A 2D Dynamic Programming Approach for Markov Random Field-based Handwritten Character Recognition

Sylvain Chevalier[1,2], Édouard Geoffrois[1], and Françoise Prêteux[2]

[1]*Centre technique d'Arcueil, FRANCE*
Edouard.Geoffrois@etca.fr
[2]*ARTEMIS Project Unit, Institut National des Télécommunications, FRANCE*
{ Sylvain.Chevalier , Francoise.Preteux }@int-evry.fr

Abstract

This paper presents the use of a new 2D dynamic programming approach for handwritten character recognition. The theoretical natural extension of the well-known 1D dynamic programming algorithm has been presented recently within an hidden Markov random field modeling framework. This principle has been adapted to a handwritten character recognition task and the performances are analyzed on the MNIST database for which spectral local features are extracted. Preliminary results exhibit an error rate similar to the ones reported in the literature.

1. Introduction

Dynamic Programming (DP) techniques have been extensively and successfully applied to solve a great variety of problems in one-dimension [20, 24]. In the field of speech recognition, most of the systems are based on hidden Markov models (HMM) and the Viterbi algorithm which is a direct application of DP [15, 24]. Several attempts were carried out to apply this principle to bi-dimensional tasks as those typically encountered in image processing. However, none of them can be considered as a true 2D approach [19, 23, 29]. Very recently, the direct extension of DP to the multi-dimensional case was presented [11, 10] and it is expected to be able to solve a large range of open issues in the field of image processing.

In this paper, the target application is handwritten character recognition. In this area, Markov chains are commonly used to solve cursive script recognition tasks [1, 5]. In addition, several attempts to use hidden Markov random fields for handwriting recognition have been made [12], specially for Chinese characters [27]. The models used are pseudo-2D Markov models or planar HMM, or causal MRF. Pseudo-2D Markov models are a combination of two Markov chains (one for each spatial direction) [12, 22]. In causal Markov random fields, the local dependency allows a 1D scanning of the image [21, 27]. A Markov random field model with a truly four-nearest-neighbor system was developed by Xiong and *al.* [28] and applied to handwritten Chinese character recognition with a traditional ICM algorithm.

The paper is organized as follows. The general principle of the multi-dimensional dynamic programming will be first explained as developed in [11, 10]. Then, we detail how to combine such an approach within a hidden Markov random field modeling framework. Our application of 2D DP to a handwriting recognition task recognition is then given together with the implementation issues and a description of the database we used. The results obtained and a conclusion follow.

2. Principle of the Multi-Dimensional Dynamic Programming

Dynamic programming is based on Bellman's optimality principle: if a path between A and B is optimal and if C belongs to this path, then paths between A and C and C and B are also both optimal.

Therefore, instead of exploring all possible paths (p^n paths of length n if there are p different values), for each C, the two sub-paths are explored (Figure 1). This process is iterated n times so that the exploring space is now in np^2.

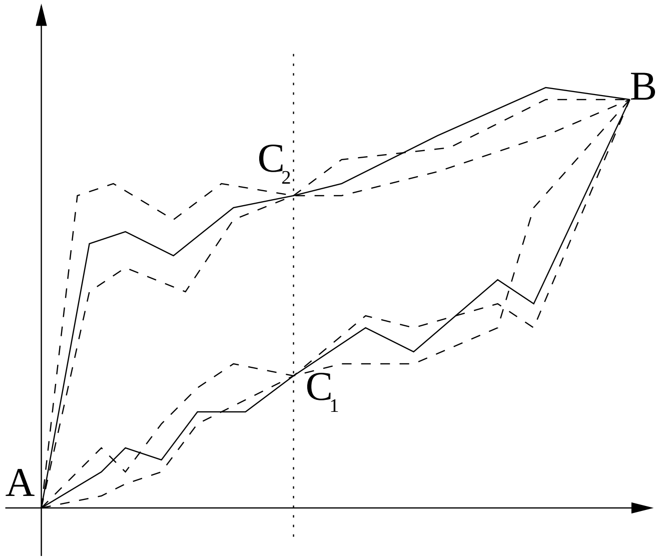


Figure 1: Dynamic Programming: optimal path and sub-paths

This principle has always been considered within a 1D framework. However, it was recently extended to the multi-dimensional case in a simple and canonic way [11, 10]. The generalized DP requires a property of local dependency given by a Markov random field model.

Markov fields are based on a graph structure which nodes are called sites and in practice usually correspond to the image pixels. Edges of the graph define the neighborhood system: two nodes $(s_1, s_2) \in S \times S$ are neighbors if they are directly linked together with an edge. In a Markovian context, the neighborhood system is usually described in terms of cliques: a clique is either a singleton or a subset of S in which every element is a neighbor of all the other elements [6]. Each site (i, j) is associated with a random variable $\omega_{(i,j)}$ with values in a discrete or a continuous space. If $V(s)$ is the set of sites which are neighbors of s, the local dependency assumption is expressed as:

$$P(\omega_s|\omega_{I\setminus(s)}) = P(\omega_s|\omega_{V(s)}). \tag{1}$$

If R_1 and R_2 make a partition of an image I, the common boundary between R_1 and R_2 can be defined as the sites that belong to a clique which contains elements from R_1 and R_2 (Figure 2). Consequently, the R_1-R_2 dependency holds only for the R_1-R_2 boundary pixels. To compute the optimal configuration, instead of exploring all the possible configurations (p^{n^2} configurations), only the optimal configuration of each region R_1 and R_2 has to be found for each configuration of the boundary. After n^2 iterations, the complexity of the algorithm decreases to $n.p^{2n}$ [10] with no loss of optimality.

3. Hidden Markov Random Fields and 2D Dynamic Programming

Markov Random Fields models are widely used in the field of image processing and computer vision mostly for applications like debluring, restoration, spatial, temporal and spatio-temporal segmentation or recognition [6, 4, 9]. The recognition procedures are achieved either by Iterated Conditional Modes (ICM) [3], which is sub-optimal, or by simulated annealing [9], whose convergence is very slow.

3.1. The general framework

The aim of the algorithm is to find the maximum *a posteriori* (MAP) of ω, *i.e.* the best configuration $\hat{\omega}$ that maximizes the posterior probability of $\omega = \hat{\omega}$ given the observation o:

$$\hat{\omega} = \arg\max_{\omega\in\Omega} P(\omega|o) = \arg\max_{\omega\in\Omega} P(o|\omega)P(\omega). \tag{2}$$

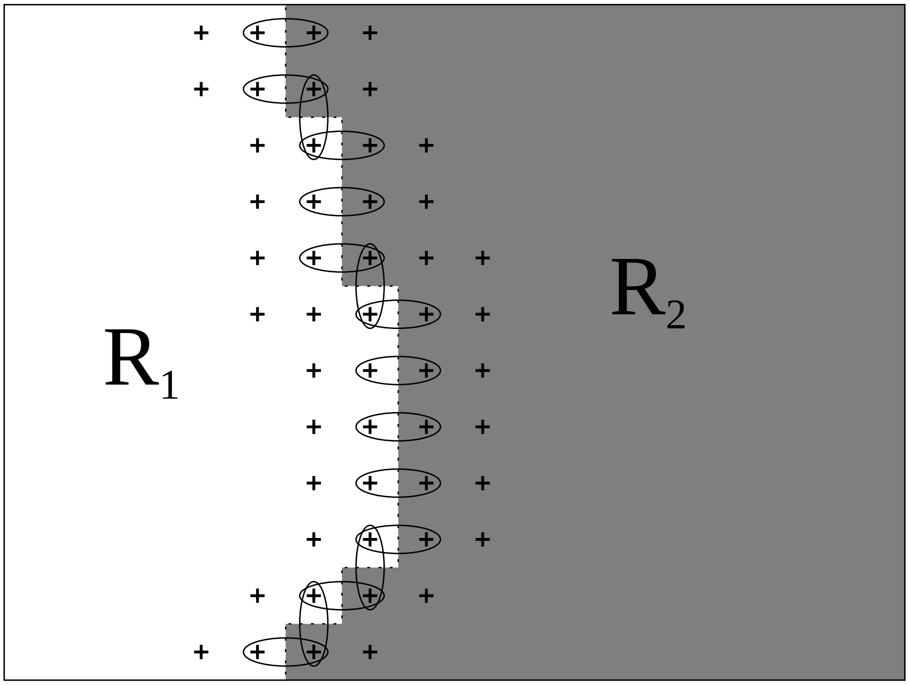


Figure 2: 2D Dynamic Programming: regions and the common boundary of R_1 and R_2 defined as pixels belonging to a clique of order 2 with one element in R_1 and one element in R_2

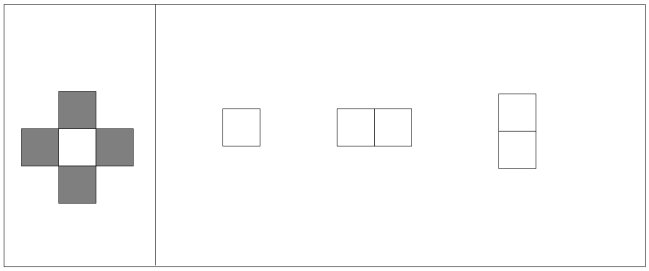


Figure 3: Clique types of the first order neighborhood

The term $P(o|\omega)$ in (2) is the marginal conditional distribution of o given the label ω (see in (4.3.2)) and $P(\omega)$ denotes the probability law for which transition probabilities will be computed.

In this paper, sites of the MRF are image pixels and cliques are associated with the four nearest neighbors as shown in Figure 3. The Markov property on the conditional probabilities of states reduces the global model to a local one:

$$P(\omega_{i,j}|\omega_{I\setminus(i,j)}) = P(\omega_{i,j}|\omega_{i-1,j}, \omega_{i+1,j}, \omega_{i,j-1}, \omega_{i,j+1}). \tag{3}$$

The Hammersley-Clifford theorem states that a MRF with the positivity property is equivalent to a Gibbs distribution in which the probability of one configuration is expressed as:

$$P(\omega) = \frac{1}{Z}\exp(-\sum_{c\in C} V_c(\omega)), \tag{4}$$

where C is the set of cliques and V is the potential function associated with this set. The normalization constant Z is unknown from a practical point of view.

In addition to the Markov property, the observed random variables given the labels are assumed to be conditionally independent:

$$P(O|\omega) = \prod_{(i,j)} P(o_{ij}|\omega_{ij}). \tag{5}$$

3.2. Transition probabilities

Let us define the interaction functions I_0, I_1, I_2 and I_3 as

$$\begin{aligned}
I_0(\omega_k, \omega_l) &= \frac{P\left(\begin{array}{|c|}\hline \omega_l \\ \hline \omega_k \\ \hline\end{array}\right)}{P(\omega_k)P(\omega_l)} \\
I_1(\omega_k, \omega_l) &= \frac{P\left(\begin{array}{|c|}\hline \omega_k \\ \hline \omega_l \\ \hline\end{array}\right)}{P(\omega_k)P(\omega_l)} \\
I_2(\omega_k, \omega_l) &= \frac{P\left(\begin{array}{|c|c|}\hline \omega_l & \omega_k \\ \hline\end{array}\right)}{P(\omega_k)P(\omega_l)} \\
I_3(\omega_k, \omega_l) &= \frac{P\left(\begin{array}{|c|c|}\hline \omega_k & \omega_l \\ \hline\end{array}\right)}{P(\omega_k)P(\omega_l)},
\end{aligned} \tag{6}$$

where

$$P\left(\begin{array}{|c|}\hline \omega_l \\ \hline \omega_k \\ \hline\end{array}\right) = P(\omega_{(i,j)} = \omega_l,\ \omega_{(i+1,j)} = \omega_k). \tag{7}$$

These interaction terms can be interpreted as a mutual information [7], which would be oriented. The set of cliques C is defined as:

$$C = C_0 \cup C_1 \cup C_2 \tag{8}$$

where,

$$C_0 = \{(i,j),\ 1 \leq i \leq M,\ 1 \leq j \leq N\}, \tag{9}$$

$$C_1 = \{((i,j),(i+1,j)),\ 1 \leq i \leq M-1,\ 1 \leq j \leq N\}, \tag{10}$$

$$C_2 = \{((i,j),(i,j+1)),\ 1 \leq i \leq M,\ 1 \leq j \leq N-1\}. \tag{11}$$

Then, potentials V_c can be defined as:

$$V_c(\omega) = \begin{cases} -log(P(\omega_k)) & \text{if } c \in C_0 \text{ with } c = (i,j) \text{ and } \omega_k = \omega_{i,j} \\ -log(I_0(\omega_k, \omega_l)) & \text{if } c \in C_1 \text{ with } \begin{cases} c = ((i,j),(i+1,j)) \\ \omega_l = \omega_{i,j} \text{ and} \\ \omega_k = \omega_{i+1,j}. \end{cases} \\ -log(I_2(\omega_k, \omega_l)) & \text{if } c \in C_2 \text{ with } \begin{cases} c = ((i,j),(i,j+1)) \\ \omega_l = \omega_{i,j} \text{ and} \\ \omega_k = \omega_{i,j+1}. \end{cases} \end{cases} \tag{12}$$

These potentials are stored in a $K \times K$ transition matrix M where K is the number of states of the model and the related coefficients are:

$$M(i,j)(k) = I_k(\omega_i, \omega_j),\ 1 \leq i \leq M,\ 1 \leq j \leq N,\ 0 \leq k \leq 3. \tag{13}$$

3.3. Potential function

By taking the logarithm of $P(o|\omega)P(\omega)$ and removing the normalization constant Z, the potential function $U(\omega)$ is introduced and expressed as:

$$U(\omega) = \sum_{(i,j)} -\log(P(o_{ij}|\omega_{ij})) + \sum_{c \in C} V_c(\omega), \tag{14}$$

and the problem is to minimize this potential:

$$\hat{\omega} = \arg\min_{\omega \in \Omega} U(\omega). \tag{15}$$

The key property of this potential is that it is the summation of local terms. If the image is divided into two regions R_1 and R_2 of configurations ω_1 and ω_2, the optimal configuration can be written as:

$$\hat{\omega} = \arg \min_{(\omega_1,\omega_2) \in \Omega_1 \times \Omega_2} U(\hat{\omega_1}) I(\omega_1, \omega_2) U(\hat{\omega_2}), \tag{16}$$

where $U(\hat{\omega_1})$ and $U(\hat{\omega_2})$ are computed for the pixels which do not belong to the boundary and $I(\omega_1, \omega_2)$ is computed only on the boundary. This is an iterative process so that it can be started with regions of one pixel, the regions being successively merged until to recover the whole image with a single region. At each step, only the best configuration for each possible configuration of the boundary is kept so that the optimal configuration is guaranteed to be reached.

3.4. Merging procedure

The principle of the 2D DP algorithm as explained in section 2 can be summarized here for any type of merging case. Every region has one list of possible configurations and each of these configurations has one likelihood value. In practice, pruning limits the number of possible configurations.

- Merging two pixels: each pixel is attached to a list of possible configurations associated to a corresponding likelihood. The resulting region is attached to a new list of possible configurations associated to a corresponding likelihood which is the sum of the likelihood of the pixels and of the interaction term I (Figure 4).
- Merging two regions: each region R_1 and R_2 is attached to a list of possible configurations associated to a corresponding likelihood. The resulting region $R_1 \cup R_2$ is attached to a new list of possible configurations associated to a corresponding likelihood which is the sum of the likelihood of the regions R_1 and R_2 and of the interaction terms I (Figure 5). These interaction terms are computed on the common boundary as expressed in section 2 and each configuration of the boundary is associated to a unique configuration of the interior pixels.

2D DP is very general and can be applied to any task that can be handled with a MRF such as image segmentation or restoration. It is now applied to handwriting recognition.

4. Application to handwriting recognition

In this section, we describe the database used, the general approach to handle with model learning and recognition, the tuning of the parameters and finally, results of the evaluation are given.

4.1. Database

As a first test of this system, it has been applied on the MNIST [1] standard database composed with 70.000 images of digits from the NIST database (Figure 6) and divided into one train set and one test set. Specificities of this database are as follows:

- all images have the same size (28 × 28),
- samples are centered and a white boundary is kept around the characters,

[1] http://yann.lecun.com/exdb/mnist

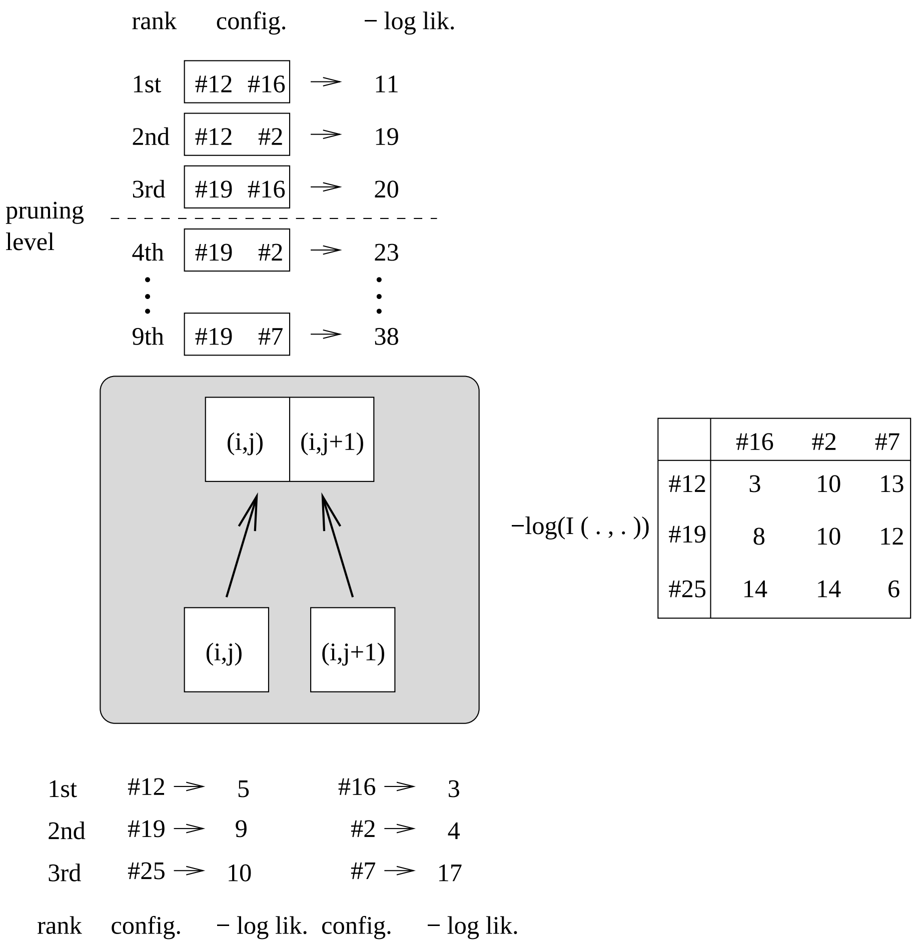


Figure 4: Merging two pixels: the resulting region is attached to a new list of possible configurations associated to a corresponding likelihood which is the sum of the likelihood of the pixels and of the interaction term I.

- gray levels result from a normalization pre-processing,
- test set and train set have been written by different writers.

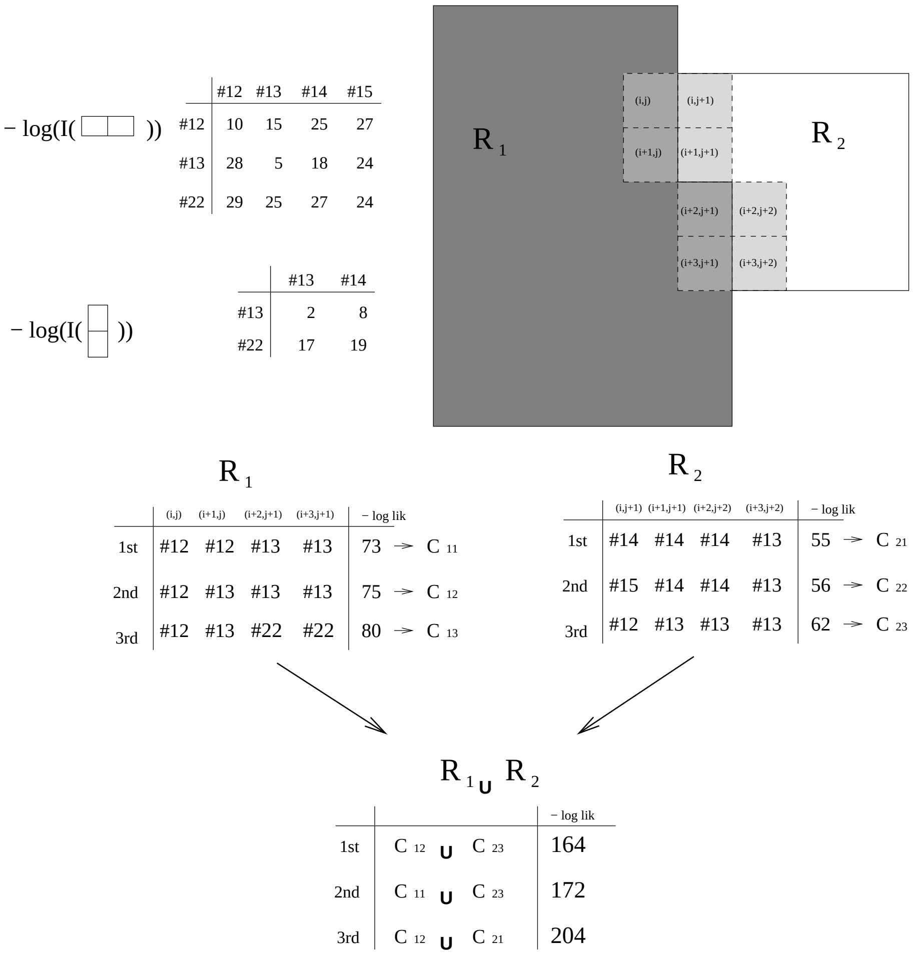
4.2. General approach

The aim of the algorithm is to segment one sample into regions related to the stroke structure of the character (extracted with convenient features computed on the samples) and to compute the likelihood of that configuration which is supposed to be the maximal one. To take into account the whole stroke structure of a letter, a 35-states-model is computed for each class and it is expected to assign a configuration to the pixels which keeps a mesh structure (Figure 10). A rectangular (5×7) mesh is used in order to be able to take into account all possible stroke structures, these values have been found to be the optimal ones on our development set of samples extracted from the train set of the MNIST database.

The use of the 2D DP algorithm to a recognition task is required at two levels: a model learning one and a recognition one. Each class of character (at least 10 classes for a digit recognition task) is associated to one hidden Markov random field model computed with a learning procedure based on an EM approach and the recognition procedure utilizes these models to compute scores of samples on models. The model that gives the highest score is then selected.

The structure of the learning and recognition algorithm are summarized:

- Learning Initialization: For each class, images of the train set are divided into a regular lattice of 5×7 regions. Each



− log(I(□□))

	#12	#13	#14	#15
#12	10	15	25	27
#13	28	5	18	24
#22	29	25	27	24

− log(I(⊟))

	#13	#14
#13	2	8
#22	17	19

R 1

	(i,j)	(i+1,j)	(i+2,j+1)	(i+3,j+1)	− log lik
1st	#12	#12	#13	#13	73 → C 11
2nd	#12	#13	#13	#13	75 → C 12
3rd	#12	#13	#22	#22	80 → C 13

R 2

	(i,j+1)	(i+1,j+1)	(i+2,j+2)	(i+3,j+2)	− log lik
1st	#14	#14	#14	#13	55 → C 21
2nd	#15	#14	#14	#13	56 → C 22
3rd	#12	#13	#13	#13	62 → C 23

R 1 ∪ R 2

		− log lik
1st	C 12 ∪ C 23	164
2nd	C 11 ∪ C 23	172
3rd	C 12 ∪ C 21	204

Figure 5: Merging two regions: the resulting region $R_1 \cup R_2$ is attached to a new list of possible configurations associated to a corresponding likelihood which is the sum of the likelihood of the regions R_1 and R_2 and of the interaction terms.

region is associated with one state and observation densities are derived with these pixels. The initial transition matrix defined in Equation 13 is also computed by counting the transitions observed on those initial segmentations and deriving the interaction terms as defined in Equation 6 (Figure 7).

- Learning Recursion: Each image of the train set is segmented with the 2D DP algorithm in order to maximize the *a priori* probability. New parameters are then computed and the process is iterated until convergence of the likelihood (Figure 8). For each iteration, the mesh structure is preserved because unlikely transition almost never occur so that they remain unlikely.

- Recognition: The best configuration of an image is computed with each model with the 2D DP algorithm. The model that gives the highest likelihood is then selected.

4.3. Algorithm tuning on development set

An important aspect of our experiment is that all optimization processes have been carried out only on a part of the train set, and just one experiment was made on the test set. Consequently, optimization has not been carried out on test samples and the result in terms of recognition rate should be valid for any real world data.

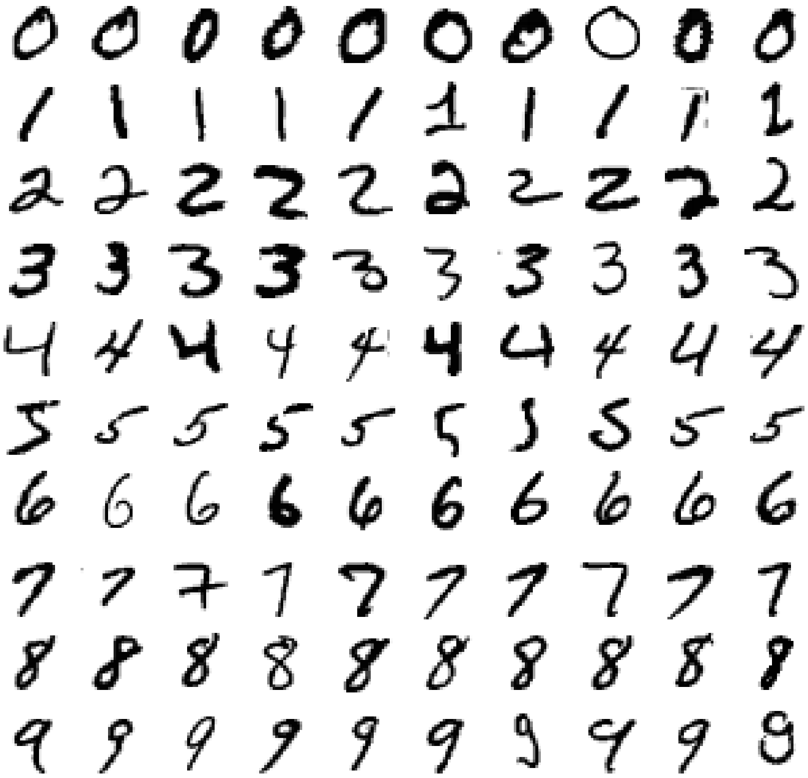


Figure 6: Samples from MNIST database

4.3.1 Feature extraction

The choice of the feature type used in handwriting recognition highly depends on the recognition procedure used by the system. A few studies have been made on the discriminant power of features [2, 25] and on the choice of the feature set [13]. Features can be divided into two classes: global and local features. Global features have been extensively described in a survey paper [26] and are mainly focused on finding moments and invariants in the whole image. Local features are extracted in windows around the pixels of the image. For isolated characters, cellular features are popular [14] to extract the stroke structure. Markov chains based algorithms usually extract features in graphemes or pseudo-letters like loops, ascenders and descenders and stroke junctions [8, 17].

The features we use are local spectral features. A Fast Fourier Transform is computed in a Gaussian window around each pixel. The logarithm of the magnitude of the transform is computed and the first five coefficients are kept (Figure 9). The first coefficient keeps the information on the energy of the window and the other four give the intensity in the four main directions (Table 1). These coefficients are related to the Gabor features based on Gabor filters [16, 18].

4.3.2 Observation densities modeling

The observation vectors, computed as explained in 4.3.1 are modeled with a multi-Gaussian function.

$$P(o \mid \omega) = \sum_{k=1}^{M} c_k G(o, \mu_{k,\omega}, \Sigma_{k,\omega}), \tag{17}$$

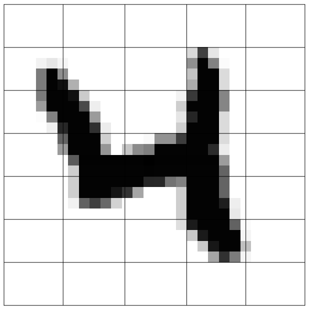


Figure 7: Initialization: initial segmentation used for the computation of the initial parameters of the states.

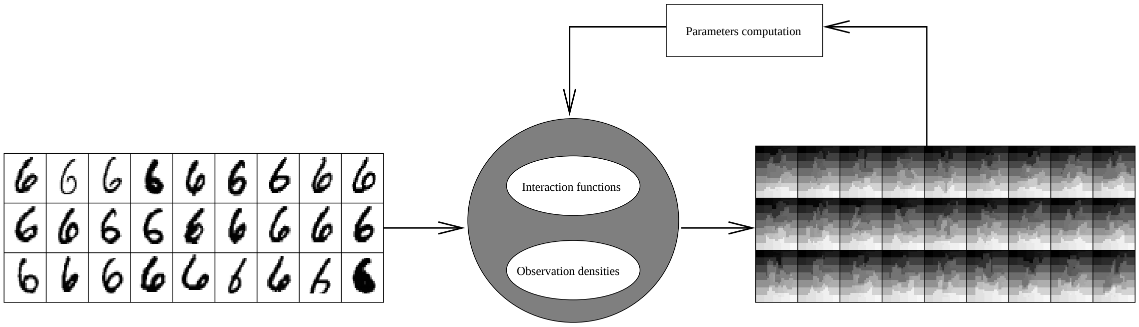


Figure 8: Model learning: at each iteration, the model parameters are computed with the results of the previous iteration before computing the 2D DP.

where $G(x, \mu, \Sigma)$ is the value in x of a Gaussian function of mean μ and covariance matrix Σ (often a diagonal matrix) and where

$$\sum_{k=1}^{M} c_k = 1. \tag{18}$$

There, the c_k, $\mu_{k,\omega}$ and $\Sigma_{k,\omega}$ are computed by performing an EM algorithm and iteratively adding Gaussian functions until the confidence reaches a given threshold. This process is commonly used in speech recognition [15]. In practice, the maximum number of Gaussian functions is set to $M = 8$ and the threshold in terms of likelihood is set to 10^{-8}. These values usually give a good matching of the distribution with the real variation (Figure 11).

4.3.3 Pruning strategy

For each pixel, the 2D DP algorithm first assigns a likelihood value for each possible configuration of that pixel by using the observation probability density before merging the pixels by adding these observation log-likelihoods with the interaction terms as expressed in 14. The DP principle decreases the size of the searching space. However, it is still much too large to be computationally tractable. Thus, a pruning strategy is introduced to keep, at each step of the computation, only the most

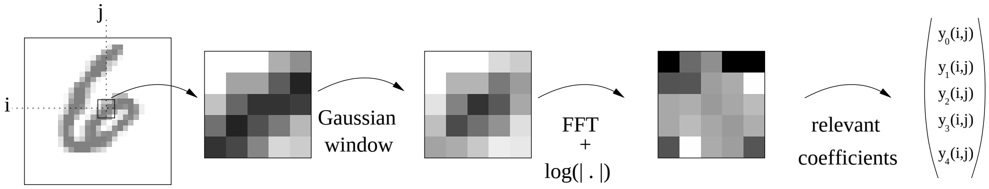
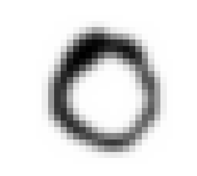
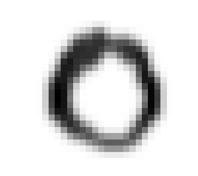
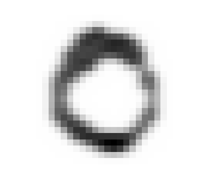
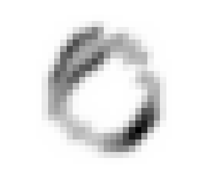
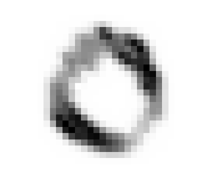
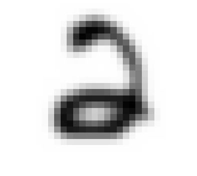
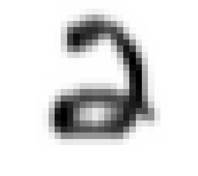
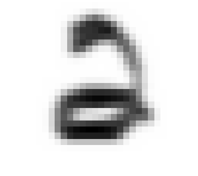
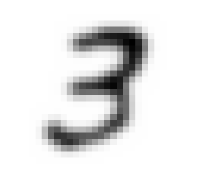
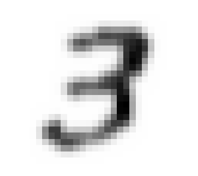
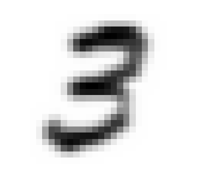
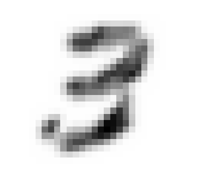
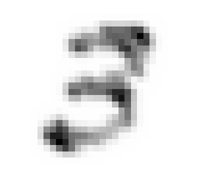
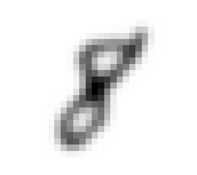
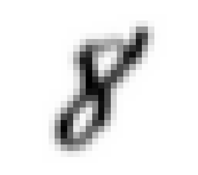
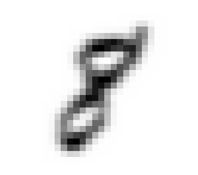
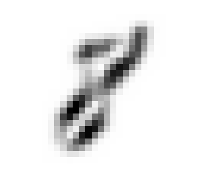
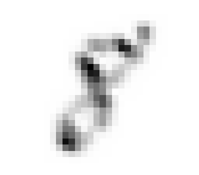


Figure 9: Feature extraction process

Table 1: The local spectral features: 5 components for each pixel

Energy	Vertical	Horizontal	First diagonal	Second diagonal
				
				
				
				

probable configurations. This pruning strategy occurs each time two regions are merged and in two ways:

- the configurations for which the log-likelihood difference before and after the pixels are merged is over a threshold, are cut,
- then, only a fixed number of different configurations for the boundary are kept.

This strategy is expected to prune only the configurations that would lead to non-optimal results because they are already far from the most promising ones. In practice, the threshold chosen in terms of log-probability is set to 5000 and usually cuts the configurations that would not preserve the mesh structure. The maximum number of configurations of the boundary kept for each region is set to 30 which results of a compromise between the quality of the segmentation and the computing time. In upcoming experiments, we are planing to focus on the influence of these parameters.

4.3.4 Merging policy

In order to compute the optimal configuration, a merging policy was chosen. To keep the mesh structure of the image configuration, the pixels of the boundary of the whole image are assigned to the relevant state in a regular lattice and they are computed first. Then pixels nearer to the center of the image are merged until only one region remains for the whole image. Table 2 shows the most promising configuration at several steps of the merging process. Each grey level correspond to one of the 35 states.

4.4. Results on test set

These very first experiments were carried out with a fixed number of iterations for the model computation. The convergence of the models can be observed on the curves of the likelihood of the training samples on the Figure 12. Our actual results

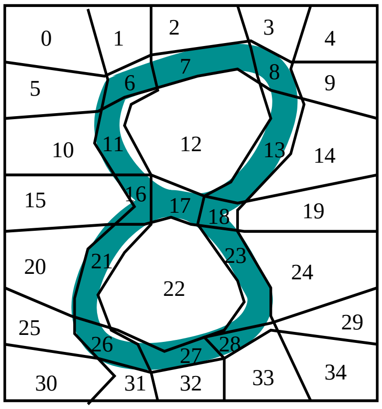


Figure 10: Expected segmentation of one sample into states: the 5×7 mesh is preserved.

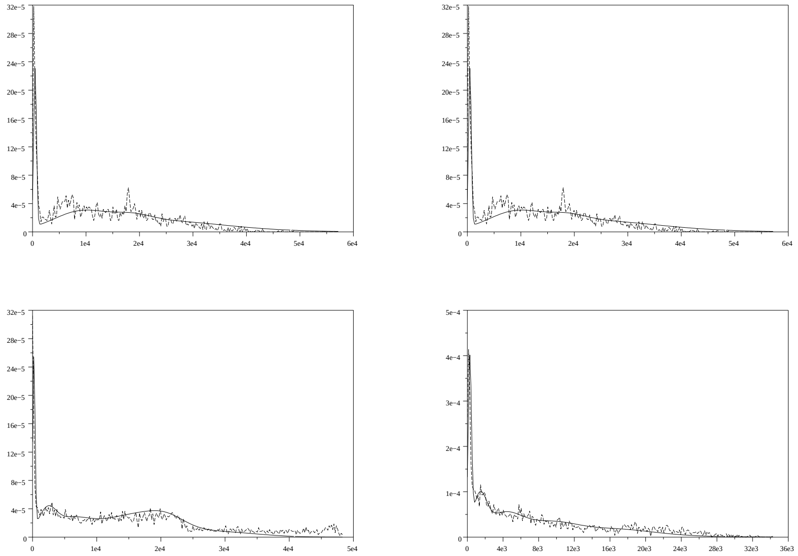


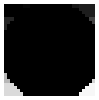
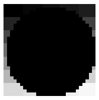
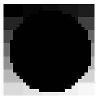
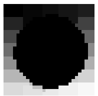
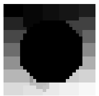
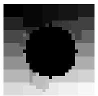
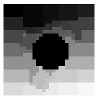
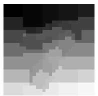
Figure 11: Histograms of training data and computed multi-Gaussian distribution.

show an error rate of 5.4 % on the whole test set of the MNIST database which is already close to the results reported in the literature. We are expecting to improve this result with a more sophisticated feature extraction process and with further studies on the pruning parameters.

5. Conclusion

A hidden Markov random field-based system for handwriting recognition has been presented. It is based on the 2D DP which has been recently developed to compute the optimal configuration of a MRF in a reasonable computing time. The process used to recognize digits with a MRF modeling has been explained and implementation issues and parameter selection have been discussed. Performances obtained are already close to the standard-of-the-art one, but this technique is expected to be really powerful for more complicated tasks such as cursive words recognition. 2D DP principle is expected to suit to each stage of a document recognition task from the analysis of the page structure to the character level.

Table 2: Merging policy: the most probable configuration of the main region is represented. Pixels that have net been merged with the main region yet are left in black.

Number of regions	700	600	500	400	300	200	100	1
								

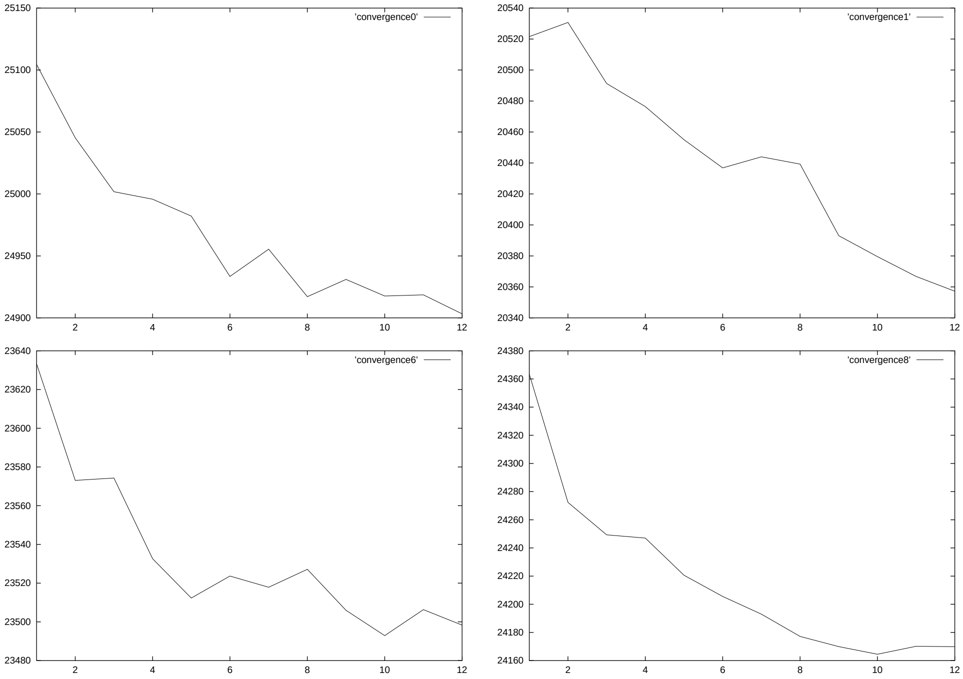


Figure 12: Convergence of the likelihood of four models: the average cost of the training data ($-log(P)$) is plotted against the number of iterations in the model computation.

References

[1] Nafiz Arica and Fatos T. Yarman-Vural. An overview of character recognition focused on off-line handwriting. *IEEE transactions on systems, man and cybernetics - part C : Applications and reviews*, 31(2):216–233, may 2001.

[2] Olivier Baret. *Régularités, singularités de représentations et leur complémentarité : Application à la reconnaissance de l'écriture manuscrite non contrainte*. PhD thesis, Université Paris 6, 1990.

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

[4] Jinhai Cai and Zhiqiang Liu. Pattern recognition using Markov random field models. *Pattern Recognition*, 35:725–733, 2001.

[5] Richard G. Casey and Eric Lecolinet. A survey of methods and strategies in character segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 18(7):690–706, july 1996.

[6] Rama Chellappa and Anil Jain, editors. *Markov Random Fields - Theory and application*. Academic Press, 1993.

[7] Richard O. Duda, Petre E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience, 2001.

[8] N. Feray and D. de Brucq. Système de reconnaissance de chiffres manuscrits hors-lignes. *Traitement du signal*, 1996.

[9] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions and the Bayesian restoration of images. *IEEE transactions on Pattern Analysis and Machine Intelligence*, (6), 1984.

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

[11] Edouard Geoffrois, Arnaud Jullian, and Christophe Debaert. Programmation dynamique 2D pour la reconnaissance d'images par modèles de Markov cachés. Technical report, DGA/DCE/CTA/GIP, 1998.

[12] Michel Gilloux. Hidden Markov models in handwriting recognition. NATO ASI Series, pages 264–288. Springer-Verlag, january 1994.

[13] F. Grandidier, R. Sabourin, C.Y. Suen, and M. Gilloux. Une nouvelle stratégie pour l'amélioration des jeux de primitives d'un système de reconnaissance de l'écriture. In *Actes du deuxième Colloque International Francophone sur l'Ecrit et le Document*, july 2000.

[14] T.H. Hildebrandt and W. Liu. Optical recognition of handwritten Chinese characters: advances since 1980. *Pattern Recognition*, 26(2), 1993.

[15] Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon. *Spoken language processing*. Prentice Hall, 2001.

[16] A. Jain and S. Bhattacharjee. Address block location on envelopes using Gabor filters. *Pattern Recognition*, 25(12), 1992.

[17] S. Kuo and O.E. Agazzi. Keyword spotting in poorly printed documents using pseudo 2-D hidden Markov models. *IEEE transactions on PAMI*, 16(8):842–848, 1994.

[18] Jouko Lampinen, Toni Tamminen, Timo Kostiainen, and Ilkka Kalliomäki. Bayesian object matching based on MCMC sampling and Gabor filters. In *Proc. SPIE Intelligent Robots and Computer Vision XX: Algorithms, Techniques and Active Vision*, volume 4572, pages 41–50, 2001.

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

[20] Magdi Mohamed and Paul Gader. Handwritten word recognition using segmentation-free hidden Markov model and segmentation based dynamic programming techniques. *IEEE transactions on pattern analysis and machine intelligence*, 18(5), may 1996.

[21] Hee-Seon Park and Seong-Whan Lee. A truly 2-D hidden Markov model for off-line handwritten character recognition. *Pattern Recognition*, 31(12):1949–1864, 1998.

[22] Lucio F. C. Pessoa. Multilayer perceptrons versus hidden Markov models: comparison and applications to image analysis and visual pattern recognition. Technical report, Georgia Institute of Technology, 1995.

[23] Georges M. Quénot. The "orthogonal algorithm" for optical flow detection using dynamic programming. In *IEEE International Conference on Accoustics, Speech and Signal Processing*, march 1992.

[24] Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of speech recognition*. Prentice Hall PTR, 1993.

[25] Jean-Claude Simon and Olivier Baret. Regularities and singularities in line images. In *Pre-proceedings SSPR90*, pages 423–439, 1990.

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

[27] Qing Wang, Zheru Chi, David D. Feng, and Rongchun Zhao. Hidden Markov random field based approach for off-line handwritten Chinese character recognition. In *Proceedings of the international conference on pattern recognition*, 2000.

[28] Y. Xiong, Q. Huo, and C. Chan. A discrete contextual stochastic model for the offline recognition of handwritten Chinese characters. *IEEE transactions on pattern analysis and machine intelligence*, 23(7), July 2001.

[29] Hiromitsu Yamada and Kazuhiko Yamamoto. Recognition of echocardiograms by a dynamic programming method. *Pattern Recognition*, 24(2), 1991.