Segmentation de documents écrits en ligne

Blanchard Julien – Artières Thierry

LIP6, Université Paris 6,
8 rue du capitaine Scott, 75015 France

Julien.Blanchard@poleia.lip6.fr

Thierry.Artières@lip6.fr

Résumé *: Dans le cadre d'applications du type cahier électronique, nous nous intéressons à la segmentation de pages d'écriture peu structurées, comme des notes manuscrites. Nous présentons des améliorations à un système existant basé sur des Grammaires Probabilistes à Caractéristiques. La nature probabiliste du système permet de générer de nombreuses hypothèses de regroupements, ce qui est intéressant pour traiter des documents assez peu structurés donc ambigus, mais induit une grande complexité algorithmique. Nos améliorations portent notamment sur la prise en compte de cette complexité et sur la définition de mesures de performances adaptées aux documents en ligne.*

Abstract *: This work concerns note-taking applications, it deals with poorly structured handwritten documents segmentation such as pages of handwritten notes. We extend an existing system based on Probabilistic Feature Grammars. The probabilistic nature of this system allows considering lots of segmentation hypothesis, which is an advantage for poorly structured documents processing, but it goes with important algorithmic complexity. Our improvements concern the handling of this complexity using genetic algorithms and the definition of performance measures that are adapted to the segmentation of on-line documents.*

Mots-clés *:* Segmentation, Documents peu structurés, Notes manuscrites, Cahier électronique, Grammaires probabilistes, Algorithmes génétiques.

Keywords *:* Segmentation, Poorly structured documents, Note-taking, Probabilistic grammars, Genetic algorithms.

1 Introduction

Notre travail se situe dans le cadre d'applications de type cahier électronique utilisant une interface stylo. Les difficultés induites par la prise de notes manuscrites libre sont multiples. Les pages sont faiblement structurées et sont caractérisées par une forte variabilité du signal. Les techniques développées pour la segmentation de documents hors-ligne du type journaux ou tables des matières [KIS 98], [NAG 84], [OGO 93] sont généralement adaptées à la segmentation de documents imprimés présentant une forte régularité et reposent sur des caractéristiques globales calculées sur un document. Les premières méthodes proposées pour la segmentation de documents manuscrits sont souvent basées sur des projections d'histogrammes pour détecter des lignes et ensuite découper celles-ci en mots ou en groupes de mots [MAR 01], [RAT 00]. Là encore, ces méthodes font des hypothèses fortes quant à la régularité des caractéristiques, notamment sur la pente des lignes ou la taille de l'interligne. Des travaux récents [JAI 01], [SHI 03] sur des types de documents assez limités relâchent ces hypothèses et tiennent davantage compte du contexte pour réaliser un traitement plus local du signal. Le système présenté dans [GAU 02] est une tentative de réponse aux problèmes soulevés par les méthodes les plus classiques et les plus spécialisées. Le formalisme choisi est basé sur des Grammaires Probabilistes à Caractéristiques (GPCs) [GOO 97], qui ont été adaptées au traitement de données bidimensionnelles. Ces grammaires permettent de prendre en compte de façon assez simple une information contextuelle, ce qui paraît essentiel si l'on souhaite traiter des documents peu structurés donc fortement ambigus. Dans la suite, nous décrivons brièvement le système initial et détaillons notre apport, l'intégration d'algorithmes génétiques pour casser la complexité algorithmique de l'approche et une évaluation expérimentale, sur une base que nous avons collectée, pour laquelle nous avons défini des mesures de performances adaptées aux documents en ligne. Dans ces expériences, nous comparons notre système à une méthode plus classique inspirée du Docstrum [OGO 93] sur un corpus de documents manuscrits contenant à la fois des documents fortement structurés et des documents faiblement structurés.

2 Utilisation de GPCs pour l'analyse de documents manuscrits

Nous décrivons brièvement ici l'application des GPCs à la segmentation de documents manuscrits en ligne, plus de détails peuvent être trouvés dans [GAU 02]. Concernant le système de segmentation, nos apports par rapport à ce dernier système se situent dans les §2.3 et §2.4. Nous commençons par donner une grammaire pour le traitement de textes. Puis, nous décrivons les lois de probabilités associées à la production des règles et l'apprentissage de leurs paramètres. Nous présentons

enfin l'algorithme de segmentation et l'introduction des algorithmes génétiques.

2.1 Une grammaire pour les documents textuels

Nous nous sommes restreints jusqu'à présent, à une grammaire relativement simple permettant de segmenter une page en paragraphes composés de lignes que nous présentons maintenant, mais le formalisme est générique et peut être utilisé avec d'autres types de grammaires, correspondant à d'autres types de documents. La grammaire est définie par les 5 règles suivantes :

1. Page → Paragraphe
2. Paragraphe → Paragraphe [Au-dessus-de] Ligne
3. Paragraphe → Ligne
4. Ligne → Ligne [A-droite-de] Mot
5. Ligne → Mot

Cette grammaire définit un document constitué d'une suite de paragraphes, eux-mêmes constitués de lignes situées les unes au dessus des autres, chaque ligne étant constituée d'une suite de mots alignés de gauche à droite. Ici, un mot représente un tracé écrit sans levée de stylo et peut en réalité n'être qu'une marque diacritique. Les opérateurs entre crochets [Au-dessus-de] et [A-gauche-de] ne font pas réellement partie de la grammaire. Ce sont des opérateurs sur lesquels s'appuient des heuristiques permettant d'implémenter efficacement les algorithmes, plus de détails peuvent être trouvés dans [GAU 02].

2.2 Probabilités de production de règles

Un des intérêts des GPCs est qu'elles permettent l'intégration d'informations contextuelles par la propagation de caractéristiques associées aux entités dérivées. Dans les GPCs les règles sont de la forme :
$X = (x_1, x_2, ..., x_g) \rightarrow Y = (y_1, y_2, ..., y_g), Z = (z_1, z_2, ..., z_g)$
où X, Y et Z sont des termes, $(x_1, x_2, ..., x_g)$, $(y_1, y_2, ..., y_g)$ et $(z_1, z_2, ..., z_g)$ sont des vecteurs de g caractéristiques associés à ces termes. Dans notre cas, les termes de la grammaires sont les constituants d'une page manuscrite: section, paragraphe, ligne, mot. Les grammaires sont vues comme des modèles génératifs de production. En notant *C(X)* les caractéristiques associées à un terme *X*, la probabilité de déclenchement de la règle ci-dessus est donnée par :

P(C(Y), C(Z) / C(X))

Dans notre système, les caractéristiques des termes sont liées à leur hauteur, leur pente, etc. Le vecteur de caractéristiques d'un terme X est constitué de trois types de caractéristiques : ses caractéristiques propres (hauteur, largeur, pente) que l'on note cp(X) ; les caractéristiques moyennes des entités qui la composent (cec(X)) ; et des caractéristiques décrivant la relation spatiale entre les entités qui la composent (rsec(X)). Un vecteur de caractéristiques associé à un terme X est donc donné par un triplet: C(X)=(cp(X), cec(X), rsec(X)). Ainsi, un paragraphe a des caractéristiques propres (sa hauteur, sa largeur et sa pente), des caractéristiques moyennes des lignes qui le composent (leur pente moyenne, leur largeur moyenne etc.), et des caractéristiques décrivant la relation spatiale entre les lignes qui le composent (la distance moyenne entre deux lignes successives). De même une ligne a des caractéristiques propres (sa hauteur, sa largeur et sa pente), des caractéristiques moyennes des tracés qui la composent (leur pente moyenne, leur largeur et hauteur moyenne, etc.), et des caractéristiques décrivant la relation spatiale entre les tracés qui la composent (la distance moyenne entre deux tracés successifs). A partir de ces caractéristiques on définit des lois de probabilité de déclenchement des règles de la grammaire en utilisant une hypothèse simplificatrice d'indépendance entre les diverses caractéristiques. Par exemple, une ligne est rajoutée à un paragraphe, avec la règle n° 2 de la grammaire présentée au §2.1, si la distance entre la dernière ligne du paragraphe et cette nouvelle ligne est similaire à l'interligne moyen dans le paragraphe, si la pente de la ligne correspond à la pente moyenne des lignes du paragraphe, etc.

2.3 Apprentissage des paramètres

Les probabilités de production de règles sont implémentées par des lois gaussiennes pour chacune des caractéristiques. Dans [GAU 02], les paramètres de ces lois avaient été estimés empiriquement sur une collection de documents. Afin de remédier à cette lacune, nous avons effectué un apprentissage statistique de ces paramètres sur une partie des documents du corpus, décrit au §3.1. Cet apprentissage repose sur un étiquetage des documents, c'est-à-dire sur un étiquetage désiré en paragraphes, lignes etc. Le principe est de construire pour chaque document l'arbre de dérivation idéal à partir de l'étiquetage puis de calculer pour chaque niveau de la grammaire (i.e. chaque règle), et pour chaque caractéristique, les moyennes et variances des caractéristiques.

2.4 Segmentation : Arbre de dérivation optimal

La segmentation d'une page produit un arbre de dérivation. Dans cet arbre, à chaque nœud correspond un terme *t* (et ses caractéristiques) et un numéro de règle *n*, le terme *t* ayant été dérivé à partir de la règle *n* et des termes des nœuds fils. La segmentation d'une page est réalisée à l'aide d'un algorithme qui détermine l'arbre de dérivation optimal, c'est à dire celui de probabilité maximale. Nous avons développé un algorithme inspiré de l'algorithme utilisé dans [STO 95], qui procède itérativement, niveau par niveau, en construisant d'abord des lignes à partir des mots de la page (un mot est un signal écrit entre deux levées de stylo), puis en construisant des paragraphes à partir des lignes trouvées etc. A chaque niveau, l'algorithme construit itérativement des ensembles de tracés de plus en plus grand en mettant à jour les hypothèses courantes, à la manière d'un algorithme de programmation dynamique comme Viterbi. Un problème de complexité se pose car nous cherchons à déterminer à la fois l'ordre de lecture des tracés et leur étiquetage, ce qui multiplie le nombre

d'hypothèses de segmentation possibles et rend impossible de conserver toutes ces hypothèses. On utilise donc une stratégie d'élagage des hypothèses les moins probables, et nous avons implémenté des opérateurs qui sélectionnent pour chaque règle les tracés susceptibles d'être agrégés à la prochaine étape. Ainsi, nous utilisons des opérateurs « bottom-up » qui déterminent à partir des données, et pour chaque règle, un ensemble limité d'entités à considérer. Ces opérateurs se basent sur la position où doit se trouver le prochain terminal (à droite des entités déjà agrégées, ou bien en dessous etc.). Par ailleurs, on utilise également une heuristique qui vise à favoriser un ordre de lecture équivalent à l'ordre dans lequel les mots ont été écrits.

Malgré ces heuristiques, le nombre de termes (e.g. lignes) générés peut être extrêmement important et nuit au traitement de pages entières. Nous avons donc ajouté dans ce système des points de rupture à la fin du traitement de chaque niveau (ligne, paragraphe…) afin d'élaguer encore plus les hypothèses. Cette optimisation consiste à ne conserver, au passage d'un niveau au suivant, que les entités servant à former des ensembles d'entités cohérents pour le niveau supérieur et regroupant le maximum de mots possible. La mesure de cohérence concerne le fait qu'aucune des entités d'un ensemble donné n'ont de tracés en commun. Par exemple, avant de passer au niveau des paragraphes, une fois que toutes les lignes (ou du moins celles suffisamment probables) ont été trouvées à partir des mots de la page, on effectue un élagage de la façon suivante. On détermine des ensembles de lignes cohérents, c'est à dire d'intersection nulle et couvrant si possible tous les mots de la page, et pouvant être regroupées en paragraphes au niveau supérieur. Seules les lignes appartenant à des ensembles cohérents les plus vraisemblables sont effectivement conservées. Cependant, trouver tous les ensembles de lignes cohérents est un problème combinatoire complexe, et nous avons utilisé une stratégie sous optimale classique pour casser cette complexité : nous utilisons un algorithme génétique de façon à sélectionner des ensembles de lignes conservées pour former des paragraphes.

Dans notre cas un individu de la population est un ensemble cohérent d'entités tel que nous l'avons évoqué ci-dessus. Nous avons tenu à employer une mesure de fitness la plus simple possible, afin d'une part de pouvoir expliciter aisément le comportement de l'algorithme et d'autre part afin d'assurer la conservation de la généricité globale du système. Ainsi la mesure de fitness est le nombre de mots couverts par l'individu, sachant que les entités sont elles même composées de mots. La fonction de crossover, elle, consiste à sélectionner aléatoirement des entités de chacun des deux individus et à les recombiner afin de former un nouvel ensemble cohérent. Enfin, la mutation consiste à remplacer au hasard une des entités de l'individu par une des entités disponibles à la fin du niveau précédent, ceci en respectant également la cohérence. Quelle que soit la taille des documents, nous avons fixé les paramètres. En effet, nous avons effectué différentes expérimentations sur des documents très variables aussi bien en taille qu'en homogénéité et les résultats se sont révélés très stables. Cependant pour les documents les plus longs, une taille de population et un nombre de générations élevés procurent de meilleurs scores sans détériorer ceux obtenus sur les petites pages. Nous avons donc choisi des valeurs convenant aux plus grands documents et impliquant un temps de calcul raisonnable (de l'ordre de la seconde au maximum). Ainsi nous avons opté pour une population de 40 individus, sur 100 générations, avec un pourcentage de mutation de 5%.

Pour connaître l'efficacité de l'algorithme génétique utilisé, nous avons mesuré à chaque niveau le nombre d'entités correctes conservées et le nombre d'entités incorrectes supprimées. Le tableau 1 présente ces résultats. On observe que, par exemple, 99% des lignes correctes sont conservées, ce qui signifie que l'algorithme génétique élimine très peu de « bonnes » lignes, tout en supprimant 76% des lignes incorrectes. Les résultats au niveau des paragraphes sont comparables (respectivement 95% et 84%). Ceci confirme que la méthode est justifiée puisqu'on élimine la plupart des entités indésirables tout en conservant la quasi-totalité des entités attendues.

Lignes correctes conservées	Lignes incorrectes supprimées	Paragraphes corrects conservés	Paragraphes incorrects supprimés
98.99%	75.99%	94.58%	83.80%

TAB. 1 - Efficacité de l'algorithme génétique

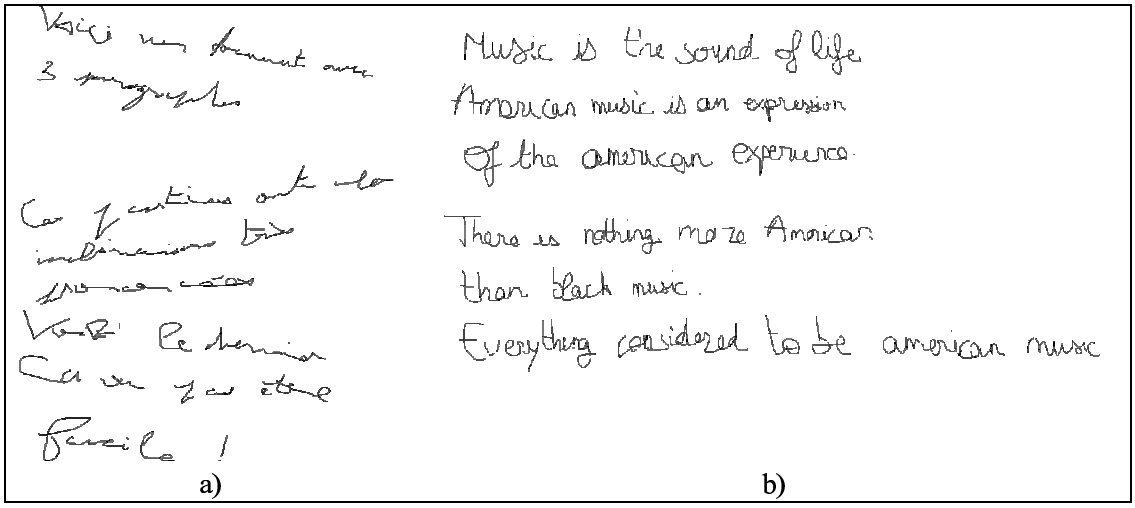
3 Résultats expérimentaux

3.1 Base de données de documents manuscrits en ligne

Nous avons collecté une base multi scripteurs étiquetée de pages de notes manuscrites dans notre laboratoire (LIP6). Les documents de la base sont des pages manuscrites écrites avec une tablette, sans contraintes. Ils possèdent de 3 à 30 lignes, chaque ligne étant composée de 2 à 20 « mots » environ. Les documents sont étiquetés par une segmentation, faite à la main, en lignes et paragraphes.

Les documents de la base sont divisés en deux catégories (voir FIG.1) ; Des pages «homogènes», du type lettres ou prises de notes soignées, présentant des caractéristiques globales assez régulières, taille d'écriture constante, lignes plutôt droites et parallèles ; Des pages «hétérogènes», de type brouillon ou «post-it», présentant des caractéristiques variables.

L'évaluation est réalisée sur un ensemble de 56 documents, répartis de manière équitable entre les deux classes présentées ci-dessus.



a) b)

FIG. 1 - Exemples de documents « hétérogène » (a) et « homogène » (b).

3.2 Mesures de performances

Le résultat du système appliqué à une page est un arbre de dérivation, on pourrait donc utiliser une mesure de similarité entre l'arbre obtenu automatiquement et l'arbre correspondant à la segmentation « à la main » de la page. Cependant, les mesures de similarité entre arbres ne sont pas forcément pertinentes pour mesurer la qualité d'un système de segmentation. De plus, la méthode de référence que nous avons implémentée (cf. §3.3) n'utilise pas d'arbre de dérivation et la comparaison entre cette méthode et notre système n'aurait pas été très simple. Nous avons préféré définir plusieurs mesures de performance permettant d'étudier le comportement du système à différents niveaux, construction des lignes, regroupements des lignes en paragraphes etc.

Pour calculer des performances au niveau de la détection des lignes, on apparie les lignes trouvées aux lignes réelles du document. On associe chaque ligne trouvée à la ligne « réelle » du document qui lui est la plus proche du point de vue de la distance d'édition. Ensuite, on compare les lignes appariées suivant deux critères. Le premier, L1, est un critère ensembliste et correspond au pourcentage de mots des lignes réelles qui se retrouvent dans les lignes trouvées correspondantes. Le second, L2, fait intervenir l'ordre de lecture et est calculé à partir d'une distance d'édition entre la ligne réelle et la ligne trouvée. On calcule le pourcentage de transformations (nombre d'insertions, de suppressions et de substitutions divisé par le nombre de mots dans la ligne réelle) nécessaires pour passer de la ligne réelle à la ligne trouvée.

Pour les performances au niveau de la détection des paragraphes, on commence par apparier les lignes trouvées aux lignes réelles du document, puis les paragraphes trouvés aux paragraphes réels. Puis, on utilise une distance d'édition pour calculer le nombre de transformations nécessaires pour passer des paragraphes réels aux paragraphes trouvés. Ce critère, P1, caractérise la capacité à regrouper des lignes en paragraphes, il est complémentaire des critères sur les lignes. Ainsi, même si certaines lignes sont incomplètement détectées, on peut obtenir une bonne performance du point de vue de ce critère pourvu que les lignes soient correctement regroupées en paragraphes.

Au niveau des documents, nous utilisons deux critères. Pour le premier, D1, après avoir apparier les lignes trouvées et les lignes réelles, on calcule une distance d'édition entre le document trouvé et le document réel, vus comme des séquences de lignes et on calcule le pourcentage de transformations pour passer de l'un à l'autre. Pour le second, D2, après avoir apparié les lignes, on apparie les paragraphes et on calcule des pourcentages de transformations pour passer des documents réels aux documents trouvés, vus comme des séquences de paragraphes.

3.3 Méthode de référence

Nous avons comparé notre approche à une méthode de référence. Le choix n'est pas aisé puisque nous ne connaissons pas de méthode, suffisamment détaillée dans la littérature, adaptée à la segmentation de documents manuscrits en ligne. Nous avons donc choisi une méthode d'analyse de documents hors-ligne relativement souple, le Docstrum [OGO 93] car contrairement à d'autres méthodes classiques basées sur les histogrammes ou la transformée de Hough, l'analyse de la page est indépendante de l'orientation ou de la pente des lignes et ne requiert pas de connaissance a priori sur la taille des caractères ou les interlignes. Le Docstrum est une méthode basée sur un partitionnement des entités de la page par l'algorithme des k-plus proches voisins. La segmentation est de type «bottom-up», c'est-à-dire que l'on part des éléments de plus bas niveaux (les tracés) pour former des mots, des lignes, puis des blocs. L'orientation, les interlignes, la taille de la police sont estimés par l'étude des histogrammes de la distribution des angles et des distances des plus proches voisins. La méthode que nous avons implémentée est inspirée du Docstrum, il s'agit d'une adaptation aux documents en ligne.

3.4 Résultats

Nous avons effectué les expériences sur un ordinateur équipé d'un AMD Athlon 900Mhz et les temps de

calculs varient suivant la taille des documents de 0,3 à 12,4 secondes, pour une moyenne de 5.3 secondes, la taille moyenne des documents étant de 116 mots. Le tableau 2 récapitule des résultats obtenus par notre système et la méthode de référence.

	Base entière		Textes homogènes		Textes hétérogènes	
	Docs.	GPCs	Docs.	GPCs	Docs.	GPCs
L_1	11.8%	**3.8%**	*3.4%*	*3.9%*	22.5%	**3.4%**
L_2	22.7%	**13.1%**	6.5%	6.9%	37.0%	**15.7%**
P_1	15.3%	**2.1%**	**1.9%**	2.6%	22.8%	**5.4%**
D_1	29.4%	**25.5%**	**18.5%**	20.7%	36.1%	**26.2%**
D_2	*15.3%*	*15.2%*	**10.6%**	19.4%	15.0%	**9.0%**

TAB.2 - Comparaison des taux d'erreurs en segmentation de notre approche et d'une méthode inspirée du Docstrum pour divers types de documents, suivant les critères définis au §3.2.

Ce tableau compare les résultats du Docstrum et de notre méthode basée sur les GPCs. Les paramètres des lois dans notre système sont appris sur les documents de la base d'apprentissage (cf. §2.3). Pour obtenir des résultats significatifs, nous avons utilisé une évaluation par validation croisée. On réalise 56 expériences, pour chacune, on utilise 55 documents de la base en apprentissage et un document en test. Les résultats du tableau 1 sont des résultats moyens sur ces 56 expériences. Le Docstrum, lui, ne se base sur aucun paramètre. Comme on le voit, le Docstrum se comporte le mieux pour la segmentation des documents homogènes, dans ce cas les deux techniques sont comparables. On observe environ 3,5% d'erreurs au niveau des lignes pour le critère ensembliste et environ 6,5% pour le critère prenant en compte l'ordre de lecture. Les résultats sont également bons au niveau de la composition des paragraphes mais le sont nettement moins au niveau des documents. Les taux d'erreurs pour les critères D_1 et D_2, signifient que des paragraphes sont mal détectés, il y a sous ou sur segmentation au niveau des paragraphes des pages.

Pour des documents hétérogènes, moins structurés, les GPCs s'avèrent plus robustes que la méthode basée sur le Docstrum. Si les résultats de notre approche sont sensiblement moins bons pour les documents hétérogènes, ceux du Docstrum chutent de façon importante, on observe par exemple 22% d'erreur pour le critère ensembliste sur la composition des lignes.

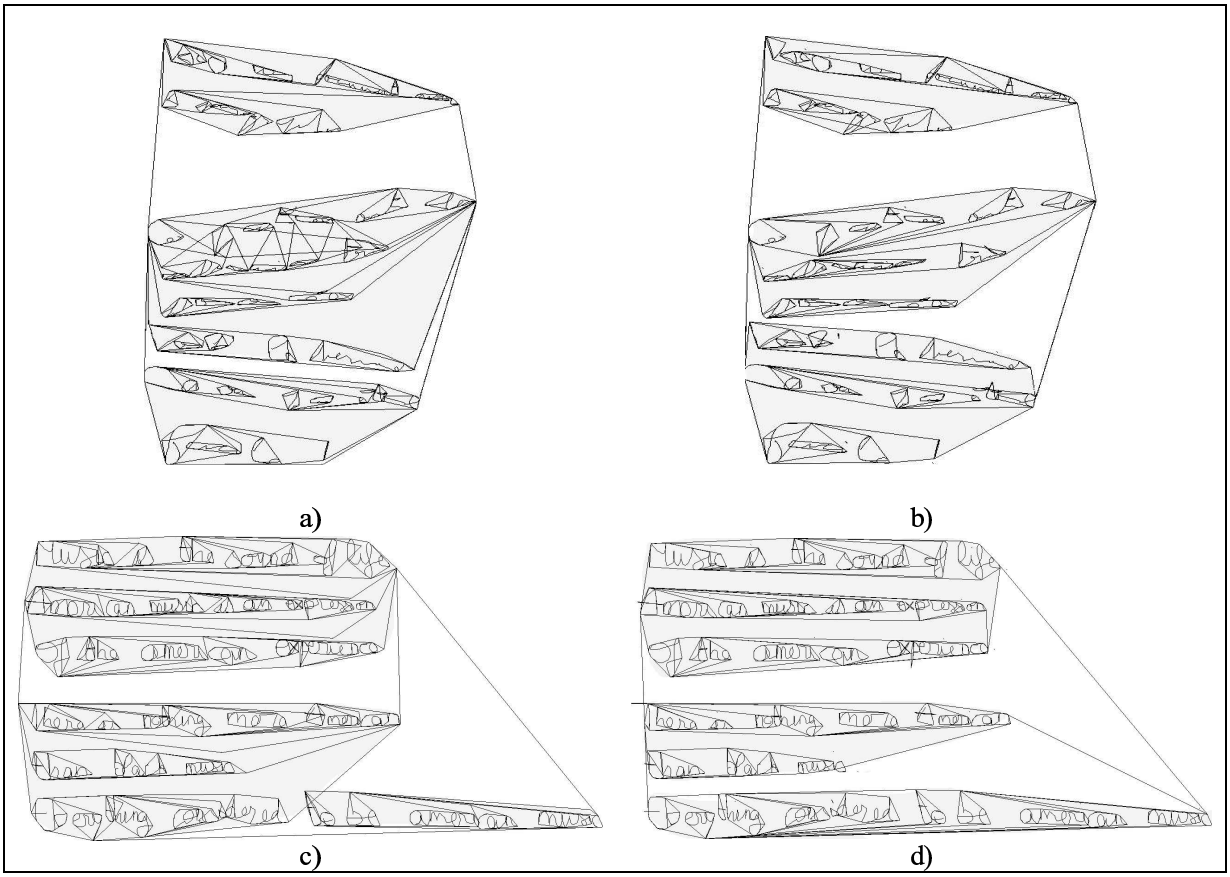


FIG.2 - Exemples de segmentations (les blocs grisés représentent des paragraphes).
a) segmentation d'un document « hétérogène » à trois paragraphes par le Docstrum.
b) segmentation du même document que a) par les GPCs.
c) segmentation d'un document « homogène » à deux paragraphes par le Docstrum.
d) segmentation du même document que c) par les GPCs

La figure 2 montre des exemples de segmentations par chacun des deux systèmes considérés. Pour les segmentations a) et b), on s'occupe d'un document plutôt hétérogène à trois paragraphes. Le Docstrum ne trouve pas correctement le deuxième paragraphe et celui-ci inclut la première ligne du troisième paragraphe. Les GPCs segmentent correctement le même texte.
Le deuxième document segmenté en c) et d) est de type « homogène » et contient deux paragraphes. Le Docstrum détecte trois paragraphes dont le dernier correspond à la fin de la dernière ligne. Les GPCs ne fournissent pas la segmentation idéale non plus, puisque trois paragraphes sont détectés également. Cependant le dernier, qui ne contient que la dernière ligne, est d'une certaine façon compréhensible. En effet la détection de ce paragraphe s'explique par le fait que la ligne qu'il contient a des caractéristiques différentes des deux lignes qui la précèdent (longueur supérieure, taille de caractères supérieure).

4 Conclusion

Nous avons étendu une approche probabiliste pour la segmentation de documents manuscrits en ligne. Cette approche, basée sur des Grammaires Probabilistes à Caractéristiques présente des avantages sur des techniques de segmentation publiées antérieurement, la prise en compte du contexte et le formalisme probabiliste qui permet d'envisager simultanément de nombreuses hypothèses de segmentation. Ces avantages induisent naturellement des inconvénients liés à la complexité algorithmique. Cette complexité ne permettait de traiter jusqu'à présent que des pages manuscrites relativement petites. Nous avons intégré dans l'algorithme de segmentation l'utilisation d'algorithmes génétiques qui permettent de casser la complexité et de traiter des pages de taille raisonnable. La méthode a été évaluée sur une base de documents de qualités variables. Nous avons défini des mesures de performances adaptées à notre problème de segmentation de pages d'écriture en ligne en faisant intervenir l'ordre de lecture. Nous avons validé notre travail en comparant notre système à une méthode de référence adaptée d'une technique classique de segmentation de documents hors-ligne. Les taux obtenus montrent un bon comportement de notre système, pour tout type de documents, contrairement à la méthode de référence, plutôt adaptée à des documents homogènes.

5 Références

[GAU 02] GAUTHIER N., ARTIERES T., Segmentation de documents peu structurés par grammaires probabilistes : application aux pages manuscrites en ligne, *CIFED*, 2002, pp 375-384.

[GOO 01] GOODMAN J., Probabilistic feature grammars, *International Workshop on Parsing Technologies,* 1997, pp 237-264.

[JAI 01] JAIN K., NAMBOODIRI A., SUBRAHMONIA J., Structure in on-line documents, *ICDAR*, 2001, pp 844-848.

[KIS 98] KISE K., SATO A., IWATA M., Segmentation of page images using the area Voronoi diagram, *Computer Vision and Image Understanding, 70*, 1998, pp. 370-382.

[LAR 90] LARI K, YOUNG S., The estimation of stochastic context-free grammars using the inside-outside algorithm, *Computer Speech and Language*, 4, 1990, pp 35-56.

[MAR 01] MARTI U., BUNKE H., Text line segmentation and word recognition in a system for general writer independent handwriting recognition, *ICDAR*, 2001, pp 159-163.

[NAG 84] NAGY G., SETH S., Hierarchical representation of optically scanned documents, *ICPR*, 1984, pp 347-349.

[OGO 93] O'GORMAN L., The document spectrum for page layout analysis, *IEEE Trans. PAMI, Vol. 15*, 1993, pp. 1162-1173.

[RAT 00] RATZLAFF E., Inter-line distance estimation and text line extraction for unconstrained online handwriting, *IWFHR*, 2000, pp 33-42.

[SHI 03] SHILMAN M., WEI Z., RAGHUPATHY S., SIMARD P., JONES D., Discerning Structure from Freeform Handwritten Notes, *ICDAR, Vol. 1*, 2003, pp 60-65.

[STO 95] STOLCKE A., An Efficient Probabilistic Context-Free Parsing Algorithm that Computes Prefix Probabilities, *Computational Linguistics, Vol. 21, No. 2*, 1995, pp. 165--201.