



**HAL**  
open science

## Reconstitution du parcours du tracé manuscrit hors-ligne de caractères isolés

Laëtitia Rousseau, Eric Anquetil, Jean Camillerapp

► **To cite this version:**

Laëtitia Rousseau, Eric Anquetil, Jean Camillerapp. Reconstitution du parcours du tracé manuscrit hors-ligne de caractères isolés. Jun 2004. <sic\_00001204>

**HAL Id: sic\_00001204**

**[https://archivesic.ccsd.cnrs.fr/sic\\_00001204v1](https://archivesic.ccsd.cnrs.fr/sic_00001204v1)**

Submitted on 7 Dec 2004

**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.



HAL Authorization

# Reconstitution du parcours du tracé manuscrit hors-ligne de caractères isolés

Laëtitia Rousseau, Eric Anquetil, Jean Camillerapp

Laboratoire IRISA  
Campus Universitaire de Beaulieu  
Avenue du Général Leclerc  
35042 RENNES Cedex

laetitia.rousseau@irisa.fr, eric.anquetil@irisa.fr, jean.camillerapp@irisa.fr

**Résumé :** Les reconnaissances en-ligne et hors-ligne sont en général traitées de façon distincte, mais il peut être intéressant d'unifier leur modélisation. Notre approche cherche à reconstruire, à partir d'une image de caractère, un signal en-ligne équivalent. Plus précisément, il s'agit de retrouver l'ordre et le sens du tracé manuscrit hors-ligne du caractère, à partir notamment de connaissances a priori sur l'écriture manuscrite. L'approche présentée dans cet article s'inspire de la méthode de reconstruction de Kato et Yasuhara. Cette méthode est une approche générique qui permet de retrouver l'ordre de parcours de tracés manuscrits. L'étude de cet algorithme dans le cadre de la reconnaissance de lettres manuscrites nous a conduit à en proposer une extension, pour résoudre les cas d'erreurs et ainsi obtenir de meilleurs taux de reconstruction. Pour valider cette extension, plusieurs expérimentations ont été effectuées sur une base de caractères manuscrits possédant à la fois le signal en-ligne et hors-ligne.

**Mots-clés :** écriture manuscrite cursive en-ligne et hors-ligne, reconnaissance de caractères isolés, reconstitution du parcours du tracé

## 1 Introduction

Les reconnaissances en-ligne et hors-ligne sont en général traitées de façon distincte, notamment parce que les signaux sont de nature différente. Certaines approches visent à associer les particularités des deux reconnaissances, chaque signal pouvant apporter des informations complémentaires. Le signal en-ligne contient des informations dynamiques absentes dans les images, telles que l'ordre dans lequel les caractères ont été formés, leur sens de parcours...

Notre projet à long terme est d'unifier la modélisation de l'écriture manuscrite associée à ces deux types de signaux pour arriver à une approche de reconnaissance homogène. Pour cela, la première phase de cette étude, présentée partiellement dans cet article, s'est focalisée sur la reconstitution d'un signal en-ligne équivalent à une image de lettre. Il s'agit donc de retrouver l'ordre et le sens du tracé dans une image à partir notamment de connaissances a priori sur l'écriture.

Cette étude s'est appuyée sur un reconnaissseur en-ligne

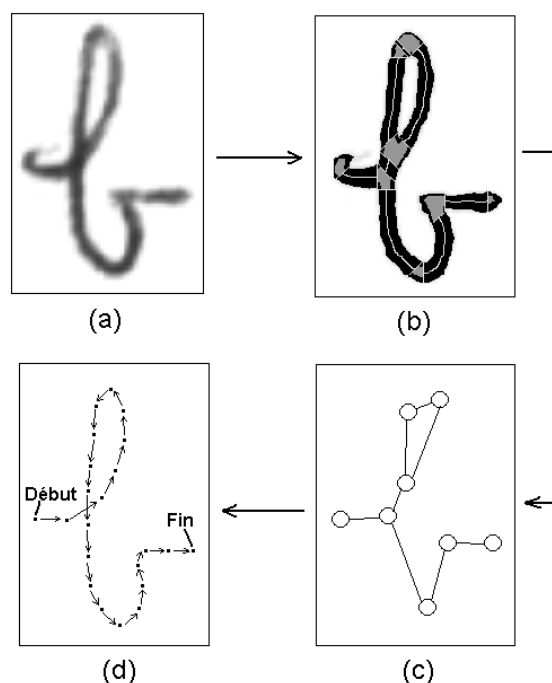


FIG. 1 – Représentations d'une lettre : (a) image en niveaux de gris (b) sous forme de zones régulières (en noir) et singulières (en gris) (c) graphe (d) suite ordonnée de points

(RESIF [ANQ 97] [ANQ 02]) et un extracteur de tracé développés par l'équipe IMADOC de l'IRISA<sup>1</sup>. L'extracteur de tracé utilise une variante de la méthode décrite dans [PET 91]. Il permet d'obtenir une représentation des lettres sous forme de zones régulières, représentant les portions longilignes du tracé, et de zones singulières, correspondant aux régions difficilement squelettisables, telles que les intersections, poser de crayon, fortes courbures... (voir figure 1(b)). Le tracé est alors représenté par un graphe (voir figure 1(c)) dans lequel les arcs représentent les zones régulières et les sommets correspondent aux zones singulières. Après ordonnancement du graphe, un signal en-ligne (voir figure 1(d)) sera généré et présenté au reconnaissseur en-ligne.

<sup>1</sup>site internet : [www.irisa.fr/imadoc](http://www.irisa.fr/imadoc)

Cet article s'inscrit dans la phase d'ordonnement du graphe. L'approche présentée cherche donc à reconstruire le sens et l'ordre du tracé manuscrit à partir du graphe (figure 1(c)) et de l'image (figure 1(b)) afin d'en déduire un signal en-ligne (séquence ordonnée de points dans le temps).

Différentes méthodes existent pour tenter de retrouver l'ordre du tracé. Certaines techniques, plutôt locales [BOC 93], cherchent à interpréter les intersections en fonction de critères de continuité des traits. D'autres, plus globales [JäG 96] [LAL 99], cherchent à minimiser la courbure globale du tracé reconstitué, ce qui peut engendrer un problème de complexité algorithmique. Plamondon et Privitera [PLA 99] étudient la continuité des contours des lettres, en utilisant des notions de psychomotricité. Doermann et Rosenfeld [DOE 95] utilisent simultanément des indices de niveau local ou global, mais leurs critères sont très liés à l'instrument d'écriture utilisé. L'algorithme de Kato et Yasuhara [KAT 00] associe les avantages des approches locales et globales, en utilisant des critères topologiques liés au graphe, tout en évitant une explosion combinatoire.

Nous nous sommes inspirés de cette dernière méthode, qui au départ est assez générique, pour la spécialiser et l'étendre à la reconstruction de tracés de lettres. La section 2 présente l'algorithme de Kato et Yasuhara et son expérimentation, la section 3 expose les problèmes de lettres mal ordonnées et propose des extensions à l'algorithme pour les résoudre. Enfin, la section 4 donne les résultats des tests effectués sur une base de caractères manuscrits possédant à la fois le signal en-ligne et hors-ligne.

## 2 L'algorithme de Kato et Yasuhara

Kato et Yasuhara ont développé un algorithme permettant de retrouver le parcours d'un tracé quelconque manuscrit. Cet algorithme cherche à reconstituer le parcours à partir de critères topologiques sur le graphe. L'algorithme repose sur plusieurs contraintes qui peuvent limiter le cadre de son utilisation.

Le tracé doit répondre aux contraintes suivantes :

- Les positions des poser et lever de crayon doivent être connues.
- Le tracé doit avoir été fait en une seule fois, c'est-à-dire qu'il n'existe qu'un poser et qu'un lever.
- Un arc peut être traversé une ou deux fois.
- Il n'existe pas de croisement de plus de deux traits, donc pas de sommet de degré<sup>2</sup> supérieur à quatre.

Le traitement est constitué de deux phases : une phase d'étiquetage, pour détecter les arcs à parcourir deux fois, puis une phase de parcours séquentiel du poser au lever, pour ordonner et orienter les arcs.

### 2.1 Phase d'étiquetage

Durant cette phase, on cherche les arcs à parcourir deux fois, que l'on étiquette comme "D-lines".

<sup>2</sup>Nous appelons degré le nombre d'arcs adjacents à un sommet

Kato et Yasuhara proposent trois sortes de D-lines :

- Looped D-line (LD-line) : ce sont des rebroussements, ils correspondent à un arc compris entre un sommet de degré un et un sommet de degré trois. (voir figure 2(a))
- Spurious D-line (SD-line) : ce sont des ponts vers une boucle, ils correspondent à un arc compris entre deux sommets de degré trois, dont l'un fait partie d'un cycle. (voir figure 2(b))
- Proper D-line (PD-line) : ces doubles lignes correspondent à un arc compris entre deux sommets de degré trois reliés par deux arcs. (voir figure 2(c))

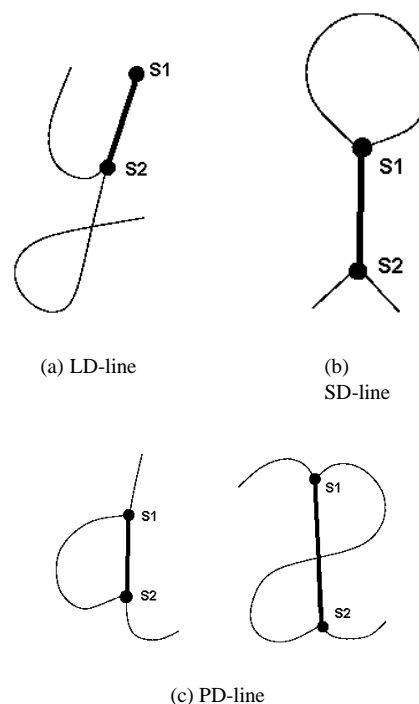


FIG. 2 – Différents types de D-line (en gras)

Un premier passage sur le graphe détecte les LD-lines. Les arcs concernés sont dupliqués.

Ensuite, on cherche à détecter les SD-lines et les PD-lines. Pour cela, on parcourt dans un ordre quelconque les sommets de degré trois non visités (n'appartenant pas à une LD-line). Si deux de ces sommets sont adjacents et que l'un possède un cycle, on a une SD-line. Si deux de ces sommets sont reliés l'un avec l'autre par deux arcs, on a une PD-line. On étiquette et on dédouble alors les arcs concernés. Pour les PD-lines, on duplique l'arc le plus court (en terme de distance géométrique) parmi les deux proposés. (voir figure 2(c))

Une PD-line peut être parcourue de deux manières différentes. Soit on la parcourt en formant une boucle (voir figure 3(a)), soit on traverse d'abord deux fois la D-line puis on parcourt la boucle (voir figure 3(b)). Deux étiquettes peuvent alors être attribuées : NORMAL-PD-line ou SPECIAL-PD-line. Ce choix d'étiquette se fait selon les angles formés au niveau des deux sommets de degré trois. On cherche à minimiser la courbure du parcours, sachant qu'il est plus natu-

rel, en écrivant, de former un rebroussement (angle de  $180^\circ$ ) plutôt qu'un virage à angle droit. Ce choix de sous-étiquette se fait durant la phase d'étiquetage.

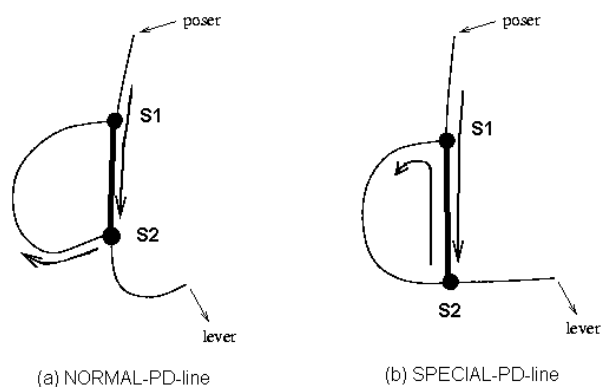


FIG. 3 – Exemple de NORMAL-PD-line et SPECIAL-PD-line.

## 2.2 Phase de parcours

Après la détection des arcs traversés deux fois, la phase de parcours commence. En partant du poser de crayon, on poursuit le tracé jusqu'au lever.

Les SD-lines sont parcourues en formant une boucle. On choisit de les parcourir comme le montre la figure 4.

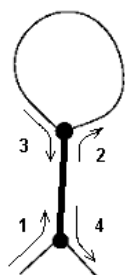


FIG. 4 – Chemin à suivre au niveau des SD-lines.

Si on rencontre un sommet de degré quatre, on choisit de poursuivre avec l'arc du milieu.

Kato et Yasuhara affirment qu'il n'existe qu'une façon de parcourir le graphe si le tracé respecte les contraintes imposées par cet algorithme. Après reconstitution du parcours, tous les arcs doivent avoir été traversés.

## 2.3 Evaluation de l'algorithme

L'algorithme de Kato et Yasuhara peut être appliqué sur des tracés manuscrits quelconques. Nous l'avons testé sur des caractères isolés issus de la base de lettres IRONOFF [VIA 99] développée à l'Université de Nantes. Elle compte plus de 10 000 lettres tracées par plus de 400 scripteurs différents. Ce sont des lettres dites "cursives". La particularité de cette base est de fournir pour chaque lettre

l'image en niveaux de gris (signal hors-ligne), et la suite de points (signal en-ligne). Ces deux informations permettent de comparer le signal en-ligne généré artificiellement à partir de l'image et le véritable signal en-ligne. Le tracé est relativement épais, ce qui peut notamment provoquer la formation de sommets de degré supérieur à quatre et d'arcs à parcourir plusieurs fois (voir figure 5).

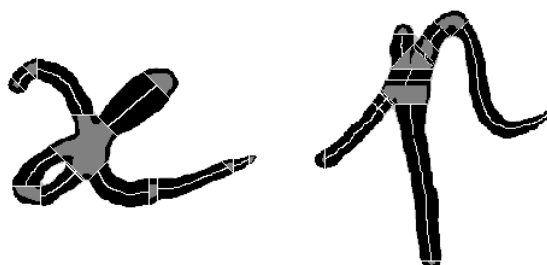


FIG. 5 – Cas particuliers présents dans la base IRONOFF

Pour évaluer l'algorithme de Kato et Yasuhara, il est nécessaire de connaître les poser et lever de crayon. Leur détection de manière automatique est un vrai problème, notamment quand ceux-ci sont cachés dans le tracé, c'est pourquoi ce problème sera résolu ultérieurement. Les poser et lever ont donc été déduits du tracé en-ligne. Pour vérifier que le parcours proposé est le bon, on a cherché à le comparer au parcours de graphe correspondant au tracé en-ligne. Un ensemble de 5 000 images a été testé.

Les résultats de l'algorithme de Kato et Yasuhara sont d'environ 94% de lettres bien ordonnancées, 3% de lettres mal ordonnancées et 3% de lettres rejetées, pour lesquelles certains arcs n'ont pas été traversés. Les résultats obtenus dans cette base de lettres étant encourageants, nous avons analysé précisément les causes d'erreurs pour proposer une extension de l'algorithme.

## 3 Extensions de l'algorithme de reconstruction du tracé

Les cas d'erreurs concernent environ 6% des lettres présentes dans la base, ce qui n'est pas négligeable. Il est donc intéressant de chercher à les résoudre.

### 3.1 Analyse des cas d'erreurs

Certaines lettres sont mal parcourues avec l'algorithme de Kato et Yasuhara. Cela se traduit par des arcs non traversés, ou des arcs mal ordonnancés. Nous avons réparti les erreurs en trois catégories, une lettre pouvant appartenir à plusieurs catégories :

- La catégorie 1 regroupe les lettres ne respectant pas les contraintes imposées par l'algorithme. Certains graphes possèdent des arcs à traverser trois fois (voir figure 6.c), concernant 30% des cas d'erreurs, ou des sommets de

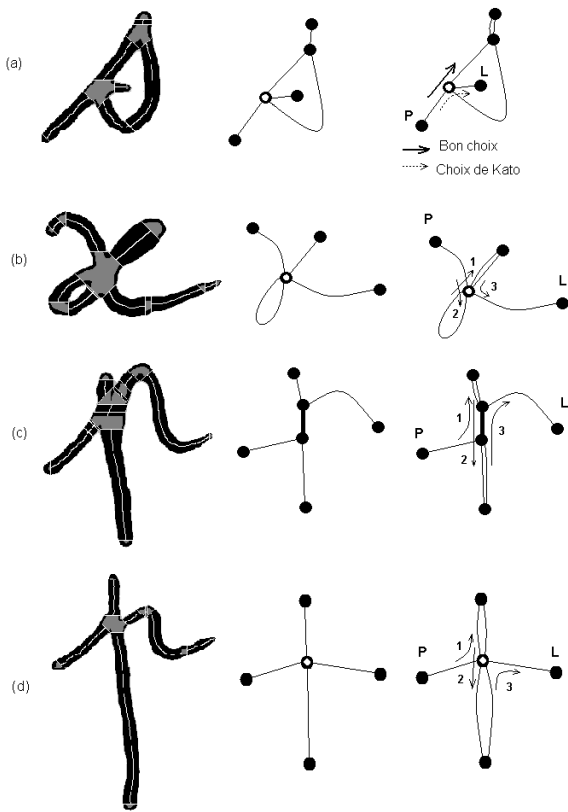


FIG. 6 – Cas non résolus avec l’algorithme de Kato et Yasuhara (a) Sommet de degré quatre représentant un contact. (b) Sommet de degré supérieur à quatre. (c) Arc à parcourir trois fois. (d) Sommet de degré quatre sans boucle

degré supérieur à quatre (voir figure 6.b ), présents dans 5% des cas d’erreurs.

- La catégorie 2 correspond à des erreurs d’étiquetage, comme des PD-lines NORMAL et SPECIAL confondues. Une troisième sorte de PD-line existe et n’est pas prise en compte par l’algorithme. (voir figure 7) 40% des cas d’erreurs appartiennent à cette catégorie.
- La catégorie 3 concerne les erreurs de la phase de parcours. Les sommets de degré quatre sont toujours interprétés comme des croisements, les cas de tangence entre traits ne sont pas pris en compte. (voir figure 6.a et figure 6.d ) 40% des cas d’erreurs sont dus à ce problème.

### 3.2 Nouvel algorithme de reconstruction proposé

Nous avons cherché à résoudre les cas d’erreurs présentés ci-dessus en relâchant les contraintes imposées par l’algorithme d’origine, en permettant les croisements de plus de deux traits et les arcs traversés trois fois.

Le relâchement de ces contraintes nous a conduit à proposer de nouvelles règles d’ordonnement des arcs pour chaque catégorie d’erreur :

- Pour les problèmes de catégorie 1, de nouvelles règles ont été ajoutées. Lors de la phase d’étiquetage, on cherche à détecter les arcs à traverser trois fois (selon des critères de

continuité) et les D-lines touchant les sommets de degré supérieur à quatre. Un principe important pour parcourir le graphe est de traverser tous les arcs une seule fois. Les arcs touchant les sommets de degré supérieur à quatre sont reliés deux à deux en évitant de relier l’arc d’entrée et l’arc de sortie.

- Pour les PD-lines, il existe des cas où la D-line doit être parcourue après la boucle, comme dans la figure 7. On ajoute donc une étiquette supplémentaire “SPECIAL2 PD-line”. Le choix du type de PD-line se fait en fonction des angles formés au niveau des arcs, comme montré dans la figure 8. Le principe est d’avoir le moins de cassures possible dans le parcours.
- Les sommets de degré quatre sont de deux types : avec ou sans boucle. En l’absence de boucle (comme sur la figure 6.d), il s’agit en fait de l’intersection de trois traits, deux arcs sont des D-lines. La première phase doit détecter ces D-lines supplémentaires. Le parcours se fait alors sur le même principe que pour les sommets de degré supérieur à quatre. En présence d’une boucle, on cherche à la parcourir avant de traverser l’arc de sortie.

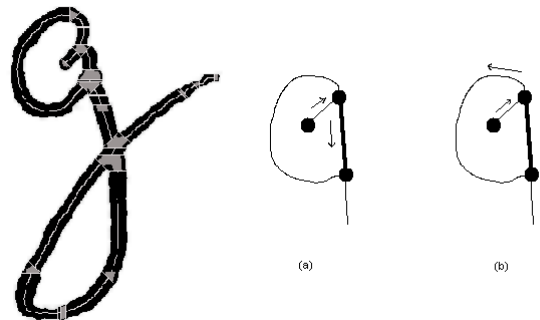


FIG. 7 – Exemple d’une PD-line ni NORMAL ni SPECIAL. La double ligne doit être parcourue après la boucle. (a) Arc choisi par l’algorithme de Kato et Yasuhara (b) Arc qui devrait être choisi.

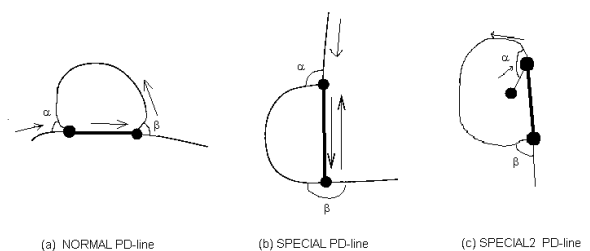


FIG. 8 – PD-line : choix de la sous-étiquette selon les angles formés.

Les décisions sont parfois trop locales pour permettre une décision immédiate sans risque d’erreurs. Une autre extension de l’algorithme est de proposer deux parcours pour un seul couple de poser et lever. En effet, il est préférable de proposer deux solutions dont l’une est bonne plutôt

qu'aucune solution. On choisira ultérieurement la meilleure solution selon d'autres critères, basés par exemple sur des connaissances *a priori*, ou encore sur la pertinence des résultats proposés par le reconnaiseur en-ligne.

## 4 Expérimentations

Les tests ont été effectués sur la base IRONOFF présentée dans la sous-section 2.3. Le même ensemble d'images a été utilisé pour comparer l'algorithme de Kato et Yasuhara d'origine et la version étendue que nous proposons. Les tests concernent un ensemble de 5000 images.

Le tableau ci-dessous présente les taux de réussite de l'ordonnement des arcs, avec l'algorithme de Kato et Yasuhara, et avec son extension. La deuxième colonne concerne les lettres dont le bon parcours est proposé parmi la ou les solutions de l'algorithme. En ce qui concerne l'extension, 2% des bons parcours résultent d'une double proposition de solutions.

	bon parcours	rejet	mauvais parcours
Kato et Yasuhara	94%	3.1%	2.9%
Extension	96.8%	2.1%	1.1%

On remarque que le taux d'images en erreur a diminué de 46.7%. Le nombre de lettres bien ordonnées augmente de 2.8% avec les solutions proposant deux parcours. L'algorithme étendu reste de complexité linéaire. Pour les lettres proposant plusieurs parcours, le temps de calcul est multiplié par deux (soit dans environ 5% des images).

## 5 Conclusion

L'objectif de notre étude est de reconstituer le parcours de caractères hors-ligne. Nous avons expérimenté l'algorithme de Kato et Yasuhara sur la base de lettres IRONOFF et mis en évidence ses limites. Nous avons diminué les 6% de cas d'erreurs en cherchant leur origine et en étendant l'algorithme à des cas non prévus. La nouvelle version de l'algorithme permet de baisser de 46.7% les cas d'erreurs. Nous proposons notamment deux parcours pour les lettres présentant une ambiguïté qui ne peut pas être résolue à ce niveau de la chaîne de traitement.

Nous travaillons actuellement sur la détection automatique des poser et lever de crayon. Cette tâche est ardue, car les poser et lever de crayon peuvent être confondus avec le tracé, tout sommet est alors un candidat potentiel. De plus, leur nombre est inconnu quand plusieurs tracés se touchent. Après utilisation de l'algorithme présenté dans l'article, nous obtiendrons un ou plusieurs signaux en-ligne, que nous fournirons au reconnaiseur en-ligne. Nous étendrons notre méthode pour reconstruire également des mots.

### Remerciements

Les auteurs tiennent à remercier les archives départementales

des Yvelines et le conseil général des Yvelines pour leur soutien et collaboration dans ces travaux de recherche.

## Références

- [ANQ 97] ANQUETIL E., LORETTE G., On-line Handwriting Character Recognition System Based on Hierarchical Qualitative Fuzzy Modeling, *Progress in Handwriting Recognition*, pp. 109-116, World Scientific, 1997.
- [ANQ 02] ANQUETIL E., BOUCHEREAU H., Integration of an On-line Handwriting Recognition System in a Smart Phone Device, *Proceedings of the 16th IAPR International Conference on Pattern Recognition (ICPR 2002)*, Quebec, 2002, pp. 192-195.
- [BOC 93] BOCCIGNONE G., CHIANESE A., CORDELLA L. P., MARCELLI A., Recovering Dynamic Information from Static Handwriting, *Pattern Recognition*, vol. 26, n° 3, 1993, pp. 409-418.
- [DOE 95] DOERMANN D. S., ROSENFELD A., Recovery of Temporal Information from Static Images Handwriting, *International Journal of Computer Vision*, vol. 15, 1995, pp. 142-164.
- [JäG 96] JÄGER S., Recovering Writing Trace in Off-Line Handwriting Recognition: Using a Global Optimization Technique, *Proceedings of ICPR '96*, vol. 3, 1996, page 150.
- [KAT 00] KATO Y., YASUHARA M., Recovery of drawing order from single-stroke handwriting images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, 2000, pp. 938-949.
- [LAL 99] LALLICAN P., Reconnaissance de l'écriture manuscrite hors-ligne: utilisation de la chronologie restaurée du tracé, PhD thesis, Université de Nantes, 1999.
- [PET 91] PETTIER J., CAMILLERAPP J., An optimal detector to localize handwriting strokes, *1st International Conference on Document Analysis and Recognition*, vol. 2, Saint-Malo, oct 1991, pp. 710-718.
- [PLA 99] PLAMONDON R., PRIVITERA C., The segmentation of cursive handwriting : an approach based on off-line recovery of the motor-temporal information, *IEEE Transactions on Image Processing*, vol. 8, n° 1, 1999, pp. 80-91.
- [VIA 99] VIARD-GAUDIN C., LALLICAN P. M., KNERR S., BINTER P., The IRESTE On/Off (IRONOFF) Dual Handwriting Database, *Proceedings of the Fifth International Conference on Document Analysis and Recognition (ICDAR'99)*, 1999, pp. 455-458.