



HAL
open science

Combinaison de caractéristiques bas niveau à travers des modèles N-flux dans le cadre de la reconnaissance de l'écriture manuscrite multi-flux.

Yousri Kessentini, Thierry Paquet, Abdelmajid Benhamadou

► To cite this version:

Yousri Kessentini, Thierry Paquet, Abdelmajid Benhamadou. Combinaison de caractéristiques bas niveau à travers des modèles N-flux dans le cadre de la reconnaissance de l'écriture manuscrite multi-flux.. Colloque International Francophone sur l'Écrit et le Document, Oct 2008, France. pp.109-114. hal-00334404

HAL Id: hal-00334404

<https://hal.science/hal-00334404>

Submitted on 26 Oct 2008

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.

Combinaison de caractéristiques bas niveau à travers des modèles N-flux dans le cadre de la reconnaissance de l'écriture manuscrite multi-scripts

Yousri KESSENTINI^{1,2}, Thierry PAQUET¹, AbdelMajid BENHAMADOU²

¹ Laboratoire LITIS EA 4108, université de Rouen, France

² Laboratoire MIRACL, université de Sfax, Tunisie

{yousri.kessentini, thierry.paquet}@univ-rouen.fr

abdelmajid.benhamadou@isimsf.rnu.tn

Résumé : Nous proposons dans cet article une approche de reconnaissance de l'écriture manuscrite multi-scripts. L'objectif étant de proposer un système indépendant de la nature du script, on procède alors sans segmentation. Des caractéristiques bas niveaux basées sur les directions et les densités des pixels sont combinées à travers une approche multi-flux. 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.

Mots-clés : Reconnaissance de l'écriture manuscrite, mutli-scripts, MMC multi-flux.

1 Introduction

La reconnaissance de l'écriture est un domaine de recherche qui est aujourd'hui bien étudié. De nombreuses approches de reconnaissance sont en effet proposées dans la littérature pour tenter de résoudre des problèmes spécifiques. Parmi ces études quelques approches sont maintenant mises en oeuvre dans des applications industrielles de reconnaissance de chèques, de formulaires, ou d'adresses. Ces succès bien que relatifs permettent toutefois de mettre en avant les principales qualités de ces systèmes : (1) Robustesse : ils sont fondés sur des techniques d'apprentissage statistiques (2) Efficacité : ils intègrent des connaissances lexicales par approche ascendante (3) Paramétrables : ils sont fondés sur des approches analytiques de modélisation en caractères (pour les langues alphabétiques) qui permettent de travailler sur des lexiques variables en fonction des besoins des applications envisagées. Ses avancées méthodologiques ne permettent pas cependant de solliciter des 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. D'autre part, les performances des systèmes de reconnaissance travaillant sur l'alphabet latin ne sont pour l'instant pas directement transposables sur d'autres alphabets tel que l'alphabet arabe par exemple. La raison principale est liée au fait que les techniques de segmentation des écritures latines sont globalement inopérantes sur les écritures arabes. De ce fait, il n'existe

pas à ce jour de système de reconnaissance capable de travailler indifféremment sur différents scripts, moyennant une paramétrisation à définir. De sorte que les très rares systèmes de lecture multi-scripts qui ont été étudiés sont en générale la combinaison de deux systèmes différents. L'un dédié à la reconnaissance de l'alphabet latin l'autre de l'alphabet l'arabe. De ce fait, les tentatives qui ont été dans le sens d'une réalisation d'un véritable système de reconnaissance multi-script n'ont à notre connaissance jamais aboutie.

Dans cette étude nous proposons un système de reconnaissance de l'écriture multi-scripts. Nous combinons plusieurs sources d'informations basées sur des caractéristiques bas niveaux à travers une approche multi-flux [BOU 97]. Cette approche particulière de combinaison a été déjà testée dans un travail préliminaire et a montré sa supériorité par rapports aux approches classiques de combinaison de représentations et de décisions [KES 08].

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 de 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.

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 traitées indépendamment jusqu'à certains points d'ancrage¹ (⊗ sur la Figure 1) où ils sont contraints à se re-synchroniser et à combiner leurs contributions partielles.

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

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-unité lexicales M_j ($j = 1, \dots, J$) (des lettres par exemple). Chaque sous-modèle M_j est composé de K modèles de Markov cachés (HMM) 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 composée chacune de K sources d'information synchrones lors des transitions entre sous-unités. On note alors :

$$X = \{(X_1^k, X_2^k, \dots, X_j^k, \dots, X_J^k), k = 1, \dots, K\}$$

où X_j^k désigne la séquence d'observation de la source k correspondant au sous modèle j .

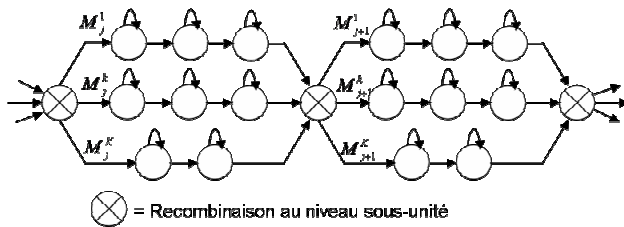


FIG. 1 : Structure générale d'un modèle multi-flux

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:

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

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:

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

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

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

2.2 Apprentissage des MMC multi-flux

Apprendre les modèles multi-flux se fait en deux étapes : Apprendre les paramètres des MMC des différents flux (probabilités de transitions, moyenne, variance 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. 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 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 3 gaussiennes. L'apprentissage des deux 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. 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 experts et fixés a priori à :

$$\alpha_k = cte = \frac{1}{K}$$

K étant le nombre de flux dans le modèle (dans notre cas $K=2$). 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 : Les poids sont estimés à partir de la segmentation forcée obtenue par l'algorithme de décodage multi-flux où on calcule le rapport entre le nombre de fois où l'expert améliore la reconnaissance et le nombre d'occurrence du caractère concerné dans la base d'apprentissage.

$$\alpha_j^k = \frac{n_{k,j}}{n_j}$$

où $n_{k,j}$ est le nombre de trame dans les données d'apprentissage ou l'expert k a la meilleure probabilité pour l'état j et n_j le nombre d'occurrence de l'état j dans la base d'apprentissage.

A 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 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, différentes solutions sont envisageables:

1- La combinaison au niveau état : bien qu'elle ne permet 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 probabilités locales sont obtenus en combinant les probabilités relatives de chaque flux à tous les instants.

2- La combinaison au niveau sous-unité lexicale : 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 décodage classique. Dans la littérature, deux algorithmes ont été proposés pour résoudre ce problème:

- Algorithme de programmation dynamique sur deux niveau 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, le meilleur empilement de sous unités.

- "HMM-recombination" [BOU 96]: 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). 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 (K=2 dans notre cas). 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.

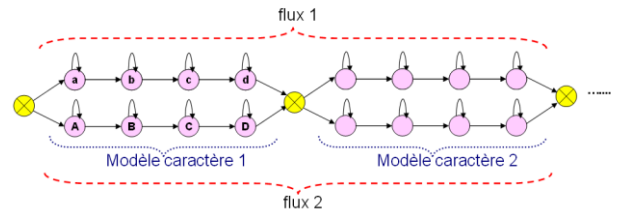


FIG. 2 : Topologie d'un modèle parallèle à 2 flux

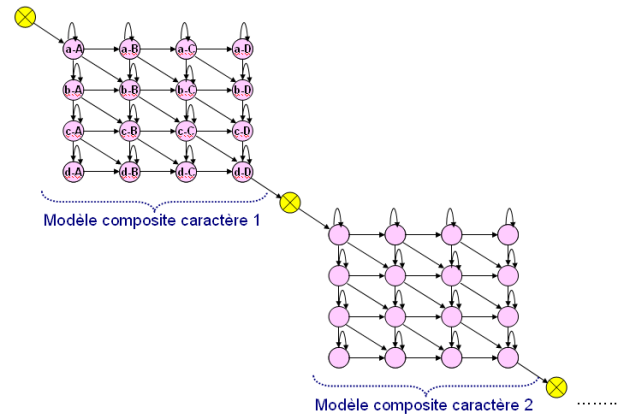


FIG. 3 : Topologie du modèle MMC résultant du modèle à 2 flux de la Figure 2.

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

$$\log P(X_1(t), X_2(t)/i) = \alpha_1^1 \log P(X_1^1(t)/S_1(i)) + \alpha_1^2 \log P(X_1^2(t)/S_2(i))$$

$X_1(t)$ et $X_2(t)$ sont les vecteurs de caractéristiques sur les deux flux à t et $s_1(i)$ et $s_2(i)$ sont des fonctions qui désignent l'état correspondant à l'état produit dans le MMC du flux 1 (resp. flux 2).

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 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 $P(A/B)$.

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 pour les deux canaux.

2.4 Extension au cas à N-flux

Combiner N flux de caractéristiques à travers une approche multi-flux revient à généraliser l'algorithme précédent de façon à combiner 2 à 2 les N flux.

Entrées

```

N: nombre de MMC parallèles.
Models[N]: Tableau de N MMC parallèles.
ProductHMM(): Fonction retournant le MMC
produit à partir de deux MMC parallèles.
Initialisation
Model = Models[0];
currentFlux = 1;
while (currentFlux != N)
    Begin
    Model =ProductHMM(Model, Models[currentFlux])
    currentFlux++;
    End
return Model;
    
```

Il est aisé de percevoir que dans le modèle composite, le nombre d'états par caractère croit considérablement en fonction du nombre de flux à combiner. Pour réduire ce nombre, et du coup accélérer le processus de décodage qui dépend fortement du nombre d'états, deux solutions sont envisagées.

1) Réduire le nombre d'états par caractère dans les MMCs parallèles : En effet, réduire le nombre d'état 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...

2) Limiter l'asynchronie entre les flux de données en restreignant le 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.

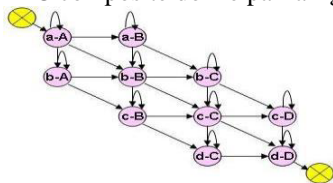


FIG. 4 : MMC composite à 1 degré d'asynchronie

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 pré-traitements et de caractérisation des images d'écriture.

3.1 Prétraitements

Avant de procéder à la reconnaissance de mot, 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 «sheering 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.

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.

3.2.1 Caractéristiques contours

Les caractéristiques que nous avons choisies 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 la direction de Freeman des points du contour supérieur (resp. inférieur), et un histogramme des directions de Freeman est calculé sur tous les points de la fenêtre, constituant ainsi un ensemble de 8 caractéristiques.

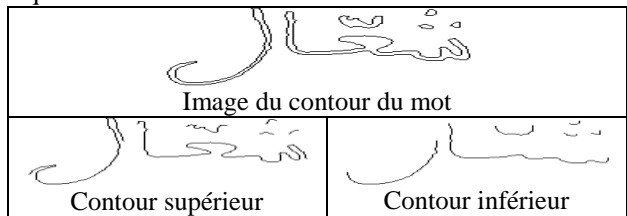


FIG. 5 : Images des contours du mot

Un deuxième ensemble de caractéristiques consiste à déterminer pour chaque point du contour supérieur (resp. inférieur) la nature du point qui lui correspond verticalement sur le contour en vis à vis. Quatre cas sont possibles:

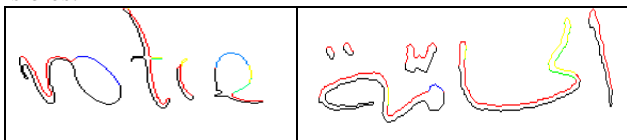


FIG. 6 : Différents types des points du contour supérieur

- Ce point appartient au contour inférieur (resp. supérieur) (points rouges sur la FIG. 6).
- Ce point appartient à une occlusion (points bleus sur la FIG. 6).
- Ce point appartient au contour supérieur (resp. inférieur) (points jaunes sur la FIG. 6).
- 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. 6).

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).

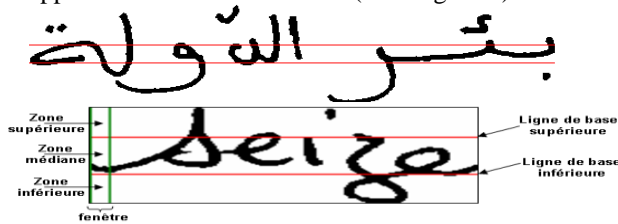


Figure. 7 : Détection des lignes de base

Un histogramme sera 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. Chaque séquence est modélisée par des modèles de caractères respectivement supérieur et inférieur.

3.2.2 Caractéristiques de densités

Un deuxième ensemble de caractéristiques est utilisé dans ce travail basé sur les densités [HAJ 05]. Chaque fenêtre étant divisée verticalement en 4 cellules et les caractéristiques suivantes sont extraites:

- f_1 : densité des pixels noirs dans la fenêtre.
- f_2 : nombre de transitions Noir/Blanc entre cellules.
- f_3 : différence de position entre les centres de gravité des pixels d'écriture dans deux fenêtres consécutives.
- f_4 à f_{11} : sont les 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, 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 Figure. 8).
- 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.

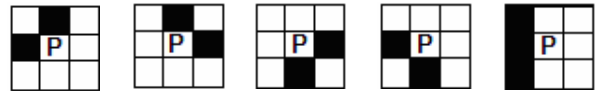


Figure. 8 : Les 5 types de configurations locales autour d'un pixel de fond P

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 différentes. 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 en s'accordant au formalisme multi-flux donné ci-dessus.

4 Expérimentations et analyses

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 Latin IRONOFF [VIA 99]. Nous présentons les résultats obtenus pour des modèles combinant de 1 à 4 flux.

4.1 La base IFN/ENIT

Cette base contient 26459 mots arabes qui sont écrits par 411 scripteurs. Les mots de cette base correspondent à un lexique de 946 noms de villes et villages tunisiens. Quatre ensembles distincts de données (a, b, c, d) sont prédéfinies dans la base IFN/ENIT, pour l'apprentissage et le test.

Modèles	Taille du lexique			
	30		100	
	Rang1	Rang5	Rang1	Rang5
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	100	96.8	99.8
2-flux 3)4)	97.6	99.6	96.4	98.6
3-flux 1)2)3)	98.6	99.8	98.4	99.8
4-flux	99.2	99.8	98.8	100

TAB.1: Performances sur la base IFN/ENIT

La table 1 présente les résultats de reconnaissance obtenus sur la base IFN/ENIT pour deux tailles différentes de lexique. Pour chaque mot à reconnaître, un

lexique de taille L est obtenu par tirage aléatoire de $L-1$ mots parmi le lexique complet de la base de mots annotés, complété par l'étiquette du mot à reconnaître.

Nous avons aussi testé notre système sur un lexique plus étendu, afin de pouvoir le comparer à d'autres systèmes. Nous obtenons un taux de reconnaissance de 86.2% sur un lexique de 500 mots en combinant seulement 2 flux d'informations. Ceci est comparable aux performances obtenues dans [HAJ 05], vainqueur de la compétition ICDAR 2005 ayant obtenu un taux de 86.51% sur un lexique de 450 mots.

4.2 La base IRONOFF

IRONOFF est une base de données duale en-ligne et hors-ligne. Le sous-ensemble IRONOFF-Chèque est 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.

Nous présentons dans la table 2 les résultats de reconnaissance que nous avons obtenus sur la base IRONOFF-Chèque en considérant les M meilleures solutions. Pour accélérer le décodage multi-flux, nous avons aussi réduit le nombre d'état par caractère à 3 e/c. Ceci affaiblit un peu les résultats mais accélère le décodage.

Modèles	4 e/c		3 e/c	
	Rang1	Rang5	Rang1	Rang5
1) Contour sup.	82.6	94.6	81.6	95.6
2) Contour inf.	78	94.4	72.4	93.2
3) Densité1	76.8	89.4	77.6	93.8
4) Densité2	75.6	86.8	78.8	93.8
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-flux 1)2)3)	94	99.6	91.6	99.6
3-flux 1)3)4)	92.6	99.2	90.8	98.8
3-flux 2)3)4)	93.4	99	89.4	98.4
3-flux 1)2)4)	93.6	99.8	92	99
4-flux	94.2	99.2	92.8	99.2

TAB. 2: Performances sur la base IRONOFF-Chèque

Nous remarquons que le système de reconnaissance proposé donne de bonnes performances aussi bien pour le script Arabe que pour le Latin. Toutefois les résultats sur l'arabe sont meilleurs, ceci peut être expliqués par le fait qu'on construit plus de 170 modèles de caractère pour modéliser les caractères arabes selon leur position dans le mot et avec ou sans la présence de la ligature "chad-da". Par contre 26 modèles de caractère seulement sont utilisés en Latin.

Les résultats montrent que l'ajout d'un flux améliore généralement la performance du système. On remarque aussi que la combinaison d'un flux contour avec un flux densité est plus intéressant que de combiner deux flux de même nature (contour ou densité).

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 : il procède sans segmentation en combinant des caractéristiques bas niveaux. Des résultats significatifs sont obtenus sur deux bases de référence : la base de mots Arabes IFN/ENIT et la base IRONOFF de mots Latins. Les résultats montrent que combiner deux flux améliore considérablement les performances. Cependant, au de las de deux flux, la combinaison n'apporte pas forcément beaucoup d'amélioration au regard du travail supplémentaire qu'il faut pour extraire les caractéristiques et le décodage.

Références

- [WEL 98] C.J.Wellekens, J.Kangasharju, C.Milesi, The use of meta-HMM in multiframe HMM training for automatic speech recognition, *Proc. of Intl. Conference on Spoken Language Processing (Sydney)*, December 1998, pp. 2991-2994.
- [BOU 97] H.Bourlard, S.Dupont, Sub-band-based Speech Recognition. *In IEEE Int. Conf. on Acoust., Speech, and Signal Processing*, 1997, pages 1251-1254.
- [POT 98] Potamianos, G., Graf, H.P. Discriminative training of HMM flux exponents for audio-visual speech recognition. *Proceeding IEEE Conference Acoustics, Speech, and Signal Processing*. pp. 3733-3736 vol.6, Seattle, WA, 1998.
- [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, pp 588- 595, 1979.
- [BOU 96] H.Bourlard, S.Dupont, and C.Riss, Multi-flux speech recognition. *Technical Report IDIAP-RR 96-07*, IDIAP, 1996.
- [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*, pp 845-848, 1990.
- [BOZ 89] R. M. Bozinovic and S. N. Srihari. "Off-line Cursive Script Word Recognition", *IEEE Transactions on PAMI*, Vol 11, N° 1, pp: 68-83, January 1989.
- [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.
- [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.
- [VIA 99] C. Viard-Gaudin, P.M. Lallican, S. Knerr, P. Binter, *The IRESTE On/Off (IRONOFF) Dual Handwriting Database*, ICDAR'99.
- [PEC 02] Pechwitz. M, Maddouri. S. S., Maergner. V., Ellouze. N., and Amiri. H., IFN/ENIT - database of handwritten Arabic words. *In Proc. of CIFED 2002*, pages 129-136, Hammamet, Tunisia, October 21-23 2002.
- [KES 08] Y. Kessentini, T. Paquet, A. BenHamadou. Combinaison d'Information pour la Reconnaissance de l'Écriture Manuscrite Hors-Ligne. *RFIA'2008*, Amiens, France.