A generic 2D approach of handwriting recognition

Sylvain Chevalier*, Edouard Geoffrois**, Françoise Prêteux* and Mélanie Lemaître*,**
*: ARTEMIS project Unit, GET/INT, Evry - France
**: Centre d'Expertise Parisien, DGA/DET/CEP, Arcueil - France
sylvain.chevalier@int-evry.fr, edouard.geoffrois@etca.fr, francoise.preteux@int-evry.fr,
melanie.lemaitre@etca.fr

Abstract

In this paper, we present a two-dimensional approach of the processing of handwriting. It combines a Markovian model, an efficient decoding algorithm, a windowed spectral features extraction scheme and a rigorous evaluation methodology. We applied this principle to a digit recognition task and to a word recognition task.

1. Introduction

The analysis of handwritten images has been performed with a wide variety of methodologies [9]. Handwritten words analysis can be efficiently processed with robust one-dimensional statistical methods based on Markov chains with good results on constrained tasks [10]. However, the 2D nature of the handwriting is obvious but no fully satisfying 2D approach has been found yet [7].

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 widely known techniques (except two-dimensional dynamic programming, explained in section 2.2) but the proposed combination is original. Section 2 gives the main theoretical background of our approach, section 3 describes how we applied these principles to a digit recognition task and section 4 briefly gives the outline of the extension to handwritten words recognition. Section 5 gives a conclusion of this work together with some of the many suggestions of improvements and extensions of the proposed approach.

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 is reduced to a local one:

$$P(\omega_{i,j} \mid \omega_{I\setminus(i,j)}) = P(\omega_{i,j} \mid \omega_{N(i,j)}),$$

where $N(i, j)$ is the set of sites which are neighbors of (i, j).

A convenient way to handle neighboring relations is to use cliques: a clique is a set of sites which are neighbors. Figure 1(a) represents a neighboring system with its associated cliques.

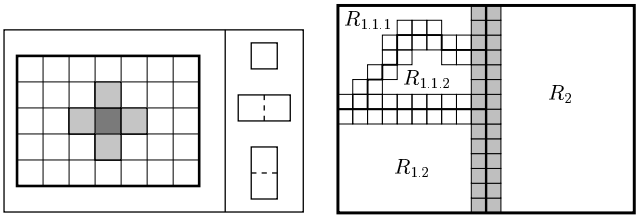

Figure 1. (a) First order neighboring system and associated cliques (b) Partition of the image into two regions

With this formalism it is possible to use the Gibbs distri-

butions which are equivalent to a Markov Random Field [1]:

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

where C is the set of cliques, V_c is a potential function associated to 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. The observed image is $O = o_{i,j}$. The observation of one site only depends on the underlying hidden state:

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

A bayesian approach is used to determine the optimal configuration which is :

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

This problem is equivalent to minimize the fonction defined by:

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

2.2. Decoding algorithm

Several methods have been proposed to perform this maximization of $P(O \mid \omega)P(w)$ 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 [12]. More recently, an extension of dynamic programming to the multidimensional case has been proposed [5, 6] 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 1(b)).

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 region R_r, and $\hat{\Omega}_r = \{\hat{\omega}_r / \partial\omega_r \in \partial\Omega_r\}$ the set of optimal configurations on the whole region for each possible configuration of its 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 1(b)).

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 is $(m \times n) \times N^m$. 2DDP thus

dramaticaly decreases the complexity of decoding, without any loss in optimality of the solution. In order to further decrease the computational cost to a tractable one, a pruning stategy can be used to remove the less promising intermediate configurations of the regions at each merging step of the algorithm. This principle is known to be very effective in speech processing with Markov chains and 1D dynamic programming [8].

2.3. Feature extraction

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 consists in computing a 2D Fourier transform in a window (regularized with a 2D Gaussian window) and extracting a certain number of coefficients of module and phase. The first coefficients (i.e. located near the center of the transformed image), the low frequency coefficients keep information on strokes and directions. Figure 2 gives an illustration of this process.

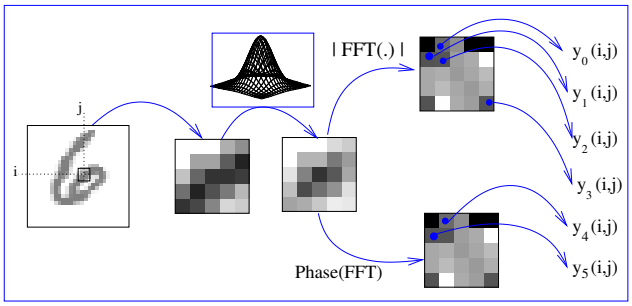

Figure 2. 2D spectral local features extraction

2.4. Observation densities modeling

For every state s, $P(o|s)$ is the observation density which is modeled using mixtures of Gaussian. These mixtures can fit any distribution with a given precision and there is an efficient algorithm EM to compute the parameters from a set of samples :

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

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 $\sum_{i=1}^{M} k_i = 1$.

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:

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

$\hat{c}$ 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 $P(O \mid c_k)P(c_k)$. 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 [11] is a widely used and publicly available database of handwritten digits.

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.

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 3 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 considering the shape of a relatively complex digit such as digit 8.

3.3. Merging policy and pruning strategy

As explained in section 2.2, the merging policy not only can influence the computational cost of decoding, but also

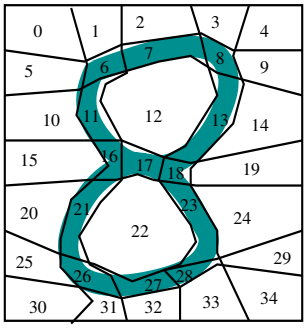

Figure 3. Expected segmentation of a sample image into 35 states.

interacts with the pruning strategy and should be carefully chosen to avoid removing promising hypotheses. As a general rule, it is more efficient to first merge 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 4.

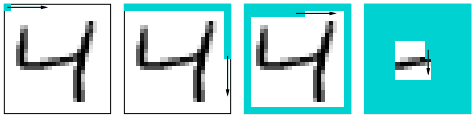

Figure 4. Merging policy: the external sites are merged first.

3.4. Features coefficients

Relevant coefficients are selected from the Fourier transform. Both phases and modules bear important information. Computing a 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 5 illustrates the first coefficients extracted from the Fourier transform, 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 of 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

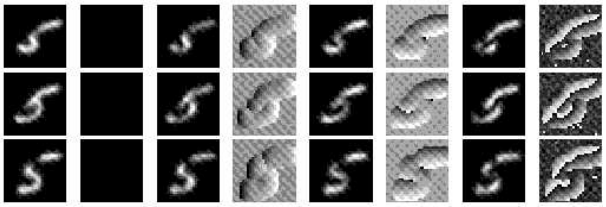

Figure 5. First coefficients of the feature extraction, alternatively module and phase.

a simplified EM approach where only the optimal configuration is kept for computing the expectation [8]. 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 6.

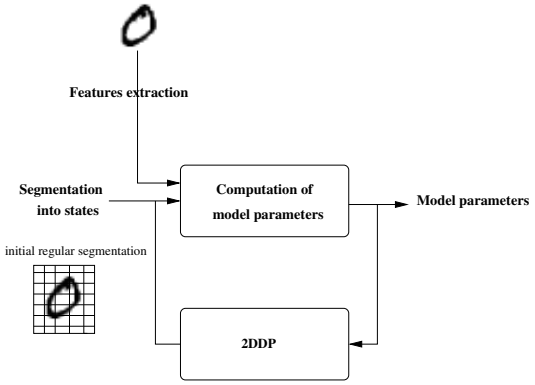

Figure 6. 2D Viterbi learning.

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 images per second on a single processor.

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

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 %	2.09 %
Error rate on testing set					**2.32 %**

experiments on words is the *Senior&Robinson* database. It is a set of 25 handwritten pages written by one scriptor and segmented into words.

4.1. Models construction

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 traditional 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.

4.2. Preliminary results

Our first experiments give interesting results in terms of segmentation of the words into letters, whereas the recognition rate must still increase. Figure 7 illustrates the segmentation part: the boundary between states 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.

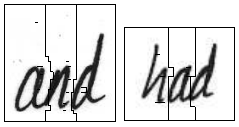

Figure 7. Segmentation of a word into letters.

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 can be applied to a word recognition task.

Many additional techniques can be explored. For example, the dictionary can be organized in tree and word models can be computed on the fly in order to improve the processing speed, contextual letter models can be computed if the database is large enough to improve the accuracy and finally a scriptor adaptation strategy can be used on the parameters of the HMRF.

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 Intelligence*, 6(6), 1984.

[5] 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.

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

[7] M. Gilloux. Hidden Markov models in handwriting recognition. In Impedovo [9], pages 264–288.

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

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

[10] 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.

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

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

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