



**HAL**  
open science

## Un Système de Vérification de Signature manuscrite en Ligne pour PDA.

Nizar Rokbani, Adel Alimi

► **To cite this version:**

Nizar Rokbani, Adel Alimi. Un Système de Vérification de Signature manuscrite en Ligne pour PDA..  
Jun 2004. sic\_00001178

**HAL Id: sic\_00001178**

**[https://archivesic.ccsd.cnrs.fr/sic\\_00001178v1](https://archivesic.ccsd.cnrs.fr/sic_00001178v1)**

Submitted on 6 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.

# Un Système de Vérification de Signature manuscrite en Ligne pour PDA.

Nizar Rokbani<sup>1</sup> – Adel Alimi<sup>1,2</sup>

<sup>1</sup> REGIM, Groupe de Recherche sur les Machine Intelligente, Ecole Nationale d'Ingénieurs de Sfax.

<sup>2</sup> Département de Génie Electrique, Ecole Nationale d'Ingénieurs de Sfax,  
Route de Soukra, Km 4, Sfax  
Tunisie

[nizar.rokbani@yahoo.fr](mailto:nizar.rokbani@yahoo.fr), [adel.alimi@ieee.org](mailto:adel.alimi@ieee.org)

**Résumé :** Dans ce papier nous présentons une méthode de vérification de la signature manuscrite en ligne fondée sur la mesure de la ressemblance graphique doublée d'une vérification paramétrique globale de type statistique. Pour la vérification de la ressemblance graphique nous proposons une nouvelle méthode basée sur la comparaison de deux trajectoires de points. Nous recherchons à associer les points les plus proches, dans la mesure du possible afin de calculer une distance simple entre les deux trajectoires. Le module de vérification paramétrique utilise dans sa version finale une vérification à deux niveaux. Notre méthode est optimisée pour des applications embarquées de type PDA, ces équipements ont besoin d'algorithme de reconnaissance du manuscrit à la fois efficace et peu complexe. Nous avons obtenu un taux de fausse acceptation de 6.43%, et un taux de faux refus de 7.33%.

**Mots-clés :** vérification de la Signature, vérification paramétrique, ressemblance graphique, PDA.

## 1 Introduction

La signature reste malgré tous les progrès technologiques le moyen le plus utilisé pour authentifier un document, valider un contrat ou une transaction financière [Jain 2002]. La signature peut être également utilisée comme un mot de passe pour consulter des documents confidentiels, ou tout simplement pour accéder à son bureau. Elle est aussi fiable que l'identification vocale ou rétinienne [Jain 99]. Les systèmes de vérification de signature sont des systèmes traitant le manuscrit ; ces derniers sont subdivisés en deux catégories les systèmes « en ligne » et les systèmes « hors ligne » [Plam 2000]. Toute fois un système hybride [Zim 2003] utilisant à la fois une représentation en ligne, suite de coordonnées (x,y), et une représentation hors ligne, image des signatures, a été proposée par Alessandro Zimmer et Lee Luang Ling (2003).

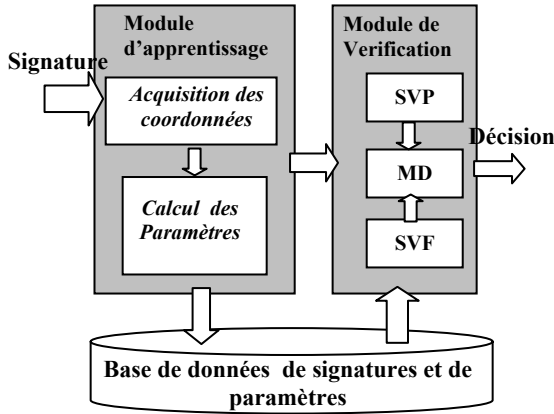
Un système « hors ligne » traite la signature à partir d'une image [Zim 2003], tel que l'émargement sur un

chèque ou sur un document. Un simple scanner suffit pour capturer la signature, Un prétraitement est indispensable à l'extraction de la signature et on ne dispose que des données statiques [Plam 93]. Dans le cas d'un système « en ligne » la signature est acquise directement sur une tablette ou un stylo électronique, on peut donc relever des caractéristiques dynamiques tel que le temps de signature, la pression ou l'inclinaison du stylo [Ohi 2000]. Pour un système de vérification en ligne l'interface d'acquisition influe sur la conception du système, ainsi une tablette simple ne peut servir à des méthodes de vérifications nécessitant des paramètres tel que la pression ou l'inclinaison du stylo, pourtant des méthodes utilisant ces paramètres sont proposées par Y.K.T Ohishi et col [Ohi 2000] ou S.Hangai et col [Han 2000]. Une tablette classique suffira pour des algorithmes utilisant des paramètres tel que la vitesse et l'accélération [plam 95], [Wu 98]. De plus en plus de méthodes comparent à la fois les paramètres et la trajectoire de la signature [Jain 2000]. L'apparition des PDAs et des tablets PCs relance la recherche sur la reconnaissance du manuscrit; ces ordinateurs personnels s'exploitent exclusivement via un écran tactile assimilable à une tablette. Là encore la signature manuscrite peut être utilisée en remplacement aux classiques mots de passe utilisés aussi bien pour la protection des documents qu'aux autorisations d'accès à certains répertoires confidentiels.

Dans ce papier nous proposons un système de vérification en ligne fondé sur une comparaison des paramètres dynamiques ainsi qu'une mesure de la ressemblance des formes des signatures. La signature capturée est mise à l'échelle, centrée et superposée à la signature spécimen. On calcule alors la distance entre le spécimen et la signature à tester. Si cette distance est en dessous d'un certain niveau la signature sera supposée authentique au niveau graphologique.

Le SVS que nous proposons est composé de deux modules, voir FIG.1. Un module d'apprentissage et un module de vérification. Ce dernier comprend un système de vérification paramétrique, SVP, et un

système de vérification des formes, SVF, qui mesure la ressemblance graphique entre les deux signatures. Pour la vérification paramétrique nous utilisons une approche statistique, convenable aux applications disposant de peut de mémoire [Nel 93]. Nous limiterons au maximum la taille des données nécessaire a notre SVS, ainsi que la complexité des algorithmes qu'il utilise dans le but d'en faire un SVS en ligne rapide pour PDA.



*SVP : Système de vérification paramétrique.*  
*SVF : Système de vérification de forme.*  
*MD : Module de décision.*

FIG. 1 – Architecture du SVS proposé

## 2 Spécification paramétrique de la signature

Nous choisissons, d'utiliser un vecteur paramètre de type Vitesse accélération, ce type de représentation est efficace pour caractériser une signature [Plam 94]. Tous les paramètres sont déduits des coordonnées de la signature. Chaque coordonnée est représentée par son abscisse et son ordonnée  $X_i = (x_i, y_i)$ . La vitesse instantanée,  $V_i$  [1], est approximés par les différentielles des coordonnées x et y entre les instants  $T_{i+1}$  et  $T_i$  [Wu 97], nous approximations les accélérations par des différentielles de la vitesse [2].

$$V_i = (V_{xi}, V_{yi}) = \frac{dX_i}{dt} = \left( \frac{dx_i}{dt}, \frac{dy_i}{dt} \right) \cong \left( \frac{x_{i+1} - x_i}{t_{i+1} - t_i}, \frac{y_{i+1} - y_i}{t_{i+1} - t_i} \right) \quad [1]$$

$$A_{xi} = (A_{xi}, A_{yi}) \cong \left( \frac{V_{x_{i+1}} - V_{x_i}}{t_{i+1} - t_i}, \frac{V_{y_{i+1}} - V_{y_i}}{t_{i+1} - t_i} \right) \quad [2]$$

A partir des vitesses instantanées, on calcule les vitesses moyennes et efficaces sur chaque axe, voir formules [3] et [4] pour l'axe X.

$$V_{x \text{ moy}} = \frac{1}{N} \sum_{i=1 \dots K} V_{xi} \quad [3]$$

$$V_{x \text{ rms}} = \frac{1}{N} \sqrt{\sum_{i=1}^N V_{xi}^2} \quad [4]$$

Ces paramètres, vitesse moyenne et efficace, dissocient les dynamiques sur chaque axe. Le produit des vitesses sur les deux axes fournit une information sur la dépendance de ces deux paramètres [5]. Nous rajoutons des paramètres permettant de tenir compte des vitesses de départ et d'arrivée [6] [7].

$$V_{xy} = \frac{1}{N^2} \sum_{i=1}^N V_{xi} * V_{yi} \quad [5]$$

$$V_{startx} \cong \frac{1}{n_0} \sum_{i=1}^{n_0} \frac{x_{i+1} - x_i}{t_{i+1} - t_i} \quad [6]$$

$$V_{endx} \cong \frac{1}{n_0} \sum_{i=last - po \text{ int} - n}^{last \text{ po int}} \frac{x_{i+1} - x_i}{t_{i+1} - t_i} \quad [7]$$

$n_0$  Étant un entier inférieur à 10.

$V_{strx}$  : indique la vitesse moyenne de départ suivant l'axe x.  
 $V_{endx}$  : indique la vitesse moyenne terminale suivant l'axe x.

Nous définirons les paramètres d'accélération par analogie aux paramètres vitesses, Nous calculons également les accélérations moyennes et efficaces sur chaque axe, ainsi que le produit des accélérations par les formules analogues à celles utilisées pour calculer ces paramètres en vitesse [3], [4] et [5]. Les directions de départ, et d'arrivée suivant les deux axes,  $dstrx$ ,  $dendy$  pour l'axe Y, sont des paramètres binaires indiquant si la signature commence suivant l'orientation de l'axe ou à son opposée. Ces paramètres sont regroupés pour former le vecteur paramètre  $Zid$ , qui inclus les temps de signature,  $T_s$ , le temps total de levée de stylo,  $T_{off}$ , le temps de signature utile,  $T_{on}$ , ainsi que le rapport longueur sur largeur de la signature.

$$Zid = (T_s, T_{on}, T_{off}, V_{xmoy}, V_{ymoy}, V_{xrms}, V_{yrms}, A_{xmoy}, A_{ymoy}, A_{xrms}, A_{yrms}, V_{xy}, A_{xy}, \underline{dstrx, dstry, dendx, dendy, V_{strx}, V_{endx}, V_{strx}, V_{endx}, R_{xy}}) \quad [8]$$

Pour tenir compte de la variation naturelle rendre le vecteur paramètre plus représentatif de la signature nous avons calculé la moyenne et l'écart type de chaque paramètre sur un nombre fini de signatures de références authentiques. On définit ainsi un vecteur paramètre moyen  $Z\mu$ .

$$Z\mu = \frac{1}{K} \sum_{i=1}^K Zid \quad [9]$$

Avec K le nombre des signatures de référence.

Il est important de remarquer que pour permettre au système d'être évolutif nous garderons la totalité des vecteurs paramètres des signatures de références ayant servies à l'apprentissage.

## 3 Vérification paramétrique de la signature

Le module procède à une vérification en deux niveaux ; en premier lieu on compare les directions de départ et de fin de la signature, si ces paramètres sont conformes, on procède à une vérification du reste du vecteur paramètre. Cette comparaison en deux niveaux cherche à réduire le nombre de faux acceptés, il s'agit de comparer les paramètres locaux aux points de départ et de fin de la signature. Si à ce niveau la signature est acceptée alors le système continue la vérification des paramètres restants. La signature est authentique si cette distance est inférieure à un seuil si non elle est frauduleuse. Nous définissons un vecteur seuil, «  $Se$  » [10], qui rassemble les seuils relatifs à chaque paramètre.

$Se = (Sts, Stn, Stf, Svxmoy, Svymoy, Svxrms, Svyrms, SVxy, Saxmoy, Saymoy, Saxrms, Sayrms, SAxy, dstrxu, dstryu, dendxu, dendyu, Svstrxu, Svendxu, Svstrxu, Svendxu, Srxxyu)$  [10]

Le seuil de chaque paramètre est ajusté a part, indépendamment des autres seuils, pour accepter 90% des signatures d'apprentissages.

## 4 Vérification des formes des signatures

Un système de vérification en ligne paramétrique reste aveugle, ne tenant pas compte de la représentation graphique de la signature. D'où l'idée de le compléter par un système qui vérifie les formes des signatures. Nous sommes donc amenés à comparer deux graphes qui ont rarement le même nombre de points, des tailles différentes... Pour résoudre ce problème nous proposons une méthode qui compare deux signatures en six étapes :

1. Calculer la taille de la signature référence.
2. Calculer la taille de la signature à tester.
3. Mettre à l'échelle et superposer les deux signatures.
4. Associer les points.
5. Calculer la distance entre les deux signatures
6. Décider l'authenticité de la signature.

Ces étapes seront détaillées dans les paragraphes suivants

### 4.1 Mise à l'échelle et superposition des signatures

Soit les graphes de la figure 2(a), si on note  $w_{x1}$  et  $w_{x2}$  respectivement la taille sur l'axe x de la première et de la deuxième signature, on constate que  $w_{x1} > w_{x2}$ . Pour résoudre ce problème nous proposons de multiplier la seconde signature par un facteur d'échelle  $c$ .

$$c = \frac{w_{x2}}{w_{x1}} \quad [11]$$

Les nouvelles coordonnées de la deuxième signature  $Xechi$  seront obtenues en multipliant les coordonnées du point  $(X_i)$  par le facteur d'échelle  $c$ .

$$Xechi = X_i * c \quad [12]$$

$i = 1, 2, \dots, N$ ,  $N$  étant le nombre de points de la signature

Nous proposons ensuite de superposer les deux centres de gravités des signatures.

$$G_k = \frac{1}{N} \sum_{i=1..N} X_{ki} \quad [13]$$

Le vecteur  $\vec{GTG2}$  sera donc le vecteur translation qui déplacera la deuxième signature pour la superposer à la première. La figure 2(a) montre les deux signatures affichées dans la même fenêtre avant la mise à l'échelle, la figure 2(b) montre les deux signatures après mise à l'échelle et superposition. Nous tenons à remarquer que nous superposons les deux centres de gravités des signatures.

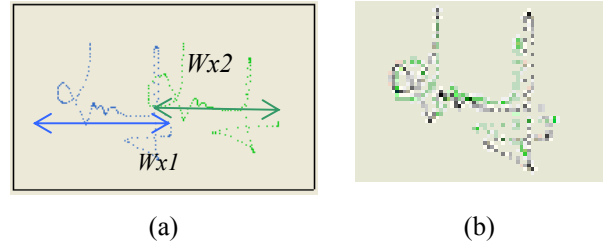


FIG. 2 – (a) Signatures Originales, (b) Signatures après mise a échelle et superposition.

### 4.2 Association des points et calcul de la distance entre signatures

Soit deux signatures ayant préalablement été superposées, nous comparons deux trajectoires en calculant la distance qui les sépare. Nous rappelons également que les signatures sont considérées comme des graphes constitués chacun par une série de points. Nous définissons la dimension d'un graphe comme le nombre de points composant ce graphe. Dans le cas de deux graphes n'ayant pas la même dimension, nous cherchons à associer tous les points du graphe ayant la plus petite dimension aux points les plus proches sur le graphe ayant la plus grande dimension.

L'association des points est réalisée par un algorithme simple ; nous considérons le graphe de plus petite dimension comme graphe de référence. On parcourt alors les points du graphe de référence en recherchant sur le second graphe le point le plus proche en terme de distance géométrique sans tenir compte du temps.

Nous proposons d'utiliser un algorithme de type Kppv (2 plus proches voisins) pour rechercher et associer les points correspondants. Chaque point associé ne pouvant être associé qu'à un seul voisin.

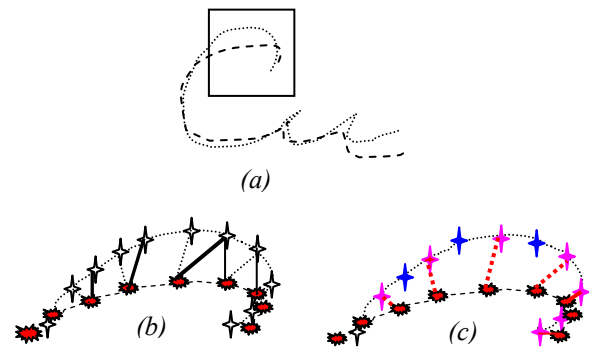


FIG. 3 – Fonctionnement de l'algorithme d'association des points. (a) deux graphes a associer, (b) calcul des distances entres points et recherche des voisins (c) association des points des deux graphes.

La figure 3 (a) montre deux signatures superposées, la figure 3 (b) montre les différentes distances calculées entres les points des deux graphes, finalement nous observons sur la figure 3 (c) les points associés qui serviront à calculer la distance entre les deux graphes.

### 4.3 Calcul de la distance entre deux graphes.

La distance entre deux graphes, de dimension respective  $N_1$  et  $N_2$ , est calculée en additionnant les distances correspondantes aux points associés par paires des deux graphes. Cette distance est exprimée par la formule [15].

$$Dist_{12} = \frac{1}{N} \sum_{i=1}^N (dist_{12})_i + P * (|N_1 - N_2|) \quad [15]$$

Avec  $N = \min(N_1, N_2)$ ,  $P$  est la pénalité qui permet de tenir compte de la différence entre les dimensions des deux graphes et  $(dist_{1,2})_i$  est la distance entre le point (i) du graphe 1 et 2.

$$P = \text{Max}_{i=0}^N (dist_{1,2})_i. \quad [16]$$

$$(dist_{12})_i = \sqrt{(X_i^1 - X_i^2) * (X_i^1 - X_i^2)^T} \quad [17]$$

La FIG.4 montre la distance entre deux signatures authentiques de la même personne.

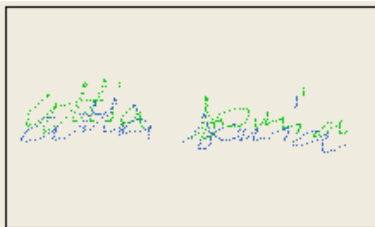


FIG. 4 - Calcul de la distance entre deux signatures,  $d=14.89$

Pour décider de l'authenticité d'une signature on se propose de calculer la distance moyenne qui sépare les graphes représentant toutes les signatures authentiques d'une même personne. Nous croisons deux à deux les signatures, nous calculons à chaque fois la distance qui les sépare avant de calculer la distance moyenne [18] séparant les signatures authentiques. Cette distance sera utilisée pour définir le seuil d'acceptation  $St$  du module de vérification de la trajectoire. Les formules suivantes détaillent le calcul de la distance moyenne [18] et du seuil d'acceptation [20].

$$Dist_{\mu} = \frac{1}{l} \sum_{i=1}^l Dist_i \quad [18]$$

$$E_{dist} = \frac{1}{l} \sqrt{\sum_{i=1}^l (Dist_{\mu} - Dist_i)^2} \quad [19]$$

$$St = (1 + a) * E_{dest} \quad [20]$$

$a$ : étant un coefficient réel d'ajustement du seuil d'acceptation,  $a \in [0,1]$ .

$l$ : le nombre de combinaisons de 2 des spécimens de signatures de références.

Pour ajuster le seuil d'acceptation nous proposons une méthode qui consiste à incrémenter le facteur d'ajustement ( $a$ ) [20], pour obtenir accepter 90% des spécimens d'apprentissages. La signature est supposée authentique si la plus petite distance qui la sépare des signatures Spécimens est inférieure au seuil

d'acceptation ; si non elle sera considérée comme fausse.

## 5 Amélioration des modules de vérification en cours d'usage

La signature étant un graphe très personnel, ce graphe subit naturellement des modifications dépendantes de la personnalité de son producteur, de son état de stress, ou son état psychique de façon générale. Pour tenir compte des ces modifications nous proposons deux méthodes. La première utilise une nouvelle série de signatures, donc un nouveau cycle d'apprentissage. La seconde ajuste les paramètres et les spécimens pour tenir compte d'une nouvelle signature authentique.

### 5.1 Réapprentissage volontaire

Cette méthode est proposée pour permettre au signataire d'introduire volontairement une nouvelle série de spécimens de signatures. Les paramètres ainsi calculés seront rajoutés aux paramètres précédents ; un nouveau vecteur paramètre moyen est calculé. Pour limiter la taille du fichier paramètre, nous détruirons les paramètres les plus anciens et garderons les nouveaux paramètres, FIG.5. Une nouvelle série de signatures de références est élue. Nous garderons les spécimens ayant les paramètres les plus centrés, les plus proches de la moyenne.

#### Début

Etape 1. {Saisir nouvelle série

Calculer paramètres}

Etape 2. {Recalculer le nouveau vecteur paramètre moyen.}

Etape 3. {Choisir nouvelles signatures de références « spécimens »}

Etape 4. {Détruire ancienne série de paramètres}

#### Fin

FIG. 5 - Algorithme du réapprentissage volontaire.

### 5.2 Glissement de paramètre

Cette méthode permet d'ajuster légèrement le vecteur paramètre moyen en le recalculant pour tenir compte du vecteur paramètre d'une nouvelle signature authentique. La méthode recalcule le vecteur paramètre moyen et inclue la nouvelle signature à la série de signatures de références si elle est plus proche de la moyenne qu'une signature existante, voir FIG.6.

#### Début

Etape 1. {Saisir nouvelle signature}.

Etape 2. {Ajuster vecteur paramètre moyen.}

Etape 3. {Si  $dist(i, \mu) < dist(r, \mu)$  alors remplacer signature de référence par la signature(i)}.

#### Fin

FIG.6 - Algorithme de glissement des paramètres

L'ajustement du vecteur paramètre moyen peut se faire par la formule suivante :

$$Z_{\mu} = \frac{1}{k+1} (k \times Z_{\mu} + Z_i) \quad [21]$$

k étant le nombre de spécimens de références ayant servis au calcul du vecteur paramètre moyen ;  $Z_i$  le vecteur paramètre de la nouvelle signature.

## 6 Résultats et perspectives

### 6.1 Description de l'échantillon et des conditions du test

Le test a été réalisé avec 15 personnes (13 hommes et 2 femmes) ; chaque personne a fourni 65 signatures authentiques. Chaque participant fournit quatre imitations des signatures de ses collègues, il fournira donc (4\*14) fausses signatures. Nous disposons de 56 imitations de chaque signature. Pour la mesure des FRR, taux de faux refus, et FAR, taux de fausses acceptations, nous partageons les signatures authentiques en deux groupes. Le premier groupe est composé de 525 signatures, soit 35 signatures par personnes qui serviront à l'apprentissage. Le second groupe est composé de 450 signatures soit 30 signatures par personne et qui serviront à mesurer le taux de faux refus FRR. Pour le taux de fausses acceptations FAR, nous utilisons les imitations soit 56 signatures par personnes. Pour évaluer globalement les performances du système nous calculons la moyenne des ces taux par rapport à chaque participant. Pour ce rapprocher des conditions naturelle d'utilisation nous déployons le SVS développé sous C#.NET<sup>1</sup> sur le simulateur de PDA de Microsoft Studio.net<sup>2</sup>, voir FIG.7(b). La FIG.7 (a) montre l'interface utilisateur du module d'apprentissage. Les signatures sont récoltées via une tablette.

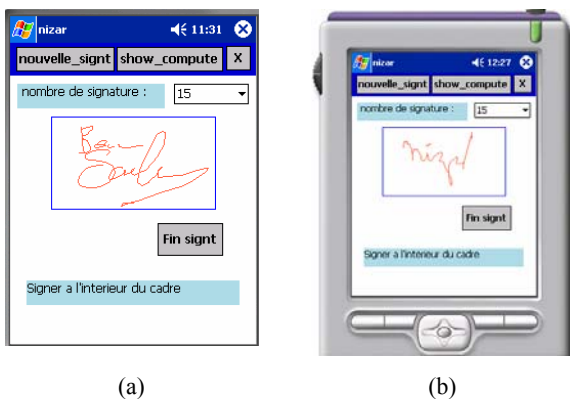


FIG.7- (a) Interface utilisateur du module d'apprentissage, (b) Le simulateur de PDA de Microsoft MS Studio. net

Le système peut travailler avec 10, 15 ou 20 signatures d'apprentissage. Nous étudions les performances du système de vérification paramétrique seul, du système de vérification de forme seul, ainsi que deux propositions de systèmes cascades. La première consiste à mettre la vérification paramétrique en premier lieu,

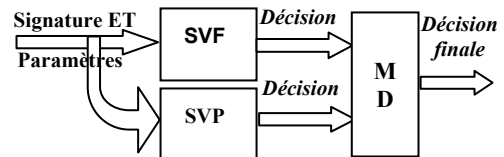
elle sera suivie d'une vérification de la forme de la signature, SVPF. La seconde proposition consiste à placer le système de vérification de forme en premier lieu suivi du système paramétrique, SVFP. Le SVPF, système de vérification paramétrique forme, ne procède à la vérification de la forme de la signature que si ces paramètres sont authentiques. Le TAB.1, résume les résultats obtenus.

Type de vérification	FRR (%)	FAR(%)
Vérification paramétrique	12.33 %	10.82 %
Vérification de la forme	8.66 %.	10.83%
VPF (Paramétrique forme)	12.33%,	9.15 %.
VFP (Forme paramétrique)	10.21%,	8.2 %.

TAB.1- performances des différents systèmes de vérification des la signature pour 10 signatures d'apprentissages

### 6.2 Performance du SVP en fonction du nombre des signatures d'apprentissages.

Le SVP, système de vérification parallèle, vérifie l'authenticité paramétrique et les forme des signatures. A l'opposée des propositions précédentes, chaque sous système de vérification procède a sa propre expertise. Les décisions sont alors transmises a un module dédié qui remet une décision finale, voir FIG.8.



SVF : Système de vérification des formes.  
SVP : Système de vérification paramétrique.  
MD : Module de décision.

FIG.8 - Structure du SVS parallèle.

Pour le module de décision, MD, une signature n'est authentique que si le SVP et le SVF s'accordent à la reconnaître en tant que tel, si non elle est frauduleuse. Nous présentons ici les FAR et FRR moyen relatif à un apprentissage de 10, 15 et 20 signatures prises au hasard dans la base des signatures d'apprentissages. Les taux moyens sont résumés au TAB.2. La moyenne des FAR et FRR pour un apprentissage à 10 spécimens sont respectivement  $FAR_{\mu} = 13.68 \%$ ,  $FRR_{\mu} = 13.33 \%$ . Ces taux sont attendus vu que nous utilisons une méthode statistique pour la définition des seuils d'acceptations paramétrique et de forme. Les résultats relatifs à 15 signatures d'apprentissages, voir TAB.2, confirme cette hypothèse en effet nous observons une baisse du FRR qui passe à  $FRR_{\mu} = 11.50 \%$ , soit une amélioration de 1.83%. Le FAR suit la même loi d'évolution et baisse à  $FAR_{\mu} = 11.22 \%$ . Les meilleurs résultats sont obtenus pour 20 signatures d'apprentissages, voir tableau 2. Nous obtenons un FRR de 7.33% et un FAR de 6.42%.

<sup>1</sup> C#.Net est. une marque de Microsoft company

<sup>2</sup> Microsoft Studio.net est. Une marque de Microsoft company

Nombre de signatures	FRR(%)	FAR(%)
10	13.33 %	13.68%
15	11.50 %	11.22 %
20	7.33 %	6.42 %

TAB.2 - FAR et FRR du SV pour PDA

### 6.3 Evaluation du module de réapprentissage

Les modules de glissement de paramètre et de réapprentissage peuvent améliorer progressivement les performances en phase d'utilisation. Pour évaluer le module de réapprentissage nous avons appliqué le protocole suivant : nous sommes partis d'un apprentissage à 20 signatures, nous avons appliqué la procédure de réapprentissage en utilisant le jeu de 15 signatures authentiques restantes dans la base de donnée avant de relever à nouveau les FRR et FAR du système. Nous remarquons une légère amélioration des taux de faux refus et de fausse acceptation visible au TAB.3. On en déduit également que le système converge vers des performances stables, autour **FRR 7.2%** et un **FAR de 6.12%**.

Nombre de signatures	FRR(%)	FAR(%)
20	7.33 %	6.42 %
+15 réapprentissage	7.2 %	6.12 %

TAB.3- FAR et FRR suite à un réapprentissage.

## 7 Conclusion

Nous avons proposé une méthode pour la vérification de la signature en ligne. Cette méthode compare à la fois les formes et les paramètres des signatures elle a la particularité d'utiliser des algorithmes de faible complexité et de travailler avec une base de donnée réduite à un fichier paramètres et à vingt fichiers binaires pour le stockage des signatures de références. Nous avons étudié les performances du SVP et SVF Par rapport à dix signatures d'apprentissages nous pouvons conclure que le système de vérification de forme SVF, donne le taux de faux refus, FRR, le moins élevé. Ce taux n'excède = 8.66%, contre 12.33% pour le système paramétrique, SVP. Ce dernier affiche le meilleur taux de fausse acceptation, ce taux est de 10.82 % contre 10.83% pour le SVF.

Le SVS embarqué que nous proposons utilise un SVF et un SVP. Il affiche un taux de faux refus de 7.33%, et un taux de fausse acceptation de 6.42 %. Ces taux sont obtenus par un apprentissage de vingt signatures.

L'embarquement sur PDA du SVS a été réalisé sous C#.net smart édition. Les résultats obtenus nous ouvrent plusieurs perspectives de perfectionnement de notre SVS. L'optimisation du module de décision qui permettra de pondérer les décisions entre la vérification paramétrique et la vérification graphologique, de forme, l'amélioration de la vérification paramétrique par des analyses complémentaires autour des points particuliers des signatures, ou encore l'analyse des

singularités graphiques de la signature, sont des axes de prospections envisageables.

## 8 Bibliographie

- [Han 2000] Hangai.S, Yamanaka.S et Hamamoto.T., "On Line Signature Verification Based on Altitude and Direction of pen Movement", ICME, International Conference on multimedia and Expo, 2000.
- [Jain 2002] Jain. A.K, Griess.F.D et Connell.S.D., "On Line Signature Verification", Pattern Recognition letters, vol 35, 2002, p. 2663-2972.
- [Jain 99] Jain. A.K, Pankanti.S et Bolle.R., "Biometrics", New York, Edition Kluwer, 1999.
- [Nel 93] Nelson.W, Turin.W et Hastie.T., "Statistical methods for on line signature verification". Progress in automatic signature verification, New york, Editions World Scientific, 1994, p. 109-130
- [Ohi 2000] Ohishi. Y.K.T et .Matsumoto.T., "On line Signature Verification Using Pen Position, Pen Pressure and Pen Inclination trajectories", ICPR, International Conference on Pattern Recognition, 2000, p. 547-550.
- [Plam 2000] Plamondon.R et Srihani.S, "On Line and Off line handwriting Recognition : A comprehensive survey", IEEE Trans. on Pattern Analysis and Machine intelligence, vol 22, 2000, p. 63-84.
- [Plam 94] Plamondon.R et Parizeau.MA., "Signature verification from position, velocity and acceleration signals: a comparative study", Progress in automatic signature verification, New york, Editions World Scientific, 1994,
- [Plam 93] Plamondon.R., "The Design of an On Line Signature Verification System: From Theory to Practice", Progress in Automatic Signature Verification, New york, Editions World Scientific, 1994.
- [Wu 98] Wu. Q.Z, Lee. S.Y et Jou.I.C., "On Line Signature Verification Based ON Logarithmic Spectrum", Pattern Recognition Letters, 1998, Vol 31, p. 1871-1998.
- [Wu 97] Wu.Q.Z, Yin.S et Jou. I.C., "On line Signature Verification based on Split and Merge Matching Mechanism". Pattern Recognition letters, 1997, vol 18, p. 665-673.
- [Zim 2003] Zimmer.A, Luan Ling.L., "A Hybrid On/Off line Handwritten Signature Verification System", ICPR, International Conference on Pattern Recognition, 2003, p. 424-428.