



**HAL**  
open science

## Localisation des ratures dans les formulaires papier numérisés en ligne

Alain Wiart, Thierry Paquet, Laurent Heutte

► **To cite this version:**

Alain Wiart, Thierry Paquet, Laurent Heutte. Localisation des ratures dans les formulaires papier numérisés en ligne. Sep 2006, pp.169-174. hal-00134733

**HAL Id: hal-00134733**

**<https://hal.science/hal-00134733>**

Submitted on 5 Mar 2007

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Localisation des ratures dans les formulaires papier numérisés en ligne

Alain Wiart, Thierry Paquet, Laurent Heutte

Laboratoire LITIS EA 4051,  
Université de Rouen, 76800 Saint-Etienne du Rouvray, France

alain\_wiart@yahoo.fr, {thierry.paquet, laurent.heutte}@univ-rouen.fr

**Résumé :** *Dans cet article, nous présentons un système de prétraitement capable de localiser les ratures dans des documents papier manuscrits numérisés en ligne. Notre approche a été de concevoir un moteur de reconnaissance des ratures placé en amont de la reconnaissance d'écriture. Ce système classe des couples de strokes connectées, grâce à un ensemble de caractéristiques de bas niveau et un perceptron multi couches. Une partie de l'étude s'attache à produire une définition précise de la notion de rature, et aboutit à une subdivision des deux classes originelles du problème. Le système issu de cette première approche présente des performances encourageantes. Il bénéficie en outre d'un niveau de tolérance paramétrable lui permettant de s'adapter à différents moteurs de reconnaissance.*

**Mots-clés :** .

## 1 Introduction

Selon le mode de numérisation des documents, on dispose ou non d'information temporelles sur les données manuscrites à reconnaître. Contrairement aux documents physiques scannés (numérisation hors-ligne), les dispositifs numérisant en temps réel les mouvements du stylo restituent une description chronologique des tracés (numérisation en-ligne). La représentation temporelle de l'écriture permet alors de s'affranchir de certains écueils propres à la reconnaissance hors ligne. Elle facilite l'interprétation de caractères se chevauchant, apporte des informations relatives à la dynamique de l'écriture, et échappe de manière générale aux altérations occasionnées par une numérisation hors ligne (variabilité d'épaisseur de tracés, tracés parasités par le fond de page...) [TAP 90]. Les approches en-ligne permettent ainsi la plupart du temps d'obtenir de meilleurs résultats de reconnaissance que pour les approches hors ligne [PLA 00]. La nature strictement temporelle du signal issu d'une numérisation en ligne occasionne néanmoins certaines difficultés de traitement, notamment pour le traitement des marques diacritiques, des signes de ponctuation ou des ratures. L'ordre d'apparition de ces composantes n'étant pas toujours cohérent avec l'ordre de lecture, leur traitement constitue un aspect important du développement des approches de reconnaissance en ligne. Dans ce domaine, de nombreuses méthodes ont été proposées pour le traitement des marques diacritiques,

mais les méthodes de traitement des ratures et corrections sont plus rares. Effectivement, le mode d'acquisition en ligne étant principalement mis en œuvre dans le cadre d'applications interactives (PDA, tablet PC), l'interface propose en général la possibilité de corriger certains tracés en les annulant. Cependant, de nouveaux dispositifs de saisie en ligne, fondés notamment sur la technologie Anoto, amènent à reconsidérer la détection et la prise en charge des ratures. Cette technologie, mariant stylo à vision électronique et papier tramé, présente l'avantage de concilier numérisation en ligne avec le support papier. Le stylo est capable de connaître en temps réel sa position sur la feuille, par capture vidéo de la portion de trame avoisinant sa pointe. Il restitue ainsi la numérisation en-ligne des pages de texte sous forme d'une séquence de trajectoires (strokes), où toutes les informations relatives à la dynamique de l'écriture sont conservées. C'est une alternative intéressante aux appareils tels les tablet PC, dont l'utilisation peut s'avérer inappropriée dans certaines circonstances, de par leur fragilité ou leur volume. Avec cette technologie, la reconnaissance est la plupart du temps différée de la saisie, ceci empêchant la supervision par l'utilisateur au moment de la reconnaissance. Elle est techniquement impossible dans la plupart des cas, et se montrerait de toute façon peu compatible avec des applications dédiées à la prise de note.

Pour ce type de saisie en-ligne notamment, il est préférable de ne pas transmettre sans précautions les tracés correspondant à des ratures au moteur de reconnaissance. L'ordonnement strictement chronologique des tracés au sein du signal entraîne en général une mauvaise interprétation des ratures par le moteur de reconnaissance. En effet, lorsque le scripteur applique des corrections lors de sa saisie, les tracés correcteurs succèdent aux tracés corrigés selon des configurations extrêmement variables. Sans prétraitement dédié, il est difficile d'établir la correspondance entre tracés correcteurs et tracés corrigés, qui seront transmis séparément au moteur de reconnaissance. Il est donc souhaitable de confier la détection des ratures à un dispositif autonome, en amont de la reconnaissance d'écriture.

C'est un système de ce type qui est présenté dans cet article. Indépendant du moteur de reconnaissance, il

permet de scruter toutes les pages avant leur transmission au processus de reconnaissance afin de localiser les ratures éventuelles. L'usage de ce type de prétraitement est particulièrement justifié pour certaines catégories de documents. En l'occurrence, notre contexte d'étude s'inscrit dans le traitement de masse de formulaires médicaux. Collectés grâce à la technologie Anoto, des milliers de formulaires sont centralisés puis traités à des fins d'études statistiques. Le recours à la lecture automatique est alors salutaire. Sur le terrain, les circonstances de saisie privilégient la réactivité, aux dépens de la forme. Par conséquent, il n'est pas rare d'observer dans ces formulaires des corrections et ratures diverses qui, si elles ne sont pas détectées préalablement, constituent une source d'erreurs notable.

La première partie de cet article positionne notre thématique de travail et ses spécificités vis-à-vis des rares travaux proposés sur le traitement des corrections, et plus généralement sur la gestion des composantes retardées (delayed strokes). Les aspects techniques et méthodologiques de notre approche sont décrits dans la seconde partie, dans laquelle on s'attache également à définir la notion ambiguë de rature. Les résultats obtenus grâce à cette première approche sont analysés et commentés dans la troisième partie.

## 2 Problématique liée à la détection des ratures

Les approches de reconnaissance en ligne exploitent une séquence temporelle de tracés, appelés « strokes », transmis selon un ordre qui n'est pas toujours coordonné avec l'ordre de lecture. Le cas des symboles diacritiques illustre bien cette difficulté : certaines composantes de l'écriture, telles que les accents, barres de « t » et points de « i » apparaissent de manière variable au cours du temps. Les positions relatives des strokes n'étant pas explicites dans un signal issu d'une numérisation en ligne, la prise en compte de ces composantes retardées n'est pas immédiate. En conséquence, différentes méthodes existent pour le traitement des diacritiques : assimilation des marques diacritiques à des caractères en ajoutant au lexique toutes les orthographes alternatives induites [HU 94][HU 96], suppression des marques diacritiques après signalisation de leurs positions dans le vecteur de caractéristiques [JAE 00], combinaison des informations en ligne et hors ligne grâce à l'ajout de caractéristiques extraites des bitmaps [MAN 94], ou combinaison de moteurs de reconnaissances en ligne et hors ligne [ALI 97], et enfin recherche du meilleur agencement des strokes retardées, parallèlement au processus de reconnaissance, grâce à un algorithme de recherche de type forward [SEN 99].

Les travaux sur les composantes retardées dues à des ratures sont plus rares. La plupart du temps, les systèmes de reconnaissance en ligne disposent d'une supervision par l'utilisateur, qui peut vérifier lui-même les sorties du moteur de reconnaissance. La prise en compte automatique des ratures n'est alors pas d'une importance capitale : si la rature intervient juste après la saisie du mot, l'utilisateur n'a qu'à annuler sa dernière saisie puis la recommencer. S'il s'agit de modifier le contenu d'un

document longtemps après sa saisie, c'est le contenu résultant de la reconnaissance qui pourra être modifié.

La prise en charge automatique des ratures présente donc un intérêt limité pour les systèmes interactifs. C'est davantage l'ergonomie qui est favorisée pour le traitement des corrections. Certains systèmes permettent ainsi l'effacement, non plus grâce à un bouton, mais en mettant à disposition de l'utilisateur une série limitée de gestes du stylo, qui seront reconnus et interprétés comme une commande d'effacement (RESIFCar) [ANQ 02]. D'autres solutions se proposent d'améliorer encore la souplesse de ce type de solution, en minimisant les contraintes imposées à l'utilisateur. C'est le cas de l'interface intelligente de correction d'erreurs dédiée à NPEN++ [HUR 98]. Ce système se propose de prendre en charge simultanément toutes les composantes retardées, qu'elles soient des marques diacritiques, de la ponctuation ou des corrections. Dans cette approche, une segmentation des strokes sur les extrema horizontaux fait apparaître les composantes retardées, qui sont alors caractérisées par la régression singulière qu'elles occasionnent sur l'axe des abscisses. De cet ensemble sont extraites les strokes réparatrices, qui sont identifiées grâce à une série d'heuristiques. Lorsque le système identifie une action du scripteur potentiellement réparatrice, il cherchera à déterminer si sa nature est suppressive, ou complétive. Tout au long de ce processus, le contrôle est gardé par l'utilisateur, si bien que si une correction apposée n'est pas interprétée correctement par le système, alors l'utilisateur répètera sa manœuvre de correction. Cette stratégie de répétition est interprétée par le système. Le caractère semi autonome de ce système de prise en compte des ratures est intéressant. Dans cette approche, c'est la machine qui s'adapte aux stratégies de réparations humaines et non l'inverse.

Pour les applications reposant sur la technologie Anoto, les informations de numérisation sont restituées par le stylo selon leur ordre de saisie. Si aucune manœuvre n'est faite au préalable, les écrits seront transmis dans cet ordre au moteur de reconnaissance. Dans le cas d'une rature franche, en supposant qu'un mot soit écrit et barré aussitôt, il y a des chances que le moteur de reconnaissance se prononce sur l'identification du mot, et laisse de côté la stroke correctrice. Ce mot apparaîtra dans le contenu issu de la reconnaissance, alors que l'utilisateur souhaitait l'éliminer. Par ailleurs, si le tracé correcteur est saisi bien après mot lui-même, un autre problème se pose. Certes, pour le cas des marques diacritiques, le moteur de reconnaissance d'écriture est capable de prendre en charge certaines variations limitées de la répartition temporelle des tracés. Cependant, si les tracés intervenant sur le même mot ne se succèdent pas immédiatement dans le temps, la tâche est plus ardue. Il faut alors identifier au sein de la page plusieurs tracés correspondant à un même événement spatial, en considérant uniquement leurs positions géographiques.

Une des solutions à cette problématique est de localiser au cours d'une phase de prétraitements toutes les ratures présentes dans le document, de manière à

anticiper les erreurs qu'elles induiront lors de la reconnaissance d'écriture. Le but de cette étude est donc de concevoir un système de reconnaissance dédié à l'identification des ratures, auxquelles les pages de documents seront soumises préalablement à la reconnaissance d'écriture.

Comme la reconnaissance est différée de la saisie et doit être menée sur des masses de documents centralisées sur un serveur, aucune interactivité n'est possible, et le système se doit d'être totalement autonome. Par ailleurs, comme aucune contrainte de forme des ratures n'a été imposée pour la saisie, il est nécessaire de concevoir un système idéalement capable de s'adapter à toutes les stratégies de correction humaine. Nous avons donc conçu un système de reconnaissance des ratures, tirant son expertise de l'analyse de documents réels.

Notre base de connaissances est extraite de quelques centaines de documents. Une analyse de cet échantillon met rapidement en évidence le caractère ambigu de la notion de rature : si certaines corrections sont des ratures manifestes, d'autres résultent de retouches du scripteur, très localisées. Une catégorisation précise s'impose alors pour fournir une définition de la rature. C'est pourquoi notre choix s'est porté sur une subdivision des deux classes « rature » et « écriture standard » en sous classes (fig. 4). Plutôt que d'étiqueter simplement les données en tant que « rature » ou « écriture standard », un certain nombre de sous classes sont définies. Elles sont sensées rassembler des cas de ratures ou d'écriture standard similaires, autorisant des groupements compacts dans l'espace de représentation. En regroupant de différentes façons ces sous classes, on peut agir précisément sur le comportement du système, en déplaçant par exemple une des sous classes de la classe rature vers la classe écriture standard.

Une première différenciation des données raturantes est faite selon leurs modes correctifs. Certaines corrections visent à annuler le ou les tracés qu'elles recouvrent, et l'on parlera alors de rature suppressive

(fig. 1a et 1b), alors que d'autres, dites sur-écritures viennent les compléter (fig. 1c et 1d). Dans le premier cas, toutes les strokes localisées dans la zone de rature doivent être écartées du contenu. Dans le second, la ou les strokes correctrices sont constitutives de la morphologie du mot. Il sera alors nécessaire de faire appel à des traitements plus évolués, comme une suppression sélective des strokes, une réorganisation temporelle, ou une fusion des tracés par une approche de reconnaissance hors ligne.

D'autres catégorisations interviennent ensuite. Pour le cas des ratures suppressives, on considère des critères comme la densité de la correction (rature dense : fig. 1a, ou barré : fig. 1b), son étendue d'action (rature complète : fig. 2a, rature partielle : fig. 2b ou sous-rature : fig. 2c). Pour le cas des sur-écritures, on s'intéresse au rôle du tracé correcteur : s'il vient remplacer (surécriture suppressive : fig. 1d) ou compléter (surécriture complétive : fig. 1c) un ou plusieurs caractères, ou s'il s'agit d'une surcharge, c'est-à-dire si le scripteur a répété la saisie d'une correction pour la rendre prépondérante (fig. 3).

On procède également à une subdivision de la classe « écriture standard », qui différenciera entre autres les cas de marques diacritiques connectées, les croix, les signatures et croquis, les débordements entre lignes. Au total, 19 sous classes ont été conservées après fusion des catégories sous représentées (Barrés partiels et Débordement rature : voir fig. 5). Une classification directe des couples de strokes parmi ces 19 sous classes (fig. 4) serait trop ambitieuse. On espère au moins pouvoir distinguer les cas de ratures des cas d'écriture standard, et, dans le meilleur des cas, les cas de ratures suppressives des cas de sur-écritures. Les exigences sont variables selon que le système est entièrement automatique, ou qu'il soumet ses résultats de détection à une vérification humaine.

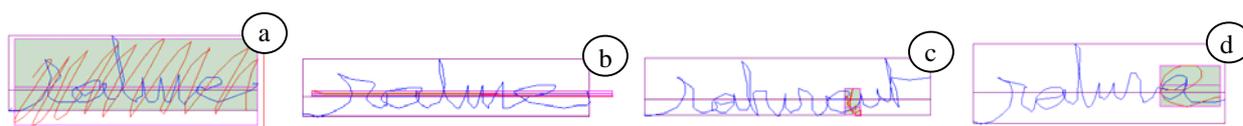


FIG. 1 - ratures suppressives, et sur-écritures.

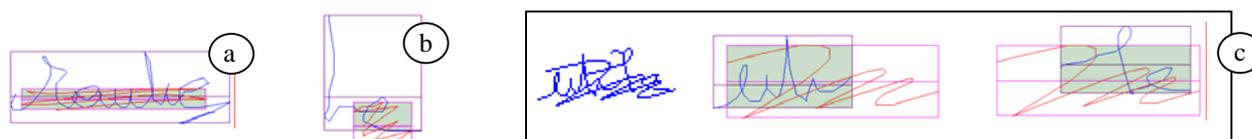


FIG. 2 - étendue d'action de la rature.

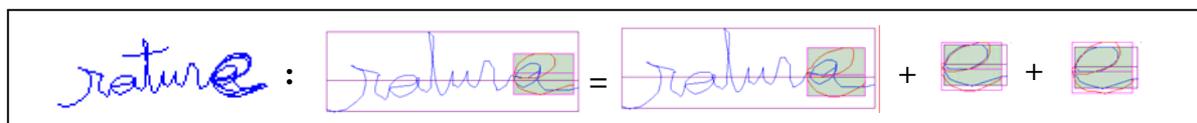


FIG. 3 - Sur-écriture et surcharges.

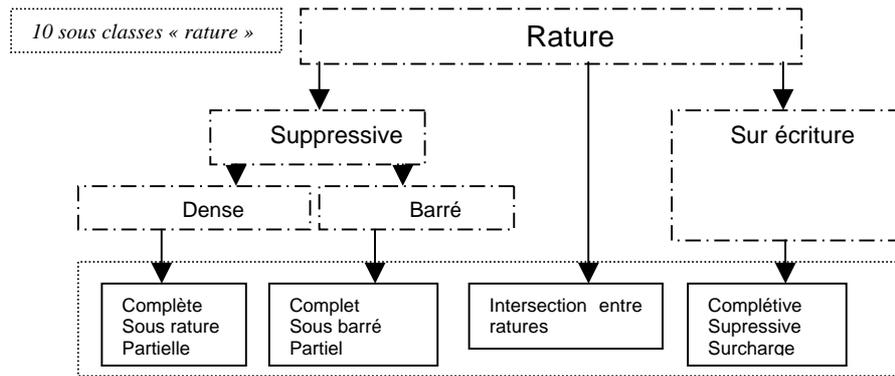


FIG. 4 – Hiérarchisation des sous-classes.

Ratures		Non ratures	
Ratures Suppressives Denses Complètes	10	Barres Horizontales	1669
Barrés Suppressifs Complètes	2	Barres Verticales	190
Sous Ratures Suppressives Denses	141	Barres Diverses	40
Sous Barrés Suppressifs	34	Croix	1017
Ratures Suppressives Denses Partielles	7	Signatures	84
		Croquis	127
Intersection entre Ratures	3	Connexion entre caractères	1336
Sur-écritures Suppressives	142		
Sur-écritures Complétives	37	Débordement entre Lignes	20
Surcharges	263	Encadrés, Entourés, Soulignés	35
		Intersection Point-Texte	106
<b>Total</b>	<b>609</b>	<b>Total</b>	<b>4529</b>
<b>Total : 5138</b>			

FIG. 5 – Effectif des sous-classes.

### 3 Méthodologie

Notre méthode est fondée sur l'hypothèse qu'une rature naît du chevauchement de plusieurs strokes. Au cours d'une première phase, les strokes isolées sont donc écartées des données. Ne restent que les strokes présentant une ou plusieurs connexions avec leurs voisines, qui sont assemblées par couples. Ce sont ces couples de strokes que le système doit classer, selon qu'il estime que ceux-ci participent ou non à une rature. Nous décrivons maintenant les différentes étapes de la reconnaissance des couples de strokes.

Filtrage. Une stroke est définie comme étant la portion de tracé continue recueillie entre le posé et le levé du stylo. Si l'on part du principe qu'il faut au moins deux strokes s'entrecoupant pour obtenir une rature, il est inutile de chercher des ratures parmi les strokes

totalemment isolées de leurs voisines géographiques. On filtre donc l'ensemble des strokes pour diminuer le nombre de candidats à la rature, en supprimant ces strokes isolées (fig. 6).

Le filtrage est réalisé selon deux critères. Le premier, peu coûteux en temps de calcul, consiste à rechercher des chevauchements entre les boîtes englobant les strokes. Les intersections entre strokes sont ensuite recherchées dans cette pré-sélection. Au cours de cette opération, les strokes présentant entre elles au moins une intersection sont assemblées par couples (fig. 7). A l'issue de cette phase de filtrage, la page est représentée sous forme d'une séquence de couples de strokes. Au sein de chaque couple, c'est la deuxième stroke, la plus tardive, qui est potentiellement réparatrice.

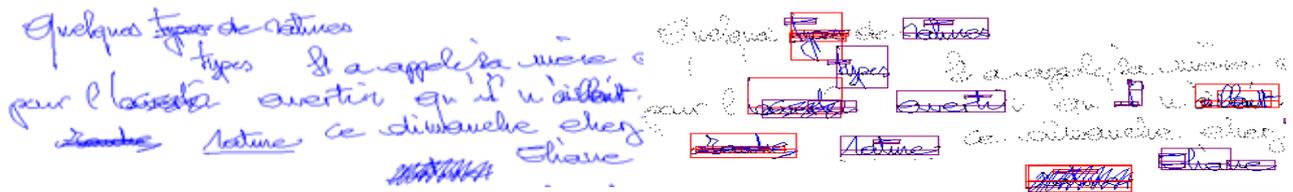


FIG. 6 – Filtrage.

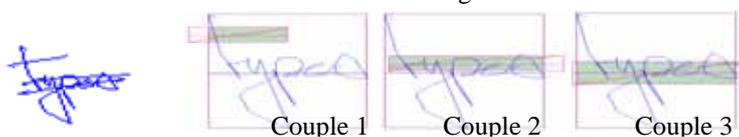


FIG. 7 – Décomposition en couples de strokes.

Extraction de caractéristiques : Onze caractéristiques sont extraites pour chaque couple de strokes. Elles rassemblent des mesures relatives aux intersections, aux positions relatives des deux strokes, ainsi qu'à la quantité de strokes entrant en contact avec la stroke potentiellement correctrice.

Le positionnement relatif des deux strokes est quantifié grâce à des mesures de distances entre les barycentres de deux strokes, ainsi que des mesures de recouvrement surfacique, et horizontal. Les ratures ont des chances de produire une distribution singulière d'intersections entre les strokes qu'elles mettent en jeu. Cinq autres mesures qualifient donc ces intersections : leur quantité, leur densité, et leur répartition. Enfin, deux dernières caractéristiques complètent le vecteur, qui sont la longueur totale déployée de la seconde stroke, et le nombre de strokes avec lesquelles celle-ci entre en contact.

Décision : La classification des couples de strokes est confiée à un perceptron multi couches, affectant à chaque couple de strokes ses scores d'appartenance aux 19 sous classes. Lors de l'apprentissage, l'application d'une matrice de coût minimise les erreurs de classification intervenant parmi les sous classes d'un même groupe. Le perceptron produit une classification correcte parmi les 19 classes dans 75,01% des cas, en validation croisée. Les scores globaux des classes « rature » et « écriture standard » sont obtenus en sommant les scores de leurs sous classes associées. Le score de la

Pour notre test, 528 pages de documents réels ont été examinées. 117 de ces pages comportent une ou plusieurs ratures, recensées au nombre de 444. Dans le cadre du traitement de masse de documents avant leur soumission à une reconnaissance du manuscrit, les pages identifiées comme ne contenant pas de ratures sont transmises telles quelles, tandis que les autres sont temporairement écartées afin d'être soumises au préalable à un traitement approprié. Il est donc utile de considérer le taux d'erreur au niveau page (fig. 8(b)) en plus des résultats de reconnaissance au niveau rature (fig. 8(a)). Lors de l'analyse des résultats, l'identification des ratures a requis un certain niveau d'interprétation, afin de définir fermement les événements correctifs présents dans chaque page, avant la détection. Idéalement, il faudrait que le système ne localise que les corrections qui poseront problème lors de la reconnaissance, et ce, indépendamment du moteur de reconnaissance. La meilleure des solutions est alors d'agir sur le seuil de reconnaissance selon le comportement observé du moteur de reconnaissance, jusqu'à obtenir le meilleur compromis.

Plusieurs tendances se dégagent dans les ratures collectées au fil des pages. Certaines ratures résultent d'une démarche réfléchie du scripteur. La plupart de ces cas ne posent pas problème lors de la détection tant la saisie est évidente, et sont détectés sans problème avec le seuil minimal de reconnaissance. D'autres observations manifestent davantage une correction réflexe, réalisée au fil de la saisie, ou simplement d'un style d'écriture saccadé ou hésitant, où les caractères sont régulièrement repris et surchargés. Dans ce

classe « écriture standard » est simplement comparé à un seuil pour fournir la décision finale, sur le mode suivant : « Si scoreécriture\_std < seuil alors décision = rature sinon décision = écriture standard ». Ce seuil, paramétrable, permet alors d'agir sur la tolérance du système. Après qu'ils ont fourni la décision globale, les scores des sous classes sont réutilisés pour scinder le groupe « rature » en deux sous groupes : « suppression » et « sur-écriture », offrant ainsi la possibilité de différencier les cas de ratures suppressives des cas de surécritures.

Bien que les données manipulées lors de la classification sont des couples de strokes, les résultats de classification à ce niveau reflèteraient mal le comportement du système. Dans les faits, une erreur de classification sur un couple de strokes n'implique pas forcément d'erreur de détection sur un événement rature. A l'issue du traitement d'une page, les couples de strokes raturant sont donc regroupés en entités « rature ». Il s'agit de reconstituer les ratures en regroupant les couples de strokes raturants participant au même événement, avant d'introduire ces nouvelles informations dans le code source de la page concernée. Un score global est alors affecté à chaque rature, et son rôle de rature suppressive ou de sur-écriture peut être précisé si la différence de score est significative. Notons qu'il est préférable de minimiser les classements erronés en tant que « rature suppressive », puisqu'ils pourront aboutir à un effacement des strokes concernées. contexte, les corrections sont parfois trop épurées pour se distinguer correctement de certaines configurations d'écriture standard. Pour des valeurs trop faibles du seuil de reconnaissance, le système montre un comportement aléatoire sur ce type de correction. En fonction du comportement du moteur de reconnaissance, s'il est nécessaire de localiser ces corrections, il faudra alors augmenter le seuil de reconnaissance, au prix d'une quantité de fausse alarme accrue.

## 4 Conclusion

Nous avons proposé une méthodologie dédiée à la minimisation des erreurs de reconnaissance dans des documents issus d'une numérisation en ligne, et dont le contexte de saisie favorise la présence de ratures et corrections. Notre système est capable de détecter et localiser les groupes de tracés problématiques en se prononçant sur leur nature correctrice. La démarche a permis de mettre en évidence qu'une frontière stricte entre écriture standard et rature est impossible à définir. Idéalement, le système doit anticiper toutes les corrections qui déclencheront un comportement erroné du moteur de reconnaissance, et ignorer celles qui seront prises en charge sans problème. En conséquence, notre choix a été de doter le système d'une certaine souplesse, grâce à un fractionnement des deux classes de notre problème, et à l'introduction d'un critère ajustable de tolérance.

L'approche présentée est un système de prétraitement construit comme un système de reconnaissance à part entière. La méthode, basée sur

une extraction de caractéristiques simples, tire son expertise d'une base d'exemples relativement limitée. Les résultats de détection de cette première approche sont très honorables. A titre de perspective, il pourrait être bénéfique d'améliorer la catégorisation des cas de rature en analysant les erreurs de classification au sein des sous classes. Une nouvelle campagne d'étiquetage

pourra être menée, dans de meilleures conditions. Le but étant, à terme, d'obtenir un système doté d'un niveau supérieur de décision lui permettant non seulement de localiser, mais également d'affranchir le document de ces tracés problématiques, en conservant son contenu effectif.

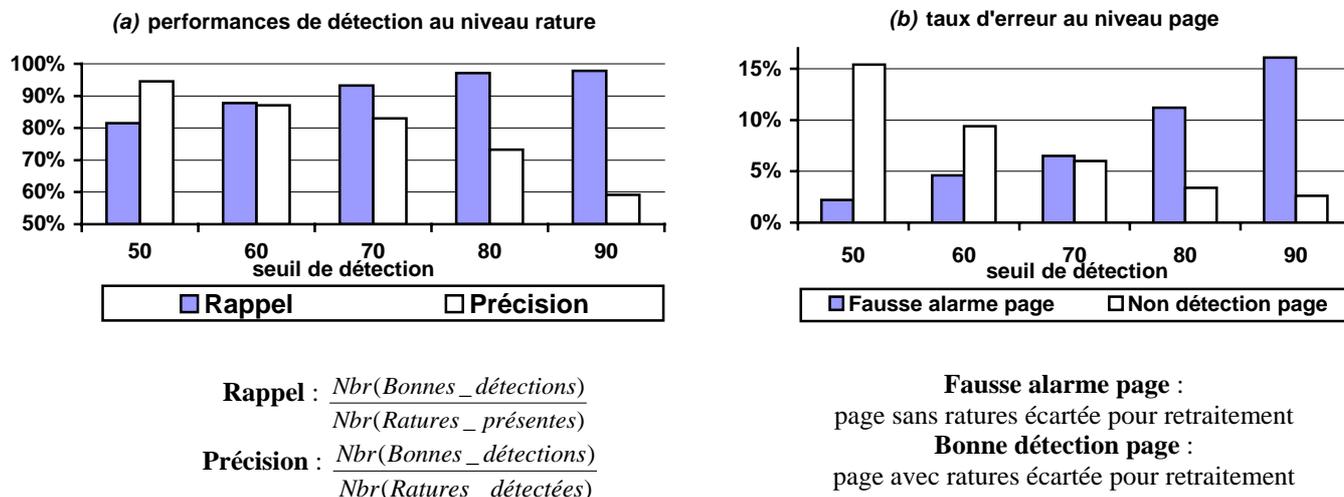


FIG. 9 – résultats de reconnaissance au sens des ratures (a) et taux d'erreur au niveau page (b).

## Références

- [ALI 97] F. Alimoglu, E. Alpaydin, « Combining multiple representations and classifiers for pen-based handwritten digit recognition », Proceedings of the Fourth International Conference on Document Analysis and Recognition (ICDAR 97), Ulm, Germany, 1997, pp 637-640.
- [ANQ 02] E. Anquetil, H. Bouchereau. « Integration of an on-line handwriting recognition system in a smart phone device ». Sixteenth IAPR International Conference on Pattern Recognition (ICPR 2002), Quebec City, Canada, 2002, pp 192-195.
- [TAP 90] C.C. Tappert, C.Y. Suen, T. Wakahara. « The state of the art in on-line handwriting recognition ». IEEE Trans. Pattern Analysis and Machine Intelligence, 1990, vol. 12, pp 787-808.
- [JAE 00] S. Jaeger, S. Manke, A Waibel, « NPEN++ : an on line handwriting recognition system » Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition, Amsterdam, Netherlands, 2002, pp 249-260.
- [HU 94] J. Hu, M. K. Brown, W. Turin. « Handwriting recognition with hidden Markov models and grammatical constraints ». Proceedings of the International Workshop on Frontiers in Handwriting Recognition, (IWFHR), 1994, Taipei, Taiwan, pp 195-205.
- [SEN 99] Giovanni Seni, John Seybold: « Diacritical processing for unconstrained online handwriting recognition using a forward search ». International Journal on Document Analysis and Recognition, (IJ DAR), 1999, vol. 2, nbr. 1, pp 24-29.
- [HU 96] J. Hu, M. K. Brown. « On-line handwriting recognition with constrained n-best decoding ». Proceedings of the International Conference on Pattern Recognition, (ICPR), 1996, Vienna, Austria, vol. 3, pp 23-27.
- [MAN 94] S. Manke, M. Finke, A. Waibel, « Combining Bitmaps with Dynamic Writing Information for On-Line Handwriting Recognition », Proceedings of the International Conference on Pattern Recognition, (ICPR), 1994, Jerusalem, pp 596-598.
- [PLA 00] R. Plamondon, S.N. Srihari. « On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey ». IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, vol. 22, nbr. 1, pp 63-84.
- [HUR 98] W. Hürst, J. Yang, A. Waibel. « Error Repair in Human Handwriting - An Intelligent User Interface for Automatic On-Line Handwriting Recognition », IEEE International Joint Symposia on Intelligence and Systems, 1998, pp 389-395.