



HAL
open science

Synthèse de caractères manuscrits en-ligne pour la reconnaissance de l'écriture

Harold Mouchère, Eric Anquetil

► **To cite this version:**

Harold Mouchère, Eric Anquetil. Synthèse de caractères manuscrits en-ligne pour la reconnaissance de l'écriture. International Francophone sur l'Écrit et le Document, 2006, Fribourg, Suisse. pp.187-192. hal-00113590

HAL Id: hal-00113590

<https://hal.science/hal-00113590>

Submitted on 13 Nov 2006

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.

Synthèse de caractères manuscrits en-ligne pour la reconnaissance de l'écriture

Harold Mouchère¹ – Éric Anquetil²

¹ IRISA / INSA de Rennes / CNRS

² IRISA / INSA de Rennes
Campus Universitaire de Beaulieu
Avenue du Général Leclerc
35042 Rennes, France

{Harold.Mouchere, Eric.Anquetil}@irisa.fr

Résumé : *Dans de précédents travaux nous avons proposé une stratégie qui permet d'adapter automatiquement un système de reconnaissance de caractères isolés en-ligne au scripteur courant. La difficulté pour améliorer encore la vitesse d'adaptation du système est que très peu de données sont disponibles au début de l'utilisation. Nous explorons donc dans cet article les possibilités d'utiliser la synthèse de caractères en-ligne par déformation d'exemples pour augmenter la quantité disponible de caractères représentatifs du style d'écriture de l'utilisateur. Nous proposons pour cela l'utilisation de déformations dédiées au tracé en-ligne qui permettent des résultats meilleurs que les déformations classiquement utilisées dans le domaine hors-ligne.*

Mots-clés : apprentissage, génération de données, reconnaissance de caractères en-ligne.

1 Introduction

Avec l'émergence des assistants personnels numériques (PDA) et des téléphones mobiles de nouvelle génération (smartphone) utilisant des interfaces orientées stylo, les performances des systèmes de reconnaissance de caractères en-ligne se doivent d'être de plus en plus hautes. Une des manières d'augmenter la précision de ces systèmes de reconnaissance est d'adapter le système au style d'écriture de l'utilisateur courant. Cette adaptation doit être rapide et transparente pour l'utilisateur. La difficulté est d'apprendre un nouveau style d'écriture avec très peu de données. L'objectif de ce papier est donc d'éviter à l'utilisateur de devoir saisir plusieurs dizaines de fois chaque caractère pour réaliser cet apprentissage.

Dans de précédents travaux [MOU 05] nous avons déjà exploré la possibilité d'adapter automatiquement un système de reconnaissance omni-scripteur appris sur des centaines de styles différents pour l'optimiser à la reconnaissance d'un seul style d'écriture. Le système de reconnaissance est basé sur des prototypes flous chacun représentant un modèle de caractères. L'adaptation se faisant au fur et à mesure de l'utilisation, les prototypes flous sont déplacés dans l'espace des caractéristiques et déformés pour mieux représenter les données spécifiques de l'utilisateur. Les performances de notre mécanisme d'adaptation permettent de passer de 88% de re-

connaissance avant adaptation à 98% en moyenne.

La synthèse de données en-ligne va permettre d'améliorer cette phase d'adaptation selon deux axes. Le premier axe est d'accélérer la vitesse d'adaptation. En effet pour atteindre 70% de l'adaptation totale, notre système a besoin d'avoir utilisé 10 fois chaque classe de caractères, soit à peu près 40 mots saisis par l'utilisateur. La synthèse de données pourra permettre de réduire ce nombre et donc d'accélérer la vitesse d'adaptation. Le deuxième axe concerne la création de nouveaux prototypes de caractères. En effet, lorsque l'utilisateur a un style d'écriture radicalement différent de ceux appris par notre classifieur, il sera plus avantageux de créer un nouveau prototype que de modifier les prototypes existants. Or cette création nécessite une assez grande quantité d'exemples. C'est à cause de cet inconvénient que cette voie n'a pas encore été explorée dans les travaux précédents et la synthèse de caractères en-ligne pourra permettre de générer ces exemples nécessaires.

L'objectif de ce papier est donc d'étudier les possibilités de synthèse automatique de tracés manuscrits (ici plus particulièrement des caractères) en utilisant les particularités de l'écriture en-ligne. La qualité de cette génération de caractères sera évaluée en mesurant les performances monoscripteur d'un classifieur appris à l'aide de caractères synthétisés à partir de très peu d'exemples de son style d'écriture (de 1 à 10 caractères).

Dans un premier temps la section 2 fait un bilan des possibilités existantes de synthèse de caractères. Ensuite la section 3 présente des stratégies de déformations classiques utilisées dans le domaine hors-ligne. Nous proposons dans la section 4 d'utiliser des déformations dédiées aux tracés en-ligne pour obtenir de nouvelles possibilités de déformations des caractères. La section 5 explique comment nous utilisons ces déformations pour synthétiser de nouveaux caractères. Enfin la section 6 présente comment nous avons choisi la meilleure stratégie de synthèse.

2 État de l'art sur la synthèse de caractères

La synthèse de caractères est un problème qui a été souvent abordé dans le domaine de la reconnaissance de l'écriture.

ture hors-ligne. La synthèse se fait à partir d'un ou plusieurs exemples qui sont déformés par différentes opérations. En hors-ligne le classifieur travaille sur une image du caractère à reconnaître, donc les déformations utilisées sont des déformations d'images. Pour la reconnaissance de caractères hors-ligne les auteurs de [CAN 02] utilisent comme déformations l'inclinaison, l'étirement et la variation de l'épaisseur du trait. D'autres types de déformations peuvent être produites. Dans [HA 97] les auteurs supposent que les caractères à reconnaître ont subi au préalable une déformation. Ils étudient donc différentes déformations d'images pour essayer de retrouver l'image avant déformation afin de faciliter la reconnaissance. On peut utiliser ces travaux pour définir, en plus des deux précédentes, deux nouvelles déformations : la mise en perspective et la rotation. Dans [SIM 03] les images sont déformées par distortions élastiques pour augmenter la taille de la base d'apprentissage d'un réseau de neurones. Ceci a pour effet de créer des "vagues" sur l'image. La synthèse de données est aussi utilisée pour augmenter les bases d'apprentissage de systèmes avec beaucoup de paramètres comme les MMC (Modèles de Markov Cachés). Ainsi dans [HEL 03, VAR 03] des lignes de textes sont déformées en utilisant les déformations classiques différemment tout au long de la ligne de texte. De plus dans ce contexte la ligne de base de l'écriture peut être aussi modifiée.

Il y a peu, à la connaissance des auteurs, de travaux portant sur la synthèse de données spécifiques aux caractères manuscrits en-ligne. Néanmoins il existe des travaux connexes. Dans le domaine en-ligne, le tracé est considéré comme une fonction paramétrique. Un tracé en-ligne est donc une suite de points $p(t)$ à intervalles de temps réguliers. Chaque point est une position où le crayon du scripteur s'est trouvé posé. Les travaux sur la synthèse de l'écriture [PLA 98] expliquent que des modifications de la fonction paramétrique entraînent des variations dans le style de l'écriture. Dans [SCH 96] des transformations du signal en-ligne sont étudiées pour élaborer une distance qui tient compte de ces possibles transformations. Cette distance est alors utilisée dans un classifieur qui devient robuste à ces transformations. Dans [VAR 05] les travaux de [PLA 98] sont utilisés pour augmenter la taille de la base d'apprentissage d'un MMC qui reconnaît des phrases hors-ligne. Les auteurs utilisent un modèle unique de caractères en-ligne qu'ils déforment et concatènent pour générer ensuite des images de phrases à partir d'un texte ASCII. Pour l'instant dans ces travaux un seul modèle de caractère obtenu manuellement par un expert est utilisé. Cette dernière approche est assez proches de la notre mais n'est pas directement applicable à nos travaux. En effet dans le cadre de l'adaptation en-ligne au scripteur, il faut que la synthèse de caractères soit automatique et doit se faire à partir des caractères saisis par l'utilisateur.

Toutes les déformations présentées dans cette section ne sont pas utilisables dans le contexte de l'écriture en-ligne. Par exemple la variation de l'épaisseur du tracé n'a pas de sens ici. De plus certains classifieurs effectuent des normalisations sur les caractères avant de les traiter pour rendre plus robuste la classification, donc les déformations correspondantes ne permettront pas d'apporter de nouvelles données. Nous étudierons dans cet article des déformations ins-

pirées du domaine hors-ligne ainsi que des déformations spécifiques à l'écriture en-ligne. Les déformations du type hors-ligne, l'étirement et l'inclinaison, sont décrites dans la section 3. La section 4 présentera des déformations spécifiques au domaine en-ligne.

3 Déformations de type hors-ligne

Les déformations issues de la génération de données hors-ligne sont basées sur la déformation d'image. Nous présentons dans cette section, après une brève description du tracé en-ligne disponible, les plus utilisées : l'étirement et l'inclinaison de l'image.

Même si nous étudions des déformations du type hors-ligne, le signal disponible est en-ligne. Nous ne considérons pas les mouvements du crayon lorsqu'il ne touche pas la surface sensible. Le tracé n'est pas modifié après la saisie, à une translation près. Chaque point $p(t)$ est défini par ses coordonnées $x(t)$ et $y(t)$. De plus pour plus de simplicité, nous supposons que :

$$\min_t(x(t)) = 0, \quad (1)$$

$$\min_t(y(t)) = 0. \quad (2)$$

3.1 Étirements

Une image peut être étirée suivant l'axe x ou l'axe y . L'opération d'étirement dépend donc de deux paramètres α_x et α_y respectivement le coefficient de déformation suivant l'axe x et le coefficient de déformation suivant l'axe y . Pour un tracé en ligne, les coordonnées des points sont modifiées suivant les équations :

$$x'(t) = \alpha_x x(t), \quad (3)$$

$$y'(t) = \alpha_y y(t). \quad (4)$$

La figure Fig. 1 donne un exemple de caractère déformé par étirement suivant les axes x et y .

3.2 Inclinaison

L'inclinaison d'une image est différente de la rotation d'une image. En effet cette déformation se rapproche plus de l'écriture "penchée". L'inclinaison dépend d'un paramètre α_i . Comme le montrent les équations 5 et 6 la coordonnée en y ne change pas mais la variation de $x(t)$ dépend de la valeur de $y(t)$:

$$x'(t) = x(t) + \alpha_i * y(t), \quad (5)$$

$$y'(t) = y(t). \quad (6)$$

La figure Fig. 2 montre une lettre déformée par inclinaison dans les deux directions (vers la gauche et vers la droite). On peut constater que la forme de la lettre ne change pas mais que les lignes verticale sont inclinées.

4 Déformations de type en-ligne

Les déformations dites en-ligne utilisent des informations sur le tracé à déformer qui ne sont pas disponibles dans l'image correspondant à ce même tracé. En effet le tracé manuscrit n'est pas ré-échantillonné. Si les points sont plus éloignés à certains moments c'est que le scripteur est allé plus

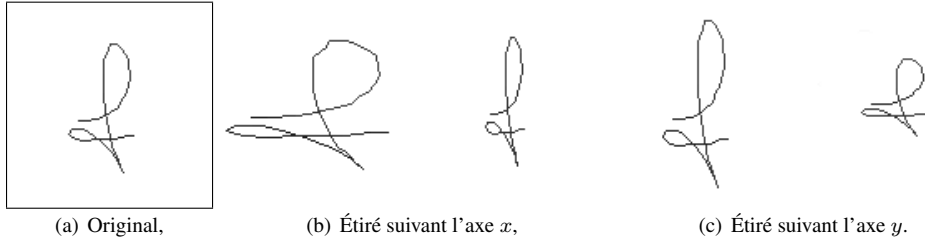


FIG. 1 – Exemple de caractère déformé par étirement.

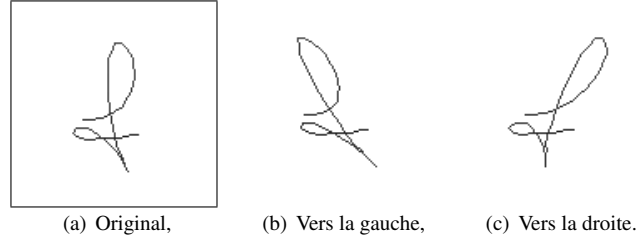


FIG. 2 – Exemple de caractère déformé par inclinaison.

vite. Ainsi nous conservons la cinétique du tracé. L'ordre des points du tracé et leur cinétique peuvent être exploités dans les stratégies de déformations. Nous présentons dans cette section des déformations permettant de modifier la vitesse du tracé suivant sa direction ou de modifier la courbure du tracé.

D'autres déformations non présentées dans ces travaux sont possibles. Par exemple avec des caractères composés de plusieurs tracés il est possible de faire varier la position relative des deux tracés. Il est aussi possible de déphaser les fonctions $x(t)$ et $y(t)$ pour obtenir des courbes différentes comme expliqué dans [PLA 98].

Les déformations présentées dans cette section peuvent bien sûr être utilisées conjointement avec les déformations du type hors-ligne, comme c'est le cas dans nos expérimentations.

4.1 Modification de la vitesse

L'objectif de cette déformation est de modifier la longueur des traits verticaux et horizontaux, ce qui devrait permettre de fermer ou d'ouvrir des lettres comme le e ou le s et de déplacer certaines parties du caractère par rapport à d'autres.

Une des possibilités offerte par la connaissance de la cinétique du tracé est de pouvoir modifier la vitesse de ce tracé. Si la vitesse est modifiée de façon homogène sur l'ensemble de la suite des points il en résultera une simple homothétie. La vitesse du tracé à l'instant t est donnée par le vecteur séparant les deux points $p(t)$ et $p(t+1)$. Nous avons donc choisi de modifier la taille des vecteurs $\vec{V}_{(t)} = (x_{(t+1)} - x_{(t)}, y_{(t+1)} - y_{(t)})$ suivant leur direction. Ainsi les vecteurs dont les directions sont proches des axes principaux pourront augmenter ou diminuer selon un paramètre α_v . Les vecteurs seront considérés comme proches d'un axe principal s'ils forment un angle inférieur à $\frac{\pi}{8}$ avec un des axes. Les équations 7 à 9 donnent les coordonnées des points du tracé après cette déformation :

$$x'_{(t)} = x'_{(t-1)} + \beta * (x_{(t)} - x_{(t-1)}), \quad (7)$$

$$y'_{(t)} = y'_{(t-1)} + \beta * (y_{(t)} - y_{(t-1)}) \quad (8)$$

$$\beta = \begin{cases} 1 & \text{si } \arg(\vec{V}_{(t-1)}) \left[\frac{\pi}{2} \right] \in \left[\frac{\pi}{8}, \frac{3\pi}{8} \right] \\ \alpha_v \sin \theta & \text{sinon} \end{cases} \quad (9)$$

La figure Fig. 3 donne un exemple du type de déformation obtenue en modifiant ainsi la vitesse du tracé. On peut constater sur le f déformé que la longueur du segment vertical du tracé change suivant la déformation ce qui a pour effet de rapprocher ou d'éloigner les deux boucles du f. De plus, le caractère s peut être plus ouvert ou même fermé suivant la déformation effectuée. Ces nouveaux caractères n'auraient pas pu être obtenus par des déformations du type hors-ligne.

4.2 Modification de la courbure

La déformation précédente permet d'agir surtout sur les lignes droites horizontales et verticales du tracé. Nous proposons donc ici une déformation qui agit sur les courbes du caractère à modifier et non sur les parties droites. Ainsi la structure du tracé est conservée tout en étant déformé.

La modification de la courbure du tracé se fait en fonction d'un paramètre α_c . La courbure du tracé à l'instant t correspond à l'angle $\hat{\theta}$ formé par le point $p(t)$ avec le point précédent et le point suivant (équation 10). Pour ne pas complètement modifier le caractère, il y a certaines parties du tracé qu'il ne faut pas modifier, par exemple les lignes droites et les points de rebroussement du tracé. Ainsi la déformation sera maximale pour des angles proches de $\frac{\pi}{2}$ et nulle pour les angles 0 et π . Les équations 10 à 12 donnent la position des points $p'_{(t)}$ après déformation :

$$\hat{\theta}_{(t-1)} = \widehat{(p_{(t-2)}, p_{(t-1)}, p_{(t)})} \in] -\pi, \pi], \quad (10)$$

$$p'_{(t)} = p^* \text{ tel que } \widehat{(p'_{(t-2)}, p'_{(t-1)}, p^*)} = \hat{\theta}_{(t-1)} - \beta, \quad (11)$$

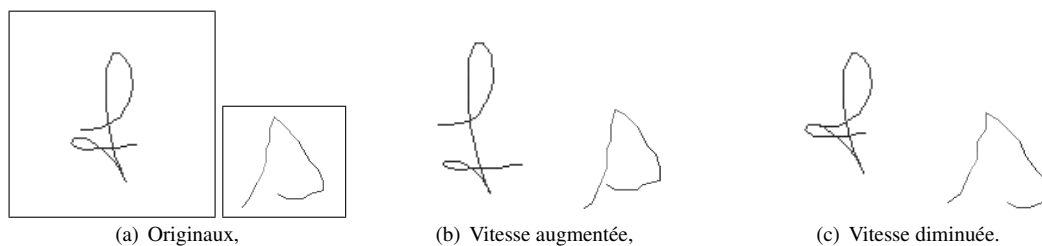


FIG. 3 – Exemple de caractères déformés par modification de la vitesse.

$$\beta = \alpha_c * 4 * \frac{|\hat{\theta}_{(t-1)}|}{\pi} * \left(1 - \frac{|\hat{\theta}_{(t-1)}|}{\pi}\right). \quad (12)$$

La figure Fig. 4 donne deux exemples de caractères déformés par la modification de la courbure. Ainsi on voit que les boucles du f ont changé de forme et la boucle du q se ferme ou s'ouvre suivant la déformation. Ces déformations n'auraient pas été possibles avec les déformations hors-ligne ni avec la variation de vitesse précédemment présentée. En effet la boucle de la lettre q n'ayant pas comme celles du f de longues parties horizontales ou verticales, la variation de la vitesse ne l'aurait presque pas déformé.

5 Synthèse de caractères

Il s'agit de générer plusieurs variations d'un même caractère à partir d'un seul exemple de ce caractère en-ligne. Pour cela on applique sur l'exemple disponible une ou plusieurs des déformations présentées ci-dessus. À chaque nouvelle génération il faut choisir des paramètres de déformation différents pour obtenir une certaine variabilité dans les données générées. Mais il ne faut pas non plus utiliser des paramètres engendrant des déformations trop importantes pour ne pas risquer de synthétiser un caractère qui ne ressemble plus à celui d'origine. En effet il s'agit de tourner autour du style du scripteur mais non de générer un nouveau style. Pour chaque paramètre α_x , α_y , α_i , α_v et α_c une borne minimum et une borne maximum est fixée pour limiter les déformations générées. Ces bornes sont fixées empiriquement en visualisant pour chacune d'elles les déformations obtenues pour plusieurs classes de caractères. Une fois fixées, les bornes restent les mêmes tout au long des expérimentations.

6 Expérimentations

La finalité de ces expérimentations est d'étudier la possibilité d'utiliser la synthèse de caractères en-ligne pour améliorer la rapidité et les performances de notre système d'adaptation automatique au scripteur [MOU 05]. Le but n'est donc pas d'obtenir un classifieur mono-scripteur extrêmement performant mais de trouver quelles sont les déformations les plus efficaces pour synthétiser des caractères représentant bien le style d'écriture du scripteur avec très peu d'exemples. Pour cela, nous mesurons et comparons les capacités des déformations présentées à synthétiser de nouvelles données d'apprentissage dans un contexte de reconnaissance mono-scripteur à partir de très peu de d'exemples de caractères. De plus le classifieur utilisé est donc le même

que dans nos travaux précédents. Il s'agit d'un Système d'Inférence Floue utilisant dans chaque règle un prototype flou. Ce prototype flou est défini dans l'espace des caractéristiques par un centre et une matrice de covariance. L'apprentissage du système est basé sur la classification non supervisée possibiliste et nécessite donc une certaine quantité de donnée pour être efficace, d'où l'intérêt de la synthèse de donnée. Une description détaillée du classifieur est disponible dans [MOU 05].

La section suivante décrit en détails le protocole de ces expérimentations puis la section 6.2 expose et commente les résultats.

6.1 Protocole

Nous avons utilisé les saisies de 12 scripteurs différents, chacun ayant entré 40 fois chaque caractère sur un PDA. Chaque saisie est séparée en une base de 30 fois chaque caractère qui servira pour l'apprentissage du classifieur et une base de 10 fois chaque caractère pour le test. Les expérimentations se déroulent en deux étapes.

D'abord on utilise toute la base d'apprentissage de chaque scripteur pour apprendre 12 classifieurs spécialisés. Le taux de reconnaissance est alors évalué sur la base de test de chaque scripteur respectivement. Ce taux de reconnaissance servira de référence pour comparer le principe de la synthèse de caractères à une technique classique d'apprentissage d'un système mono-scripteur.

Ensuite un sous ensemble volontairement petit des exemples d'apprentissage (1, 2, 3, 5 ou 10 caractères par classe) est tiré au sort et ces exemples sont ensuite déformés avec la ou les opérations choisies jusqu'à obtenir 360 caractères de chaque classe. Le classifieur est alors appris en utilisant les données tirées au sort complétées avec les caractères synthétisés. L'opération est répétée quatre fois (pour avoir des tirages différents) pour chaque scripteur. Les résultats présentés sont les moyennes des 4 tirages pour les 12 scripteurs.

6.2 Résultats

La première étape des expérimentations a permis de constater qu'en apprenant le classifieur sur toutes les données de la base d'apprentissage, la moyenne obtenue en reconnaissance mono-scripteur est en moyenne de 79,36% sur les bases de tests de chaque scripteur. Ce faible taux est dû à un sur-apprentissage du classifieur, en effet les résultats sur les bases d'apprentissage sont proches de 99%. Les données utilisées en apprentissage (30 par classe de caractères) de chaque utilisateur ne se suffisent donc pas à elles seules pour

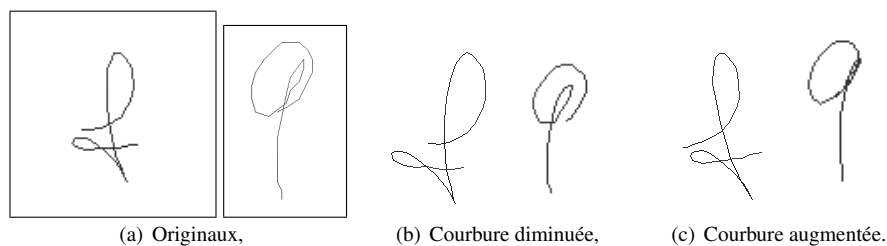


FIG. 4 – Exemple de caractères déformés par modification de la courbure.

Stratégie	Mini. (%)	Moy. (%)	Maxi. (%)
Référence	72,3	79,4	89,2
Hors-ligne	67,7	75,5	83,9
Vitesse	82,2	86,3	92,3
Courbure	72,8	80,9	86,6
Tous	80,0	86,9	93,7

TAB. 1 – Valeur minimum, moyenne et maximum des taux de reconnaissance moyens de chaque scripteur pour les différentes stratégies avec seulement 3 caractères d'origine utilisés.

apprendre un classifieur mono-scripteur performant.

La figure Fig. 5 compare les résultats obtenus pour les différentes déformations suivant le nombre de données utilisées pour la génération de la base d'apprentissage. L'utilisation de un ou deux caractères par classe correspond à un cas idéal pour une exploitation dans un cadre d'adaptation automatique car elle serait rapide et par conséquent très peu coûteuse pour l'utilisateur. Par contre l'utilisation de 10 caractères par classe (soit environ 50 mots) commence à être plus fastidieux pour l'utilisateur au niveau du temps de saisie. Nous comparons de plus dans ce graphique les déformations du type hors-ligne (section 3, notées *Hors-ligne* dans la légende) avec la combinaison de celles-ci et des déformations en-ligne (section 4, notées *Vitesse* et *Courbure*). La combinaison de toutes les déformations, notée *Tous*, est aussi testée. Elle consiste à choisir aléatoirement à chaque synthèse d'un nouveau caractère soit la déformation *Courbure* ou la déformation *Vitesse*. La ligne horizontale correspond à la *Référence* calculée à l'étape précédente.

Le tableau Tab. 1 résume les résultats en utilisant seulement 3 caractères pour la synthèse des données d'apprentissage (correspond à la ligne pointillée de la figure Fig. 5).

Il faut d'abord constater que l'utilisation de la synthèse de caractères permet de dépasser rapidement le taux de référence obtenu avec les 30 caractères d'origine de chaque classe, ceci quelque soit les déformations utilisées, avec seulement 2 à 5 caractères par classe suivant les déformations. La stratégie consistant à synthétiser des caractères en les déformant pour augmenter la quantité disponible de caractères représentatifs d'un style d'écriture pourra donc sûrement permettre d'améliorer la vitesse de l'adaptation en-ligne.

De plus les déformations du type en-ligne permettent de créer plus de diversité avec peu de caractères utilisés que les déformations du type hors-ligne seules, puisqu'avec seulement 3 caractères on atteint 86,3% et 80,9% de reconnais-

sance pour la modification de la vitesse et la modification de la courbure au lieu de 75,5% pour les déformations du type hors-ligne. Mais c'est l'utilisation de toutes les déformations conjointement qui permet d'obtenir les meilleurs résultats (68,9% avec un seul caractère et 86,9% avec trois). Les minima et maxima du tableau Tab. 1 confirment cette observation. C'est donc cette dernière stratégie qui sera utilisée dans le cadre de futurs travaux sur l'amélioration de l'adaptation en-ligne au scripteur.

7 Conclusion et travaux futurs

L'objectif de cette étude est de trouver une stratégie qui permet de s'adapter automatiquement au scripteur même si très peu de caractères sont disponibles. Nous avons montré que l'utilisation de la synthèse automatique de caractères en-ligne par déformation d'exemples permet d'obtenir une base d'apprentissage fidèle au style d'écriture du scripteur avec seulement deux ou trois exemples disponibles par classe. De plus nous avons montré que l'utilisation de déformations utilisant les spécificités du tracé en-ligne, notamment sa dynamique, permet d'atteindre de meilleurs résultats que les déformations habituellement utilisées dans le domaine hors-ligne.

Les travaux futurs consisteront donc à utiliser cette stratégie de synthèse de caractères pour augmenter la vitesse d'adaptation de notre système existant [MOU 05] ainsi que pour la création dynamique de nouveaux prototypes de caractères. De plus il sera intéressant d'explorer d'autres types de déformations dédiées aux caractères en-ligne mais aussi d'évaluer les possibilités d'utilisation de ces stratégies pour améliorer les performances des systèmes omni-scripteur appris sur des centaines de styles d'écriture.

8 Remerciements

Les auteurs souhaitent remercier Guy Lorette, Professeur à l'Université de Rennes I, pour ses précieux conseils. Ces travaux sont en partie financés par la région Bretagne.

Références

- [CAN 02] CANO J., PÉREZ-CORTES J.-C., ARLANDIS J., LLOBET R., Training Set Expansion in Handwritten Character Recognition., *Proceedings of the 9th International Workshop on Structural and Syntactic Pattern Recognition (SSPR) and 4th Statistical Pattern Recognition (SPR)*, 2002, pp. 548-556.
- [HA 97] HA T., BUNKE H., Off-line, Handwritten Numeral Recognition by Perturbation Method, *IEEE Transaction*

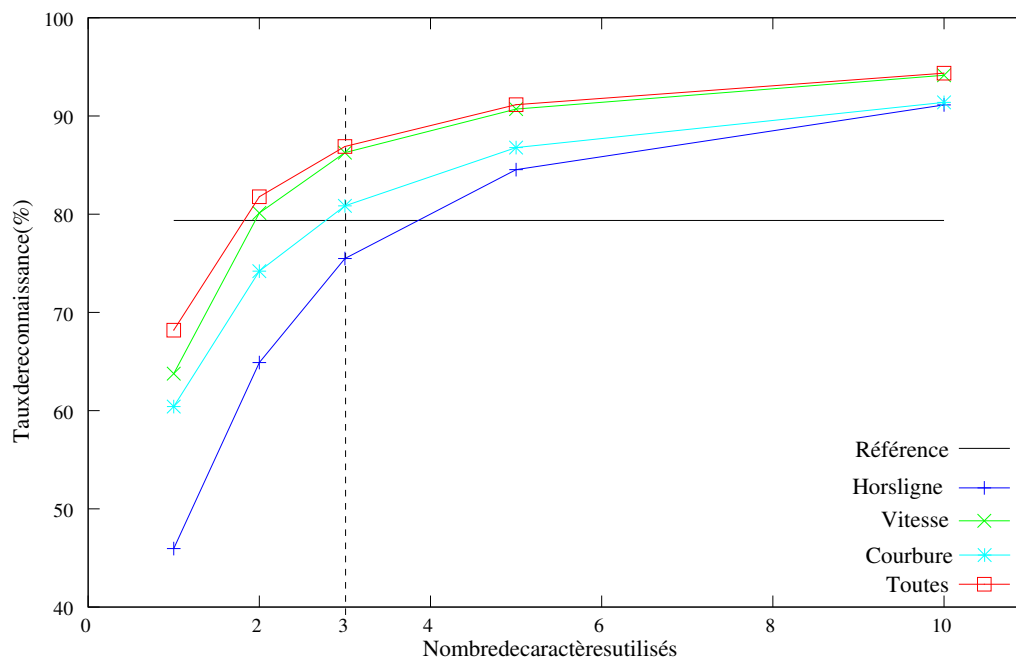


FIG. 5 – Taux de reconnaissance obtenus pour la reconnaissance mono-scripteur en utilisant 1, 2, 3, 5 ou 10 caractères par classe pour la synthèse de données d'apprentissage.

on *Pattern Analysis and Machine Intelligence*, vol. 19, n° 5, 1997, pp. 535-539.

Conference of the International Graphonomics Society (IGS), 2005, pp. 206-211.

[HEL 03] HELMER M., BUNKE H., Generation and Use of Synthetic Training Data in Cursive Handwriting Recognition, *Proceedings of 1st Iberian Conference Pattern Recognition and Image Analysis (IbPRIA)*, 2003, pp. 336-345.

[MOU 05] MOUCHÈRE H., ANQUETIL E., RAGOT N., On-line Writer Adaptation for Handwriting Recognition using Fuzzy Inference Systems, *Proceedings of the 8th International Conference on Document Analysis and Recognition (ICDAR)*, vol. 2, 2005.

[PLA 98] PLAMONDON R., GUERFALI W., The generation of handwriting with delta-lognormal synergies, *Biological Cybernetics*, vol. 78, 1998, pp. 119-132.

[SCH 96] SCHWENK H., MILGRAM M., Constraint Tangent Distance for On-Line Character Recognition, *Proceedings of the 13th International Conference on Pattern Recognition (ICPR)*, 1996.

[SIM 03] SIMARD P., STEINKRAUS D., PLATT J., Best Practice for Convolutional Neural Network Applied to Visual Analysis, *Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR)*, 2003.

[VAR 03] VARGA T., BUNKE H., Generation of Synthetic Data for an HMM-based Handwriting Recognition System, *Proceedings of 7th International Conference on Document Analysis and Recognition (ICDAR)*, 2003, pp. 618-622.

[VAR 05] VARGA T., KILCHHOFFER D., BUNKE H., Template-based Synthetic Handwriting Generation for the Training of Recognition Systems, *Proceedings of 12th*