2D Markovian modeling for character recognition and segmentation

Sylvain Chevalier*, Edouard Geoffrois** and Françoise Prêteux*
*: ARTEMIS project Unit, GET/INT, Evry - France
**: Centre d'Expertise Parisien, DGA/DET/CEP, Arcueil - France

Abstract: Processing text components in multimedia contents remains a challenging issue for document indexing and retrieval. More specifically, handwritten characters processing is a very active field of pattern recognition. This paper describes an innovative two-dimensional approach for character recognition and segmentation. The method proposed combines Markovian modeling, efficient decoding algorithm together with a windowed spectral features extraction scheme. A rigorous evaluation methodology is achieved to analyse and discuss the performances obtained for digit and word recognition.

Key-Words: Handwriting recognition, Markov Random fields, 2D dynamic programming.

1. Introduction

Handwriting analysis has been performed by applying a wide variety of methodologies [8]. Geometric approaches [13] are simple to implement but their globality is a serious limitation. Handwritten words analysis can be efficiently processed with robust one-dimensional statistical methods based on Markov chains with good results on constrained tasks [9]. However, the 2D nature of the handwriting is obvious but no fully satisfying 2D approach has been found yet: statistical models such as pseudo-2D [6] and causal 2D models [12] attempt to solve this problem but they are limited by directional independence and causal hypothesis respectively.

We propose a fully 2D approach of handwriting recognition that can be applied to every step of document processing and we apply it to handwritten digits and handwritten words recognition. Most of the techniques performed are well-known (except two-dimensional dynamic programming, explained in Section 2.2) but the proposed combination is original. Markov chains, spectral features and dynamic programming have been successfully used in speech processing while Markov random field modeling and local feature extraction are key tools for image analysis. The synopsis of these interactions is shown on Figure 1. Section 2 addresses the main theoretical background of our approach while Section 3 describes how these principles are applied to a digit recognition task. Section 4 extends the proposed method to handwritten words recognition. Section 5 concludes the paper and opens new perspectives for improving and extending the presented 2D Markov modeling.

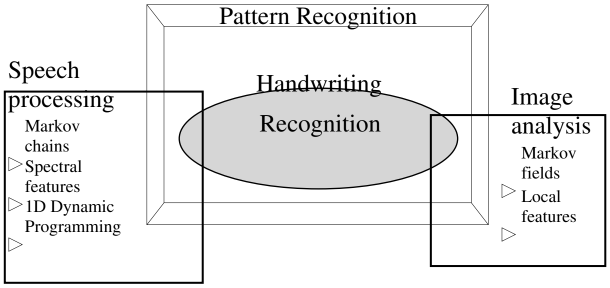

Figure 1. Synopsis of the synergies between Handwriting recognition, image analysis and speech processing.

2. Approach

The framework of our approach is based on Markov models which are popular statistical models for pattern recognition.

2.1. Markov Random Fields

Markov models are widely used for a variety of problems in pattern recognition [3]. It is based on the markovian assumption of short term dependency which seems to be valid for most of the images encountered in computer vision.

In this context, an image I is a set of sites (i,j) associated to labels $\omega_{i,j} \in S$, where $S = \{s_1, s_2, ..., s_N\}$ is the set of states of the model. A region R is a subset of adjacent sites of an image, and the associated set of labels is the configuration of the region ω_R.

The markovian assumption assumes that the dependency between the states of the sites reduces to a local one:

$$P(\omega_{i,j} \mid \omega_{I\backslash(i,j)}) = P(\omega_{i,j} \mid \omega_{N(i,j)}), \qquad (1)$$

where $N(i,j)$ is the set of sites which are neighbors of (i,j). N is a neighboring function if, for every pair of sites (i,j) and (k,l),

$$(i,j) \in N(k,l) \Leftrightarrow (k,l) \in N(i,j). \qquad (2)$$

A convenient way to handle neighboring relations is to use cliques: a clique is a set of sites which are neighbors. In 4-connexity, cliques correspond to single sites and vertical and horizontal pairs of sites.

With this formalism it is possible to use the Gibbs distributions which are equivalent to a Markov Random Field [1]:

$$P(\omega) = \frac{1}{Z}\exp(-\sum_{c\in C} V_c(\omega)) \qquad (3)$$

where C is the set of cliques, V_c is a potential function associated with cliques c and Z is a normalisation constant so that $\sum_\omega P(\omega) = 1$.

Hidden Markov Random Fields (HMRF) are a class of Markov fields with an observation layer. Each site of an image is associated to an observation which can be a number or a vector. Let us denote the observed image $O = \{o_{i,j}\}$. The observation of one site only depends on the underlying hidden state:

$$P(O \mid \omega) = \prod_{i,j} P(o_{i,j} \mid \omega_{i,j}). \qquad (4)$$

The problem of finding the optimal configuration reduces to the problem of finding $\hat{\omega}$ that minimizes:

$$U(\omega) = \sum_{(i,j)} -\log(P(o_{(i,j)} \mid \omega_{(i,j)})) + \sum_{c\in C} V_c(\omega). \qquad (5)$$

2.2. Decoding algorithm

Given a Markov model, the decoding procedure aims at assigning labels to the sites. Given the parameters of the model, the optimal configuration is defined as:

$$\hat{\omega} = \arg\max_{\omega} P(O \mid \omega). \qquad (6)$$

Several methods have been proposed to perform this maximization such as simulated annealing [4], which is very slow, or Iterated Conditional Modes (ICM) [2], which give a sub-optimal solutions. More restrictive assumptions such as causality in the Markov modeling can reduce the decoding to a 1D problem that can be easily solved with dynamic programming [11]. More recently, an extension of dynamic programming to the multi-dimensional case has been proposed [5] and can be easily applied for the decoding of Markov Random Fields. This work is the first application of this 2D Dynamic Programming (2DDP) algorithm to handwriting recognition.

Let us consider a partition of an image into two regions R_1 and R_2. Let ∂R_1 and ∂R_2 be the boundaries of these regions, that is the sites belonging to cliques that contain sites from two different regions (Figure 2).

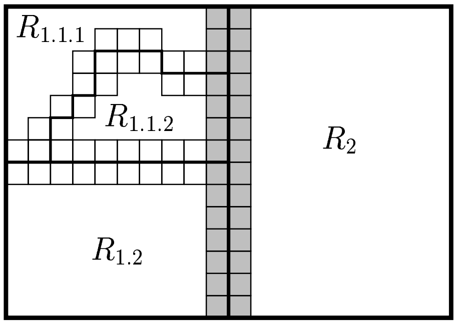

Figure 2. Partition of the image into two regions (R_1 and R_2) that can be divided into sub-regions. Only the sites belonging to boundaries are represented.

For a given configuration ω, let ω_1, ω_2, $\partial\omega_1$ and $\partial\omega_2$ be the restrictions of this configuration respectively to R_1, R_2, ∂R_1 and ∂R_2. The function to minimize U can be written with different terms for the two regions and an interaction term I associated to the sites of the boundary.

$$U(\omega) = U(\omega_1) + I(\partial\omega_1, \partial\omega_2) + U(\omega_2).$$

The notations $U(\omega_1)$ and $U(\omega_2)$ are simplified notations for $U_{R_1}(\omega_1)$ and $U_{R_2}(\omega_2)$ and correspond to the terms of $U(\omega)$ that depend on only one region. The term $I(\partial\omega_1, \partial\omega_2)$ is a simplified notation for $I_{\partial R_1, \partial R_2}(\partial\omega_1, \partial\omega_2)$ and corresponds to the remaining terms, associated to cliques that cross the boundary.

Let us consider two configurations ω and ω' that have the same configurations on the boundary (i.e.

$(\partial\omega_1, \partial\omega_2) = (\partial\omega_1', \partial\omega_2'))$. In this case, we have:

$$\left.\begin{array}{l} U(\omega_1) < U(\omega_1') \\ U(\omega_2) < U(\omega_2') \end{array}\right\} \Rightarrow U(\omega) < U(\omega').$$

Hence, for a given configuration of the boundaries $(\partial\omega_1, \partial\omega_2)$, it can be seen that:

$$\left.\begin{array}{l} \hat{\omega}_1 = \arg\min U(\omega_1) \\ \hat{\omega}_2 = \arg\min U(\omega_2) \end{array}\right\} \Rightarrow \hat{\omega}_1 \cup \hat{\omega}_2 = \arg\min U(\omega_1 \cup \omega_2),$$

that is,

$$\hat{\omega} = \hat{\omega}_1 \cup \hat{\omega}_2.$$

So that it is not necessary to compute the summations $U(\omega_1) + I(\partial\omega_1, \partial\omega_2) + U(\omega_2)$ for every ω_1 and ω_2 to find the optimal configuration. Only the optimal configurations $\hat{\omega}_1$ and $\hat{\omega}_2$ must be stored for every configuration of the boundaries $\partial\hat{\omega}_1$ and $\partial\hat{\omega}_2$.

Let $\partial\Omega_r$ $(r = 1, 2)$ be the set of possible configurations of the boundaries of regions R_r, and $\hat{\Omega}_r = \{\hat{\omega}_r / \partial\omega_r \in \partial\Omega_r\}$ the optimal configurations of the sites inside of the regions for each configuration of the boundary. The global optimum $\hat{\omega}$ is obtained by combining the configurations of $\hat{\Omega}_1$ and $\hat{\Omega}_2$ and selecting the minimum:

$$\hat{\omega} = \arg \min_{(\hat{\omega}_1, \hat{\omega}_2) \in \hat{\Omega}_1 \times \hat{\Omega}_2} U(\hat{\omega}_1) + I(\partial\omega_1, \partial\omega_2) + U(\hat{\omega}_2).$$

This process can be iterated: $\hat{\Omega}_1$ can be computed from $\hat{\Omega}_{1.1}$ and $\hat{\Omega}_{1.2}$ the same way. Only one part of the boundaries of $R_{1.1}$ and $R_{1.2}$ remains in the new boundary of the region R_1 (in grey on Figure 2).

At each step, for a region R_r, $\hat{\Omega}_r$ can be computed from the optimal configurations of two sub-regions $\hat{\Omega}_{r.1}$ and $\hat{\Omega}_{r.2}$, and so on and so forth until elementary regions of one site are reached. At this point elementary regions can be initialized as being in any of the N states.

From a set of elementary regions, regions are merged two by two by keeping only the best configuration of each configuration of the boundary until the whole image is in one region.

The order in which the regions are merged (called the merging policy) can be of any type. It will not influence the result but it can influence the computational cost. For a $m \times n$ image, considering every configuration would have a computational cost of $N^{m \times n}$. Using 2DDP, if regions are merged line by line, the cost would be $(m \times n) \times N^m$. In practice, this number is usually too high, but a pruning strategy can decrease this cost to a tractable one by removing less promising intermediate configurations of the regions.

2.3. Feature extraction and observation densities modeling

The values of the observation O are directly extracted from the original image. A great variety of feature extraction types have been proposed in the literature, that highly depend on the type of modelization used [13].

In the context of a HMRF modeling, 2D local features must be extracted. A windowed analysis of the image can extract observations that are represented as vectors. We use a 2D windowed spectral features extraction that is fully continuous and extracts information about the main directions in the image. It consist in computing a 2D Fourier transform in a window (regularized with a 2D Gaussian window) and extracting selected coefficients in module and phase. The first coefficients (i.e. located near the center of the resulting image), the low frequency coefficients keep information on strokes and directions. Figure 3 gives an illustration of this process.

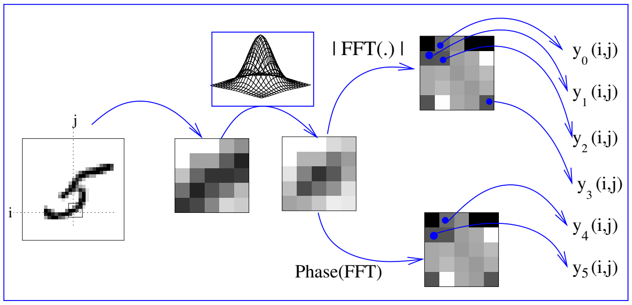

Figure 3. 2D spectral local features extraction.

For every state s, $P(o|s)$ is the observation density which is efficiently stored using mixtures of Gaussians. These mixtures can fit any real distribution. The EM algorithm is efficient to compute the parameters from a set of samples. There we have:

$$P(o|s) = \sum_{i=1}^{M} k_i G(o, \mu_{i,\omega}, \Sigma_{i,\omega}),$$

where $G(o, \mu, \Sigma)$ is the value in o of a Gaussian function of mean μ and covariance matrix Σ (in practice, a diagonal matrix), and where $\sum_{i=1}^{M} k_i = 1$. An example of a real distribution with the corresponding mixture of Gaussians can be seen on Figure 4.

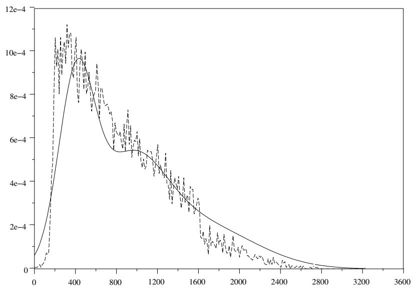

Figure 4. Real distribution with corresponding mixture of Gaussians.

3. Application to handwriting recognition

The simplest way to handle a short vocabulary task (such as digit recognition) is to perform a model discriminant approach of recognition:

$$C = \arg\max_{c_k} P(c_k \mid O) = \arg\max_{c_k} P(O \mid c_k)P(c_k).$$

C is the most probable class of the pattern among the c_k given that O is observed. If we have a set of models for these c_k, 2DDP can perform the computation of $P(O \mid c_k)$. The probabilities $P(c_k)$ is known from the statistics of the training set.

Hence, the remaining issues are the choice of the database, of the state space of the HMRF, of the merging policy of 2DDP and of a strategy for the training of the models (observations densities and cliques potentials).

3.1. Database

The MNIST database [10] is a widely used and publicly available database of handwritten digits. A few samples extracted from the database can be seen on Figure 5

There is a training set of $60,000$ samples and a testing set of $10,000$ samples. For the development and tuning of the algorithm, we divided the training set into a development set and validation set, so that we only performed few evaluations on the testing set. Performing more evaluations on the testing set would include knowledge from the testing set into the algorithms and give results which are not completely realistic. For class i, the validation set is the last n_i samples of the training set where n_i is the number of samples in the corresponding testing set.

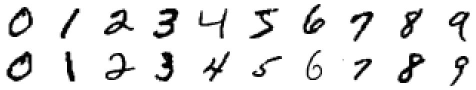

Figure 5. Samples from the MNIST database.

3.2. State space

To capture the shape of characters, models must keep information on the strokes and particularly their direction and relative position. To capture this information, states can be associated to homogeneous portions of strokes in the image. The features described in section 2.3 are efficient to extract the local features in terms of directions. Cliques potentials (cf. section 2.1) can keep the information about the relative position of these strokes.

Figure 6 illustrates the expected segmentation into states. Each of the 35 states is associated with an homogeneous portion of the image in terms of position and stokes directions. In our experiment, the 5×7 states models gave the best results as it could be expected regarding to the shape of digit 8.

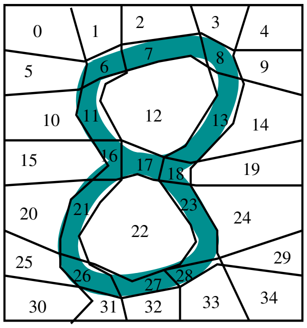

Figure 6. Expected segmentation of a sample image into 35 states.

3.3. Merging policy and pruning strategy

As explained in section 2.2, the merging policy should not have any influence on the results but on the computational cost. In practice, a real 2DDP decoding is not tractable, so that a pruning strategy must be performed. It consists in removing the less promising configurations that would probably give sub-optimal final configurations. This principle is known as being very efficient in speech processing with Markov chains and 1D dynamic programming [7].

In this case where configurations are removed, it is important to merge first the regions where the uncertainty is less important. Our merging policy merges the sites on the external boundary of the image first and then the ones closer to the center. An illustration of this merging policy is given by Figure 7.

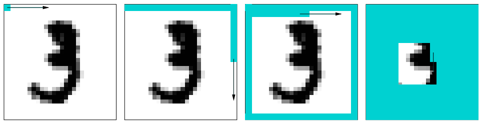

Figure 7. Merging policy: the external sites are merged first.

3.4. Features coefficients

Relevant coefficients are selected from the Fourier transform. Both phases and modules keep important information. Computing on vector for each pixel is not necessary, since the different windows are overlapping. We found that using 14×14 images of 10 dimensional vectors gives good results (cf. section 3.6). Figure 8 illustrates the first coefficients extracted from the Fourier transform, alternatively module and phase.

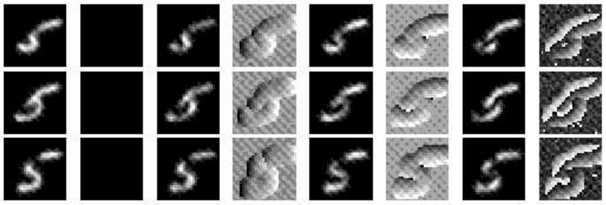

Figure 8. First coefficients of the feature extraction, alternatively module and phase.

3.5. Learning strategy

In order to perform the recognition of the digit samples, a set of models must be available. A digit model is composed a set of observation densities functions (one for each state) as well as a set of cliques potentials. The available ground truth for this database reduces to the class of the samples so that no information of segmentation of the training set is available.

A common and efficient way to come through this issue with 1D problems is to perform a Viterbi learning which is a simplified EM approach where only the optimal configuration is kept for computing the expectation [7]. A first model is computed by using a regular segmentation of the training samples into 35 states. These first segmentation allows the computation of initial models (observation densities and transition probabilities). This models are then used to process a 2DDP decoding and getting new segmentations which will give new model parameters. This process is then iterated until convergence. This learning strategy is illustrated Figure 9.

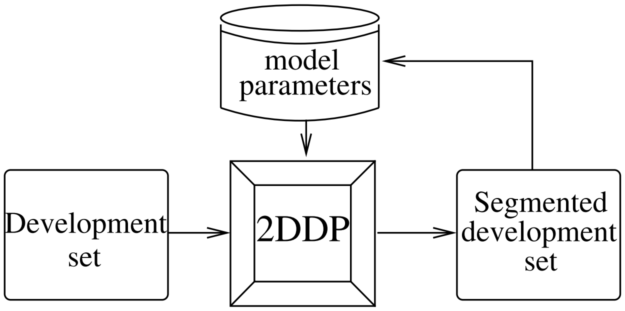

Figure 9. 2D Viterbi learning.

3.6. Results

Table 1 summarises the results on the validation set for different types of feature vectors as well as the final result on the testing set. These results are without a rejection process. The processing speed of our algorithm is about 3 samples per second on a single processor.

Table 1. Error rate for different types of feature vectors

Number of module and phase coefficients	(0,2)	(4,0)	(4,2)	(4,4)	(8,2)
Error rate on validation set	4.92 %	3.56 %	2.38 %	2.61 %	2.34 %
Error rate on testing set	-	-	-	-	**2.48 %**

4. Extension to handwritten words recognition

This proposed approach is very general and can be easily applied to a wide variety of recognition and segmentation tasks. In this section, we propose the extension to a handwritten words recognition. The database used for these first experiments on words is the *Senior&Robinson* database. It is a set of 25 handwritten pages written by one scriptor and segmented into words.

A simple way to extend our approach to word processing is to build word models by concatenating letter models. One model is built for each word of the vocabulary but this is different from a *holistic* approach since only letter models are trained.

The idea is to build the word models with a concatenation procedure where the transition probabilities are adjusted between the states on the right hand side of the first letter and the states on the left hand side of the second letter. This process can be iterated to build any word model.

Once a word image has been segmented into states, it is possible to cut the images into letters according to this segmentation, the set of letter images can then be used to train letter models as explained in section 3.

Our first experiments give interesting results in terms of segmentation of the words into letters, whereas the recognition rate must still increase. Figure 10 illustrates the segmentation part: the boundary between sates belonging to different letters is drawn, and gives the boundary between letters. It can be seen that this line is not a straight line as it would be obtained with a Markov chain modeling.

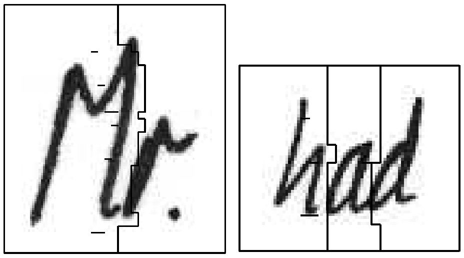

Figure 10. Segmentation of words into letters.

5. Conclusion and perspectives

We presented an approach of handwriting recognition based on Markov Random Fields models and 2D dynamic programming. It is a fully 2D model with an efficient feature extraction procedure and algorithms are available for training and recognition. It has been successfully applied to handwritten digits recognition and interesting results on word recognition are expected in the near future.

With such a generic method, many perspectives arise. To improve the processing speed, the dictionary can be organized in tree and word models can be computed on the fly. Contextual letter models can be computed if the database is large enough to improve the accuracy. A scriptor adaptation strategy can be used on the parameters of the HMRF. Finally, document processing can be performed in a similar way with two models of word and non-word and a language model (n-grams).

References

[1] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *J. Roy. Stat. Soc., Ser. B*, 36:192–236, 1974.

[2] J. Besag. Statistical analysis of dirty pictures. *Journal of the Royal Statitical Society*, 1986.

[3] J. Cai and Z. Liu. Pattern recognition using Markov random field models. *Pattern Recognition*, 35:725–733, 2001.

[4] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE transactions on Pattern Analysis and Machine Intelligen ce*, 6(6), 1984.

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

[6] M. Gilloux. Hidden Markov models in handwriting recognition. In Impedovo [8], pages 264–288.

[7] X. Huang, A. Acero, and H.-W. Hon. *Spoken language processing*. Prentice Hall, 2001.

[8] S. Impedovo, Editor. *Fundamentals in handwriting recognition*. NATO ASI Series. Springer-Verlag, January 1994.

[9] S. Knerr, V. Anisinov, O. Baret, N. Gorski, D. Price, and J.-C. Simon. The A2iA intercheque system: courtesy amount and legal amount recognition for French checks. *International journal of pattern recognition and artificial intelligence*, 11(4):505–548, June 1997.

[10] Y. LeCun. http://yann.lecun.com/exdb/mnist .

[11] E. Levin and R. Pieraccini. Dynamic planar warping for optical character recognition. In *IEEE International Conference on Accoustics, Speech and Signal Processing*, Mar. 1992.

[12] G. Saon and A. Belaïd. High performance unconstrained word recognition system combining HMMs and Markov random fields. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI) Special Issue on Automatic Bankcheck Processing*, 1997.

[13] O. Trier, A. Jain, and T. Taxt. Feature extraction methods for character recognition - a survey. *Pattern Recognition*, 29(4):641–662, 1996.

Acknowledgments

The authors would like to thanks Emmanuèle Grosicki and Mélanie Lemaître for their fruitful help.