IJDAR (2002) 4: 281–289

International Journal on Document Analysis and Recognition IJDAR

Cross-learning in analytic word recognition without segmentation

C. Choisy, A. Belaïd

LORIA/CNRS, Campus scientifique, BP 239, 54506 Vandoeuvre-les-Nancy, France; e-mail: {choisy,abelaid}@loria.fr

Received: March 31, 2000 / Accepted: January 9, 2002

Abstract. In this paper a method for analytic handwritten word recognition based on causal Markov random fields is described. The word models are HMMs where each state corresponds to a letter modeled by a NSHP-HMM (Markov field). The word models are built dynamically. Training is operated using *Baum-Welch* algorithm where the parameters are reestimated on the generated word models. The segmentation is unnecessary: the system determines itself during training the best repartition of the information within the letter models. First experiments on two real databases of French check amount words give very encouraging results up to 86% for recognition without rejection.

Keywords: HMM, NSHP-HMM, Cross-learning, Metamodels, *Baum-Welch* Algorithm

1 Introduction

Research on writing recognition recently showed the superiority of 2D models on 1D ones. As asserted by different works [2,4,7,13,20], they take better into account the plane nature of the writing. The literature shows three types of 2D models: Neural Network (NN), Planar-HMM (PHMM) and Hidden Markov Mesh Random Fields (HMMRF). An original work uses 2D probability matrices to estimate letter positions in word images [12].

The NN can be applied either on letters [20] or on graphems [7]. They are used in [11] to model inter-character confidence. Their major drawback is their lack of elasticity: having a fixed input size, they cannot adapt to length variability, and they are very sensitive to important distortions. To deal with lentgh variability the use of specific NN such TDNN and recurrent NN were proposed [18,19]. The backdraw of this approach lies in the difficulty to automatically label the netwok observations according to the current observed letter; this information is necessary to correctly train the NN.

The PHMM was successfully applied in many works [2, 4]. Composed of secondary HMMs and a principal HMM for the correlation, this model has interesting 2D elasticity properties. But it requires an independence hypothesis between the secondary models which is not realistic in practice.

The HMMRF was applied on handwritten hangul characters recognition with good performances [13]. But it needs some non-realistic hypothesis to be tractable and its use remains very costly in computational time. Some other works deal with 2-dimensional warping under some specific constraints with interesting results [22,21].

G. Saon proposed in [16] a 2D model combining Markov fields and HMMs the NSHP-HMM (Non-Symmetric Half-plane Hidden Markov Model, cf. Sect. 2). Applied on binary images, it takes better into account the 2D writing nature by using 2D neighborhoods. The HMM part confers to it a horizontal elasticity enabling it to adapt to the analyzed samples length. Using a 2D neighborhood for the pixel observation, it overcomes the column independence hypothesis of the PHMMs. Its use as a global approach showed some limits. Particularly, the NSHP-HMM needs a high number of parameters (cf. Sect. 2). Furthermore, the efficiency of this approach is proved only for restricted and distinct vocabulary (similar words will lead to misclassification, small differences being absorbed by the models).

To overcome these limits, an analytic approach is proposed. It is based on a concatenation of letter models, allowing to work with a large vocabulary (words) by using restricted components (letters). Each letter is modeled by a NSHP-HMM. This also reduces the global complexity of the approach, which is limited to letter modeling.

Classically, the analytic word recognition approaches are leaned on grapheme segmentation [5,7,9] which cannot be 100% reliable because it is usually based on topological criterions [5,1]. For this reason, it seemed better to us to let the system decide itself which part of the image belongs to which letter. The use of the *Baum-Welch* algorithm [3,14] allows the system to find the best pa-

rameters repartition in the letter models, knowing only the label of the words learned [6].

The reestimation of letter models and transitions between letters is made by cross-learning. This technique is directly derived from the *Baum-Welch* reestimation formulas. It was used in [15] to automatically learn the graphems label in a segmentation-based approach; in this work transitions between letters are estimated on the word labels of the database. This approach needs to know the exact label of each word of the database. In our case no segmentation is necessary and all the parameters for letter and word models are estimated in the same time. No exact knowledge on word image labels is necessary: the only need is to model all the possible orthographies of each word in the corresponding class model.

2 Non-symmetric half-plane hidden Markov model

The NSHP-HMM is a stochastic model combining the properties of Markov fields and HMM. The observation probabilities in the HMM states (NSHP-HMM state) is estimated by a Markov Random Field (MRF). This probability is performed as the product of elementary probabilities performed on each pixel in the observed column. The elementary probability is determined by the MRF according to a 2D neighborhood fixed in the half plane previously analyzed. Figure 1 illustrates the application of such a model on an image (here a letter image manually segmented). Training and recognition methods are described in [16] and will be remembered in the next section.

The determinant NSHP-HMM parameters are: column height, neighborhood size (i.e. model order), HMM state number. To force the observation repartition in the NSHP-HMM states, two specific states D and F are added to the HMM. They allow to model the probability of beginning and ending in each state.

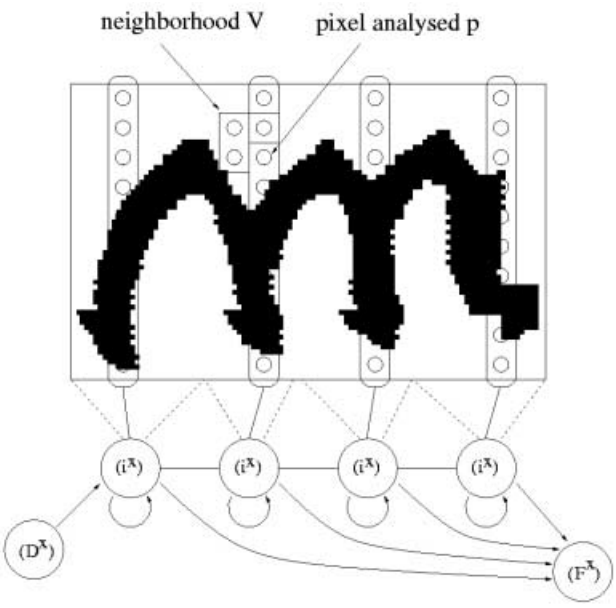

Fig. 1. Example of a NSHP-HMM applied on a letter, associated to the meta-state x

2.1 NSHP-HMM *learning*

The NSHP-HMM learning is based on the *Baum-Welch* algorithm. This algorithm ensures the convergence to a local optimum on the learning set. Let N be a NSHP-HMM with S normal states and the specific states D and F. K is the number of word images analyzed, O^k is the kth word image, T_k is the column number of the word image k, $P_k = P(O^k|n)$. a_{ij} is the transition probability between states i and j, where $i, j \in S$. $b_i(O_t^k)$ is the observation probability of the column t for the state $i \in S$, $i \neq D, F$. $\alpha_t^k(i)$, $\beta_t^k(i)$ and $P(O^k|n)$ are derived from [14] as follows:

1. $\alpha_1^k(i) = a_{Di}\ b_i(O_1^k)$
2. $\alpha_t^k(i) = \left(\sum_{j=1}^{S} \alpha_{t-1}^k(j)\ a_{ji} \right)\ b_i(O_t^k)$

1. $\beta_T^k(i) = a_{iF}$
2. $\beta_t^k(i) = \sum_{j=1}^{S} \beta_{t+1}^k(j)\ b_j(O_{t+1}^k)\ a_{ij}$

$$P(O^k|n) = \sum_{i=1}^{S} \alpha_t^k(i)\ \beta_t^k(i) = \sum_{i=1}^{S} \alpha_T^k(i)\ a_{iF}$$

The transition probabilities between the NSHP-HMM state are reestimated similarly as a classical HMM:

– for the transitions leaving the specific state D:

$$\overline{a_{Di}} = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{P_k}\ \alpha_1^k(i)\ \beta_1^k(i) \tag{1}$$

– for the transitions between normal states:

$$\overline{a_{ij}} = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i)\ a_{ij}\ b_j(O_{t+1}^k)\ \beta_{t+1}^k(j)}{\sum_{k=1}^{K} \frac{1}{P_k} \left[\sum_{t=1}^{T_k-1} \alpha_t^k(i)\ \beta_t^k(j) + \alpha_{T_k}^k(i)\ a_{iF} \right]} \tag{2}$$

– for the transitions towards the specific state F:

$$\overline{a_{iF}} = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \alpha_{T_k}^k(i)\ a_{iF}}{\sum_{k=1}^{K} \frac{1}{P_k} \left[\sum_{t=1}^{T_k-1} \alpha_t^k(i)\ \beta_t^k(j) + \alpha_{T_k}^k(i)\ a_{iF} \right]} \tag{3}$$

The observation probability reestimation consists in a counting of the neighborhood configurations in each state

for each pixel of each column analyzed. Let $V(p_t, p_y)$ be the neighborhood of the pixel at position (p_t, p_y) in the current analyzed sample. Let V be the structure of the neighborhood for which probability is reestimated. The reestimation is based on a counting of the pixels of color c observed with a such neighborhood V. The two values for c are *black* and *white* and the probability of a *white* pixel knowing V is $1 - P(black|V)$; thus the reestimation for the black pixels observation is sufficient. The observation probability at vertical position y for color c with a neighborhood V in the state i is:

$$\overline{b_i}(y, V, c) = \frac{\displaystyle\sum_{k=1}^{K} \frac{1}{P_k} \sum_{\substack{t=1 \\ V(p_t,p_y)=V \\ (p_t,p_y)\ is\ c}}^{T_k} \alpha_t^k(i)\beta_t^k(i)}{\displaystyle\sum_{k=1}^{K} \frac{1}{P_k} \sum_{\substack{t=1 \\ V(p_t,p_y)=V}}^{T_k} \alpha_t^k(i)\beta_t^k(i)} \tag{4}$$

3 Word modeling

For word modeling we use meta-HMMs in which each meta-state represents a letter (see Fig. 2) but without loop on the states for technical reasons. A meta-HMM is an HMM modeling a meta-level representation. Here the meta-HMM describes the word-level representation.

Let m be a meta-model representing a word with S_m normal states and the specific states D_m and F_m. At each meta-state $x \in S_m$ is associated a letter $l(x)$ to which corresponds a NSHP-HMM l with S^x normal states and the specific states D^x and F^x; i^x denotes the state i of the letter model associated to the meta-state x (cf. Fig. 1).

Starting from a meta-model, a global NSHP-HMM is built by connecting the NSHP-HMM associated to the meta-model letter states. Each state sequence of type $i^x \rightarrow F^x \rightarrow D^y \rightarrow j^y$ is replaced by one transition $i^x \rightarrow j^y$, whose value is the product of the transitions between these states:
$P(j^y|i^x) = P(j^y|D^y) * P(D^y|F^x) * P(F^x|i^x)$
Following the same idea, the beginning and ending transitions are given by:
$P(i^x|D_m) = P(i^x|D^x) * P(D^x|D_m)$
$P(F_m|i^x) = P(F_m|F^x) * P(F^x|i^x)$
Traduced in term of transitions these formulas give:
$a_{i^x j^y} = a_{D^y j^y}\ a_{F^x D^y}\ a_{i^x F^x}$

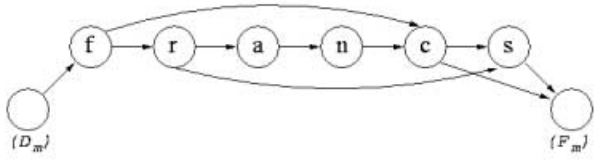

Fig. 2. Example of a meta-model architecture for the word "*francs*", including frequent misspellings as "franc", "frans", and abbreviations as "frans", "frs", etc.

$a_{D_m i^x} = a_{D^x i^x}\ a_{D_m D^x}$
$a_{i^x F_m} = a_{F^x F_m}\ a_{i^x F^x}$

Figure 3 shows the substitution of the meta-states by the corresponding NSHP-HMM and the transition transformation where only the NSHP-HMM states are drawn. The consideration of self-loop on a meta-state allows to build transitions between states in the corresponding letter model. This will erase the existing transitions. For this reason, the feedback on the meta-states is forbidden.

4 Cross-learning

Cross-learning for transitions in the letter models is described using the same notation than seen in Sect. 2.1 and in Sect. 3. This description will be followed by that of the observation probability reestimation which follows the same principle.

During the construction of the global model, the specific states D^x and F^x are removed. The reestimation of the transitions $a_{D^x i^x}$ and $a_{i^x F^x}$ is made using the transitions built with them.

To simplify the next equations we note:
$\omega^k(i^x, j^y, t) = \alpha_t^k(i^x)\ a_{i^x j^y}\ b_{j^y}(O_{t+1}^k)\ \beta_{t+1}^k(j^y)$.

Intuitively $\omega^k(i^x, j^y, t)$ is the sum of all paths using the transition $a_{i^x j^y}$ to go from column O_t^k to column O_{t+1}^k.

For a model associated to the meta-state x:

- the transition $a_{D^x i^x}$ is removed when the transitions $a_{j^y i^x}$, $y \neq x$, and $a_{D_m i^x}$ are built
- the transition $a_{i^x F^x}$ is removed when the transitions $a_{i^x j^y}$, $y \neq x$, and $a_{i^x F_m}$ are built
- the internal transitions $a_{i^x j^x}$ leave unchanged.

The principle of the cross-reestimation is to synthesize this information for all the models associated with a same letter in the various meta-models. For the internal transitions the *Baum-Welch* formulae can be applied directly by summing the transitions over all the occurrences of a letter model in all the word models. For the transitions $a_{D^x i^x}$ and $a_{i^x F^x}$ this sum is made through the sum of the paths containing the transitions built with them. M is the number of word models.

For a model associated to the meta-state x:
The transition $a_{D^x i^x}$ is used to build:

- the transitions $a_{j^y i^x}$, $y \neq x$
- the transition $a_{D_m i^x}$

The balanced sum on all the paths using this transition in the model m is:

$$W_{D^x i^x}^m = \sum_{k=1}^{K} \frac{1}{P_k} \left[\begin{array}{c} \displaystyle\sum_{t=1}^{T_k-1} \sum_{\substack{y=1 \\ y\neq x}}^{S_m} \sum_{j^y=1}^{S^y} \omega^k(j^y, i^x, t) \\ +a_{D_m i^x} b_{i^x}(O_1^k)\beta_1^k(i^x) \end{array} \right] \tag{5}$$

The transition $a_{i^x F^x}$ is used to build:

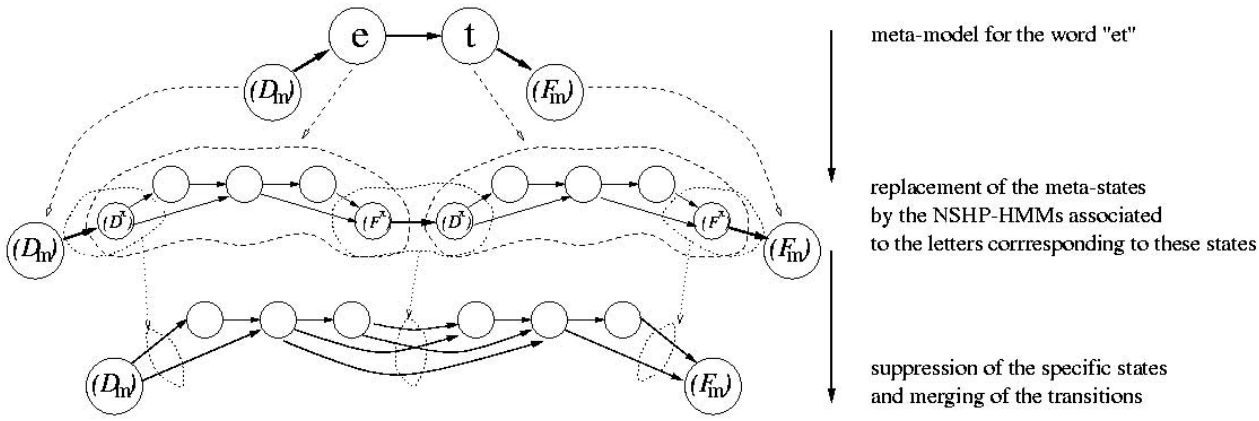

Fig. 3. Construction principle of the global NSHP-HMM corresponding to the word "et"

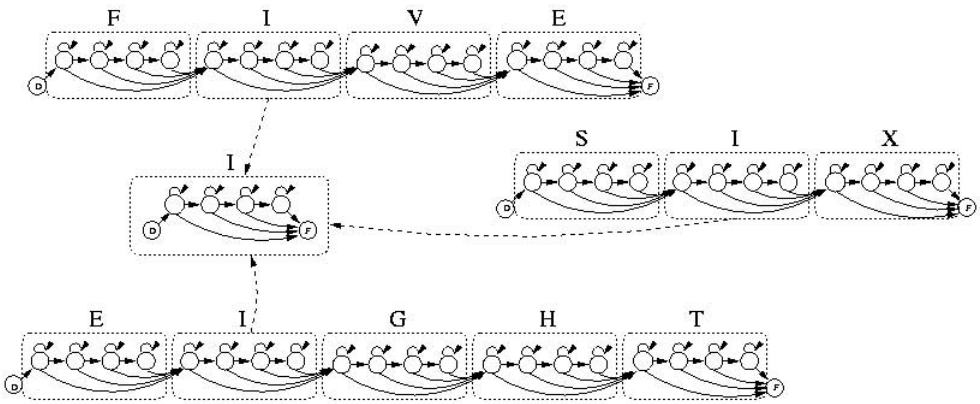

Fig. 4. Example of a cross-learning for the letter "i" model

– the transitions $a_{i^x j^y}$, $y \neq x$
– the transition $a_{i^x F_m}$

The balanced sum on all the paths using this transition in the model m is:

$$W^m_{i^x F^x} = \sum_{k=1}^{K} \frac{1}{P_k} \left[\begin{array}{c} \displaystyle\sum_{t=1}^{T_k-1} \sum_{\substack{y=1 \\ y\neq x}}^{S_m} \sum_{j^y=1}^{S^y} \omega^k(i^x, j^y, t) \\ +\alpha^k_{T_k}(i^x) a_{i^x F_m} \end{array} \right] \tag{6}$$

For the internal transitions $a_{i^x j^x}$ of the model x, we have classically:

$$W^m_{i^x j^x} = \sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k-1} \omega^k(i^x, j^x, t) \tag{7}$$

$l(x)$ is the letter associated with a meta-state x. According to the *Baum-Welch* reestimation formulas, the transitions for a model of letter l are reestimated as follows:

– for the transitions leaving the specific state D:

$$\overline{a^l_{Di}} = \frac{\displaystyle\sum_{m=1}^{M} \sum_{\substack{x=1 \\ l(x)=l}}^{S_m} W^m_{D^x i^x}}{\displaystyle\sum_{m=1}^{M} \sum_{\substack{x=1 \\ l(x)=l}}^{S_m} \sum_{j^x=1}^{S^x} W^m_{D^x j^x}} \tag{8}$$

– for the transitions between normal states:

$$\overline{a^l_{ij}} = \frac{\displaystyle\sum_{m=1}^{M} \sum_{\substack{x=1 \\ l(x)=l}}^{S_m} W^m_{i^x j^x}}{\displaystyle\sum_{m=1}^{M} \sum_{\substack{x=1 \\ l(x)=l}}^{S_m} \left[\sum_{k^x=1}^{S^x} W^m_{i^x k^x} + W^m_{i^x F^x} \right]} \tag{9}$$

– for the transitions towards the specific state F:

$$\overline{a^l_{iF}} = \frac{\displaystyle\sum_{m=1}^{M} \sum_{\substack{x=1 \\ l(x)=l}}^{S_m} W^m_{i^x F^x}}{\displaystyle\sum_{m=1}^{M} \sum_{\substack{x=1 \\ l(x)=l}}^{S_m} \left[\sum_{k^x=1}^{S^x} W^m_{i^x k^x} + W^m_{i^x F^x} \right]} \tag{10}$$

The observation reestimation is similar to the reestimation of the transitions between normal states. It is a counting of the neighborhood configurations in each state for each pixel of each column analyzed. Let remember the notations of Sect. 2.1. Let $V(p_t, p_y)$ be the neighborhood of the pixel (p_t, p_y) in the current analyzed sample. Let V be the structure of the neighborhood of which we reestimate the probability. The reestimation is a counting of the pixels of color c observed with such neighborhood V. The two values for c are *black* and *white* and the probability of a *white* pixel knowing V is $1 - P(black|\mathrm{V})$; thus the reestimation for the black pixels observation is sufficient. $W^m_{i^x}(y, V, c)$ is the balanced sum of the pixels colored in c at height y for which $V(p_t, p_y) = V$ on all paths using the state i^x in the meta-model m, where t is the column currently observed by the state i^x:

$$W^m_{i^x}(y, V, c) = \sum_{k=1}^{K} \frac{1}{P_k} \sum_{\substack{t=1 \\ V(p_t,p_y)=V \\ (p_t,p_y)\ is\ c}}^{T_k} \alpha^k_t(i^x)\beta^k_t(i^x) \tag{11}$$

Thus the observation probability for the pixel y of color c with a neighborhood V in the state i of a letter l is:

$$\overline{b^l_i}(y, V, c) = \frac{\displaystyle\sum_{m=1}^{M} \sum_{\substack{x=1 \\ l(x)=l}}^{S_m} W^m_{i^x}(y, V, c)}{\displaystyle\sum_{m=1}^{M} \sum_{\substack{x=1 \\ l(x)=l}}^{S_m} \begin{bmatrix} W^m_{i^x}(y, V, black) \\ +W^m_{i^x}(y, V, white) \end{bmatrix}} \tag{12}$$

The boundary conditions are not described here. Intuitively, we can consider that the pixels of $V(p_t, p_y)$ that aren't in the image are white. A post-processing is added to correct the probabilities lower than a threshold. This correction allows the neighborhoods V that rarely appear in the learning set to have a non-null probability. This threshold was experimentally fixed to 0.01.

5 Meta-model reestimation

lobal models are build from meta-models. These are HMMs and the transitions between the meta-state of a meta-model m from the information of the generated models can be reestimated. Indeed, for $x, y \in S_m$, we obtain by construction:
$a_{xy} = a_{F^x D^y},\ a_{D_m x} = a_{D_m D^x},\ a_{xF_m} = a_{F^x F_m}$.

- the transition a_{xy}, $x \neq y$, is used to build the transitions $a_{i^x j^y}$
- the transition $a_{D_m x}$ is used to build the transitions $a_{D_m i^x}$
- the transition a_{xF_m} is used to build the transitions $a_{i^x F_m}$

As for the cross-learning, according to the *Baum-Welch* formulas, the reestimation of a meta-transition of a meta-model m is made by summing on all the paths containing the transition using it:

$$\overline{a_{xy}} = \frac{\displaystyle\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k-1} \sum_{i^x=1}^{S^x} \sum_{j^y=1}^{S^y} \omega^k(i^x, j^y, t)}{\displaystyle\sum_{k=1}^{K} \frac{1}{P_k} \begin{bmatrix} \displaystyle\sum_{t=1}^{T_k-1} \sum_{i^x=1}^{S^x} \sum_{\substack{z=1 \\ z\neq x}}^{S_m} \sum_{j^z=1}^{S^z} \omega^k(i^x, j^z, t) \\ \displaystyle + \sum_{i^x=1}^{S^x} \alpha^k_{T_k}(i^x) a_{i^x F_m} \end{bmatrix}} \tag{13}$$

$$\overline{a_{D_m x}} = \frac{\displaystyle\sum_{k=1}^{K} \frac{1}{P_k} \sum_{i^x=1}^{S^x} a_{D_m i^x} b_{i^x}(O_1) \beta^k_1(i^x)}{\displaystyle\sum_{k=1}^{K} \frac{1}{P_k} \sum_{z=1}^{S_m} \sum_{i^z=1}^{S^z} a_{D_m i^z} b_{i^z}(O_1) \beta^k_1(i^z)} \tag{14}$$

$$\overline{a_{xF_m}} = \frac{\displaystyle\sum_{k=1}^{K} \frac{1}{P_k} \sum_{i^x=1}^{S^x} \alpha^k_{T_k}(i^x) a_{i^x F_m}}{\displaystyle\sum_{k=1}^{K} \frac{1}{P_k} \begin{bmatrix} \displaystyle\sum_{t=1}^{T_k-1} \sum_{i^x=1}^{S^x} \sum_{\substack{z=1 \\ z\neq x}}^{S_m} \sum_{j^z=1}^{S^z} \omega^k(i^x, j^z, t) \\ \displaystyle + \sum_{i^x=1}^{S^x} \alpha^k_{T_k}(i^x) a_{i^x F_m} \end{bmatrix}} \tag{15}$$

6 Experiments

The first experiments are made on French bank check words. The lexicon contains 26 word classes. Each class can contain some orthographical variations that are modeled by the corresponding meta-model (cf. Fig. 2). A meta-model is associated to each lexicon entry. The global models associated to each word class are dynamically built using the class meta-models and the letter NSHP-HMM. While normalizing proportionally the images at a fixed height the scan resolution has only influence on the sample quality.

The system was tested on a database of 7031 word images given by the SRTP [1], and a database of 25260 word images from an industrial real application. The parameters of the NSHP-HMM for the letter models are: height of 20 pixels, 3 pixels for the neighborhoods; the number of normal states for the NSHP-HMM corresponding to a letter is $\overline{n}/2 + 1$, where $\overline{n}$ is the average number of columns

[1] Service de Recherche Technique de la Poste: French Post Research Team

of normalized samples for the letter (tis number was estimated from a database manually segmented).

For each meta-model, 4 NSHP-HMMs are created corresponding to the 4 flips of images. The probability of an image is the product of the 4 probabilities, obtained by each model. Our meta-models synthesize the frequent misspellings found in the words.

Each learning step is carried out as follows:

- global word models are built from letter models and word meta-models
- each global model is trained on the samples of the corresponding class
- information is crossed between the different global models to reestimate both letter models and word meta-models

Recognition is carried out as follows:

- global word models are built from letter models and word meta-models
- the sample to recognize is analyzed by each global model
- the model obtaining the best score determines the sample class

6.1 Pre-processings

For the two databases, two preprocessing are applied to reduce the writing variability. The first one is a slant correction, as proposed in [17]. The second one normalizes the word height to 20 pixels (height of letters models), by normalizing the three writing bands in three equal vertical parts. This second preprocessing allows a better synchronization of the information between the different samples, increasing the redundancies of information and thus improving the learning and recognition quality. To normalize a sample image, the median band (lower case band) of writing is detected using the product of the histogram of the horizontal projection of black points with the histogram of writing natural length [10,8] (number of transitions between black and white pixels) for each line of the sample. The histogram obtained is smoothed. The limits of the median band of writing are determined when the histogramme value falls below a threshold (1/4 of the maximum value of the histogram). The three bands separated by these limits are put in three equal parts of the analyzed sample. Figure 5 shows the effect of this normalization for two samples of the word "deux".

Two tests are performed to validate the cross-learning principle. The interest of this method is that all the models and the meta-models can theoretically be learn in the same time.

6.2 Experiments on the SRTP database

The database was approximately split in 66% (4627 words) for cross-learning and 34% (2404 words) for recognition tests. Word meta-models and letter models are generated with equal transition and observation probabilities (no specific initialization) and the cross-learning is applied at several steps. The results are related in Table 1.

Table 1. Average word recognition rates for different numbers of learning steps

cross learning	Top 1	Top 2	Top 3	Top 4	Top 5
5 steps	81.66%	89.98%	92.64%	94.68%	95.80%
10 steps	84.36%	91.51%	93.97%	95.97%	96.92%
15 steps	84.82%	91.26%	94.13%	96.26%	97.00%
20 steps	85.15%	91.60%	94.09%	96.05%	97.34%
25 steps	85.07%	91.68%	94.34%	96.05%	97.17%
35 steps	85.23%	91.93%	94.47%	96.30%	97.13%
45 steps	85.73%	92.01%	94.68%	96.26%	97.17%
55 steps	85.82%	92.14%	94.76%	96.34%	97.21%
65 steps	86.02%	92.14%	94.76%	96.51%	97.25%
global approach	90.08%	- - -	92.60%	- - -	

Table 2. Average word recognition rates on learning set

cross learning	Top 1	Top 2	Top 3	Top 4	Top 5
5 steps	88.02%	94.43%	96.84%	97.96%	98.50%
10 steps	91.47%	96.51%	98.04%	98.63%	99.08%
15 steps	92.55%	96.92%	98.21%	98.79%	99.08%
20 steps	92.60%	96.84%	98.21%	98.92%	99.13%
25 steps	92.60%	96.76%	98.21%	98.88%	99.21%
35 steps	92.60%	96.84%	98.25%	98.88%	99.21%
45 steps	92.85%	96.80%	98.29%	98.96%	99.25%
55 steps	92.93%	96.96%	98.29%	99.00%	99.21%
65 steps	92.93%	97.13%	98.38%	99.13%	99.42%

The results are relatively good remembering that no initialization are made on the letter models (equal probabilities). We can consider 20 learning steps for a correct stabilization of the system. This first test suffers from a too small database with a lot of classes badly represented. The results can be compared to the global approach on the same database in the same table [16]. As shown, cross-learning gives lower result than global approach in top 1, but top 3 is higher. Another interesting comparison can be made with Table 2. The high difference between scores reflect the lack of samples in this database. Thus the training set is not representative of the whole set of possible samples. With a more complete database, the recognition scores for test set should be improved.

6.3 Experiment on the industrial database

For this database the sample distribution gives 16660 words (66%) for cross-learning and 8600 words (34%) for recognition tests. Experiment conditions are similar to the previous test: word and letter models are initialized with equal probabilities. The results are related in Table 3.

Fig. 5. Normalization in 3 bands for 2 samples of the word "deux". The lower case band of the two word images is centered in the same zone. This increases redundancies between samples

Table 3. Average word recognition rates for the industrial database

cross learning	Top 1	Top 2	Top 3	Top 4	Top 5
5 steps	79.14%	87.56%	90.93%	92.95%	94.37%
10 steps	81.55%	89.50%	92.53%	94.22%	95.50%
15 steps	82.30%	89.84%	92.92%	94.53%	95.78%
20 steps	82.67%	90.05%	93.02%	94.73%	96.02%
25 steps	82.73%	90.16%	93.20%	94.80%	96.14%
30 steps	82.81%	90.20%	93.33%	94.84%	96.20%
35 steps	82.95%	90.28%	93.38%	94.94%	96.20%
40 steps	83.03%	90.28%	93.34%	94.91%	96.22%
45 steps	83.09%	90.33%	93.31%	94.95%	96.21%
50 steps	83.16%	90.31%	93.38%	94.98%	96.23%
global approach	82.50%	89.56%	92.72%	94.57%	95.74%

Table 4. Average word recognition rates on learning set

cross learning	Top 1	Top 2	Top 3	Top 4	Top 5
5 steps	80.16%	87.88%	91.28%	93.34%	94.70%
10 steps	82.85%	89.94%	92.97%	94.87%	96.09%
15 steps	83.38%	90.50%	93.29%	95.24%	95.24%
20 steps	83.84%	90.55%	93.42%	95.42%	96.49%
25 steps	84.08%	90.63%	93.48%	95.57%	96.60%
30 steps	84.12%	90.66%	93.60%	95.63%	96.60%
35 steps	84.14%	90.65%	93.66%	95.60%	96.58%
40 steps	84.09%	90.67%	93.71%	95.58%	96.60%
45 steps	84.12%	90.76%	93.66%	95.64%	96.66%
50 steps	84.13%	90.91%	93.70%	95.67%	96.66%

Results are low comparing to the precedent test. They show the quality difference between the two databases. Particularly the preprocessing quality seems to be very different between the two databases (samples seem cleaner and have a better look in first database). Table 4 shows the scores on the training set.

The comparison between results on training and test set shows that they are close. It indicates that the training set is relatively representative of the most real cases. Some classes remain poorly represented leading to a bad letter context learning for corresponding words. Results can be compared with the global approach on another industrial database [16]. Learning conditions are somewhat better (90% for learning / 10% for testing, 36829 samples). Comparison shows that the analytic approach is efficient on industrial cases.

In these two experiments the initial database contains 28 words, the shortcuts "frs" and "cts" being separated of the words "francs" and "centimes". These classes were mixed to have the same conditions of the test on the SRTP database. Some further tests could show the impact of the separation of these classes in the learning set and training set.

6.4 Complexity

This approach allows the reduction of the complexity of the system proposed by G. Saon in [16,17]. The number of floating point operations is highly dependent of the number of states in the models because each state analyzes each column of a word when using the *Baum-Welch* algorithm (cf. α function Sect. 2.1). The α function complexity for a word NSHP-HMM having n states and observing y pixels and a sample having t columns is $\mathcal{O}(n^2 t)$ additions and $\mathcal{O}((n^2+n)t)$ multiplications for the transition part, and $\mathcal{O}(nyt)$ for the observation part. This function is sufficient to perform a sample probability. By reducing the global state number, we reduce the observation calculation part. More, while considering letter models, this part becomes vocabulary size independent. The global approach proposed by G. Saon has a high number of states, based on the mean size of words. Our approach limits this number by considering the mean size of the letters. A first estimation gives a reduction of a factor 7 for the number of states in our approach: G. Saon used 615 states with his approach and the analytic approach reduces this number to 87 states; that divides by 7 the floating point multiplications necessary for the observation calculation by pre-performing for each state

of each letter model the observation probability of each column of the sample.

At the same time, we observed that a neighborhood of size $v = 4$ is too high for the letter recognition and we choose a size of $v = 3$. The number of observation parameters for a letter NSHP-HMM having n states and y pixels observed in each state is $ny2^v$. The neighborhood size reduction divides by 2 this number. The combination of these two factors allows a reduction of a factor 14 for the number of parameters to estimate for state observations. The transition part of the NSHP-HMM requires less than $n^2 + 4n$ (case of a fully connected NSHP-HMM with D and F). In practice the NSHP-HMM used have left-right structure and thus need $3n + 1$ transitions. Thus transition parameters are divided at least by 7.

7 Conclusion

We proposed a new approach for analytic word recognition based on a dynamic generation of global models. This approach reduces the number of observation parameters of the global approach by a factor 14, and the complexity in floating point multiplications for the observation probabilities calculation by 7. The first experiments give very encouraging results on industrial database of real application. The letter models learning is made through the word models and the *Baum-Welch* algorithm ensures the optimal learning in good conditions. More tests need to be made with bigger databases in order to evaluate in better conditions such an approach. Some tests with a lower neighborhood could show the importance of its size.

The word models are dynamically generated, corresponding to meta-models. This method can easily be extended to entire amounts of checks with another level of meta-models in which states would represent words. This extension needs that we can find the best path between words. This problem is the same with a generalization of our method to unconstrained vocabulary recognition. For such a task, we need to find the sequence of states in the meta-model than best describe the word analyzed. Some studies are necessary to find the best method to get the path in the meta-models.

References

1. A. Agarwal, A. Negi, K. S. Swaroop: A correspondence based approach to segmentation of cursive words. In: Third International Conference on Document Analysis and Recognition (ICDAR'95), Montréal, 1995
2. O. E. Agazzi, S. Kuo: Hidden Markov model based optical character recognition in the presence of deterministic transformation. Pattern Recognition 26(12):1813–1826 (1993)
3. L. E. Baum: Statistical inference for probabilistic functions of finite state Markov chains. Ann Math Stat 37:1554–1563 (1968)
4. R. Bippus: 1-dimensional and pseudo 2-dimensional HMMs for the recognition of German literal amounts. In: Fourth International Conference on Document Analysis and Recognition (ICDAR'97), vol. 2, pp 487–490, Ulm, Germany, 1997
5. M.-Y. Chen, A. Kundu, J. Zhou: Off-line handwritten word recognition using a hidden Markov model type stochastic network. IEEE Trans Pattern Recognition Mach Intell 16(5):481–497 (1994)
6. C. Choisy, A. Belaïd: Analytic word recognition without segmentation based on Markov random fields. In: Seventh International Workshop on Frontiers in Handwriting Recognition (IWFHR-VII), The Netherlands, Sept. 2000
7. M. Gilloux, B. Lemarié, M. Leroux: A hybrid radial basis function network/hidden Markov model handwritten word recognition system. In: Third International Conference on Document Analysis and Recognition (ICDAR'95), pp. 394–397, Montréal, 1995
8. G. Kaufmann, H. Bunke, M. Hadorn: Lexicon reduction in an hmm-framework based on quantized feature vectors. In: Fourth International Conference on Document Analysis and Recognition (ICDAR'97), vol. 2, pp. 1097–1101, Ulm, Germany, 1997
9. F. Kimura, M. Shridhar, Gilles Houle: Handwritten word recognition using lexicon free and lexicon directed word recognition algorithms. In: Third International Conference on Document Analysis and Recognition (ICDAR'95), pp. 82–85, Montreal, 1995
10. S. Madvanath, V. Govindaraju, V. Ramanaprasad, D. S. Lee, S. N. Shriari: Reading handwritten US census forms In: Fourth International Conference on Document Analysis and Recognition (ICDAR'97), Ulm, Germany, 1997
11. M. Mohamed, P. D. Gader, J.-H. Chiang: Handwritten word recognition with character and inter-character neural networks. IEEE Trans Syst Man Cyber 27(2):158–164 (1997)
12. M. A. Ozdil, F. T. Yarman-Vural: Optical character recognition without segmentation. In: Fourth International Conference on Document Analysis and Recognition (ICDAR'97), Ulm, Germany, 1997
13. H. S. Park, S. W. Lee: An HMMRF-based statistical approach for off-line handwritten character recognition. In: Proc IEEE 2:320–324 (1996)
14. L. R. Rabiner: A tutorial on hidden Markov models and selected applications in speech recognition. Proc IEEE 77(2): (1989)
15. R. Sabourin, A. El-Yacoubi, M. Gilloux, C. Y. Suen: An HMM-based approach for off-line unconstrained handwritten word modeling and recognition. IEEE Trans Pattern Recognition Mach Intell 21(8):752–760 (1999)
16. G. Saon: Modèles markoviens uni- et bidimensionnels pour la reconnaissance de l'écriture manuscrite hors-ligne. PhD thesis, Université Henri Poincaré - Nancy I, Vandœuvre-lès-Nancy, 1997.
17. G. Saon, A. Belaïd: Off-line handwritten word recognition using a mixed HMM-MRF approach. In: Fourth International Conference on Document Analysis and Recognition (ICDAR'97), vol. 1, pp. 118–122, Ulm, Germany, 1997
18. A. W. Senior: Off-line handwriting recognition: a review and experiments. Technical report, Cambridge University Engineering Department, Cambridge, 1992
19. A. W. Senior, A. J. Robinson: An off-line cursive handwriting recognition system. IEEE Trans Pattern Recognition Mach Intell 20(3):309–321 (1998)

20. J. C. Simon, O. Baret, N. Gorski: A system for the recognition of handwritten literal amounts of checks. In: Internal Association for Pattern Recognition Workshop on Document Analysis System (DAS'94), Kaiserlautern, Germany, pp. 135–155, September 1994
21. S. Uchida, M. A. Ronee, H. Sakoe: Handwritten character recognition using piecewise linear two-dimensional warping. In: Sixth International Conference on Document Analysis and Recognition (ICDAR'2001), pp. 39–43, Seattle, Washington, USA, Sept. 10–13, 2001
22. S. Uchida, H. Sakoe: An efficient two-dimensional warping algorithm. IEICE Trans Inf Syst E82-D(3):693–700 (1999)

Christophe Choisy studied at the University Louis Pasteur of Strasbourg where he obtained his Bachelor's degree in computer sciences in 1996. He continued his studies in Nancy, France, where he obtained his M. Sc. in computer sciences in 1997 at the University of Nancy 2. He is now doing his Ph.D thesis in this university (4rd year) on the use of Markov fields in analytic handwriting recognition. His other research interests are in the use of elastic models for image normalization, applied to analytic handwriting recognition. He presented these two aspects in international conferences.

Abdel Belaïd received his Ph.D degree in Computer Science in 1979 and his D.Sc. in 1987 from the University Henri Poincaré Nancy I, France. After a few years as Assistant Professor in the University HP and Nancy 2, he joined the National Center for Scientific Research (CNRS) as a Research Scientist in 1984. He leads since 1992 a research group at the UMR LORIA 7503 working on Document Analysis and Text Recognition. His areas of research include Image Processing, Pattern Recognition, Document Analysis and Handwriting Recognition where he has authored over 120 articles which have been published in international journals and conferences. He is the co-author of a book, Pattern Recognition: Methods and Applications. He has developed retro-conversion techniques for document structure recognition using multi-agent systems, emergent architectures and part of speech tagging. He developed handwriting recognition systems based on stochastic modeling, for linear and bi-dimensional representations like HMMs, planar-HMMs and random fields. With his group, he was involved in different national and European projects such as the MORE project on Digital Libraries and has a big interaction and collaboration with French companies. He belongs to the scientific committees of many national and international conferences. He organized and chaired different conferences and workshop in the field of document analysis. He is member of different groups and associations in France (GRCE, ASTI, GDR-PRC I3) related to pattern recognition and writing recognition.