

# Reconnaissance de l'écriture manuscrite arabe et latine par des modèles de Markov cachés multi-Flux

Arabic and Latin handwriting recognition  
using multi-stream hidden Markov models

Yousri Kessentini<sup>1,2</sup>, Thierry Paquet<sup>1</sup> et AbdelMajid Ben Hamadou<sup>2</sup>

<sup>1</sup>Laboratoire LITIS EA 4108, université de Rouen, France  
{yousri.kessentini, thierry.paquet}@univ-rouen.fr

<sup>2</sup>Laboratoire MIRACL, université de Sfax, Tunisie  
abdelmajid.benhamadou@isimsf.rnu.tn

Manuscrit reçu le 15 décembre 2009

Résumé et mots clés

Dans cet article nous proposons une approche de reconnaissance de l'écriture manuscrite. L'objectif étant de proposer un système indépendant de la nature du script, nous procédons alors sans segmentation. Des caractéristiques bas niveaux, basées sur les directions des contours et les densités de pixels, sont combinées à travers une approche multi-flux. Nous évaluons l'apport de l'approche multi-flux ainsi proposée et nous la comparons aux approches classiques de combinaison par fusion de représentations et par fusion de décisions. Pour valider l'approche proposée nous avons effectué des expérimentations sur deux bases de données de référence, la base de mots arabes IFN/ENIT et la base IRONOFF de mots latins. Les résultats montrent que le système proposé donne de bons résultats comparables aux meilleurs approches rapportées dans la littérature, aussi bien sur le Latin que sur l'Arabe.

Reconnaissance de l'écriture manuscrite hors-ligne, écriture arabe, écriture latine, combinaison d'information, modèles de Markov cachés multi-flux.

Abstract and key words

In this paper, we present a multi-stream approach for off-line handwritten word recognition. The multi-stream formalism presents many advantages: it can combine several kinds of independent features. The combination can be adaptive: some sources of information can be weighted, or even rejected if they are not reliable. The topology of the HMM can be adapted to each source of information. It also allows asynchronous modelling of streams.

The proposed approach combines low level feature streams namely, density based features extracted from 2 different sliding windows with different widths, and contour based features extracted from upper and lower contours.

Significant experiments have been carried out on two publicly available word databases: IFN/ENIT benchmark database (Arabic script) and IRONOFF database (Latin script).

In order to model the Latin characters, we built 26 uppercase character models and 26 lowercase character models. In the case of Arabic characters, we built up to 159 character models. An Arabic character may actually have different shapes according to its position within the word (beginning, middle, end word position). Other models are specified with additional marks such as "shadda". In both Latin and Arabic script, each character model is composed of 4 emitting states. The observation probabilities are modelled with Gaussian Mixtures (3 per state). Embedded training is used where all character models are trained in parallel using Baum-Welch algorithm applied on word examples. The system builds a word HMM by concatenation of the character HMM corresponding to the word transcription of the training sample.



The recognition step is done allowing the HMM-recombination algorithm that consists in building the product HMM and using a classical Viterbi decoding algorithm. We investigate the extension of 2-stream approach to  $N$  streams ( $N=2, \dots, 4$ ) and analyze the improvement in the recognition performance. The computational cost of this extension is discussed. The developed system has been tested on two publicly available databases. For both scripts the results show significant improvement while using a multi-stream approach. The comparison of the multi-stream performances to the classical combination strategies namely, fusion of features and fusion of decisions shows the superiority of the multi-stream approach. Moreover, the proposed recognition system provides significant results comparable to the best results reported in the literature on both databases.

Off-Line handwriting recognition, Hidden Markov Models, Latin script, Arabic script, multi-stream, information combination.

## 1. Introduction

**S** Ces dernières années ont connu une explosion du nombre de documents papier générés par des activités administratives et économiques. Pour faciliter l'archivage, le traitement et le transfert de ces documents, des systèmes de Gestion Electronique de Documents ont été développés. Les documents papier sont alors numérisés et peuvent être stockés et transférés électroniquement. Dans ces conditions, le traitement automatique de documents, qui vise à automatiser la lecture des contenus, a connu un essor rapide. De nombreuses approches de reconnaissance sont en effet proposées dans la littérature pour tenter de résoudre ces problèmes. Parmi ces études quelques approches sont maintenant mises en œuvre dans des applications industrielles de reconnaissance de chèques [KNE 98], ou d'adresses [SRI 00] [EL-Y 02].

Cependant, ces avancées méthodologiques ne permettent toujours pas de solliciter les moteurs de reconnaissance dans des conditions plus générales telles que la reconnaissance de texte par exemple, car les performances sont encore insuffisantes pour travailler sur de grands lexiques.

Les modèles de Markov cachés (MMC) sont devenus un outil très populaire en reconnaissance d'écriture manuscrite grâce d'une part, à leur capacité d'intégration du contexte et absorption du bruit, et d'autre part à la grande expérience accumulée dans l'utilisation de ces modèles pour la reconnaissance automatique de la parole [BEL 97, VIN 04]. Leur application à la reconnaissance du signal d'écriture En-Ligne acquis au moyen d'une tablette ou d'un stylo électronique a été très largement étudiée. Il convient de souligner que dans ce cadre les MMC de séquences sont particulièrement appropriés pour modéliser le signal mono-dimensionnel. En revanche l'application de ces mêmes modèles à la reconnaissance d'images d'écriture n'est pas aussi naturelle puisqu'elle oblige à transformer l'information d'entrée bi-dimensionnelle en une représentation mono-

dimensionnelle. Dans ce cadre dit Hors-Ligne, le module d'extraction de caractéristiques a donc une grande importance. C'est pourquoi nous pensons que de ce point de vue il est nécessaire et possible de faire progresser les approches Hors-Ligne utilisant les MMC afin de les rendre plus adaptées à la modélisation des informations bi-dimensionnelles. Dans cette étude nous nous intéressons à la combinaison de différentes sources d'informations mono-dimensionnelles dans la perspective de mieux modéliser la représentation 2D.

Traditionnellement, deux grandes approches sont envisagées en matière de combinaison d'information : la fusion des représentations [OKA 98] et la fusion des décisions [PRE 03]. La première famille rassemble les techniques utilisant une projection des représentations des différentes sources d'informations dans un espace de représentation conjoint avant d'utiliser des modèles classiques, comme les modèles de Markov cachés (MMC). Ceci conduit dans ce cas à un espace de dimension élevée, d'où une augmentation de la variance lors de l'estimation des paramètres des modèles et donc une diminution des performances du système de reconnaissance. Les méthodes de la deuxième famille d'approches s'appuient sur la combinaison de décisions prises indépendamment sur les différentes sources d'informations. Plusieurs travaux ont été réalisés dans le domaine de la combinaison de classificateurs, en particulier les travaux de [PRE 03, KIM 00] qui ont montré l'importance d'avoir des solutions robustes pour les problèmes de reconnaissance de l'écriture manuscrite.

Dans la littérature, et plus précisément dans le domaine de la reconnaissance de la parole, il a été proposé d'envisager la combinaison d'information au niveau des sous-unités mises en jeu dans les modèles. Alors que la décision finale est prise au niveau des mots, la combinaison des sources d'information est envisagée au niveau des modèles de caractères ou de phonèmes. Parce que ces approches permettent de désynchroniser les différentes sources d'informations, elles laissent envisager la possibilité de mieux modéliser la spatialisation des différentes caractéris-

tiques. En ce sens, leur application à la reconnaissance de l'écriture Hors-Ligne apparaît prometteuse. Ces approches basées sur une modélisation par des modèles de Markov cachés, sont désignées sous le vocable d'approches multi-flux. Les avantages potentiels de ces approches sont multiples [BOU 96] :

- Elles offrent un moyen pour combiner différentes sources d'information.
- Cette combinaison peut être adaptative, certaines sources d'informations pouvant être sous pondérées, voir rejetées si elles sont identifiées comme très peu fiables.
- La topologie des modèles de Markov cachés peut être adaptée à chaque source d'information.
- Les différents flux peuvent se désynchroniser jusqu'à certains points lexicaux prédéfinis.

Malgré toutes ces possibilités, l'utilisation des modèles multi-flux dans le domaine de la reconnaissance de l'écriture est restée très limitée. Dans ce cadre, des travaux récents ont été proposés pour la reconnaissance de l'écriture En-Ligne. Dans [GAU 01, ART 03] les modèles multi-flux sont utilisés pour combiner des représentations temporelles et spatiales du signal d'écriture En-Ligne.

Dans cette étude, nous proposons de combiner différentes sources d'informations à travers une approche multi-flux. Nous comparons l'approche multi-flux aux méthodes classiques de combinaison par fusion des représentations et par fusion des décisions. L'évaluation du système de reconnaissance est établie sur deux bases de mots publiques.

La présentation est organisée comme suit : dans la section 2, nous décrivons le formalisme de l'approche multi-flux ainsi que les techniques d'apprentissage et de décodage utilisées. La section 3 est consacrée à la description de la phase de prétraitement et d'extraction des caractéristiques. Une évaluation expérimentale est donnée dans la section 4 présentant les résultats de reconnaissance obtenus sur deux bases de données de référence: la base de mots arabes IFN/ENIT et la base IRONOFF pour les mots latins. Finalement dans la section 5, nous terminons par une conclusion et quelques perspectives.

## 2. L'approche multi-flux

L'approche multi-flux initialement proposée dans [WEL 98, BOU 97] est une méthode adaptative permettant de combiner différentes sources d'information en utilisant des modèles de Markov coopératifs. Les modèles des différentes sources sont traités indépendamment jusqu'à certains points d'ancrage<sup>1</sup> (désignés par  $\otimes$  sur la Figure 1) où ils sont contraints à se re-synchroniser et à combiner leurs contributions partielles. Les états de recombinaisons permettent donc d'introduire une certaine dépendance entre les sources d'information tout en permettant de désynchroniser ces mêmes sources pour tous les autres états du modèle.

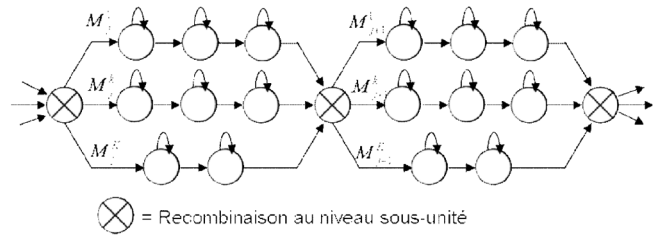


Figure 1. Structure générale d'un modèle multi-flux.

### 2.1. Formalisme

Soit  $K$  le nombre de sources d'informations et soit  $M$  le modèle composé d'une séquence de  $J$  sous-modèles qui correspondent à des sous-unités lexicales  $M_j$  ( $j = 1, \dots, J$ ) (des lettres par exemple). Chaque sous-modèle  $M_j$  est composé de  $K$  modèles de Markov cachés (MMC) notés  $M_j^k$ . Dans le cadre des modèles multi-flux, les  $K$  modèles des  $K$  sources d'information d'une même sous-unité sont considérés indépendants entre eux. En revanche la modélisation multi-flux introduit une dépendance entre les sources d'information en imposant un synchronisme entre les  $K$  sources lors des transitions entre sous-unités. Ainsi, si  $X$  désigne la séquence d'observations multi-flux, celle-ci peut se décomposer en  $J$  sous-séquences associées chacune au sous-modèle  $M_j$  et composées chacune de  $K$  sources d'information synchrones lors des transitions entre sous-unités (entre les lettres dans le cas qui nous intéresse). On note alors :  $X = \{(X_1^k, X_2^k, \dots, X_j^k, X_j^k), k = 1, \dots, K\}$  les  $K$  flux d'information, où  $X_j^k$  désigne la séquence d'observation de la source  $k$  correspondant au sous modèle  $j$ .

Si  $C_j$  désigne la séquence d'états multi-flux associée à la séquence d'observations multi-flux  $X_j$ , alors par définition du modèle multi-flux, la vraisemblance  $P(X_j, C_j | M_j)$  est calculée sur la base des vraisemblances associées à chacune des sources d'information selon la relation suivante :

$$P(X_j, C_j | M_j) = f(\{P(X_j^k, C_j^k | M_j^k), k = 1, \dots, K\}) \quad (2.1)$$

Par exemple, pour une fonction de combinaison  $f$  linéaire, la vraisemblance de la sous-séquence  $X_j$  par rapport au modèle de sous-unité  $M_j$  et pour le chemin  $C_j$  s'écrit :

$$\log P(X_j, C_j | M_j) = \sum_{k=1}^K a_j^k \log P(X_j^k, C_j^k | M_j^k) \quad (2.2)$$

où  $a_j^k$  représente la fiabilité du flux  $k$  pour la sous unité  $j$ . On déduit alors la vraisemblance de la séquence  $X$  en sommant sur toutes les sous-unités.

$$\log P(X, C | M) = \sum_{j=1}^J \sum_{k=1}^K a_j^k \log P(X_j^k, C_j^k | M_j^k) \quad (2.3)$$

1. États de combinaison correspondants aux frontières des caractères ou des syllabes par exemple.

## 2.2. Apprentissage des MMC multi-flux

Apprendre les modèles multi-flux se fait en deux étapes. Il faut tout d'abord apprendre les paramètres des MMC des différents flux (probabilités de transitions, vecteurs moyens, matrice de covariance et poids des gaussiennes) et estimer leurs poids de combinaison.

Les paramètres des MMC multi-flux peuvent être estimés soit indépendamment soit conjointement. Dans la première approche, les paramètres des MMC sont estimés indépendamment dans chaque flux avant d'effectuer le produit des probabilités. La deuxième approche consiste à profiter de la représentation sous forme de MMC produit pour estimer conjointement tous les paramètres du modèle.

Dans ce travail, nous avons opté pour un apprentissage indépendant des flux. Chaque caractère est donc modélisé par un MMC et les modèles des différents flux sont entraînés par maximisation de la vraisemblance grâce à l'algorithme de Baum-Welch. Le nombre d'états dans le modèle est le même pour chaque caractère. La Topologie est de type gauche-droite, n'autorisant que les transitions bouclantes et vers l'état suivant. Les densités de probabilité des observations dans chaque état sont modélisées par un mélange de gaussiennes. L'apprentissage des différents modèles est effectué de manière indépendante. L'apprentissage est embarqué: tous les modèles de caractères sont entraînés simultanément sur l'ensemble des mots de la base d'apprentissage. Une fois appris, ces modèles pourront alors être concaténés de manière à obtenir le modèle correspondant à chaque mot du lexique.

Apprentissage des poids de combinaison

Dans l'approche multi-flux proposée, la combinaison des décisions locales sur les états de recombinaison est assurée par une fonction de combinaison linéaire en logarithme de vraisemblance (c'est-à-dire par moyenne géométrique pondérée de vraisemblances). Dans ce cas, l'estimation par maximisation de vraisemblances des poids de combinaison échoue. En effet, la vraisemblance globale des données d'entraînement s'écrit :

$$P(X, C/M) = \prod_{k=1}^K \prod_{j=1}^J P(X_j^k, C_j^k / M_j^k)^{\alpha_k} \quad (2.4)$$

$$= \prod_{k=1}^K \prod_{j=1}^J (P_k)^{\alpha_k}$$

Maximiser  $P(X, C/M)$  sous la contrainte  $\sum_k \alpha_k = 1$  conduit à la solution suivante :

$$\begin{cases} \alpha_k = 1 & \text{si } k = \arg \max_k P_k \\ \alpha_k = 0 & \text{sinon} \end{cases} \quad (2.5)$$

Dans [POT 98], les auteurs introduisent des contraintes supplémentaires pour remédier à ce problème. Dans notre cas, nous

avons choisi d'optimiser les paramètres de combinaison selon les deux stratégies suivantes :

- Approche de combinaison par poids égaux : cette stratégie consiste à attribuer des poids égaux pour les différents flux. Les poids sont indépendants des flux et fixés *a priori* à :

$$\alpha_k = \text{cte} = \frac{1}{K} \quad (2.6)$$

$K$  étant le nombre de flux dans le modèle. L'avantage de cette stratégie est qu'elle ne nécessite ni de données ni de temps additionnel pour estimer les poids.

- Approche par fréquence relative : dans ce cas, les poids sont estimés à partir de la segmentation forcée obtenue par l'algorithme de décodage multi-flux où l'on calcule le rapport entre le nombre de fois où le modèle du flux considéré est de vraisemblance maximum par rapport aux autres modèles des autres flux, et le nombre d'occurrence du caractère concerné dans la base d'apprentissage.

$$\alpha_j^k = \frac{n_{k,j}}{n_j} \quad (2.7)$$

où  $n_{k,j}$  est le nombre de trames dans les données d'apprentissage ou le modèle du flux  $k$  à la vraisemblance maximum pour l'unité lexicale  $j$ , et  $n_j$  le nombre d'occurrences de l'unité lexicale  $j$  dans la base d'apprentissage. Cela revient donc à pondérer chaque source d'information en fonction de la fiabilité du modèle qui lui correspond.

À ce niveau, on détermine un poids pour chaque modèle de caractère. Cette approche peut être généralisée pour affecter un poids à chaque état dans le modèle de caractère. Il faut noter que l'évaluation de ces deux stratégies d'optimisation des poids de combinaison (par fréquence relative ou à poids identiques) a montré des résultats similaires dans le cadre de la reconnaissance de mots manuscrits.

## 2.3. Décodage d'un modèle multi-flux

Le processus de décodage des modèles multi-flux revient à calculer la vraisemblance du meilleur chemin de la séquence de vecteurs d'observations des différentes sources d'informations. Il s'agit donc d'un décodage selon le principe du calcul du meilleur chemin, semblable à l'algorithme de Viterbi. Pour ce faire, deux solutions sont envisageables :

1- La combinaison au niveau état: bien qu'elle ne permette pas l'asynchronie entre les différents flux, l'implémentation de cette méthode est relativement simple et revient à appliquer le décodage standard de Viterbi pour lequel les vraisemblances locales sont obtenues en combinant les vraisemblances relatives de chaque flux à tous les instants.

2- La combinaison au niveau sous-unités lexicales: cette méthode de combinaison permet de contrôler le degré d'asynchronie entre les différents flux de données. Le décodage nécessite dans ce cas une adaptation sérieuse de l'algorithme de déco-

dage classique. Dans la littérature, deux algorithmes ont été proposés pour résoudre ce problème :

- Algorithme de programmation dynamique sur deux niveaux dit «Two-Level» [SAK 79]: le décodage est séparé en deux niveaux imposant deux phases de traitement. Un premier processus de programmation dynamique opère au niveau des sous-unités et permet de déterminer un score pour chaque sous-modèle pour toutes les positions possibles de début et de fin des caractères sur les trames. Un deuxième processus de programmation dynamique détermine, en se servant des résultats du premier processus, la meilleure séquence de sous unités.

- Algorithme dit «HMM-recombination» [BOU96]: cet algorithme est une adaptation de l'algorithme «HMM-decomposition» [VAR 90] utilisé dans le domaine de la reconnaissance de la parole permettant de décomposer le signal audio en deux composantes indépendantes (son et bruit). C'est ce dernier algorithme qui a été retenu dans ce travail et dont le principe est de construire un MMC composite à partir des MMC parallèles du modèle multi-flux (voir Figure 2). Le MMC composite (voir Figure 3) est considéré comme le produit des MMC associé à chaque flux de données. Chaque état du MMC composite correspond à un  $K$ -uple d'états des  $K$  modèles coopératifs. La topologie de ces modèles composites possède donc  $K$  dimensions et est définie de façon à pouvoir représenter tous les schémas possibles étant données les topologies des modèles coopératifs initiaux. Les contributions locales d'un état composite sont définies comme une combinaison des contributions locales des états intervenant dans cet état composite. La combinaison peut donc se faire à chaque trame et le décodage est tout à fait classique.

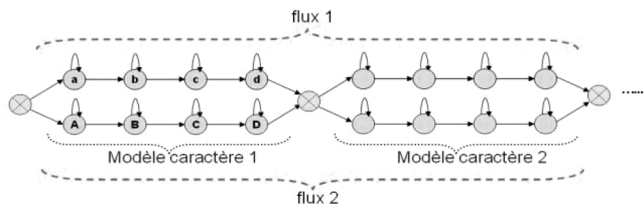


Figure 2. Topologie d'un modèle de Markov caché à 2 flux.

Dans le modèle produit, la log-vraisemblance conditionnelle des données pour l'état  $i$  est donnée par :

$$\log P(X_1(t), X_2(t)/i) = \alpha_i^1 \log P(X_1^i(t)/S_1(i)) + \alpha_i^2 \log P(X_2^i(t)/S_2(i)) \quad (2.8)$$

Où  $X_1(t)$  et  $X_2(t)$  sont les vecteurs de caractéristiques sur les deux flux à l'instant  $t$  et  $S_1(i)$  et  $S_2(i)$  sont des fonctions qui désignent l'état du MMC du flux 1 (resp. flux 2) correspondant à l'état produit  $i$ .

Les probabilités de transitions de chaque état du modèle produit sont déterminées à partir des probabilités de transitions des états correspondants dans le modèle parallèle. Par exemple, la transition de l'état  $a - A$  vers l'état  $a - B$  dans le MMC composite correspond à une transition de l'état  $a$  vers lui-même sur le

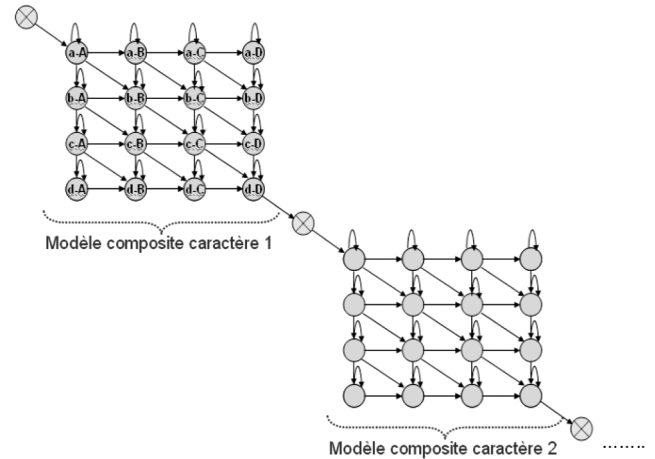


Figure 3. Topologie du modèle MMC résultant du modèle à 2 flux de la Figure 2.

MMC 1 du modèle parallèle et de l'état A vers l'état B dans celui du MMC 2. La probabilité de transition  $P(a - A/a - B)$  se calcule alors comme étant le produit des probabilités de transition  $P(a/a)$  et  $P(A/B)$  grâce à l'hypothèse d'indépendance des deux flux entre les états de synchronisation.

Notons que sur la Fig. 3, les mouvements gauche-droite correspondent à des transitions dans la chaîne  $\{A, B, C, D, E\}$ , les mouvements haut-bas correspondent à des transitions dans la chaîne  $\{a, b, c, d, e\}$ , et les mouvements diagonaux à des transitions simultanées sur les deux flux d'états.

## 2.4. Extension au cas à $N$ -flux

Combiner  $K$  flux de caractéristiques à travers une approche multi-flux revient à construire le MMC produit défini par ses états composites, ses probabilités de transitions et d'émissions. Soit  $S^i = \{s_j^i; j = 1, \dots, n_i\}$  l'ensemble des états du modèle mono-flux  $i$ . Le MMC composite est défini dans l'espace produit des états noté comme suit :

$$S^\otimes = S^1 \otimes S^2 \dots \otimes S^K = \{s_i^\otimes = (s_j^1, s_m^2, \dots, s_l^K); j = 1, \dots, n_1, m = 1, \dots, n_2, l = 1, \dots, n_k\}$$

Dans ce modèle, la log-vraisemblance conditionnelle des données pour l'état  $S_i^\otimes$  est calculée par une combinaison linéaire des log-vraisemblance de chaque flux :

$$\log P(X_1(t), X_2(t), \dots, X_K(t)/S_i^\otimes) = \sum_{k=1}^K \alpha_i^k \log P(X_k(t)/f[k](S_i^\otimes)) \quad (2.9)$$

$f$  est une fonction qui désigne l'état  $s_j^k$  du MMC du flux  $k$  et correspondant à l'état produit  $S_i^\otimes$  tel que  $f[k](S_i^\otimes) = s_j^k$ .

Les probabilités de transitions de chaque état du modèle produit sont déterminées à partir des probabilités de transitions des états correspondants dans le modèle parallèle par un produit des pro-

babilités de transition grâce à l'hypothèse d'indépendance des flux entre les états de recombinaison.

$$P(S_i^{\otimes} | S_j^{\otimes}) = \prod_{k=1}^K P(f[k](S_i^{\otimes}) | f[k](S_j^{\otimes})) \tag{2.10}$$

Le nombre d'états d'un modèle à  $K$  flux croit exponentiellement avec  $K$  puisqu'on travaille dans l'espace produit des états. Pour limiter la complexité d'un modèle à  $K$ -flux, deux solutions peuvent être envisagées.

1) Réduire le nombre d'états par caractère dans les MMCs parallèles : En effet, réduire le nombre d'états par caractère dans les MMC parallèles de la figure 2 de 4 à 3 permet d'obtenir des MMC composite à 9 états par caractère au lieu de 16 dans le cas de 2 flux, et à 27 états par caractère au lieu de 64 dans le cas de 3 flux etc...

2) Limiter l'asynchronie entre les flux de données en restreignant l'espace produit aux seuls états (produits) valides : on mesurera l'asynchronie, dans un état produit comme la valeur absolue de la différence entre les numéros des états des éléments du produit. En réduisant le degré d'asynchronie à 1 dans le MMC produit de la figure 3, on obtient le MMC composite donné par la figure 4.

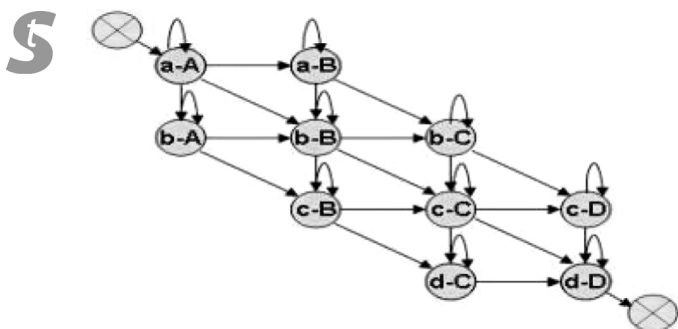


Figure 4. MMC composite à 1 degré d'asynchronie.

### 3. Approche proposée

Dans ce travail, nous proposons un système de reconnaissance hors-ligne de mots manuscrits multi-scripts et mettons en évidence l'apport d'une approche multi-flux. Nous présentons dans ce qui suit la chaîne de prétraitements et de caractérisation des images d'écriture.

#### 3.1. Prétraitements

Avant de procéder à la reconnaissance des mots, il est nécessaire d'effectuer une série de prétraitements sur l'image de celui-ci en vue d'éliminer (ou du moins de réduire) les sources de variabilité ou de bruit, et également de simplifier la procédure

d'extraction de caractéristiques. Nous appliquons une procédure de normalisation dans le but de ramener l'orientation du signal d'écriture à l'horizontale [KIM 94]. Cette normalisation s'effectue en trois étapes : une estimation de l'inclinaison est tout d'abord réalisée d'après le contour du mot. Ensuite l'image est corrigée par décalage des lignes de l'image (transformation par cisaillement, ou «shearing transform»). Cette transformation génère du bruit sur le contour de l'image. On applique alors une dernière étape de lissage du contour par une opération de morphologie mathématique : la fermeture.

L'estimation de l'inclinaison des lettres est réalisée par une analyse du contour du mot. Ce contour est extrait par une méthode de suivi de contour et est encodé sous la forme de «chaincode» selon les 8 directions de freeman [KIM 94]. Quatre directions sont envisagées pour les segments de contour (0°, 45°, 90° et 135°). L'inclinaison  $\alpha$  est alors calculée comme suit :

$$\alpha = \arctan \frac{n_1 + n_2 + n_3}{n_1 - n_3} \text{ où } n_i \text{ est le nombre d'occurrences de la direction } i \text{ dans le chaincode.}$$

#### 3.2. Extraction des caractéristiques

Deux ensembles de caractéristique sont proposés dans ce travail, un premier ensemble basé sur les contours supérieur et inférieur de l'image et un deuxième basé sur les densités. Ces deux familles de caractéristiques ont été retenues pour leurs bonnes performances pour la reconnaissance de l'écriture en caractère latin (contours) et de l'écriture arabe (densité). Nous précisons ces caractéristiques dans les paragraphes qui suivent.

##### 3.2.1 Caractéristiques contours

Ces caractéristiques sont basées sur les points du contour du mot. Ces points sont séparés en un premier ensemble formant le contour supérieur et un second formant le contour inférieur du mot (voir Fig. 5). Lors de l'extraction des caractéristiques, nous travaillons sur des fenêtres glissantes positionnées sur les contours. Une fenêtre est caractérisée par une largeur et un taux de recouvrement entre deux positions successives. Pour chaque position de la fenêtre, nous déterminons l'histogramme des directions de Freeman des points du contour supérieur (resp. inférieur) dans cette fenêtre. On obtient alors un premier ensemble de 8 caractéristiques.

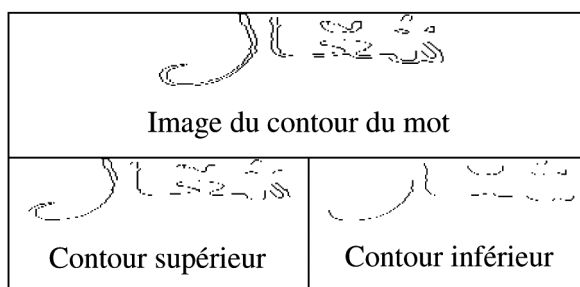


Figure 5. Images des contours du mot.

Un deuxième ensemble de caractéristiques est défini par la caractérisation en chaque point du contour supérieur (resp. inférieur) du point de contour en vis-à-vis selon la direction verticale descendante (resp. ascendante). Quatre cas sont possibles :



Figure 6. Différents types de points du contour supérieur.

- Ce point appartient au contour inférieur (resp. supérieur) (points rouges sur la Fig.).
- Ce point appartient à une occlusion (points bleus sur la Fig.).
- Ce point appartient au contour supérieur (resp. inférieur) (points jaunes sur la Fig.).
- Pas de point, ce qui veut dire que le point du contour supérieur (resp. inférieur) est un point extrême dans la direction horizontale (points verts sur la Fig.).

Les points noirs sur la Figure 6 représentent les points du contour inférieur.

Ces 4 catégories permettent de caractériser les occlusions, les rebroussements horizontaux (barre de 't' par exemple), les traits simples et les points extrêmes sur l'image du mot. Ces informations locales sont ensuite sommées par catégorie sur toute la largeur de la fenêtre d'analyse pour constituer un second histogramme formant ainsi un deuxième ensemble de 4 caractéristiques.

Un troisième ensemble de caractéristiques indique la position des points du contour supérieur (resp. inférieur) dans la fenêtre. Pour cela, nous localisons la zone médiane de l'image du mot en se basant sur l'analyse de l'histogramme de projection horizontale. Cette opération consiste principalement à déterminer la ligne de base supérieure et la ligne de base inférieure dans l'image du mot en appliquant la méthode décrite dans [BOZ 89]. Ainsi, nous déterminons pour chaque point du contour supérieur (resp. inférieur) sa position dans la fenêtre, 3 cas sont possibles : 1) le point appartient à la zone médiane, 2) le point appartient à la zone supérieure, 3) le point appartient à la zone inférieure (voir Figure 7).

Un histogramme est calculé pour chaque fenêtre d'où un troisième ensemble de 3 caractéristiques.

Nous disposons ainsi de deux séquences d'observations constituées chacune d'un vecteur de 15 caractéristiques continues, l'une provenant du contour supérieur, l'autre du contour inférieur.

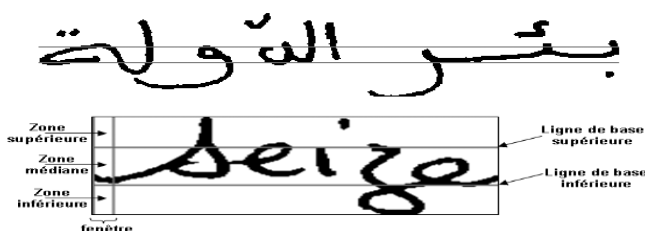


Figure 7. Détection des lignes de base.

### 3.2.2 Caractéristiques de densités

Un deuxième ensemble de caractéristiques est construit par l'analyse des densités de pixels selon la méthode proposée dans [HAJ 05]. L'image est balayée par une fenêtre glissante de largeur  $w$  (voir Figure 8). Chaque fenêtre est divisée verticalement en cellules de hauteur  $h$  ( $h$  est déterminée empiriquement et fixée à 4 pixels dans notre cas) et les caractéristiques suivantes sont extraites :

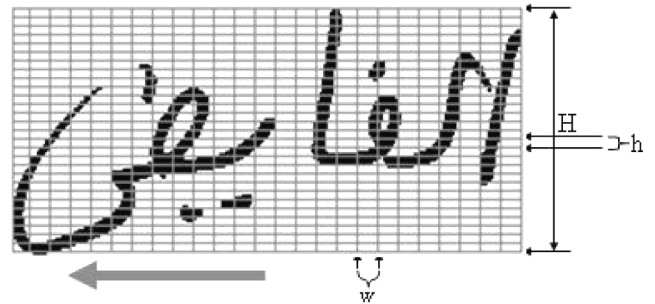


Figure 8. Exemple d'une image de mot divisée en cellules.

- $f_1$  : densité des pixels noirs dans la fenêtre.
- $f_2$  : nombre de transitions Noir/Blanc entre cellules.
- $f_3$  : différence de position verticale entre les centres de gravité des pixels d'écriture dans deux fenêtres consécutives.
- $f_4$  à  $f_{11}$  : sont 8 densités de pixels d'écriture dans chaque colonne de la fenêtre (ici une fenêtre est de largeur 8).

Les caractéristiques suivantes dépendent de la position estimée des lignes de base.

- $f_{12}$  : position verticale normalisée du centre de gravité des pixels d'écriture dans la fenêtre, par rapport à la ligne de base basse.
- $f_{13}$  -  $f_{14}$  : densités des pixels d'écriture au-dessus et au-dessous de la ligne de base basse.
- $f_{15}$  : nombre de transitions Noir/Blanc entre les cellules situées au dessus de la ligne de base basse.
- $f_{16}$  : zone à laquelle appartient le centre de gravité de l'écriture dans la fenêtre (zone supérieure  $f_{16} = 1$ , zone médiane  $f_{16} = 2$ , zone inférieure  $f_{16} = 3$ ).
- $f_{17}$ - $f_{21}$  : liées aux configurations locales des pixels d'écriture dans chaque fenêtre (5 configurations possibles voir 9).
- $f_{22}$  à  $f_{26}$  sont liées aux configurations locales des pixels d'écriture situés dans la zone médiane de l'écriture.

Cet ensemble est indépendant du langage utilisé. Ces caractéristiques peuvent être utilisées pour tout type d'écriture cursive comportant des hampes et des jambages comme l'écriture arabe et l'écriture latine.

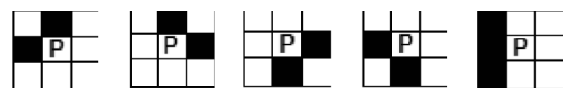


Figure 9. Les 5 types de configurations locales autour d'un pixel de fond P.

Ces deux ensembles de caractéristiques nous permettront d'extraire 4 flux de données basés respectivement sur des caractéristiques extraites sur les deux contours supérieur et inférieur et des caractéristiques de densités extraites avec deux tailles de fenêtre. Un apprentissage indépendant des modèles est réalisé par la suite sur les différents flux de données. Une fois appris, le décodage simultané des différents modèles est réalisé conjointement par coopération selon le formalisme multi-flux que nous avons présenté ci-dessus.

## 4. Expérimentations et analyses

Nous avons décrit ci-dessus les différentes étapes de notre système de reconnaissance de mots manuscrits. L'approche proposée est indépendante de la nature du script, elle opère sans segmentation explicite en graphèmes. Des caractéristiques bas niveaux sont extraites sur l'image du mot quelle que soit la langue. Nous analysons les performances des différents systèmes de reconnaissance que nous pouvons construire par combinaison multi-flux des différents jeux de caractéristiques proposés. Dans chacun des cas, l'évaluation du moteur de reconnaissance de mots est réalisée sur deux bases publiques différentes : la base de mots arabes IFN/ENIT [PEC 02] et la base de mots latins IRONOFF [VIA 99].

Durant toute la phase d'expérimentation, la topologie du modèle de caractère retenue est de type gauche-droite, n'autorisant que les transitions bouclantes et vers l'état suivant. Le nombre d'états dans le modèle est le même pour chaque caractère et est fixé à 4 états. Les densités de probabilité des observations dans chaque état sont modélisées par un mélange de 3 composantes gaussiennes. Pour modéliser l'écriture latine, 52 modèles ont

été construits correspondant aux caractères minuscules et majuscules. Dans le cas de l'écriture arabe ce nombre s'élève à 178 pour modéliser les différentes formes de caractères selon leur position dans le mot. C'est là la particularité principale de l'écriture arabe par rapport à l'écriture latine.

### 4.1. La base IFN/ENIT

La base de données de référence IFN-ENIT a été collectée et distribuée par l'IFN (Institute for Communications, de l'université de Braunschweig - Allemagne), et l'ENIT (École Nationale d'Ingénieurs de Tunis). Cette base comporte 32492 mots arabes écrits par 411 scripteurs. Les mots de cette base correspondent à un lexique de 946 noms de villes et villages tunisiens. Cinq ensembles distincts de données (*a, b, c, d, e*) sont prédéfinis dans la base IFN/ENIT, pour l'apprentissage et le test.

Pour tester notre moteur de reconnaissance sur la base IFN/ENIT, les sous-ensembles *a, b, c* et *d* ont été utilisés pour l'apprentissage et le sous-ensemble *e* pour le test. Le tableau 1 présente les résultats de reconnaissance obtenus pour les quatre modèles : contour supérieur, contour inférieur et les densités avec deux tailles de fenêtre différentes (8 pixels pour densité 1 et 14 pixels pour densité 2). Ces résultats sont obtenus en utilisant un décodage mono-flux avec l'algorithme de Viterbi.

Tableau 1. IFN/ENIT: Résultats de reconnaissance avec un seul flux.

Modèles	Rang			
	1	2	5	10
1) Contour supérieur	70,5	78,6	86,3	90,4
2) Contour inférieur	63,5	73,1	82,6	86,4
3) Densité 1	65,1	73	80,6	83,2
4) Densité 2	68,7	78,1	83,3	86,9

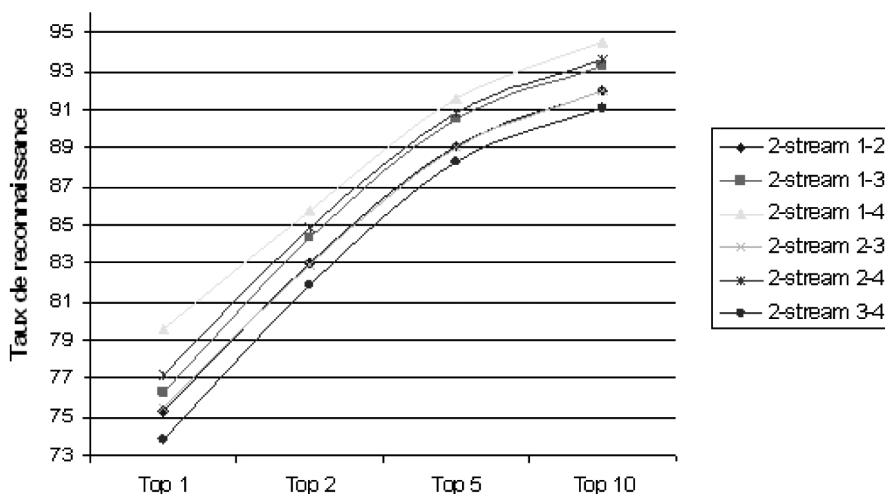


Figure 10. Résultats de reconnaissance avec 2 flux, base IFN/ENIT.



Les meilleurs résultats sont obtenus avec le modèle du contour supérieur avec 70,5 %. Ceci montre que le contour supérieur est le plus informatif pour la reconnaissance de mots manuscrits arabes.

Pour améliorer ces performances, nous avons combiné 2 à 2 ces 4 sources d'informations dans le cadre d'une approche 2-flux. Partant de 4 flux, différentes combinaisons sont alors possibles, les résultats obtenus sont graphiquement représentés sur la Figure 10.

Les résultats montrent que l'approche multi-flux améliore les taux de reconnaissance comparée à l'approche mono-flux. Le meilleur résultat est obtenu en combinant les flux correspondant au contour supérieur et Densité 2 avec un taux de reconnaissance de 79,6 %. Le gain ici est de 9,1 % comparé au meilleur taux obtenus par une approche mono-flux. Notons aussi que la combinaison de deux flux de nature différente (ici contour et densité) est plus intéressante que tout autre combinaison de flux. En ce sens, ces deux flux apparaissent les plus complémentaires pour la reconnaissance.

Des expérimentations supplémentaires ont été menées afin de comparer l'apport d'une approche multi-flux par rapport aux approches classiques de combinaison par fusion des représentations et par fusion des décisions de reconnaissance. Le tableau 2 présente les résultats obtenus pour ces différentes approches de combinaison. Notons que pour l'approche par combinaison des scores, la même règle de combinaison que celle appliquée dans le formalisme multi-flux est utilisée (*i.e* somme pondérée). Pour la fusion des représentations, un vecteur de taille 41 a été construit en concaténant les caractéristiques du contour supérieur avec celles de Densité 1, le même nombre de gaussiennes par état a été utilisé (*i.e* 3 gaussiennes).

Les résultats montrent la supériorité de l'approche multi-flux par rapport aux approches classiques de combinaison.

Tableau 2. IFN/ENIT: Comparaison de l'approche multi-flux aux approches par fusion des décisions et par fusion des représentations.

Modèles	Rang			
	1	2	5	10
Fusion des scores	75,4	83,2	89,5	92,2
Fusion des représentations	74,1	82,6	88,4	90,8
<b>2-flux</b>	<b>79,6</b>	<b>85,7</b>	<b>91,6</b>	<b>94,5</b>

Pour comparer nos résultats de reconnaissances aux autres systèmes travaillant sur la même base, nous reportons dans le tableau 3 les résultats obtenus lors de la compétition ICDAR'05 [MÄR 05]. Cinq groupes ont soumis leurs systèmes pour la compétition. Les systèmes participants ont été entraînés sur les quatre sous-ensembles a, b, c et d de la base de référence. Ces systèmes ont ensuite été testés sur le même sous-ensemble e

Tableau 3. IFN/ENIT: Comparaison avec les systèmes présentés dans la compétition ICDAR'05.

Système	Rang 1	Rang 5	Rang 10
ICRA	65,74	83,95	87,75
SHOCRAN	35,70	51,62	51,62
TH-OCR	29,62	43,96	50,14
UOB	75,93	87,99	90,88
REAM	15,36	18,52	19,86
ARAB-IFN	74,69	87,07	89,77
<b>Système proposé</b>	<b>79,6</b>	<b>91,6</b>	<b>94,5</b>

contenant 6033 images non connues par les groupes participants. Les résultats montrent bien la supériorité de notre système de reconnaissance qui dépasse le système ayant gagné la compétition [HAJ 05] de 3,67 % en Top 1.

Lors de l'édition ICDAR'07, la compétition a été reconduite [MÄR 07]. Les différents systèmes participants ont été entraînés cette fois sur les 5 sous-ensembles *a, b, c, d* et *e* et ont été testés sur un nouveau sous-ensemble *f*. À ce jour, ce dernier ensemble *f* n'est pas publiquement disponible, c'est pourquoi nous ne pouvons pas comparer nos résultats de reconnaissance à ceux de la compétition ICDAR'07. Par contre, d'autres tests ont été réalisés sur les sous-ensembles *d* et *e*, faisant partie de la base d'apprentissage sur laquelle les systèmes ont été entraînés. Bien que ceci ne soit pas complètement significatif, nous avons choisi de nous placer dans ces conditions pour comparer notre système aux systèmes proposés. Pour cela nous avons entraîné

Tableau 4. IFN/ENIT: Comparaison avec les systèmes présentés dans la compétition ICDAR'07 (ID 01: MITRE; IDs 02-04: CACI; ID 05: CEDAR; ID 06: MIE; IDs 07-08: SIEMENS; IDs 09-12: UOB-ENST; ID 13: ICRA; ID 14: PARIS V).

ID	Ensemble d	Ensemble e
	Rang 1	Rang 1
01	66,34	64,89
02	40,45	37,73
03	70,62	68,62
04	48,68	44,04
05	68,07	57,37
<b>06</b>	<b>93,63</b>	<b>86,67</b>
07	91,23	84,27
<b>08</b>	<b>94,58</b>	<b>87,77</b>
09	90,02	81,80
10	92,12	83,52
11	92,38	83,92
<b>12</b>	<b>93,32</b>	<b>85,13</b>
13	88,33	83,87
14	89,80	80,24
<b>Notre Système</b>	<b>92,82</b>	<b>84,33</b>

de nouveau notre système sur les 5 sous-ensembles *a, b, c, d* et *e* et nous l'avons testé sur les sous-ensembles *d* et *e*. Le tableau 4 présente une comparaison de notre système avec ceux rapportés dans la compétition de 2007 [MÄR 07].

Le système le plus performant est celui de Siemens [SCH 08] avec un taux de reconnaissance de 94,54% pour le sous-ensemble *d* et de 87,77% pour le sous-ensemble *e*. Notre système se positionne quatrième parmi les 14 systèmes avec des taux très proches des 3 meilleurs systèmes.

#### 4.2. La base IRONOFF

La base de mots IRONOFF est collectée et distribuée par l'université de Nantes (IRESTE, France), en collaboration avec la société VISION OBJECTS. C'est une base de données duale En-Ligne et Hors-Ligne. Elle a été collectée auprès d'environ 700 scripteurs différents et elle comporte 31 346 mots manuscrits isolés représentant un lexique de 197 mots. Les images sont en niveau de gris avec une résolution de 300 dpi. La base comporte 28 657 échantillons écrits en Français et 2689 échantillons écrits en Anglais.

Pour tester notre système sur la base IRONOFF, nous avons binarisé les images de la base en utilisant une méthode de seuillage global classique. 20,898 images ont été utilisées pour l'apprentissage et 10,448 pour le test. Nous présentons dans le tableau 5 les résultats de reconnaissance obtenus par les 4 flux utilisant un décodage mono-flux.

Nous remarquons que les résultats de reconnaissance du contour inférieur sont très faibles comparés aux autres flux.

Cela s'explique par le fait que le contour inférieur est généralement peu informatif sur les lettres latines. On notera que cette particularité n'a pas été observée sur l'Arabe. En revanche, alors que le contour supérieur donne des performances sur l'Arabe nettement supérieures aux autres flux, il n'y a pas vraiment un flux qui surpasse les autres dans le cas du Latin.

Pour améliorer ces résultats, les quatre sources d'informations sont combinées 2 à 2 dans le cadre d'une approche 2-flux. Les résultats obtenus sont graphiquement représentés sur la figure 11.

Tableau 5. IRONOFF: Résultats de reconnaissance avec un seul flux.

Modèles	Rang			
	1	2	5	10
1) Contour supérieur	81,2	86,8	91,4	94,6
2) Contour inférieur	69,8	78,4	88,6	93,4
3) Densité 1	81,8	87,6	92,6	95,6
4) Densité 2	80	86,2	92,8	95,8

Sur le Latin, les résultats confirment l'apport de l'approche multi-flux observé durant les expérimentations réalisées sur la base IFN/ENIT de mots arabes. Le meilleur taux de reconnaissance est de 89.8 % en combinant les deux flux contour supérieur et densité 1. Le gain est de 8% par rapport aux résultats obtenus par une approche mono-flux. De la même façon que sur l'Arabe, les résultats sur le Latin confirment la complémentarité des flux contour et densité.

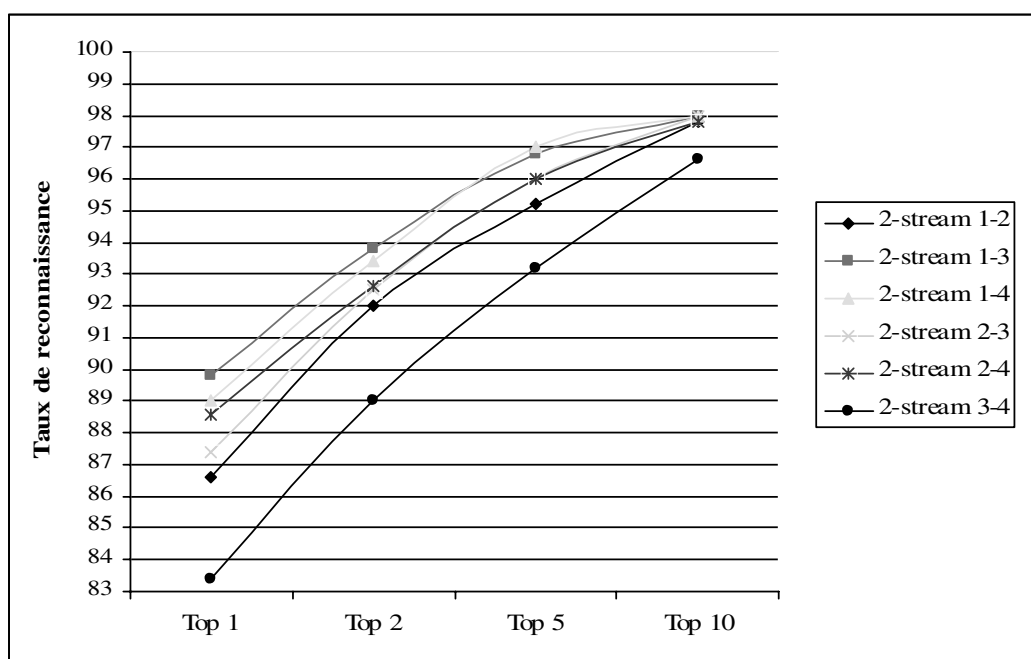


Figure 11. Résultats de reconnaissance avec 2 flux, base IRONOFF.

A nouveau, ces résultats sont comparés dans le tableau 6 aux résultats des approches classiques de combinaison par fusion de représentations et de décisions. Ce tableau montre la supériorité de l'approche multi-flux avec un gain de 3,8 % en Rang 1.

Tableau 6. IRONOFF: Comparaison de l'approche multi-flux aux approches par fusion des décisions et des représentations.

Modèles	Rang			
	1	2	5	10
<b>2-flux</b>	<b>89,8</b>	<b>93,8</b>	<b>96,8</b>	<b>98</b>
<b>Fusion des scores</b>	86	91,4	95	97,4
<b>Fusion des représentations</b>	85,2	90,6	94,4	97

Pour la base IRONOFF, il n'y a pas eu malheureusement de compétitions à l'échelle internationale. Afin d'apporter des éléments de comparaison, nous avons cependant collecté les principaux résultats qui sont rapportés sur cette base dans la littérature. Les taux de reconnaissance sont ainsi présentés dans le tableau 7 et confirment les bons résultats de notre système qui se place en deuxième position.

Tableau 7. IRONOFF: Comparaison aux systèmes présentés dans la littérature.

Auteurs	Performances	
	TOP 1	TOP 5
[Tay 01] système 1	86,6	94,2
[VIA 05] système 1	87,4	95,8
[VIA 05] système 2	89,8	97
[Tay 01] système 2	96,1	99,1
<b>Système proposé</b>	<b>89,8</b>	<b>96,8</b>

À travers ces différentes expérimentations réalisées, nous avons pu montrer l'apport de l'approche multi-flux proposée dans le cadre de la reconnaissance de l'écriture manuscrite multi-script. Pour mieux étudier les possibilités offertes par cette approche, nous avons cherché à combiner plus de deux flux. Nous étudions alors dans la suite l'apport de cette extension ainsi que la complexité et les limites de l'approche.

### 4.3. Résultats avec N-flux

Combiner  $K$  flux de caractéristiques à travers une approche multi-flux revient à généraliser l'algorithme HMM-Recombinaison de façon à combiner 2 à 2 les  $K$  flux et construire le MMC produit correspondant. Il est aisé de percevoir que dans le modèle composite, le nombre d'états par caractère croît considérablement en fonction du nombre de flux à combiner, et la méthode devient de plus en plus complexe. En

effet, la complexité de l'algorithme de Viterbi est de  $O(T.(C.N)^2)$ ,  $N$  étant nombre d'états par modèle de caractère,  $C$  est le nombre de caractères dans le modèle et  $T$  le nombre d'observations d'une trame par unité syntaxique. Dans le MMC produit, le nombre d'états par modèle de caractère croît en fonction du nombre de flux et est égale à  $C.N^K$ ,  $K$  étant le nombre de flux. La complexité de l'algorithme de décodage devient alors  $O(T.(C.N^K)^2) = O(T.C^2.N^{2K})$ . En conséquence, la complexité d'un algorithme de décodage multi-flux croît exponentiellement avec le nombre de flux.

Vu la complexité de la tâche, nous avons choisi de tester l'approche sur un lexique réduit. Les premières expérimentations sont réalisées sur le sous-ensemble IRONOFF-Chèque de la base IRONOFF, restreint à un petit lexique d'environ 30 mots utilisés pour exprimer les montants de chèques français. On dispose de 7956 images de mots pour l'apprentissage et de 3987 images pour le test. Les résultats obtenus avec 2, 3 et 4 flux sont donnés dans le tableau 8 en fonction du nombre d'états par caractère  $e/c$  (4  $e/c$  et 3  $e/c$ ).

Tableau 8. Performances sur la base IRONOFF-Chèque.

Modèles	4e/c		3 e/c	
	Top1	Top2	Top1	Top5
2-flux 1)2)	86	95,4	87	97,6
2-flux 1)3)	92,8	99,2	91	99,4
2-flux 2)3)	94	99,6	89,4	97,8
2-flux 3)4)	91	99	86,2	97,2
3-flux1)2)3)	94	99,6	91,6	<b>99,6</b>
3-flux1)3)4)	92,6	99,2	90,8	98,8
3-flux2)3)4)	93,4	99	89,4	98,4
3-flux1)2)4)	93,6	<b>99,8</b>	92	99
4-flux	<b>94,2</b>	99,2	<b>92,8</b>	99,2

Tableau 9. Performances sur l'ensemble de la base IFN/ENIT

Modèles	Lexicon size			
	30		100	
	Top1	Top2	Top1	Top5
1) Contour sup.	97,8	99,8	95,4	99
2) Contour inf.	97,8	99,8	93,4	99,2
3) Densité1	92	97,4	88,4	94
4) Densité2	92,6	98	88,8	94,4
2-flux 1)2)	98	99,8	95,6	99,8
2-flux 1)3)	98,6	99,8	98,4	99,4
2-flux 2)3)	98	<b>100</b>	96,8	99,8
2-flux 3)4)	97,6	99,6	96,4	98,6
3-flux1)2)3)	98,6	99,8	98,4	99,8
4-flux	<b>99,2</b>	99,8	<b>98,8</b>	<b>100</b>

Dans les mêmes conditions, d'autres expérimentations ont été réalisées sur la base de mots arabes IFN/ENIT. Les résultats obtenus sur un lexique réduit de 30 mots, puis de 100 mots sont reportés dans le tableau 9. La génération des lexiques est réalisée par tirage aléatoire: pour chaque mot à reconnaître, un lexique de taille  $N$  est obtenu par un tirage aléatoire de  $N - 1$  mots parmi le lexique complet, complété par l'étiquette du mot à reconnaître.

Les résultats obtenus montrent que l'ajout d'un flux améliore généralement la performance du système pour les deux scripts arabe et latin. Cependant, ceci augmente considérablement la complexité de l'algorithme de décodage. Des solutions pour optimiser et accélérer la procédure de décodage multi-flux ont été proposées dans la littérature [KOE 02, RAB 90] et peuvent être le sujet de futurs travaux tel l'algorithme level-building guidé par le lexique, l'algorithme de décodage rapide à deux niveaux, l'algorithme A\*...

## 5. Conclusion

Nous avons présenté dans ce travail une approche de reconnaissance basée sur une modélisation par des modèles de Markov cachés multi-flux. Le système proposé opère indépendamment de la nature du script. Pour cela il procède sans segmentation en combinant des caractéristiques bas niveaux basées sur les directions et les densités des pixels.

Des résultats significatifs sont obtenus, en ne combinant que 2 flux, sur deux bases de référence: la base de mots arabes IFN/ENIT et la base IRONOFF de mots latins. Les résultats montrent la supériorité de l'approche multi-flux par rapport à l'approche mono-flux ou aux approches classiques de combinaison par fusion des représentations et par fusion des décisions. Les performances obtenues sont comparables aux meilleurs résultats reportés dans la littérature sur les mêmes bases.

L'extension de l'approche à plus de deux flux a permis d'améliorer légèrement les performances, en contre partie, l'approche devient très complexe.

Comme perspectives à ce travail, nous envisageons de tester d'autres règles de combinaison. Nous proposons aussi de combiner d'autres sources d'information plus complémentaires dans le cadre de l'approche multi-flux. Une autre piste consiste à explorer la combinaison MMC et Modèle des Croyances Transférables (MCT) [SME 94] afin de bénéficier de la puissance des algorithmes des MMC et de la généralité des MCT dans le cadre de la reconnaissance de l'écriture manuscrite.

## Références

- [ART 03] T. ARTIÈRES, N. GAUTHIER, P. GALLINARI, B. DORIZZI, A Hidden Markov Models combination framework for handwriting recognition, *International Journal on Document Analysis and Recognition (IJ DAR)*, Vol 5, N° 4, p. 233-243, July (2003).
- [BEL 97] A. BELAID, G. SAON, Utilisation des Processus Markoviens en Reconnaissance de l'Écriture, *Traitement du Signal*, Vol 14, N° 2, p. 161-177 (1997).
- [BOU 97] H. BOURLARD, S. DUPONT, Sub-band-based Speech Recognition. In *IEEE Int. Conf. on Acoust., Speech, and Signal Processing*, p. 1251-1254 (1997).
- [BOU 96] H. BOURLARD, S. DUPONT, and C. RISS, Multi-stream speech recognition. Technical Report IDIAP-RR 96-07, IDIAP, 1996.
- [BOZ 89] R. M. BOZINOVIC and S. N. SRIHARI. «Off-line Cursive Script Word Recognition», *IEEE Transactions on PAMI*, Vol 11, N° 1, p. 68-83, January (1989).
- [EL-Y 02] M.A. EL-YACOUBI, M. GILLOUX & J.M. BERTILLE. A Statistical Approach for Phrase Location and Recognition within a Text Line: An Application to Street Name Recognition. *IEEE Trans. on PAMI*, vol. 24, no. 2, p. 172-188 (2002).
- [GAU 01] N. GAUTHIER, T. ARTIÈRES, B. DORIZZI, and P. GALLINARI, Strategies for combining on-line and off-line information in an on-line handwriting recognition system. *International Conference on Document Analysis and Recognition (ICDAR)*, p. 412-416 (2001).
- [HAJ 05] R. EL-HAJJ, L. LIKFORMAN-SULEM, C. MOKBEL, Arabic handwriting recognition using baseline dependent features and Hidden Markov Modeling, *ICDAR 2005*, Seoul, Corée du Sud.
- [KES 08] Y. KESSENTINI, T. PAQUET, A. BENHAMADOU. Combinaison d'Information pour la Reconnaissance de l'Écriture Manuscrite Hors-Ligne. *RFA'2008*, Amiens, France.
- [KIM 94] F. KIMURA, S. TSURUOKA, Y. MIYAKE & M. SHRIDHAR, «A Lexicon Directed Algorithm for Recognition of Un-constrained Handwritten Words». *IEICE Trans. Inf. & Syst.*, vol. E77-D, no. 7 (1994).
- [KIM 00] J. KIM, K. KIM, C. NADAL, and C. SUEN, A methodology of combining hmm and mlp classifiers for cursive word recognition. *International Conference Document Analysis and Recognition (ICDAR)*, vol 2, p. 319-322 (2000).
- [KNE 98] S. KNERR, E. AUGUSTIN, O. BARET and D. PRICE, Hidden Markov Model Based Word Recognition and Its Application to Legal Amount Reading on French Checks. *Computer Vision and Image Understanding* 70(3): 404-419 (1998).
- [KOE 02] A. KOERICH, Large vocabulary off-line handwritten word recognition. PhD Thesis (2002).
- [OKA 98] S. OKAWA, E. BOCCHIERI, and A. POTAMIANOS, Multi-band speech recognition in noisy environments. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, p. 641.644, Seattle, Washington (1998).
- [PEC 02] M. PECHWITZ, S.S. MADDOURI, V. MAERGNER, N. ELLOUZE, and H. AMIRI, IFN/ENIT-database of handwritten Arabic words. In *Proc. of CIFED 2002*, p. 129-136, Hammamet, Tunisia, October (2002).
- [POT 98] G. POTAMIANOS, H.P. GRAF, Discriminative training of HMM flux exponents for audio-visual speech recognition. *Proceeding IEEE Conference Acoustics, Speech, and Signal Processing*. vol.6, p. 3733-3736, Seattle, WA (1998).
- [PRE 03] L. PREVOST, C. MICHEL-SENDIS, A. MOISES, L. OUDOT, and M. MILGRAM, Combining model-based and discriminative classifiers: application to handwritten character recognition. In *7th International Conference on Document Analysis and Recognition*, volume 1, p. 31-35 (2003).
- [RAB 90] RABINER, L.R., A tutorial on hidden markov models and selected applications in speech recognition. *Readings in speech recognition*, pp. 267-296, 1990.
- [SAK 79] H. SAKOE, Two-level DP matching – A dynamic programming – based pattern matching algorithm for connected word recognition, *IEEE Transactions of the IECE of Japan*, Vol 27, p. 588- 595, 1979.

- [SME 94] Ph. SMETS and R. KENNES, The Transferable Belief Model. *Artificial Intelligence*, vol 66, n° 2, p. 191-234 (1994).
- [SRI 00] N.S. SRIHARI, Handwritten Address Interpretation: A Task of Many Pattern Recognition Problems. *IJPRAI* 14(5), p. 663-674, (2000).
- [TAY 01] Y. H. TAY, P.M. LALLICAN, M. KHALID, C. VIARD-GAUDIN and S. KNERR, An Offline Cursive Handwritten Word Recognition System, *IEEE Region 10 International Conference on Electrical and Electronic Technology (TENCON)*, vol. 2, p. 519- 524 (2001).
- [VAR 90] A. P. VARGA and R. K. MOORE, Hidden Markov model decomposition of speech and noise. *Proc. IEEE Internat. Conf. Acoust. Speech and Signal Process*, p. 845-848 (1990).
- [VIA 99] C. VIARD-GAUDIN, P.M. LALLICAN, S. KNERR, P. BINTER, The IRESTE On/Off (IRONOFF) Dual Handwriting Database, *ICDAR'99*, p. 455-459 (2009).
- [VIN 04] A. VINCIARELLI and S. BENGIO and H. BUNKE. Offline Recognition of Unconstrained Handwritten Texts Using {HMM}s and Statistical Language Models, *IEEE Transactions on PAMI*, Vol26, N° 6, p. 709-720, 2004.
- [VIA 05] C. VIARD-GAUDIN , P. LALLICAN , S. KNERR, Recognition-directed recovering of temporal information from handwriting images, *Pattern Recognition Letters*, v.26 n.16, p. 2537-2548, December (2005).
- [WEL 98] C.J. WELLEKENS, J. KANGASHARJU, C. MILESI, The use of meta-HMM in multiflux HMM training for automatic speech recognition, *Proc. of Intl. Conference on Spoken Language Processing (Sydney)*, p. 2991-2994, December (1998).

Yousri **Kessentini**

Yousri Kessentini est titulaire d'un doctorat en informatique de l'université de Rouen en 2009. Ses domaines de recherche s'articulent autour de l'utilisation de modèles stochastiques pour la reconnaissance de l'écriture manuscrite multi-scripts et l'extraction de la structure de documents manuscrits complexes.

Abdelmajid **Ben Hamadou**

Abdelmajid Ben Hamadou est titulaire d'une thèse de troisième cycle en informatique de l'Université de Paris Orsay et une thèse d'Etat en Informatique de l'Université de Tunis. Il est actuellement Professeur en Informatique à l'Institut Supérieur d'Informatique et du Multimédia de Sfax et Directeur du Laboratoire de recherche MIR@CL à la même institution. Abdelmajid Ben Hamadou est auteur de plus que 280 articles dans des revues et des conférences. Ses domaines de recherche s'articulent autour de la documentique, le Traitement Automatique des langues et le Web sémantique.

Thierry **Paquet**

Thierry PAQUET, titulaire d'un doctorat de l'Université de Rouen (1992), a collaboré de 1992 à 1996 à des projets industriels avec la société Matra MCS et le service de recherche technique de La Poste, pour l'automatisation du tri du courrier et la lecture de chèques. Pendant cette période il s'est également intéressé aux modèles stochastiques et à l'utilisation des critères d'Information. Depuis il est impliqué dans le développement de techniques de traitement des documents manuscrits: Biométrie, Auto-adaptation des systèmes de reconnaissance aux scripteurs, catégorisation de textes manuscrits pour des applications industrielles, analyse d'images de document de mise en page complexe. Il s'intéresse également au développement de systèmes de lectures robuste des documents anciens pour des applications aux bibliothèques numériques. Thierry Paquet est professeur à l'Université de Rouen depuis 2002. Il est président du Groupe de Recherche en Communication Ecrite (GRCE) et directeur adjoint du laboratoire LITIS.

