From Off-line to On-line Handwriting Recognition

Lallican, P.; Viard-Gaudin, C.; Knerr, S.

FROM OFF-LINE TO ON-LINE HANDWRITING RECOGNITION

P.M. LALLICAN[1], C. VIARD-GAUDIN[2], S. KNERR[1]

[1] Vision Objects, 11 Rue de la Fontaine Caron, 44300 Nantes, France

[2] Ecole Polytechnique de l'Université de Nantes, IRCCyN / UMR CNRS 6597, Rue C. Pauc, BP 60601, 44306 Nantes Cedex 3, France

E-mail : {pmlallican, stefan.knerr}@visionobjects.com, cviard@ireste.fr

On-line handwriting includes more information on the time order of the writing signal and on the dynamics of the writing process than off-line handwriting. Therefore, on-line recognition systems achieve higher recognition rates. This can be concluded from results reported in the literature, and has been demonstrated empirically as part of this work.
We propose a new approach for recovering the time order of the off-line writing signal. Starting from an over-segmentation of the off-line handwriting into regular and singular parts, the time ordering of these parts and recognition of the word are performed simultaneously. This approach, termed " OrdRec", is based on a graph description of the handwriting signal and a recognition process using Hidden Markov Models (HMM). A complete omni-scriptor isolated word recognition system has been developed. Using a dynamic lexicon and models for upper and lower case characters, our system can process binary and gray value word images of any writing style (script, cursive, or mixed).
Using a dual handwriting data base which features both the on-line and the off-line signal for each of the 30 000 words written by about 700 scriptors, we have shown experimentally that such an off-line recognition system, using the recovered time order information, can achieve recognition performances close to those of an on-line recognition system.

1 Introduction

The starting point of our work is the observation that the on-line handwriting signal contains more information on the writing process than the off-line signal, especially regarding the temporal order and the dynamic information of the writing process. Consequently, on-line handwriting recognition systems can obtain superior recognition rates [17, 19].

Motivated by this observation, several authors have attempted to automatically reconstruct the temporal order of off-line signals [2, 3, 5, 7, 8, 9, 11, 13] . Most of these approaches are based on a number of heuristics and a local analysis of the handwriting signal. The heuristics assume a general left-right and top-bottom preference for the direction of the writing process and minimal curvature at crossings of the handwriting signal. In Figure 2 for instance, latter heuristic suggests following segment 3 (and not segments 2 or 6) when arriving at the first crossing starting from segment 7. The main drawback of these approaches is that the decisions are made locally, i.e. without the word context.

In this paper, we propose a methodology, termed " OrdRec", for the reconstruction of the temporal order of the off-line handwriting signal which is based on the simultaneous time ordering and recognition of the signal at the word

In: L.R.B. Schomaker and L.G. Vuurpijl (Eds.), Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition, September 11-13 2000, Amsterdam, ISBN 90-76942-01-3, Nijmegen: International Unipen Foundation, pp 303-312

level. " OrdRec" uses (i) a graph based optimization process which generates candidates for the time ordering and (ii) Hidden Markov Models for the word recognition. Thereby, the decisions as to the time ordering of the writing signal are made globally within the word context instead of locally. As compared to other approaches [8], our graph structure has been greatly refined in order to model the handwriting signal more accurately: the topology accounts for multiple drawings of a part of the writing signal, as well as for pen-up and pen-down signals.

We show experimentally that an off-line recognition system using the recovered temporal stroke order can achieve recognition rates close to those of an on-line recognition system. For that purpose, we have collected a dual handwriting database which features both the on-line and the off-line signal for each of the 30 000 words and 25 000 characters written by about 700 scriptors. This dual database, termed IRONOFF, has been presented in more detail in [18]. We also show that the "OrdRec" approach often succeeds at recovering the true time order of the handwriting signals, even in cases where purely local analysis does not work.

2 Overview of the recognition system

Figure 1 gives an overview of the complete off-line recognition system.

The word image is first segmented into regular parts (segments) and singular parts (crossings, strong curvature points, etc.). Most approaches in the literature start by binarizing the word image and computing a skeleton of the handwriting signal [2, 7, 8]. Both preprocessing steps often result in suboptimal intermediate data representations: while binarization often leads to broken or touching letters and filled loops, skeletonization usually generates artifacts which disturb further processing. The approach we have implemented is partly based on work presented in [12], and has been presented in an earlier paper [10]. The basic idea is that the singularities can be obtained as the complement to the regular parts of the handwriting which are easier to extract [5]. First the contours of all connected components are detected starting from gray scale or black and white images. Next, the cross sections of the handwriting are detected for the regular parts by starting from cross section seeds in straight regions of the handwriting which are extended until the stroke following algorithm meets a singularity. The result of this segmentation process is a set of regular parts (segments) and singular regions which can be represented by a graph (Figure 2). Each segment is defined by two nodes in this graph; one for each extremity of the segment. Each arc in the graph either links two segments which have a common singularity (inter-link) or links the two extremities of the same segment (intra-link). Local costs express the likelihoods of inter-links and intra-links. A path through the graph corresponds to a possible pen trajectory producing the given word image.

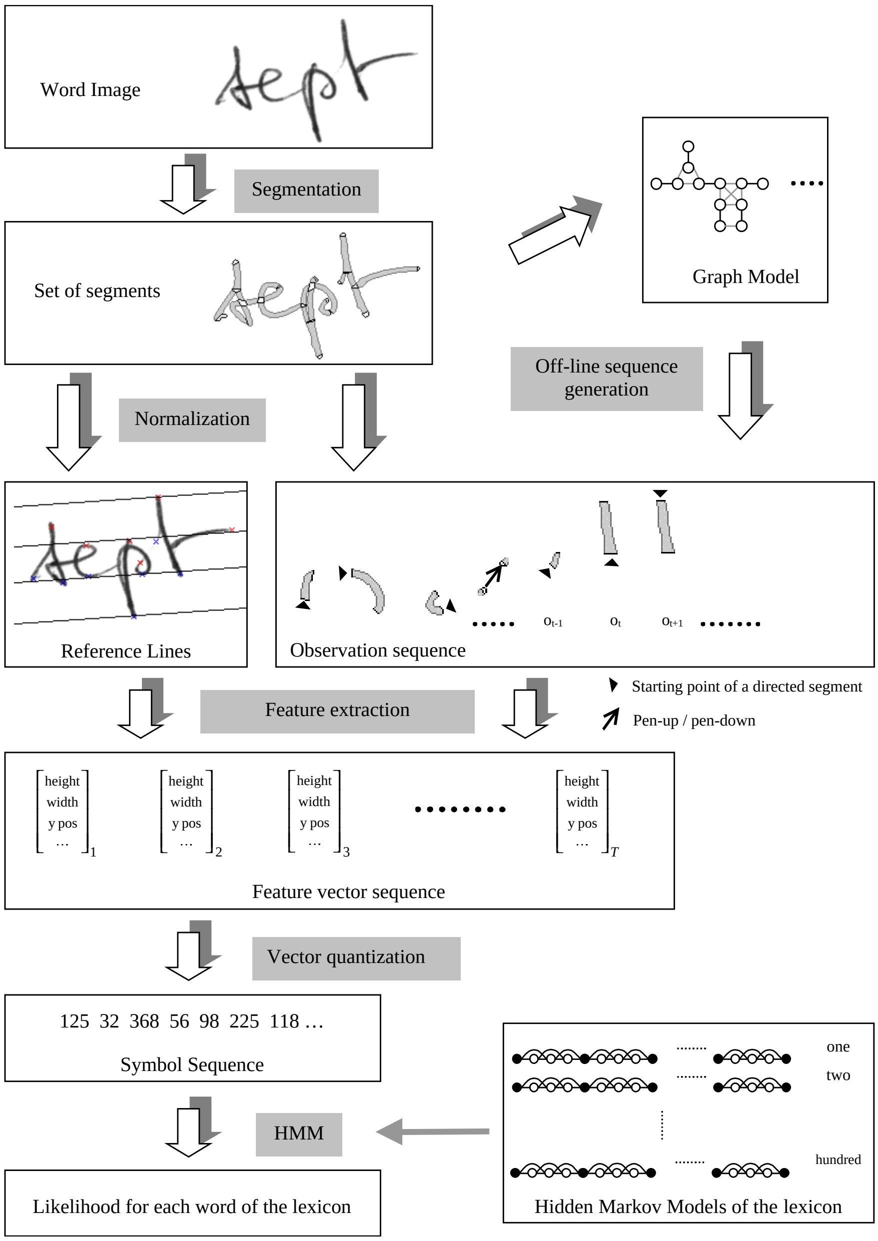


Figure 1: Overview of the off-line handwriting recognition system.

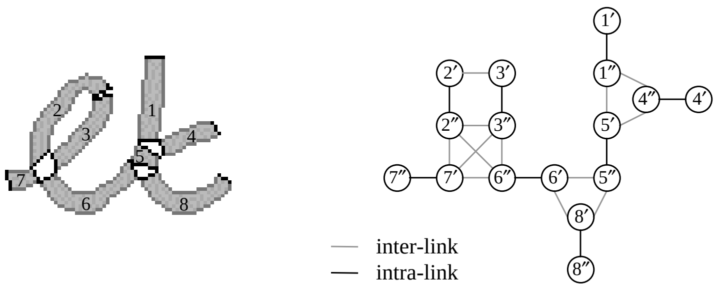


Figure 2: Segmented word image with regular and singular parts and the corresponding graph.

Note that the graph also needs to model segments which have been drawn in both directions and the invisible pen-up moves (between a pen-up and a pen-down signal). The goal of the " OrdRec" approach is to find the most likely path which ideally corresponds to the true pen trajectory.

Next, each segment of a given candidate sequence is described by a feature vector (position, length, orientation, etc.). In order to make the recognition process invariant with respect to size and skew, we normalize the word image using an algorithm reminiscent of the one used in [1]. Four straight parallel reference lines are computed which define the skew of the word as well as its size, typically determined by the height of the core zone. The EM-like optimization algorithm proceeds by matching the minima and maxima of the word contour onto the four reference lines which are defined by priors and probability distributions regarding their position and their skew [1, 4]. Each candidate sequence of feature vectors is then transformed into a sequence of symbols by means of a vector quantization.

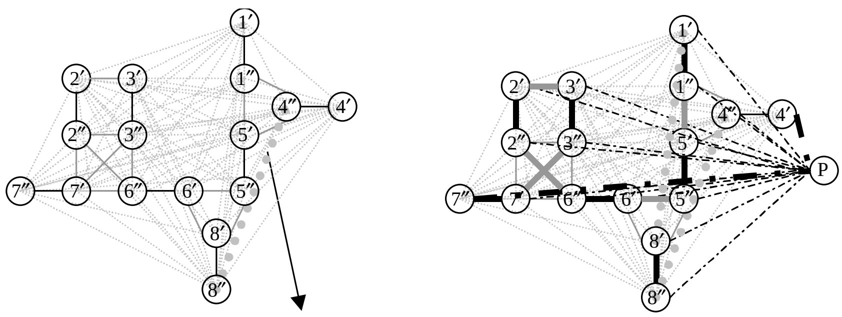
Finally, the likelihood for each word of the lexicon is computed using discrete word HMMs which can be dynamically concatenated from letter HMMs. The highest recognition score determines the most likely pen trajectory as well as the most likely word model.

3 Recovering the time order of the handwriting signal

Starting from the graph representation, we find the path through the graph with the minimum cost. Ideally, this path is equivalent to the pen trajectory produced by the writer. This time order restoration is performed by a global optimization process proceeding in two steps: in a first step, the graph is divided into a set of pen strokes, each pen stroke being a sequence of segments delimited by a pen-down at one end and a pen-up at the other end. In the second step, each of the pen strokes is oriented and the pen strokes are time ordered.

For the first optimization process a cost is assigned to each link of the graph. Inter-link costs model the likelihood that two segments are part of the same stroke crossing a singularity. We estimate this cost based on a match of the pen trajectories before and after the singularity. Each trajectory of a given segment is extrapolated across the singular region and compared with the real trajectory of the other segment taking into account the differences in position and orientation [9]. Since the graph is not directed, the extrapolation process is performed in both ways, and the two costs are averaged. Intra-link costs are set to zero since no additional cost is required when progressing inside a given segment.

The optimal path should visit each node as often as the corresponding segment has been drawn since an off-line segment may correspond to several on-line signals. The problem of visiting every node of a graph at least once is known as the search of a pre-Hamiltonian cycle, and is equivalent to the search of a Hamiltonian cycle in the completed graph which visits every node exactly once [6]. Figure 3-a shows the graph of Figure 2 after completion. The cost of a link added at completion between node A and node B is set to the value of the minimum cost path between A and B in the uncompleted graph.



cost (4″↔8″) = min [cost (4″↔5′) + ret (5) + cost (5″↔8′) + ret (8), up-down (4″, 8″)]

Figure 3: - a - The completed graph of Figure 2; - b - Final graph for the first optimization process.

Figure 3-a shows the cost associated with the completion-link 4"-8". The term ret(.) represents the retracing cost of a segment and up-down(.) characterizes the pen-up and pen-down cost between two segment extremities. The completion-link 4"-8" for example is assigned the minimum of the two following values: (1) cost of the path 4"-5′-5"-8′-8" which is the sum of the costs of the inter-links 4"-5′ and 5"-8′ and of the costs of the retraced segment 5 and 8, and (2) cost of pen-down and pen-up between nodes 4" and 8.

In fact, we are not interested in an optimal cycle in the graph but in an optimal path which starts at the pen down point and ends at the pen up point. However, it is possible to transform the search for a Hamiltonian path in a non directed graph into a search for a Hamiltonian cycle in a graph derived from the original graph by adding a complementary node P [6]. Costs of these new links are set to zero. The search for a Hamiltonian cycle in a graph can be performed by various algorithms: branch and bound techniques, simulated annealing or tabou algorithms [14]. We have used the latter. For the example of Figure 3-b, the optimal path is 7″-7′-3″-3′-2′-2″-6″-6′-5″-5′-1″-1′-8′-8″-4″-4′.

The search for an optimal path is performed on the graph of each connected component of the handwriting. The result is a set of strokes for which we know the segment order, but not yet the orientation of the segments.

The optimal path of the word “et” in Figure 3-b is constituted by two strokes. The interconnections between these two strokes are modeled by a second graph which is shown in Figure 4. Since handwriting usually proceeds from left to right, the costs associated with inter-links are mainly based on the horizontal distance between the two strokes. The cost of an intra-link is proportional to the distance between its two extremities; left-right and top-bottom links are preferred. The second graph is already complete. A supplementary node P linked to all nodes of the graph is again added in order to transform the search of a Hamiltonian path into the search of a Hamiltonian cycle. The objective is then to find the Hamiltonian path with minimal cost. Again, the tabou algorithm has been used for that purpose. For the example of Figure 4, the optimal path found is 2′-2″-1″-1′.

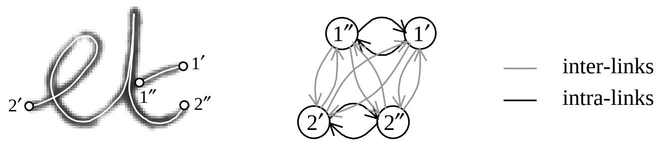


Figure 4: Second graph model for recovering stroke orientations and stroke ordering.

4 Word recognition with HMMs

At this point the handwriting signal consists of a time ordered sequence (recovered) of two types of observations: directed segments and pen-up trajectories. Each directed segment is described by a feature vector including the following information: size, y-position with respect to the reference lines, direction, and curvature. Pen-up trajectories are described by their y-position with respect to the reference lines and their direction. All geometrical features are normalized by the core zone height of the word image.

The next step is a vector quantization of the feature vectors using the K-means algorithm and a simple Euclidean metric [15]. For the segments, we have typically used K=300 clusters, and for the pen-up trajectories K=100 clusters.

The recognition of the resulting symbol sequences is achieved by using discrete HMMs. We have used 54 left-right letter models (a-z, A-Z, -, “dia”) with a number of states proportional to the average number of observations constituting the corresponding letter. “dia” designates a generic model for diacritical marks. The model topology (self loops and state transitions) is the same for all letters. The letter models are concatenated in order to build word models.

The HMMs have been trained using the Baum-Welch training algorithm. The word likelihoods can be computed by the forward-backward algorithm where the lexicon is either flat or organized in a trie structure [16].

The "OrdRec" approach proceeds by simultaneously ordering and recognizing a given off-line word image. In the first step, a list of recovered ordering candidates is established as discussed in section 3. In the second step, the system computes the recognition probability for each word in the lexicon and for each order candidate using HMMs.

The computational complexity of the "OrdRec" approach for a word with N segments is 2N!. We limit the number of order candidates to N1 candidates in the first graph, corresponding to N1 different segmentations into strokes. For each candidate, N2 stroke orders are proposed corresponding to the N2 best paths of the second graph. Thus, N=N1*N2 candidates are in competition in the recognition system which chooses the most likely candidate for each word of the dictionary.

5 Experiments and performance comparisons

For the experiments reported in this paper we have used a training set of 20 898 words and a test set of 10 448 words from a 197 word lexicon (French and English). All data is taken from the IRONOFF dual data base which has been presented in more detail in [18] and which has been collected among approximately 700 scriptors. For each word in the database, IRONOFF provides the off-line pixel image scanned with a resolution of 300 dpi, as well as the on-line signal which has been sampled at 100 points per second on a Wacom UltraPad A4.

We have conducted several experiments using different training conditions for the HMMs and different numbers N (=N1*N2) of "OrdRec" candidates. Figure 5 shows the recognition rate as a function of N. All five recognition systems tested start from the off-line images and use the same pre-processing, the same data representation, the same HMM topologies, and the same data sets. The only difference is how the time ordering of the segments is obtained. The "on-line" curve gives the recognition rate of the system using the true time order of the segments during training and test. For the IRONOFF database, the true time order can be obtained by a simple match between the off-line segments and the on-line points. Of course, the "on-line" recognition rate is independent of the number of "OrdRec" candidates N. The other four curves correspond to the recognition rates of "OrdRec" off-line recognition systems, i.e. only the recovered ordering of segments is available during test. They use respectively (i) the true on-line ordering of segments, (ii) the best recovered time ordering of segments, (iii) both the true on-line ordering and the best recovered time ordering of segments for training, and (iv) a combination of the two recognition systems "Ord. on" and "Ord. off" which proceeds by multiplying the resulting probabilities of the two systems for each of the word models in the lexicon.

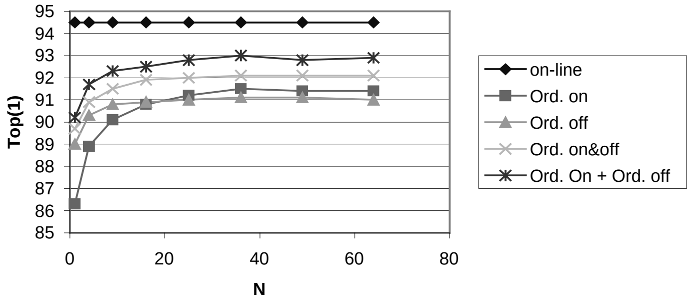


Figure 5: Recognition rates as a function of the number of "OrdRec" candidates.

As can be seen from Figure 5, all off-line recognition systems can be improved considerably by the " OrdRec" approach. For instance, the system which has been trained using the true on-line ordering available in the IRONOFF database and the best recovered ordering (Ord. on&off) achieves a 93% recognition rate with N1=N2=6, instead of the 90.2% without " OrdRec", thereby coming close to the recognition rate of the on-line system which achieves 94.5%. Increasing the number of "OrdRec" candidates beyond 40 or 50 does not bring any improvement.

For comparison, our best on-line recognition system which uses a data representation based on a resampled sequence of on-line points provides a recognition rate of 96% on the same data set. The same system obtains 97.5% when using a " RefRec" approach, i.e. several candidates for the reference lines are used and the best candidate is again found by the recognition process.

In order to evaluate the quality of the recovered stroke order, we have conducted a comparison of the word likelihoods for the true word model computed by both the on-line recognition system and the " OrdRec" system. This comparison shows that for about 80% of the samples the best " OrdRec" candidate obtains a likelihood which is close, equal or larger than the likelihood of the on-line system. Therefore, we conclude that for about 80% of the samples of the test set the "OrdRec" approach recovers the true (or close to true) time order of the handwriting signal. Note that approximately 15-20% of the test samples are correctly recognized despite an unsatisfactory restoration of the time order.

We have also analyzed the causes of the misclassification by the " OrdRec" system of samples which have been correctly classified by the on-line recognition system. Therefore, we have computed the best recovered time order with respect to the true word model. This time order candidate has been used by all word models. Since the on-line system has correctly classified the sample, and the recovered time order is optimized with respect to the true word model, a misclassification by the"OrdRec" system indicates a bad time order of the handwriting. About half of the errors made by our " OrdRec" system are due to a bad restoration of the time order. The other half are due to confusions, i.e. the time order may have been recovered correctly and the true word model may have achieved a good likelihood, but other "OrdRec" candidates, sometimes very different from the true time order, achieve even better likelihoods for word models different from the true word model at the recognition step. While adding more " OrdRec" candidates increases the chances of findiong a good time ordering, the chances of confusion increase as well. This effect may even be more important in large lexicon applications.

6 Conclusions

Using the same databases, the same data representation and the same algorithms for both on-line and off-line handwriting recognition, our experiments have confirmed that on-line recognition systems can achieve higher recognition rates than off-line systems. This shows that the time ordering of the signal contains important information as to the recognition of handwriting. We have also shown that an off-line recognition system can obtain recognition performances close to those of an on-line system using the " OrdRec" approach. Simultaneous time ordering and word recognition succeed in the vast majority of cases in recovering the true or close to true time order of the handwriting signal. Since the local decisions as to the time order, at crossings for instance, are taken at a global level by means of the entire recognition process, the " OrdRec" approach succeeds in restoring the true time order, even in cases where purely local analysis as reported in the literature does not work.

Acknowledgments

The authors wish to thank Zsolt Wimmer from Vision Objects and Tay Yong Haur from the Technical University of Malaysia for their contributions.

References

1. Y. Bengio, Y. LeCun, "Word-Level Training of a Handwritten Word Recogniser Based on Convolutional Neural Networks", *International Conference on Pattern Recognition*, p. 88-92, 1994.
2. G. Boccignone, A. Chianese, L.P. Cordella, A. Marcelli, "Recovering Dynamic Information from Static Handwriting", *Pattern Recognition*, Vol. 26, No. 3, p. 409-418, 1993.
3. H. Bunke, R. Ammann, G. Kaufmann, T.M. Ha, M. Schenkel, R. Seiler, F. Eggimann, "Recovery of Temporal Information of Cursively Handwriten

Words for On-line Recognition", *International Conference on Document Analysis and Recognition*, p. 931-935, 1997.

4. A.P. Dempster, N.M. Laird, D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm", *Journal of the Royal Statistical Society*, Vol. 39, No. 1, p. 1-38, 1977.
5. D.S. Doermann, A. Rosenfeld, "Recovery of Temporal Information from Static Images of Handwriting", *Computer Vision and Pattern Recognition*, 1992.
6. M. Gondran, M. Minoux, "Graphs and Algorithms", *Wiley*, 1984.
7. V. Govindaraju, D. Wang, S. Srihari, "Using Temporal Information In Off-Line Word Recognition", *Advanced Technology Conference*, Vol. 1, pp. 529, 1992.
8. S. Jäger, "A Psychomotor Method for Tracking Handwriting", *International Conference on Document Analysis and Recognition*, p. 528-531, 1997.
9. P.M. Lallican, C. Viard-Gaudin, " Une Méthode d'Optimisation Globale pour Reconstruire la Chronologie d'un Tracé Manuscrit Hors- ligne", *Conférence Internationale Francophone sur l'Ecrit et le Document*, p. 344-353, 1998.
10. P.M. Lallican, C. Viard-Gaudin, "Off-line Handwriting Modeling as a Trajectory Problem", *Advances in Handwriting Recognition* , Word Scientific Publishing, Vol. 34, p. 449-458, 1999.
11. H.J. Lee, M.J. Joe, "A Combined Method On The Handwritten Character Recognition", *International Conference on Document Analysis and Recognition*, Vol. 1, p. 112-115, 1995.
12. J.C. Pettier, "Extraction d'une Représentation Adaptée à la Reconnaissance de l'Écriture", *PhD thesis*, Université de Rennes I, France.
13. R. Plamondon, C.M. Privitera, "The Segmentation of Cursive Handwriting : An Approach Based on Off-line Recovery of the Motor-Temporal Information", *IEEE Transactions on Image Processing*, Vol. 8, No. 1, p. 80-91, 1999.
14. C. Prins, "Algorithmes de graphes", *Eyrolles*, 1994.
15. J. Mac Queen, "Some Methods for Classification and Analysis of Multivariate Observations", *Berkeley Symposium*, 1967.
16. L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Readings in Speech Recognnition*, p. 267-296, 1989.
17. R. Seiler, M. Schenkel, F. Eggimann, "Off-line Cursive Handwriting Recognition Compared with On-Line Recognition", *International Conference of Pattern Recognition*, p. 505-509, 1996.
18. C. Viard-Gaudin, P.M. Lallican, S. Knerr, P. Binter, "The IRESTE ON-OFF (IRONOFF) Handwritten Image Database", *International Conference on Document Analysis and Recognition*, p. 455-458, 1999.
19. Z. Wimmer, S. Garcia- Salicetti, B. Dorizzi, P. Gallinari, "Reconnaissance de Mots Manuscrits par une Approche Neuro-Markovienne : Coopération En-ligne Hors-ligne", *Colloque Int. Francophone sur Ecrit et Document* , pp. 354-363, 1998.