Étude de primitives spectrales pour la reconnaissance de caractères manuscrits dans le cadre d'une approche markovienne 2D
Spectral features for handwriting recognition using a 2D markovian approach

Sylvain Chevalier* Mélanie Lemaître[1,2] Edouard Geoffrois[1]
Sylvain.Chevalier@gmail.com Melanie.Lemaitre@etca.fr Edouard.Geoffrois@etca.fr

Emmanuèle Grosicki[1] Françoise Prêteux[2]
Emmanuele.Grosicki@etca.fr Francoise.Preteux@int-evry.fr

[1] DGA/Centre d'Expertise Parisien
16 bis avenue Prieur de la Côte d'Or
94114 Arcueil cedex

[2]GET/INT, Unité de projets ARTEMIS
9, rue Charles Fourier
91011 Evry cedex

Résumé

Dans le cadre de la reconnaissance d'écriture manuscrite, nous avons proposé une approche markovienne réellement bidimensionnelle utilisant des primitives spectrales locales. Dans cet article nous proposons d'étudier en détail la phase d'extraction de ces primitives en analysant en particulier leur influence sur les performances du système. Pour cela, nous utilisons la base de chiffres manuscrits MNIST.

Mots Clef

Écriture manuscrite, champs de Markov, extraction de primitives, primitives spectrales locales.

Abstract

Within the framework of handwriting recognition, we have proposed a really bidimensional markovian approach using local spectral features. In this article, we focus on the features extraction phase. In particular, we study the influence of the parameters of the chosen features on the performances obtained by considering the handwritten digits MNIST database.

Keywords

Handwritting, Markov random fields, features extraction, local spectral features.

1 Introduction

Le succès rencontré par les HMM (Hidden Markov Models) depuis quelques années [5, 7] a incité les chercheurs à les utiliser dans le cas de la reconnaissance de l'écriture manuscrite [8, 9]. Du fait de la complexité des algorithmes de décodage, les solutions étaient principalement monodimensionnelles ou pseudo 2D. Grâce à l'introduction de la programmation dynamique 2D [4, 3], nous avons proposé une approche par champs de Markov réellement bidimensionnelle pour la reconnaissance d'écriture manuscrite [6]. Dans cette approche, chaque classe de caractères est modélisée par une grille d'états cachés, chacun étant relié à des primitives extraites de l'images par une densité de probabilité d'observation de ces primitives.

Concernant le choix des primitives, deux types d'approches peuvent être envisagées :
– soit on utilise directement la valeur des pixels,
– soit on réalise une phase de prétraitement sur les pixels pour en extraire une information plus riche.

Alors que les champs de Markov utilisent classiquement la première approche, nous proposons d'utiliser la seconde. Dans le cadre de notre système, celle–ci consiste en une extraction originale de primitives spectrales locales présentant l'avantage d'être robustes au bruit et à la variabilité du signal d'entrée, et permettant d'obtenir des informations sur la direction des traits.

Après avoir présenté le principe général de notre système de reconnaissance, nous détaillerons la phase d'extraction des primitives ainsi que les différents paramètres qui lui sont associés. Enfin, nous étudierons l'influence de ces paramètres sur les performances du système.

2 Présentation du système de reconnaissance

L'idée générale est de créer un modèle markovien reliant des états cachés à des primitives extraites des images pour chaque classe de caractères, ces modèles étant utilisés lors de la phase de reconnaissance pour déterminer la classe des différentes images de la base de test à partir de leurs primi-

*Une partie de ces travaux a été effectuée dans le cadre de la thèse du premier auteur au Centre d'Expertise Parisien de la DGA et sous la direction de Françoise Prêteux.

tives.

L'approche proposée [6] consiste à segmenter les images en états de manière à obtenir des zones homogènes en terme de directions de traits. Pour cela, on part d'une segmentation initiale uniforme que l'on va affiner grâce à un algorithme itératif de type EM au cours duquel les paramètres associés aux différents états (densités d'observations et probabilités de transitions) sont calculés (voir figure 1).

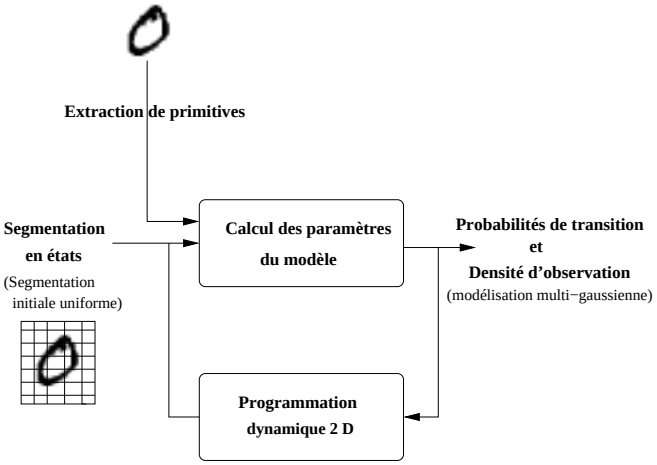


FIG. 1 – Apprentissage d'un modèle de chiffre

La figure 2 montre les segmentations en états obtenues pour deux images différentes d'une même classe. On peut constater que la répartition des états est similaire dans les deux cas, *i.e.* qu'un même état dans les deux segmentations est associé à une direction de traits et à une position identiques. Par exemple, l'état 7 est associé pour les deux images à la direction horizontale positionnée sur le haut du "3".

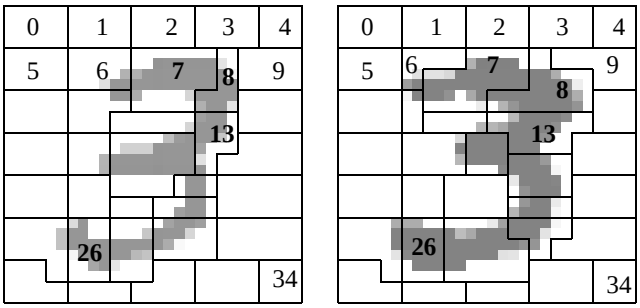


FIG. 2 – Exemples de segmentations en états de 2 images de la classe "3" obtenues avec notre algorithme

Les paramètres et les hypothèses utilisés pour notre modèle sont les suivants :

– La probabilité d'un état ne dépend que des états de son voisinage immédiat : voisinage[1] d'ordre 1 (voir figure 3).

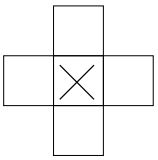


FIG. 3 – Voisinage d'ordre 1

– La densité d'observation est modélisée par des multi-gaussiennes dont le nombre de composantes maximum a été expérimentalement fixé à 20.
– On choisit un nombre d'états égal à 35 (grille 5×7), nombre d'états qui permet de modéliser les caractères les plus complexes tels que le "8" ou le "3".

Dans le cadre de notre étude, nous utilisons la base de données MNIST [13]. Cette dernière est une base standard, publique, composée de 70000 images de chiffres extraits de la base NIST (voir figure 4) et divisée en une base d'apprentissage (60000 images) et une base de test (10000 images). Les spécificités de la base de données sont :
– toutes les images ont la même taille (28×28),
– les échantillons sont centrés et un cadre blanc est gardé autour des caractères,
– des niveaux de gris résultent des prétraitements de normalisation en taille,
– les bases d'apprentissage et de test on été écrites par des scripteurs distincts.

FIG. 4 – Échantillons de la base MNIST

Le développement a été effectué uniquement sur la base d'apprentissage découpée en deux parties : une partie de

[1] D'autres topologies de voisinages (ordre 2 ou 3) sont susceptibles d'apporter des améliorations.

cette base est utilisée pour l'entrainement des modèles, et l'autre pour leur validation. La base de test n'est donc pas utilisée pour l'optimisation des paramètres, mais uniquement pour l'évaluation finale des modèles, qui sont alors appris sur la totalité de la base d'appentissage.

3 Extraction des primitives

3.1 Primitives classiques

L'objectif commun de toutes les primitives est de caractériser au mieux la forme des caractères afin de pouvoir distinguer si deux images appartiennent à deux classes différentes ou à la même classe, c'est–à–dire qu'elles doivent diminuer la variabilité intra-classe et augmenter la variabilité inter-classe. Suivant les applications et techniques utilisées pour le système de reconnaissance, les primitives extraites peuvent être très différentes.

Dans la littérature, les primitives sont classifiées de plusieurs façons différentes.

Une première distinction peut être effectuée entre les primitives globales et les primitives locales. Les primitives globales cherchent à représenter au mieux la forme générale d'un caractère et sont donc calculées sur des images relativement grandes (ex : transformée de Fourier et transformée de Hough). Les primitives locales sont calculées lors d'un parcours des pixels de l'image avec un pas d'analyse qui dépend de la modélisation, du type de primitive et de la taille de l'image.

Une seconde distinction est effectuée entre les primitives topologiques, structurelles ou statistiques [10, 11] :

– Les primitives topologiques ou métriques : elles consistent à compter dans une forme le nombre de trous, évaluer les concavités, mesurer des pentes et autres paramètres de courbures et évaluer des orientations, mesurer la longueur et l'épaisseur des traits, détecter les croisements et les jonctions des traits, mesurer les surfaces et les périmètres, ...
– Les primitives structurelles : elles ressemblent beaucoup aux primitives topologiques. La différence est qu'elles sont généralement extraites non pas de l'image brute, mais à partir du squelette ou du contour de la forme. Ainsi, on ne parle plus de trous, mais de boucles ou de cycles dans une représentation filiforme du caractère.
– Les primitives statistiques : l'histogramme, qui représente le nombre de pixels sur chaque ligne ou colonne de l'image, en est un exemple classique et simple à calculer. On peut citer également l'approche basée sur un moyennage des pixels situés à l'intérieur d'un masque rectangulaire : on construit une matrice de masques recouvrant la totalité de la forme qui permet une représentation statistique des valeurs correspondant à chaque masque.

Notre système de reconnaissance utilise des primitives spectrales locales se rapprochant des primitives statistiques. Elles sont décrites dans la section suivante.

3.2 Primitives spectrales locales choisies

Dans le cadre de notre approche, on cherche à segmenter les images en états suivant les orientations des traits. Ceci est réalisé grâce aux primitives qui doivent permettre d'extraire ce type d'information. De plus, elles doivent être les plus robustes possible au bruit et à la variabilité du signal.

Le succès rencontré par le traitement de la parole nous a incité à étudier les techniques utilisées [12] et à nous en inspirer. Les bancs de filtres et le cepstre sont souvent utilisés pour extraire les primitives. Celles–ci cherchent non seulement à extraire des caractéristiques pertinentes du signal mais aussi à réduire l'influence du bruit. Une analyse fréquentielle est donc nécessaire. De plus, du fait de la non–stationnarité du signal de parole, on réalise une analyse fenêtrée du signal (souvent fenêtre de Hamming).

Par analogie avec le traitement de la parole, nous utilisons des primitives spectrales locales. Celles–ci présentent l'avantage d'extraire des informations sur la direction des traits.

Les primitives choisies consistent (voir figure 5) à calculer une FFT dans une fenêtre gaussienne glissante centrée autour de pixels uniformément répartis dans l'image (un sur deux par exemple) (voir figure 6). Le fenêtrage par une gaussienne a été introduit afin de réduire les effets de bords dus à la FFT (il est possible de choisir une autre fenêtre : Hamming, ...). En effet, ces effets de bords sont d'autant plus gênant que l'on cherche à extraire une information directionnelle (les bords de l'image ajoutent de l'information parasite sur les directions verticale et horizontale).

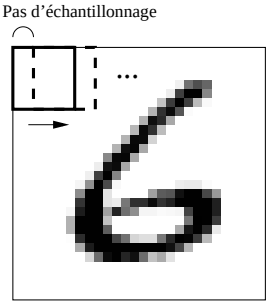


FIG. 6 – Fenêtre glissante

Un certain nombre de paramètres rentrant en compte dans le calcul des primitives influe sur les performances du système ainsi que sur la complexité du modèle :

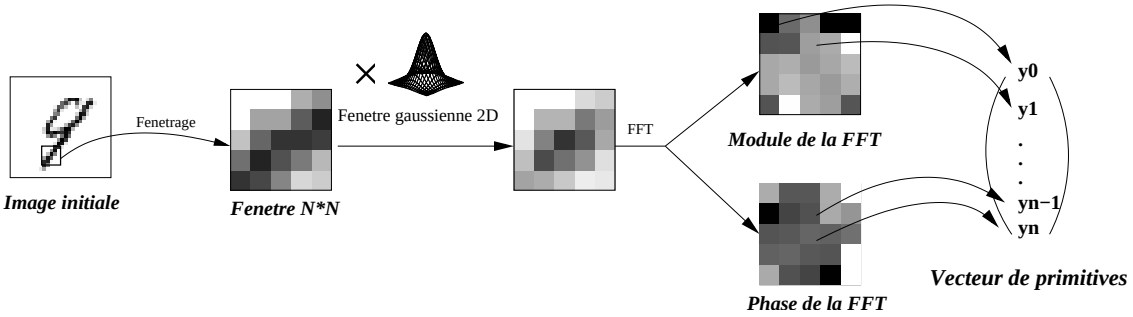


FIG. 5 – Processus d'extraction des primitives

– la taille de la fenêtre gaussienne et le pas d'échantillonnage,
– la sélection des coefficients du module et de la phase de la FFT.

Ces paramètres, notamment la taille de la fenêtre et le pas d'échantillonnage, sont directement liés à la résolution des images. L'étude des paramètres présentée par la suite s'applique au cas des images de la base MNIST (images de chiffres isolés de taille 28×28).

3.3 Choix de la taille de la fenêtre et du pas d'échantillonnage

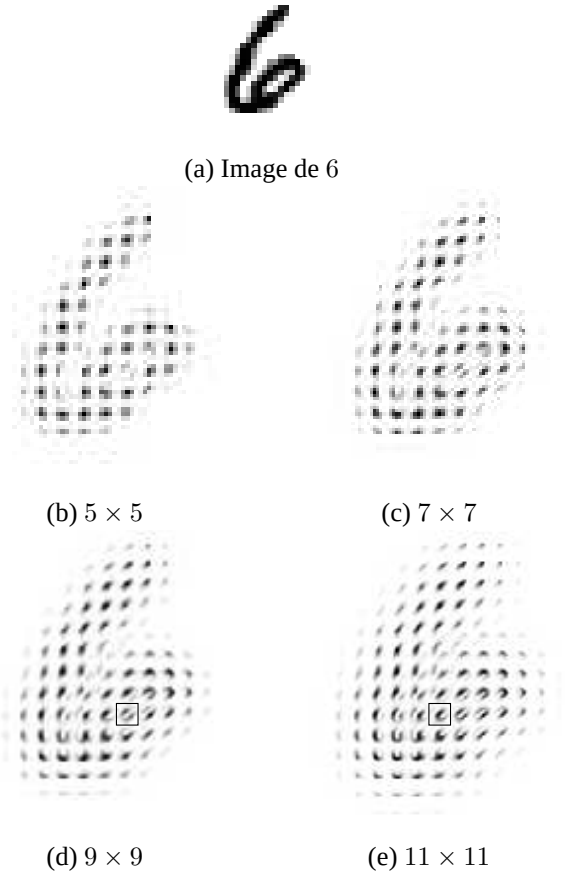
La taille des fenêtres utilisées lors de l'extraction des primitives ainsi que le décalage entre deux fenêtres consécutives (ou pas d'échantillonnage) sont des paramètres fortement corrélés. Notre choix s'est effectué en tenant compte des considérations suivantes.

Premièrement, la taille de la fenêtre doit être choisie de manière à pouvoir extraire des informations locales et précises en terme de direction de traits. Elle doit donc être à la fois suffisament grande pour permettre de distinguer sans ambiguité des directions de traits et ne pas extraire simplement une information de niveau de gris et suffisament petite pour ne pas contenir des formes trop complexes composées de plusieurs directions de traits.
Pour illustrer ce point, nous pouvons observer les imagettes extraites d'un exemple représentatif d'une image de chiffre (figure 7.a) pour des tailles de fenêtres différentes 5×5, 7×7, 9×9, 11×11 (voir figure 7) et pour un même pas d'échantillonage, fixé à 2 (*i.e.* 1 pixel sur 2).

L'observation de ces figures montre que :
– la fenêtre 5×5 ne permet pas toujours de distinguer les directions des traits,
– les fenêtres 9×9 et 11×11 peuvent contenir des formes assez complexes contenant différentes directions de traits (Cf. les encadrés noirs dans les figures correspondantes).

La fenêtre 7×7 semble représenter un bon compromis. Les expériences détaillées plus loin confirment ce choix.

Deuxièmement, le choix du pas d'échantillonage est lié au



(a) Image de 6

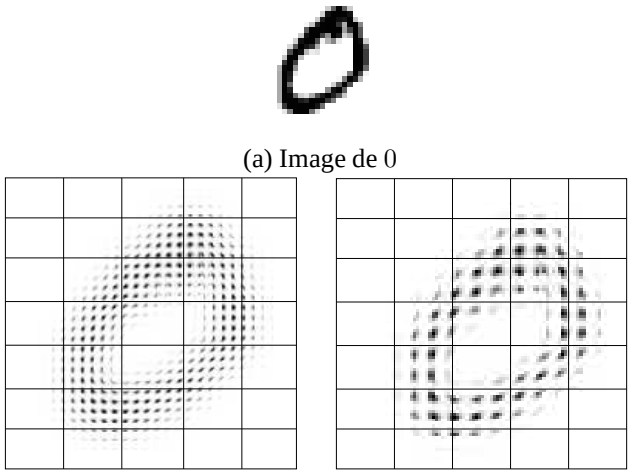
(b) 5×5 (c) 7×7

(d) 9×9 (e) 11×11

FIG. 7 – Imagettes extraites pour différentes tailles de fenêtre

choix de la taille de la fenêtre. Il doit être choisi de façon à accorder la même importance à tous les pixels tout en limitant les redondances pour limiter le temps de calcul notamment au moment du décodage. L'analogie avec le traitement de la parole nous a conduit à adopter un pas d'échantillonnage égal à environ $1/3$ de la taille de la fenêtre. Ainsi pour une fenêtre 7×7, le pas a été choisi égal à 2.

On peut observer que le choix de ces paramètres est en adéquation avec le modèle d'états développé, et en particulier



(a) Image de 0

(b) Pas de 1 (c) Pas de 2

(d) Pas de 3

FIG. 8 – Imagettes extraites avec une fenêtre 7×7 pour différents pas d'échantillonnages

avec le nombre d'états choisi (35). La figure 8.c montre que le choix d'une fenêtre 7×7 et d'un pas de 2 permet d'avoir un nombre suffisant d'observations par état comparé au cas d'une fenêtre 7×7 et d'un pas de 3 (figure 8.d). Le cas d'un pas égal à 1 (figure 8.b) n'a pas été retenu car il contient trop de redondance et multiplie par 4 le temps de calcul de décodage par rapport au pas de 2.

3.4 Étude des coefficients

Dans le cas d'une fenêtre gaussienne de taille 7×7, on note les coefficients du module de la FFT (même principe pour la phase) comme présenté figure 9.
Il y a une symétrie centrale, c'est–à–dire :

$$D^M_{i,j} = D^M_{-i,-j} \text{ (modules)},$$
$$D^\varphi_{i,j} = -D^\varphi_{-i,-j} \text{ (phases)}.$$

On note :

$$V^M_i = D^M_{i,0} \text{ et } V^\varphi_i = D^\varphi_{i,0},$$
$$H^M_j = D^M_{0,j} \text{ et } V^\varphi_j = D^\varphi_{0,j}.$$

Les différents coefficients peuvent être interprétés de la façon suivante :
– O est la moyenne sur l'imagette 7×7.
– V^M_1, H^M_1, $D^M_{-1,1}$ et $D^M_{-1,-1}$ sont les coefficients du module correspondant aux quatre directions principales

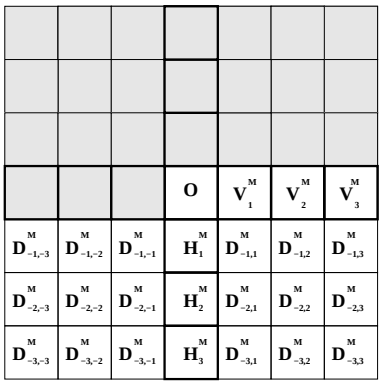


FIG. 9 – Positionnement et nom des différents coefficients du module

(respectivement direction verticale, horizontale, première diagonale et deuxième diagonale).
– Les coefficients V^M_k, H^M_k, $D^M_{-k,k}$ et $D^M_{-k,-k}$ $(k = 2, 3)$ correspondent aux mêmes directions mais à une fréquence multiple.
– Enfin, les autres coefficients correspondent à d'autres directions (voir figure 10).

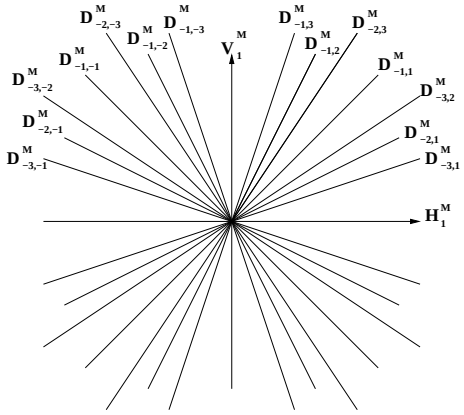


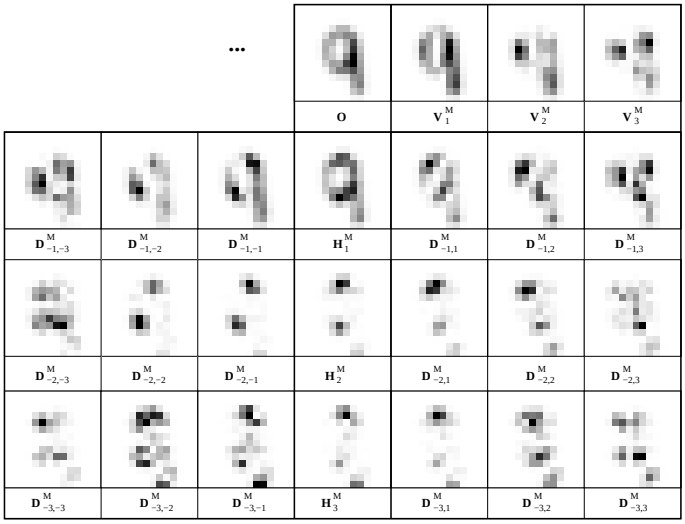
FIG. 10 – Direction des coefficients

On considère une image pour laquelle on représente les différents coefficients du module et de la phase de sa FFT (figure 11) : tous les 1 pixel sur 2, on calcule la FFT dans une fenêtre 7×7 et on note le coefficient que l'on souhaite observer.

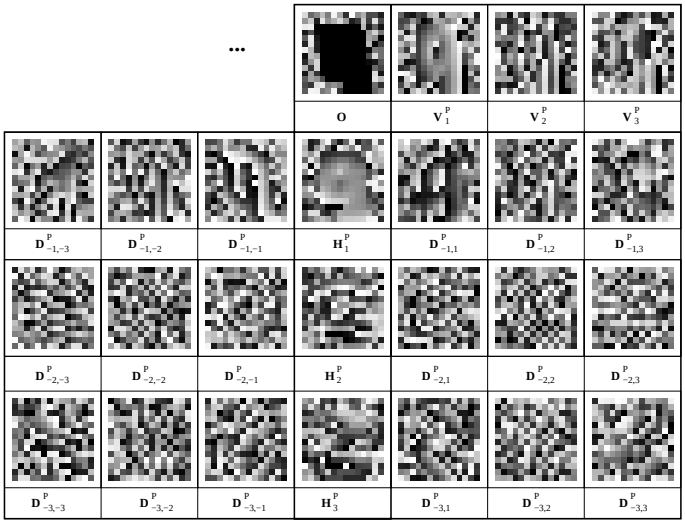
On constate bien que chaque coefficient donne une information différente sur la direction des traits. On pourra, en particulier, observer les coefficients (V^M_1, H^M_1, $D^M_{-1,1}$ et $D^M_{-1,-1}$) correspondant aux quatre directions principales.
Les coefficients de la phase semblent être moins facilement interprétables que les modules. On verra par la suite leur influence dans la reconnaisance.

Dans le processus d'extraction de primitives, nous ne conservons qu'un certain nombre de coefficients. Ceux–ci

(a) Image de 9



(b) Coefficients du module de la FFT



(b) Coefficients de la phase de la FFT

FIG. 11 – Coefficients du module et de la phase pour une image de “9”

doivent contenir suffisament d’information pour capturer la structure essentielle du caractère [10]. Pour le vérifier, nous pouvons reconstruire l’image d’origine à partir des coefficients sélectionnés. Dans la figure 12 on effectue la reconstruction à partir des 4 coefficients du module correspondant aux 4 directions principales et des 2 phases correspondant aux directions horizontale et verticale. L’image d’origine est la figure 8.a.

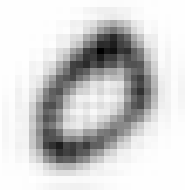


FIG. 12 – Reconstruction à partir de 4 modules et 2 phases

4 Étude de l’influence des paramètres

Dans cette partie, nous allons étudier l’influence des paramètres associés au calcul des primitives sur les performances de la reconnaissance avec la base de chiffres manuscrits MNIST.

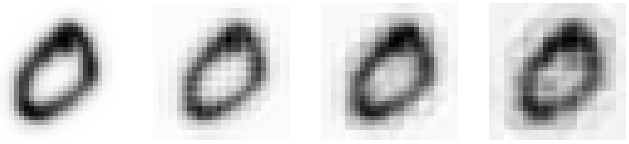
4.1 Influence de la taille de la fenêtre et du pas d’échantillonnage

Comme nous l’avons vu précédemment, le pas d’échantillonnnage est lié à la taille de la fenêtre. Ainsi, pour une fenêtre 5×5, on prend un pas d’échantillonnage de 1 pixel sur 2, pour une fenêtre 7×7 un pas de 1 pixel sur 2, pour une fenêtre 9×9 un pas de 1 pixel sur 3 et pour une fenêtre 11×11 un pas de 1 pixel sur 4. Dans le tableau 1, on donne les taux d’erreurs obtenus pour ces quatre tailles de fenêtres en utilisant un vecteur de primitives de 8 modules et de 2 phases. On constate que le meilleur résultat est obtenu avec une fenêtre 7×7.

TAB. 1 – Influence de la taille de la fenêtre sur le taux d’erreurs

Taille de la fenêtre	Taux d’erreurs
5×5	2.88%
7×7	2.09%
9×9	3.06%
11×11	4.27%

On peut le vérifier également en effectuant la reconstruction de l’image à partir des 10 coefficients, pour chaque taille de fenêtre (voir figure 13). On constate que l’image la mieux reconstruite et la plus précise est celle obtenue



5×5 7×7 9×9 11×11

FIG. 13 – Reconstructions de l'image à partir des 10 coefficients (8 modules et 2 phases) pour quatre tailles de fenêtres différentes

avec une fenêtre 7×7, les autres reconstructions sont plus floues.

4.2 Étude des coefficients

Intuitivement, nous avons débuté nos tests en utilisant les coefficients du module correspondant aux quatre directions principales (horizontale, verticale et les 2 diagonales), en effet, ceux sont celles qui, visuellement (voir figure 11) semblent apporter le plus d'information. L'apport de la phase est moins évident, c'est pourquoi, nous avons étudié l'influence de certains coefficients de celle–ci dans la reconnaissance (voir tableau 2).

TAB. 2 – Comparaison des résultats suivant les phases utilisées

Coefficients	Taux d'erreurs
4 dir principales	3.56%
4 dir principales + (V_1^{φ}, H_1^{φ})	2.38%
4 dir principales + (V_1^{φ}, H_1^{φ}, $D_{-1,1}^{\varphi}$, $D_{-1,-1}^{\varphi}$)	2.61%

Plusieurs expériences ont montré que les deux phases V_1^{φ} et H_1^{φ} améliorent le taux d'erreurs de 25% à 30% en relatif. De plus, on peut voir figure 11 que ceux sont les deux phases qui semblent contenir une information intéressante. Par contre l'ajout de phases supplémentaires n'améliorent plus les résultats.

Concernant l'influence du coefficient O (moyenne), des expériences ont montré que sa prise en compte ne permettait pas d'améliorer les résultats. De même les coefficients H_2 et V_2 ne semblent pas améliorer les performances.

Après nous être limités aux quatre coefficients du module correspondant aux quatre directions principales, nous allons étudier la prise en compte de directions supplémentaires.

Remarque : les résultats donnés ci–dessous, ont été obtenus avec les deux phases V_1^{φ} et H_1^{φ} évoquées précédemment.

1. *Quatre directions principales et deux autres directions secondaires*

   On ajoute aux quatre directions principales, deux autres directions secondaires et on observe les résultats (voir tableau 3). On constate que la prise en compte de deux directions supplémentaires améliore les résultats. En particulier, l'ajout des modules $D_{-2,-1}^M$ et $D_{-2,1}^M$ donne un taux d'erreurs de 2.14%.

2. *Quatre directions principales et quatre directions secondaires*

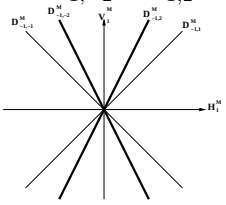
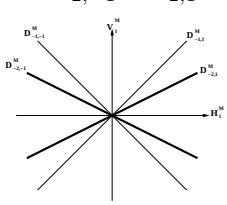
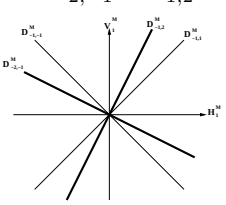
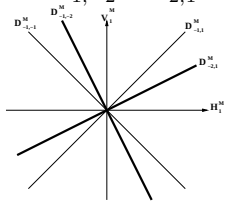
   On ajoute aux quatre directions principales les quatre directions secondaires (voir tableau 4) et l'on obtient un taux d'erreurs de 2.09%.

3. *Quatre directions principales et six autres directions*

   On ajoute aux huit directions précédentes, deux directions supplémentaires (voir tableau 5).

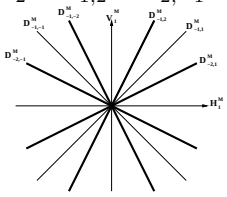
   L'ajout de directions supplémentaires ne semble plus améliorer les performances du système. Ceci peut s'expliquer par le fait que la taille des images traitées est trop petite et la résolution n'est pas assez bonne pour faire la distinction entre deux directions assez proches l'une de l'autre.

TAB. 3 – Résultats obtenus avec deux autres directions

$D_{-1,-2}^M$, $D_{-1,2}^M$  Taux d'erreurs : 2.42%	$D_{-2,-1}^M$, $D_{-2,1}^M$  Taux d'erreurs : 2.14%
$D_{-2,-1}^M$, $D_{-1,2}^M$  Taux d'erreurs : 2.27%	$D_{-1,-2}^M$, $D_{-2,1}^M$  Taux d'erreurs : 2.17%

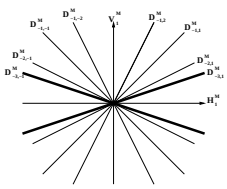
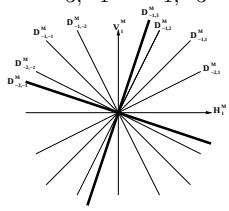
TAB. 4 – Résultats obtenus avec 4 directions principales et 4 directions secondaires

$D_{-1,-2}^M$, $D_{-1,2}^M$, $D_{-2,-1}^M$, $D_{-2,1}^M$



Taux d'erreurs : 2.09%

TAB. 5 – Résultats obtenus avec six autres directions

$D^M_{-3,-1}$, $D^M_{-3,1}$	$D^M_{-3,-1}$, $D^M_{-1,-3}$
	
Taux d'erreurs : 2.25%	Taux d'erreurs : 2.41%

Le meilleur résultat obtenu jusqu'à présent est un taux d'erreurs de $\mathbf{2.09}\%$ sur la base de validation en utilisant les quatre directions principales, les quatre directions secondaires et les deux phases. Pour les mêmes primitives, on obtient un taux d'erreurs de $\mathbf{2.32}\%$ sur la base de test.

5 Conclusion et perspectives

Nous avons présenté ici, une étude concernant les primitives spectrales locales utilisées par notre système de reconnaissance d'écriture manuscrite. Nous avons mis en évidence l'influence des différents paramètres associés à ces primitives sur les performances du système. Ainsi, nous avons obtenu un taux d'erreurs de 2.32% sur la base de test MNIST, ce qui améliore le taux d'erreurs, égal à 2.74%, obtenu intuitivement avec un vecteur de primitives composé de 4 modules (correspondant aux 4 directions principales) et 2 phases.
Plusieurs perspectives sont susceptibles d'améliorer encore les performances :

– Concernant les primitives, des améliorations telles que le type de fenêtre utilisé (fenêtre de Hamming, ...) peuvent encore être apportées.
– Concernant le modèle, on peut également faire varier certains paramètres, comme le nombre d'états.
– Enfin, on peut envisager des techniques de combinaison de systèmes. Dans le cadre de l'étude proposée ici, cette idée est renforcée par le fait que les images mal reconnues d'un système ne se retrouvent pas toutes dans les images mal reconnues d'un système donnant un taux d'erreurs plus élevé.

Références

[1] S. Chevalier, *Reconnaissance d'écriture manuscrite par des techniques markoviennes : une approche bidimensionnelle et générique*, Thèse de Doctorat, Université Paris 5, 2004.

[2] S. Chevalier, E. Geoffrois et F. Prêteux, *Programmation dynamique 2D pour la reconnaissance de caractères manuscrits par champs de Markov*, Reconnaissance des Formes et Intelligence Artificielle, 2004.

[3] E. Geoffrois, *Multi-dimensional Dynamic Programming for statistical image segmentation and recognition*, International Conference on Image and Signal Processing, 2003.

[4] E. Geoffrois, A. Jullian et C. Debaert, *Programmation dynamique 2D pour la reconnaissance d'images par modèle de Markov cachés*, Rapport technique, DGA/DCE/CTA/GIP, 1998.

[5] S. Geman et D. Geman, *Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images*. IEEE transactions on Pattern Analysis and Machine Intelligence, 1984.

[6] S. Chevalier, E. Geoffrois, F. Prêteux et M. Lemaître, *A generic 2D approach of handwriting recognition*, International Conference on Document Analysis and Recognition, 2005.

[7] H. Bunke et T. Caelli, *Hidden Markov Models Applications in computer vision*, Machine perception artificial intelligence vol. 45, 2001.

[8] A. Belaid et G. Saon, *Utilisation des processus markoviens en reconnaissance de l'écriture*, Traitement du signal, 1997.

[9] E. Augustin et S. Knerr, *A neural network–hidden Markov model hybrid for cursive word recognition*, International Conference on Pattern Recognition, 1998.

[10] I. Trier, A. Jain et T. Taxt, *Feature extraction methods for character recognition – A survey*, Pattern Recogntion, 1995.

[11] P. Dargenton, *Contribution à la Segmentation et à la reconnaissance de l'écriture manuscrite*, Thèse de Doctorat, 1994.

[12] X. Huang, A. Acero, H. Hon, *Spoken language processing*, 2001.

[13] Y. LeCun, The MNIST Database, *http ://yann.lecun.com/exdb/mnist/index.html*, 1998.