Indexation automatique de formulaires anciens par reconnaissance du patronyme manuscrit

Indexing old printed forms with handwritten last name

Jean Camillerapp[1] Laurent Pasquer[2] Bertrand Coüasnon[2]

[1]IRISA/Insa Rennes

[2]IRISA/INRIA

Adresse : 20, Avenue des buttes de Coësmes, F-35043 Rennes cedex

Email : Jean.Camillerapp@irisa.fr

Résumé

La numérisation des documents anciens protège les originaux et permet une consultation à distance, mais n'accélère pas pour autant la sélection des pages qui intéressent un lecteur parmi une grande masse de documents

Si la reconnaissance de l'écriture manuscrite dans des documents anciens reste dans la majorité des cas très difficile, il existe cependant des types de documents pour lesquels il est possible d'envisager la construction automatique d'index.

Cet article présente d'abord les caractéristiques des documents que nous avons choisi d'indexer, puis le choix que nous avons fait pour représenter les index et pour les extraire automatiquement des images. Il présente ensuite comment l'ensemble de ces index est utilisé pour répondre à la demande de consultation formulée par un utilisateur. Il se termine en donnant quelques indices de performances du système de sélection de documents anciens en fonction d'un contenu manuscrit.

Mots clef

Indexation de documents, écriture manuscrite, archive, classification, distance d'édition.

Abstract

Digitizing old documents is important to protect originals and to view them on Internet. However, it gives nothing more to readers for retrieval of the document they are looking for.

Handwriting recognition in old documents is, most of time, extremely difficult or impossible to do. Nevertheless on some kind of old documents it is possible to think about automatic indexing of handwriting.

This paper starts by a presentation of the old documents we choose to index by handwritten content. Then we propose a way to represent indexes on handwriting and a way to automatically extract them in images of document. We then show how theses indexes are used to answer to a request made by a user. The paper ends on a performance evaluation of the system of document retrieval by handwritten content.

Keywords

Document indexing, handwriting, classification, edit distance

1. Introduction

Les services des Archives tant nationales que départementales, possèdent des centaines de kilomètres de rayonnages de documents, le plus souvent manuscrits. Or en l'absence de tables d'index, leur consultation oblige le lecteur à feuilleter un grand nombre de pages avant de retrouver les pages contenant l'information qu'il recherche.

Actuellement ces services procèdent à d'importantes campagnes de numérisation. La dématérialisation du document qui en résulte, permet de protéger l'original et en facilite l'accès, [13] . Des lecteurs peuvent consulter simultanément le même document, et ils peuvent éventuellement le faire à distance si les services rendent leurs collections accessibles à travers Internet.

Mais cette numérisation ne suffit pas [1] ; il faut absolument profiter de la forme numérique des documents pour construire des outils destinés à accélérer la sélection des documents pertinents.

2. Caractéristiques des documents

Si la lecture automatique des documents anciens n'est pas actuellement réalisable sur n'importe quel document, on peut par contre envisager des traitements automatiques sur certains types de documents. Pouvoir localiser assez précisément dans la page les mots manuscrits qui servent à l'indexation, et ceci indépendamment de leur reconnaissance constitue un facteur favorable pour

réaliser une indexation automatique ; c'est, par exemple, le cas des formulaires.

Les fiches d'incorporation militaire du XIX^{e} siècle, dont il existe plusieurs millions d'exemplaires en France, sont constituées à partir de formulaires préimprimés dans lesquels le nom est relativement bien écrit. Ces fiches sont regroupées en registre contenant environ 500 fiches, les registres sont classés par année. Ces registres sont très consultés par les généalogistes, mais ils ne sont actuellement indexés que par le numéro de matricule attribué par les autorités militaires.

En général il existe à la fin de chaque année, une table alphabétique qui permet de retrouver le numéro de matricule. Cette indexation n'est cependant pas considérée comme suffisamment fiable par les services des archives.

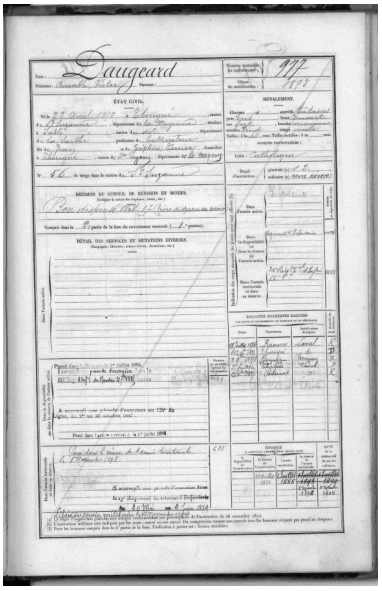


Figure 1. *Fiche d'incorporation militaire.*

Nous présentons dans cet article une méthode de construction automatique d'un index permettant de faire une recherche par le nom dans ces registres.

Vis à vis d'un traitement automatique ces documents ont un certain nombre d'avantages :

– le champ contentant le nom est parfaitement défini et localisé au sein du formulaire ;

– ce nom est relativement bien écrit, en utilisant une forme de lettres assez stable, la bâtarde, et il est écrit relativement gros.

Par contre, ils présentent un certain nombre de défauts :

– la numérisation introduit de petites rotations ;

– les images sont déformées par la reliure des registres lorsque la numérisation a été faite au moyen d'une caméra ;

– le papier présente une certaine transparence, le verso est donc partiellement visible (Figure 4) ;

– les fiches ont été endommagées, déchirées, recollées, tachées ;

– des tampons viennent perturber l'aspect visuel de la page (Figure 5) ;

– en raison de la guerre de 1914, certaines cases se sont avérées trop petites à l'usage. Les secrétaires ont donc collé de petites feuilles annexes (paperolles ou retombes) qui masquent largement la structure du document (Figure 2) ;

– sur certaines fiches, le nom est barré (Figure 6).

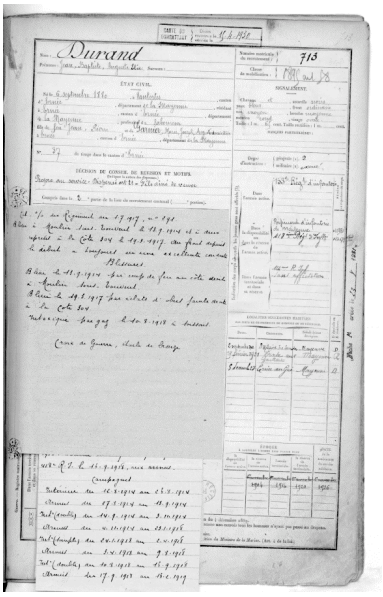


Figure 2. *Fiche comportant des retombes.*

Nous avons à notre disposition deux bases.

– La base des Archives départementales de la Mayenne contient 60.000 fiches. En réponse à une volonté politique de rendre accessible à travers Internet, la plus grande quantité possible des documents situés aux archives, ces fiches ont été découpées automatiquement mais indexées manuellement par ce service ;

– La base des Archives départementales des Yvelines contient plus de 300.000 fiches, c'est cette base qui doit être indexée automatiquement.

Dans les deux cas, les images ont été numérisées en 200 dpi, ce qui conduit à des images de 2000x3000 pixels. Avant tout traitement ces images ont été comprimées en JPEG. Si cette compression affecte peu la reconnaissance des segments, et donc celle de la structure, elle gène, par contre, la reconnaissance automatique des textes. En effet la compression JPEG induit un découpage en carré de 8 pixels qui bruitent la détection des contours des tracés un peu épais ce qui génèrent des barbules dans le squelette.

3. Architecture générale

Par rapport à l'écriture manuscrite la plus générale, l'écriture des noms en bâtarde fait apparaître une décomposition presque systématique de chaque lettre en traits.

Figure 3. *Exemples de nom écrit en bâtarde, avec un tracé segmenté en haut et un tracé lié en bas.*

La Figure 3 donne deux exemples d'écriture en lettres bâtardes. Dans l'image du haut, les lettres sont bien séparées et on perçoit même la décomposition d'une lettre en ses différents traits de plume. Dans l'image du bas, au contraire le tracé est cursif, mais l'existence des pleins et des déliés permet de retrouver la décomposition des lettres en traits.

La figure précédente montre des cas favorables, les figures ci-dessous donnent des exemples de difficultés rencontrées pour traiter ces images.

Figure 4. *Verso partiellement visible*

Figure 5. *Présence d'un tampon*

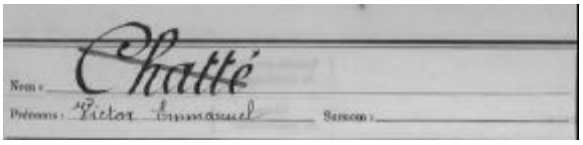


Figure 6. *Fiche dont le nom est barré.*

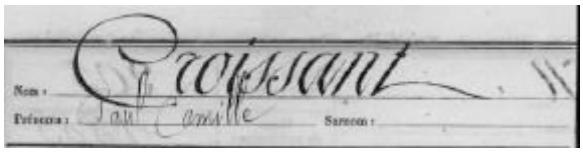


Figure 7. *Chevauchement du nom et du prénom.*

Figure 8. *Autre style de lettres et biais de l'image.*

Compte tenu de la qualité de l'écriture, on pourrait envisager de s'inspirer des techniques utilisées en OCR pour reconnaître les caractères, mais cela suppose une segmentation de l'image en caractères et donc la gestion de différentes hypothèses de segmentation.

Comme Seiler et *al* ont montré dans [16] la supériorité de la reconnaissance en-ligne sur la reconnaissance hors-ligne, on peut aussi rechercher la dynamique du tracé, comme le fait Lallican [9] .

La reconnaissance de l'écriture manuscrite en ligne distingue la reconnaissance des caractères isolés et la reconnaissance des mots.

Pour utiliser des méthodes de reconnaissance de caractères isolés, nous devrions comme précédemment découper l'image en caractères et gérer les différentes hypothèses de segmentation.

La reconnaissance de mots manuscrits appartenant à un grand vocabulaire incorpore la gestion de la segmentation, mais s'appuie, pour avoir des performances suffisantes, sur des connaissances lexicales. Ces connaissances peuvent utiliser un dictionnaire ou se limiter à la fréquence d'apparition des bigrammes ou des trigrammes. Dans le cadre de notre application, les mots à reconnaître sont des noms propres inconnus a priori, ce qui exclut l'usage d'un dictionnaire.

De toute manière, le processus de recherche dans les index doit tenir compte d'éventuelles erreurs de lecture et incorporer pour cela des distances de chaînes comme celle de Levenshtein [11] entre la requête et les index, ce qui introduit à nouveau du combinatoire.

L'étude visuelle des fiches montre que les traits qui composent les lettres bâtardes appartiennent à un vocabulaire de base assez limité. Par la suite nous désignerons chacun de ces traits sous le nom de graphème.

Comme il nous a semblé que la segmentation de l'image en graphèmes pouvait se faire avec un contexte réduit, mais que par contre le regroupement de ces graphèmes pour former des lettres implique d'examiner un certain nombre d'hypothèses de regroupement, nous avons choisi d'indexer les images par une chaîne ordonnée de graphèmes, et de transformer également la requête alphabétique de l'utilisateur en une chaîne de graphèmes.

Cette approche s'apparente aux techniques de représentation des mots par des silhouettes que l'on retrouve dans la structuration des dictionnaires [14] .

Afin de valider ce choix important, nous avons construit une première version pour l'indexation par le nom. La chaîne des traitements a été conçue de façon suffisamment modulaire, pour que les différents maillons qui la composent puissent être remplacés par d'autres. En particulier le choix du classifieur des graphèmes pourrait être facilement remis en cause.

Avoir une chaîne complète présente l'intérêt de fournir un indicateur sur les performances globales et permet donc d'évaluer l'impact de telles ou telles modifications.

Nous proposons de découper la présentation de notre approche de l'indexation des registres matricules en quatre parties :

– localisation de la structure du document,
– extraction des graphèmes,
– reconnaissance des graphèmes,
– utilisation des index.

4. Localisation de la structure du document

Ces fiches regroupent des informations nominatives dont la diffusion est protégée par la loi ; il y a par exemple un délai de 150 ans pour les informations médicales.

Pour mettre à la disposition du public la totalité des registres, il faut soit se limiter aux images de certaines cases, soit être capable de masquer les cases susceptibles de contenir des informations protégées. Il est donc nécessaire de reconnaître et de localiser précisément la structure de toute la fiche. De plus la vérification automatique de la cohérence globale de cette détection valide la localisation de chaque case.

De nombreuses méthodes ont été développées pour reconnaître des structures tabulaires. Lopresti en donne un aperçu dans [12] , mais nous n'avons pas trouvé dans la littérature de résultats évoquant des formulaires anciens, altérés ou partiellement masqués.

Dans l'équipe nous avons développé une méthode générique de reconnaissance de documents structurés : la méthode DMOS.

Cette méthode, dont on trouvera une description détaillée dans [6] s'appuie sur :

– un langage grammatical, EPF (*Enhanced Position Formalism*), de description de documents qui permet de modéliser la connaissance a priori,
– un analyseur produit automatiquement à partir de la description du document en EPF,
– l'équivalent d'analyseurs lexicaux, pour reconnaître les éléments terminaux du langage, qui dans le cas présent sont constitués par des segments de droites.

La généricité de cette méthode a déjà été validée en produisant un analyseur adapté aux partitions musicales et un autre dédié aux formules mathématiques. En s'appuyant sur une description adéquate, cette méthode est également capable de reconnaître une structure en tableau, comportant un nombre indéterminé de lignes ou de colonnes et dont les tailles sont inconnues a priori. Pour obtenir des performances satisfaisantes, il faut cependant posséder des documents de bonne qualité et dans lesquels la structure est nettement apparente.

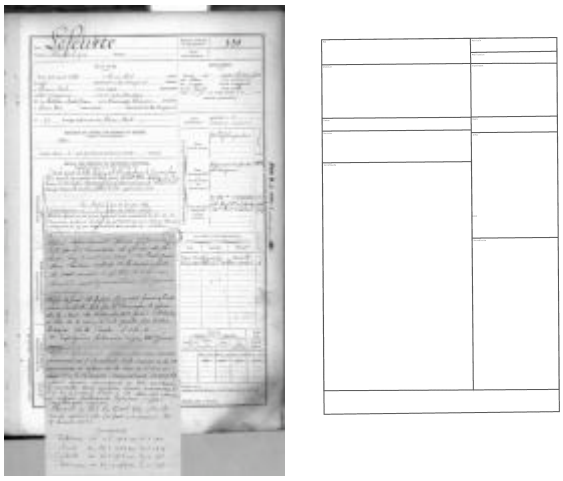


Figure 9. *Fiche avec des retombes et la structure reconnue.*

La structure des fiches est restée stable sur une quarantaine d'années, par contre la taille des cases a varié de 1 à 2 cm selon les éditions et certaines sont masquées par des retombes.

Pour compenser les dégradations importantes des documents il a suffi de restreindre la description très générique des tableaux de DMOS et d'incorporer des informations plus spécifiques. On a indiqué notamment que le document comporte

deux colonnes, qui sont elles-mêmes décomposées en cases. Par contre, il n'est pas nécessaire de connaître exactement la taille de ces différentes parties, ce qui permet une adaptation automatique à la variation de la hauteur des cases.

Nous avons traité les 60.000 pages des registres matricules de la Mayenne. Parmi ces pages 0,4% ont été automatiquement jugées non traitables par le logiciel, il s'agit effectivement de pages mal numérisées, trop abîmées ou ne correspondant pas à la structure. Il faut noter qu'il n'y a eu aucune erreur dans les pages acceptées. Cette fiabilité est un élément important pour une chaîne automatique de traitement.

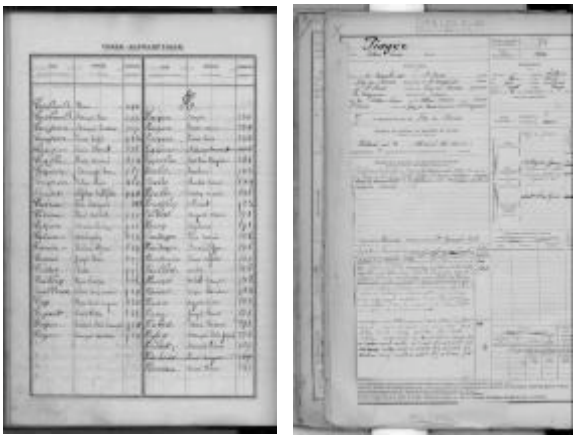


Figure 10. *Images rejetées : table alphabétique, page décalée.*

Le traitement d'une page nécessite environ 18 secondes (6 s de traitement d'image et 12 s d'analyse) sur une station SunBlade 100.

Cette partie du traitement, totalement autonome, fournit dans un fichier les coordonnées des coins de chaque case de la structure.

5. Extraction des graphèmes

La reconstitution de la dynamique du tracé dans des images de mots manuscrits a déjà suscité l'attention des chercheurs. Le problème majeur se situe dans les critères utilisés pour raccorder les tracés aux endroits des intersections.

Boccignone et *al* [4] utilisent des critères de bonne continuité du tracé qui ne prennent en compte que des informations très locales aux endroits des intersections.

Jäger [8] utilise un critère global en recherchant le chemin qui minimise la longueur totale et la courbure globale.

Doerman et Rosenfeld [7] combinent des critères locaux et des critères globaux.

L'écriture bâtarde n'atteint pas le degré de complexité de l'écriture cursive ordinaire. La plupart des intersections de traits sont des intersections à 3 branches, dont l'une des branches correspond à la liaison que le scripteur est tenté de mettre entre les lettres. C'est pourquoi nous avons principalement cherché à détecter ces liaisons, ce qui induit indirectement la continuité du tracé entre les deux autres branches.

Dans la suite de cette section, nous décrivons les différentes étapes qui permettent de construire la chaîne de graphèmes associée à la case du patronyme.

5.1. Binarisation

La reconnaissance de la structure fournit les coordonnées des coins de la case. Ces informations permettent :

– d'extraire précisément une sous-image d'environ 1000x300 pixels,

– de localiser les parties imprimées, comme les filets encadrant la fiche ou marquant la ligne de base.

Figure 11. *Sous image contenant le nom (noter la visibilité du verso par transparence)*

On associe à cette image la liste des points noirs (binarisation), mais l'image en niveaux de gris reste toujours accessible aux autres étapes du traitement.

Pour cela nous estimons le contraste local calculé comme la différence entre luminosité d'un pixel et la moyenne des luminosités dans un voisinage. Cette approche permet de compenser les variations de luminosité aussi bien d'une image à l'autre qu'à l'intérieur d'une même image. Afin de limiter les effets de seuil, la structure représentative des points noirs garde une localisation sub-pixelle de la séparation entre un pixel blanc et un pixel noir.

Le contraste local ne constitue pas un critère suffisant pour séparer les traits du recto de ceux du verso apparaissant par transparence. En effet pour les traits relativement fins la valeur de ce contraste est souvent inférieure à celle d'un tracé épais vu par transparence. La distinction ne pourra se faire que dans un contexte plus large : lorsque les pixels auront été regroupés pour former des tracés.

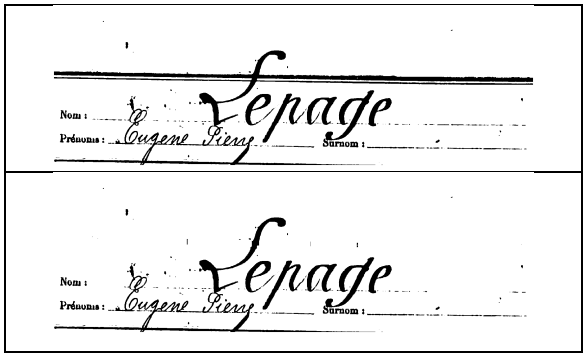


Figure 12. *Visualisation des points noirs, avec ou sans les points des filets.*

Les informations de positions extraites lors de la localisation de la structure permettent ensuite de repérer les points noirs qui peuvent être interprétés comme des points des filets ou de la ligne de base. Ces points sont soit retirés de la liste quand il n'y a pas d'ambiguïté d'interprétation, soit marqués dans les zones d'intersection avec le tracé.

5.2. Extraction du squelette

Dans le traitement de l'écriture manuscrite, il est relativement classique de représenter la trace laissée par la plume, par une ligne infiniment fine. C'est dans cette optique que les notions d'axe médian ou de squelette ont été introduites. Les méthodes de squelettisation ne manquent pas [10], mais beaucoup oublient l'objectif initial : représenter les objets longilignes. C'est pourquoi nous pensons qu'il est important de distinguer dans les images deux familles de regroupement de pixels :

– les zones régulières, zones dans lesquelles la notion d'axe médian est valide. Ces zones sont représentées par les pixels qui composent leur axe médian.
– les zones singulières, zones qui correspondent le plus souvent à des intersections du tracé et dans lesquelles le calcul d'un axe médian n'a aucune signification. Ces zones sont représentées par une liste de points noirs.

Nous avons vu que l'extraction des points noirs garde la distance de chaque point à la séparation noir blanc dans les 8 directions. Cette information permet de repérer la direction de la largeur du trait et la position de l'axe médian. Elle fournit donc les pixels squelettes auxquels sont associées la direction de la largeur du trait, et la valeur de cette largeur.

Une zone régulière correspond à un regroupement de points squelettes dont les largeurs sont cohérentes en direction et en valeur.

Les zones singulières sont déterminées après localisation des zones régulières. Elles regroupent les points noirs qui touchent les extrémités des zones régulières. Elles peuvent correspondre à :

– des débuts ou des fins de trait,
– des intersections de traits,
– des zones de forte courbure du tracé dans lesquelles les critères de cohérence entre les points squelettes ne sont pas respectés.

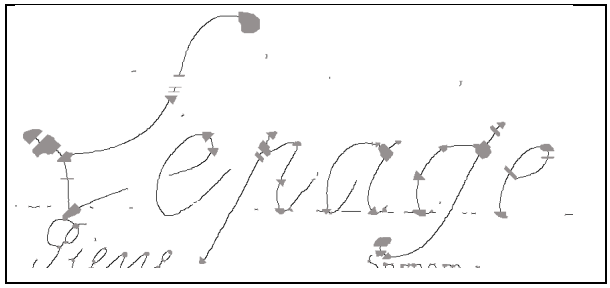


Figure 13. *Décomposition en zones régulières (les traits) et en zones singulières (les taches).*

Afin de préparer la suite du traitement on associe à chaque zone régulière une approximation polygonale de la liste ordonnée des points squelettes qui la constituent.

5.3. Traitement des intersections

Les zones régulières qui viennent d'être détectées ne correspondent pas encore aux graphèmes, car le tracé est trop segmenté.

Dans un premier temps, toutes les zones singulières qui ne connectent que deux zones régulières sont éliminées et les approximations polygonales sont concaténées.

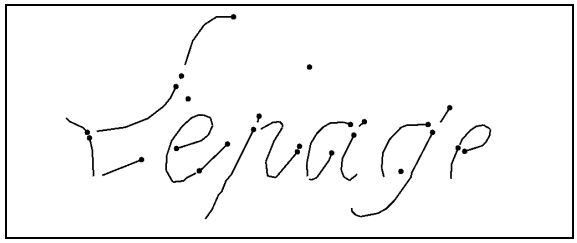


Figure 14. *Après suppressions des zones singulières ne reliant que deux zones régulières.*

Ensuite, à chaque extrémité d'une zone régulière, nous utilisons des règles locales s'appuyant sur l'épaisseur du tracé et sur sa direction pour repérer la présence d'une liaison entre lettres. Si c'est le cas, la connexion de cette extrémité de la zone régulière avec la zone singulière est rompue. Dans les cas d'intersection à trois branches, qui sont les plus fréquents, cette rupture entraîne implicitement la connexion des branches restantes.

D'autres règles, également locales, permettent de traiter les quelques cas qui restent encore pour former des traits.

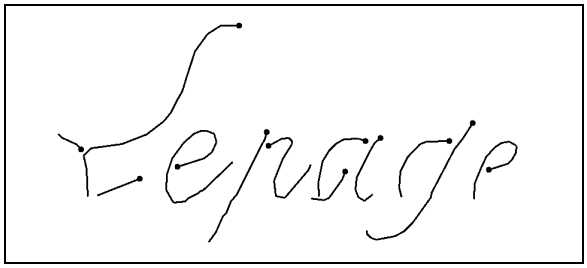


Figure 15. *Tracé après traitement des liaisons.*

Les traits trop petits, mal placés ou insuffisamment contrastés, sont éliminés, les traits restant sont ordonnés horizontalement. Ils vont fournir les graphèmes.

Une fois calculées, les caractéristiques de chaque graphème sont transmises au classifieur. Dans cette première version nous ne retenons que le nom de la première classe fournie par le classifieur. Nous ne tenons compte ni des autres réponses, ni de la valeur des indices de vraisemblance qui pourraient fournir des indications de rejet.

Figure 16. *Les graphèmes servant à l'indexation de l'image, les points symbolisent le début de chaque graphème.*

L'ensemble des traitements décrits ci-dessus qui permettent de construire l'index à partir de la sous-image prend environ 3 secondes par fiche.

6. Reconnaissance des graphèmes

6.1. Calcul des caractéristiques

Les caractéristiques que nous avons utilisées, s'inspirent plus de caractéristiques de tracé en-ligne que de caractéristiques liées à une image. Elles sont calculées à partir de l'approximation polygonale. Cette approximation polygonale est orientée dans le sens présumé du tracé. Compte tenu de la relative simplicité des graphèmes, la définition de cette orientation ne pose pas de problèmes particuliers.

La taille des lettres varie d'un facteur deux entre les différentes fiches, il est donc nécessaire de détecter le corps du mot qui servira de facteur d'échelle dans les caractéristiques liées à la taille des lettres.

Comme le scripteur est très influencé par la ligne de base imprimée dans le formulaire nous n'avons pas jugé utile de faire un ajustement complémentaire.

Nous avons déterminé le corps du mot en regardant la variation du nombre d'intersections en fonction de la position verticale d'une parallèle à la ligne de base.

Nous avons calculé une douzaine de caractéristiques, elles évaluent la taille (la hauteur du rectangle englobant), la position par rapport à la ligne de base, la position du point de départ du tracé et sa direction, le caractère plus ou moins rectiligne du tracé.

6.2. Choix du classifieur

Pour cette version nous avons utilisé, en raison de ces bonnes propriétés, un classifieur à base de réseaux de neurones [3] et plus particulièrement de RBF (Radial Basis Functions) [5] [15] . Mais ce classifieur [2] pourrait être remplacé par un système d'inférence floue ou par des arbres de décision.

6.3. Apprentissage des modèles

La constitution d'une base d'apprentissage ou d'une base de test constitue une charge lourde et relativement ingrate. Nous avons donc voulu tirer parti du travail fait par les Archives de la Mayenne qui ont créé manuellement l'association {fichier image, patronyme} sur l'ensemble de leurs fiches pour constituer automatiquement la base d'apprentissage du classifieur.

Comme la taille de cette base est importante, de l'ordre de 60.000 fiches, nous pouvons nous permettre de procéder par élimination automatique des images indésirables.

En raison de difficultés potentielles sur la reconnaissance de la majuscule initiale, nous utilisons la dernière lettre du nom et pour chaque graphème nous sélectionnons à partir du patronyme un nombre suffisant d'images (plus de 100) réparties uniformément sur l'ensemble des registres.

Dans une première passe nous localisons, dans ces images, les graphèmes qui sont sensés correspondre à la dernière lettre du patronyme, et nous en calculons les caractéristiques.

Après un examen visuel rapide, nous avons décidé de regrouper les graphèmes en 7 classes.

Nom	Description succincte
fl	Dépasse largement le corps du mot, se retrouve dans b, f, h, k, l
dt	Dépasse un peu le corps du mot, se retrouve dans d et t
nmu	Trait vertical de la taille du corps du mot, graphème très fréquent
gy	Contient une hampe descendante, se retrouve dans g, j, p, q, y
ca	Courbe de la taille du corps du mot avec une ouverture vers la droite, forme de base du c mais aussi du début des a, d, g, q
e	Graphème du e
s	Graphème du s

Comme la localisation des graphèmes de la dernière lettre est entachée d'erreurs, en raison des traits visibles par transparence, des tampons ou des erreurs de segmentation, nous ne gardons dans la base d'apprentissage que les images assez vraisemblables au vu de la valeur de quelques caractéristiques simples comme la hauteur par rapport au corps du mot, ou le caractère plus ou moins rectiligne du tracé. Par exemple, pour construire la base des « *e* », on ne garde que les images associées à des patronymes se terminant par un *e* et dont la hauteur du dernier graphème est voisine de la hauteur du corps du mot.

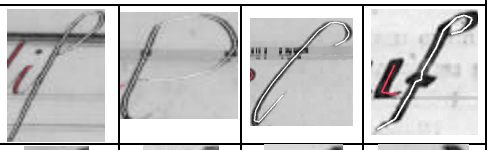
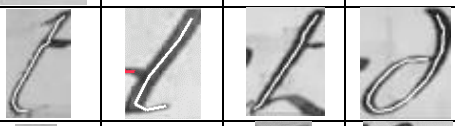
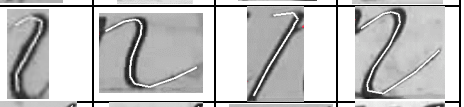
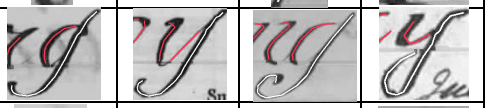
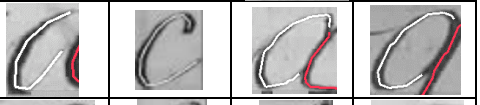
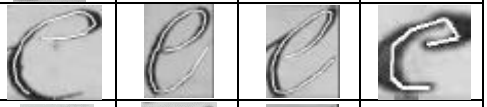
Nom	Images de graphèmes
fl	
dt	
nmu	
gy	
ca	
e	
s	

Figure 17. *Exemples de graphèmes, en blanc l'approximation polygonale, en noir le contexte.*

Le classifieur fonctionnant en apprentissage reçoit les caractéristiques des graphèmes sélectionnés et l'étiquette associée. Il en déduit les modèles les mieux adaptés.

Bien qu'il soit préférable de calculer les performances d'un classifieur sur un ensemble autre que l'ensemble d'apprentissage, nous avons évalué ces performances sur l'ensemble d'apprentissage, ce qui permet d'avoir également une idée de la qualité de la base d'apprentissage.

	Résultat du classifieur						
	fl	dt	nmu	gy	ca	e	s
fl	59,2	12,7	10,6	16,7	0,4	0,4	
dt	2,7	61,8	28,1	0,2	2,7		4,5
nmu	0,1	2,8	88,9	1,1	4,9		2,0
gy	0,6	0,4	6,6	91,3	0,6		0,2
ca	1,8	1,4	21,0	1,7	72,1	0,4	1,5
e	2,5	0,6	14,2	0,6	8,0	70,4	3,7
s	0,5	4,2	11,9	0,5			82,7

Figure 18. *Performance du classifieur sur la première base d'apprentissage.*

Avec un taux de reconnaissance de l'ordre de 80%, ce premier classifieur n'obtient pas des performances remarquables, car la base d'apprentissage qui lui a été fournie contient entre 10 et 15% de graphèmes mal étiquetés. Ces performances sont cependant suffisantes pour réaliser un nouveau filtre qui ne garde dans la base d'apprentissage que les graphèmes correctement reconnus par ce premier classifieur.

	Résultat du classifieur						
	fl	dt	nmu	gy	ca	e	s
fl	98,8	0,6	0,6	0,1		0,4	
dt	3,1	93,8	3,1				0,2
nmu	0,2		99,5	0,1	0,1		0,2
gy	0,2	0,2	1,4	98,1			
ca	0,2		3,6		96,1	0,2	
e						100	
s			4,4	0,7			94,9

Figure 19. *Performance du classifieur sur la deuxième base d'apprentissage.*

Le classifieur obtenu à partir de cette nouvelle base d'apprentissage, supposée correctement étiquetée, obtient des taux de reconnaissance bien plus élevés, de l'ordre de 98 %. Ce taux reste indicatif, car nous ne disposons pas, pour le mesurer, d'une base fiable de graphèmes, seule l'association {fichier image, patronyme} est connue.

7. Utilisation des index

Pour rechercher les images qui correspondent à un certain patronyme, l'utilisateur formule sa requête sous la forme de la chaîne des caractères qui composent le patronyme.

Cette chaîne est normalisée en mettant toutes les lettres en minuscules, en remplaçant les lettres accentuées par leur lettre de base (ainsi é devient e) et en supprimant la première lettre car celle-ci est toujours écrite en majuscule.

Ensuite nous utilisons une table pour transformer systématiquement chaque lettre en graphèmes. Ainsi un *d* dans le patronyme est transformé en deux graphèmes : *ca*, *dt*.

Par exemple la requête pour retrouver Renard conduit à rechercher dans les index la chaîne de graphèmes :{*e, nmu, nmu, ca, nmu, nmu, ca, dt* }.

Pour comparer cette chaîne avec tous les index nous avons utilisé la distance classique de Levenshtein et nous nous sommes inspirés de l'algorithme de Wagner et Fisher [18] .

Nous sommes restés, pour l'instant, sur une version très simple dans laquelle les coûts d'insertion, de suppression et de substitution sont égaux.

Des améliorations des performances pourront être obtenues en affinant ces coûts grâce à une connaissance statistique plus précise sur la fréquence des ces différentes possibilités [17] .

L'ensemble des images (ou un sous-ensemble) triées selon leur distance par rapport à la requête est alors transmis à l'utilisateur.

Actuellement, il faut moins d'une seconde pour rechercher un nom parmi 5.000 images.

Bien que le facteur temps ne soit pas primordial, nous pensons qu'il n'est pas raisonnable de vouloir interroger de la sorte une base de 50.000 à 100.000 images. Un regroupement des registres par année et la possibilité d'une interrogation limitée à quelques années nous paraîtrait plus judicieuse.

8. Résultats

Afin de pouvoir évaluer l'influence de différentes modifications sur les performances de la chaîne complète nous avons mis en place un indicateur. En profitant de l'association {fichier image, patronyme} de la base de la Mayenne.

Etant donné un sous-ensemble d'images ***E*** et un sous-ensemble ***N*** des noms présents dans ***E***, nous pouvons interroger la base ***E*** avec les noms de ***N***.

Comme chaque requête sur un nom de ***N*** produit une liste triée des images, il est possible de connaître le rang de la première (ou de la dernière) image qui est associée avec le nom recherché.

L'ensemble ***N*** des requêtes est trié en fonction du rang auquel se trouve la bonne réponse. Ce tri est représenté sous la forme d'une courbe qui donne le rang de la bonne réponse en fonction de la place de la requête dans la liste triée.

La courbe ci-dessous correspond à un ensemble de plus de 350 images dans lequel un patronyme n'existe qu'en un seul exemplaire.

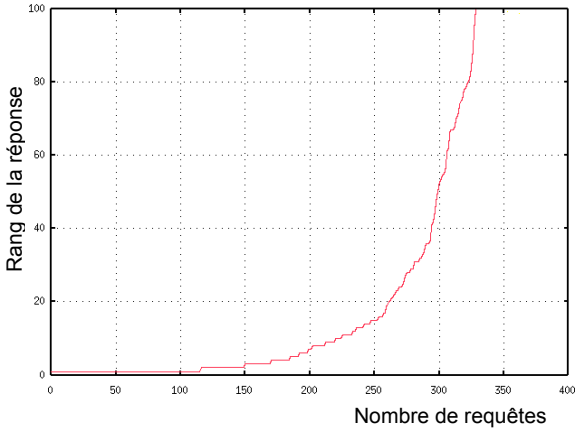


Figure 20. *Classement de la bonne réponse*

Sur cette courbe on peut voir, par exemple, que la bonne réponse est en première position dans plus d'un tiers des requêtes (jusqu'au rang 120), qu'elle est dans les 10 premières pour plus de deux tiers des requêtes (jusqu'au rang 240). On constate également qu'il existe des requêtes pour lesquelles les performances sont catastrophiques. Sur certaines images, comme celle de la Figure 21, ces contre-performances s'expliquent aisément. Les autres nous ont permis de mettre en évidence des erreurs résiduelles dans l'extraction des graphèmes que nous allons corriger dans la version suivante.

Figure 21. *Image dont l'indexation automatique est difficile.*

Il serait également intéressant de savoir estimer la probabilité qu'une fiche ne soit pas présentée dans un rang raisonnable alors qu'elle est présente dans la base.

9. Conclusion et perspective

Les résultats obtenus dans cette première version confirment la validité d'une indexation par les graphèmes. Ils permettent déjà d'envisager de rendre accessible au public des documents d'archives en les sélectionnant sur un contenu manuscrit.

L'étude détaillée des performances suggère des possibilités d'amélioration.

Un accroissement modéré du nombre des classes de graphèmes, avec si besoin le calcul de nouvelles caractéristiques devrait augmenter la distance entre une requête et les patronymes qui ne lui ressemblent pas au sens alphabétique. Cela se traduirait par un recul des mauvaises réponses dans la liste triée des réponses.

Une meilleure estimation des coûts utilisés dans la distance de Levenshtein devrait également être une source d'amélioration.

Enfin il nous semble important d'associer un indice de lisibilité à chaque image, cela permettrait de pouvoir présenter à l'utilisateur les images pour lesquelles l'indexation ne donnera pas de bons résultats (voir Figure 21).

Les auteurs tiennent à remercier les Archives départementales de la Mayenne, les Archives départementales des Yvelines, la Région Bretagne, le Conseil Général des Yvelines pour l'aide qu'ils leur ont apportée dans ce travail.

10. Bibliographie

[1] J. André, M.A. Chabin, *Numériser les documents anciens : et après ?*, 1999, Les documents anciens, numéro spécial de Document Numérique, vol. 3, n°1-2, p. 7-11.

[2] E. Anquetil, B. Coüasnon, F. Dambreville, *A Symbol Classifier able to Reject Wrong Shapes for Document Recognition Systems,* 2000, Graphics Recognition, Recent Advance, vol. 1941, Lecture Notes in Computer Science, Springer, p. 209-218.

[3] C.M. Bishop, *Neural Networks for Pattern Recognition*, Chapitre 5, Oxford University Press.

[4] G. Boccignone, A. Chianese, L. P. Cordella, A. Marcelli, *Recovering Dynamic Information from Static Handwriting*, 1993, Pattern Recognition, Vol. 26, No. 3, p. 409-418.

[5] D.S. Broomhead, D. Lowe, *Multivariable functionnal interpolation and adaptative networks*, 1988, journal Complex system 2, p. 321-355.

[6] B. Coüasnon, J. Camillerapp, *Une méthode générique de rétroconversion de documents pour la constitution de dossiers numériques*, 2002, Document Numérique, 6, 1-2, p. 129-144.

[7] D.S. Doermann, A. Rosenfeld, *Recovery of Temporal Information from Static Images of Handwriting*, 1995, International Journal of Computer Vision, 15, p. 143-164.

[8] S. Jäger, *Recovering Writing Trace in Off-Line Handwriting Recognition : Using a Global Optimization Technique*, 1996, 13th ICPR, vol 3, p. 150-154.

[9] P.M. Lallican, *Reconnaissance de l'écriture manuscrite hors-ligne : utilisation de la chronologie restaurée du tracé*, 1999, thèse de l'Université de Nantes.

[10] S.-W. Lee, L. Lam, C.Y. Suen, *Performance Evaluation of Skeletonization Algorithms for Document Image Processing*, 1991, ICDAR'91, p. 260-271.

[11] V.I. Levenshtein, *Binary Codes Capable of Correcting Deletions, Insertions and Reversals*, 1966, Soviet Physics Doklady, vol 10, p. 707-710.

[12] D. Lopresti, G. Nagy, *A tabular survey of automated table processing*, 2000, Graphics Recognition, Recent Advance, vol. 1941, Lecture Notes in Computer Science, Springer, p. 93-120.

[13] R. Lorie, *Information digitale : archivage à long terme*, 2000, CIFED'2000, p. 1-9.

[14] S. Madhvanath, V. Krpäsundar, *Pruning Large Lexicons Using Generalized Word Shape Descriptors*, 1997, ICDAR'97, p. 552-555.

[15] J. Moody, C.J. Darken, *Fast leraning in networks of locally-tuned processing units*, 1989, journal Neural Computation 1, p. 281-294.

[16] R. Seiler, M. Shenkel, F. Eggimann, *Off-line Cursive Handwrittin Recognition Compared with On-Line Recognition*, 1996, 13th ICPR, vol 4, p. 505-509.

[17] G. Seni, R.K. Srihari, N. Nasrabadi, *Large Vocabulary Recognition of on-line Handwritten Cursive Words*, IEEE Transaction on Pattern Analysis and Machine Intelligence, 1996, vol 18-7.

[18] R. Wagner, M. Fisher, *The String-to-String Correction Problem*, 1974, Journal JACM vol 21-1, p. 168-173.