES DES AUTEURS :

   – adresse postale :

   LaBRI - Université de Bordeaux - CNRS UMR 5800 - 341 Cours de la libération, 33400 Talence

   – téléphone : 00 00 00 00 00

   – télécopie : 00 00 00 00 00

   – e-mail : Roger.Rousseau@unice.fr

7. LOGICIEL UTILISÉ POUR LA PRÉPARATION DE CET ARTICLE :
   LaTeX, avec le fichier de style `article-hermes.cls`,
   version 1.2 du 2000/12/01.

8. FORMULAIRE DE COPYRIGHT :
   Retourner le formulaire de copyright signé par les auteurs, téléchargé sur :
   `http://www.revuesonline.com`

SERVICE ÉDITORIAL – HERMES-LAVOISIER
14 rue de Provigny, F-94236 Cachan cedex
Tél : 01-47-40-67-67
E-mail : revues@lavoisier.fr
Serveur web : http://www.revuesonline.com