Off-Line Handwritten Numeral String Recognition

Thien M. Ha and Horst Bunke
University of Berne
Institut für Informatik und Angewandte Mathematik
Neubrückstr. 10, CH-3012 Bern, Switzerland
Email: {haminh,bunke}@iam.unibe.ch

November 3, 1997

Abstract

This report describes work on handwritten numeral string recognition. In particular, we address the building of a continuous recognition system based on hidden Markov models and the combination of this system with a previously developed discrete recognition system. We describe experiments with a simple combination scheme, called score summation, that turns out to be capable of improving the recognition rates of both individual systems. Further research directions are also proposed.

CR Categories and Subject Descriptors: I.4.7 [Image Processing]: Feature Measurement; I.4.10 [Image Processing]: Image Representation; I.5.0 [Pattern Recognition]: General; I.5.1 [Pattern Recognition]: Models; I.5.2 [Pattern Recognition]: Design Methodology; I.5.4 [Pattern Recognition]: Applications; I.5.m [Pattern Recognition]: Miscellaneous.
Additional Keywords: document processing, pattern recognition, Optical Character Recognition (OCR), integrated segmentation and recognition, discrete approach, continuous approach, hidden Markov model, combination scheme.

Contents

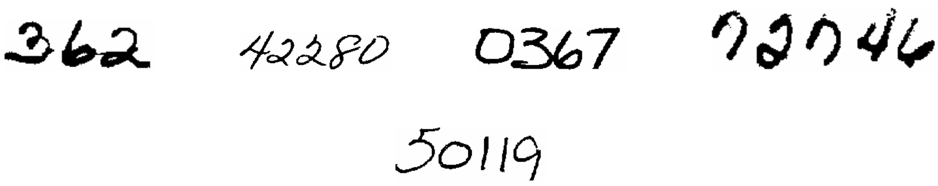

Figure 1: Numeral String Samples from NIST SD3 Database

1 Introduction

Automatic character recognition is a subfield of pattern recognition and can either be on-line or off-line. On-line recognition refers to those systems where the data to be recognised is input through a tablet digitiser, which acquires in real-time the position of the pen tip as the user writes. In contrast, off-line systems input the data from a document through an acquisition device, such as a scanner or a camera. Off-line character recognition is moreover divided into two categories, namely, machine-printed and handwritten. This report is concerned with one particular problem in off-line handwriting recognition, namely, the recognition of sequences of numerals where individual numerals may touch or overlap each other. Such sequences are usually produced under unconstrained, i.e. 'natural', handwriting; see Fig. 1. Notice that this recognition task is different from recognising words from a dictionary in that virtually no context is available, i.e., any numeral can follow any other one.

Off-line handwritten numeral string recognition is an important and necessary step in many document processing applications. For instance, we can think of the reading and processing of checks, mail addresses, tax forms, and census forms. Although the recognition of isolated numerals has somehow reached a saturation after decades of research, the automatic reading of numeral strings has still ample room for improvements. All solutions that have been described in the literature are not mature enough for application to tasks from the real world.

A new trend has recently appeared in the field of character recognition, which consists in combining multiple independent systems to compensate for individual system's weaknesses, and thus provides the combined system with superior performance. This approach has proven remarkably efficient in improving the accuracy of isolated character recognition, but has not yet been used for numeral string recognition. Therefore, we propose to investigate the use of the combination scheme in numeral string recognition.

To study combination schemes, it is necessary to have at least two independent systems. Currently, handwritten numeral string recognition systems are either based on the *discrete* or the *continuous* approach. Since we had already implemented one system based on the discrete approach [15], we investigated and implemented another one based on the continuous approach, and then studied appropriate schemes to combine the two systems.

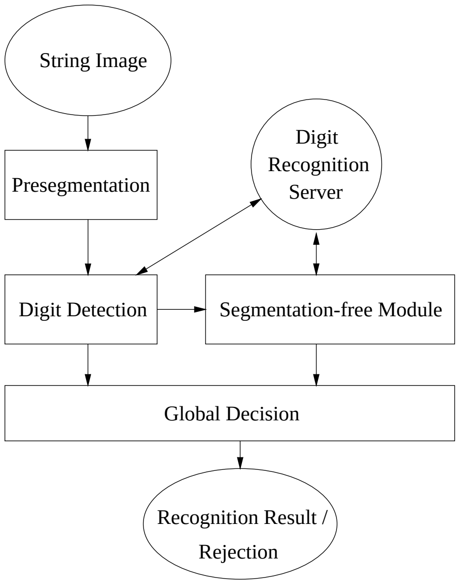

Figure 2: Discrete recognition system.

Section 2 summarises the discrete recognition system that had been implemented in our previous works. Section 3 describes the continuous recognition system based on Hidden Markov Models. Section 4 discusses the combination of the two systems, including experimental results and Section 5 concludes the report.

2 Discrete Recognition System (DRS)

This section provides a brief description of our discrete recognition system. The full description can be found in [15, 17]. In the discrete approach, the numeral isolation (separation) takes place at a number of points where the image exhibits some special characteristics. For instance, the analysis of the vertical projection of black pixels provides a simple (but not always correct or sufficient) way to segment the input string into numerals or groups of numerals. Other special characteristics are line end-points, crossings, and T-joints [24].

Our DRS is able to simultaneously segment and recognise totally unconstrained data. The system is built upon a number of components, namely, a presegmentation module, an isolated numeral recogniser, a segmentation-free module and a merging module; see Fig. 2. Presegmentation consists in dividing the input numeral string image into groups of numerals, each of which represents an integer number of numerals. For each group, the actual number of numerals and their identity are then determined by a cascade of two recognition-based tests: isolated numeral and segmentation-free. The last technique is able to recognise a numeral group of any length. All results from all groups are eventually merged yielding the final interpretation of the input numeral string. We also introduced the concept of dummy symbol in order to overcome the problem of noisy parts that cannot be eliminated

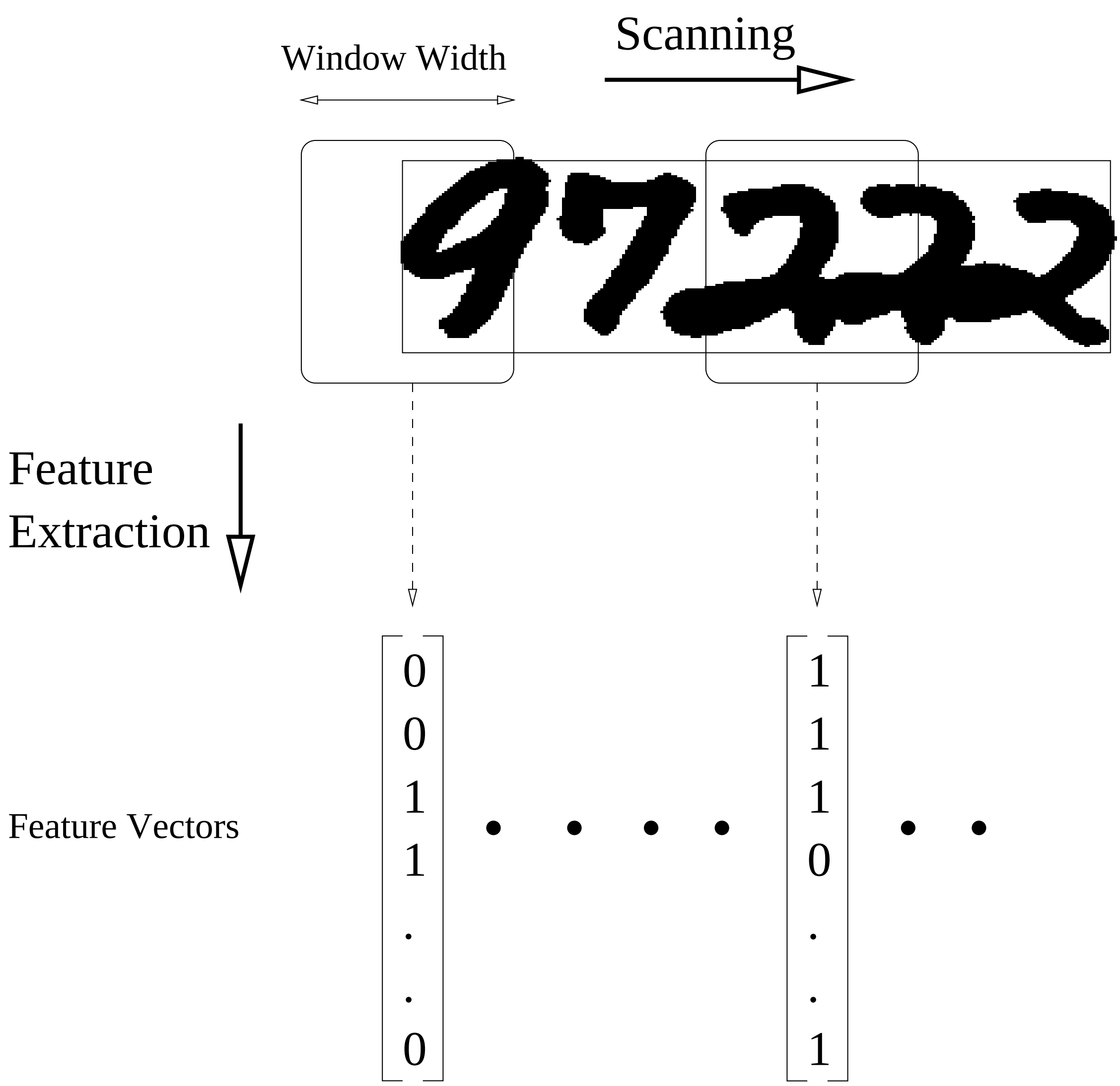

Figure 3: Sliding window approach.

by standard filtering algorithms. Experiments on totally unconstrained data showed that the results compare favorably to other published methods [15, 17]. More details will be provided in Section 4.

3 Continuous Recognition System (CRS)

In the continuous approach, a sliding window scans the input image from left to right. Each position of the window defines a sub-image which is extracted and analysed. Thus the input image is converted into a sequence of feature vectors each of which represents the sub-image within the sliding window at a given position; see Fig. 3. Since it is generally impossible to have a window width that fits all individual numerals in a string, some mechanism must be provided to handle this variability.

For instance, the window width is chosen to be roughly two times the average width of individual numerals so as to be almost certain that it covers any numeral in a string at least once through the scanning [22]. Another method consists in using a recurrent neural network (a general architecture that includes time-delay neural networks) to encode information between neighbouring vertical strips (thin vertical sliding windows) [20]. Other similar mechanisms have also been used [18, 23].

In our work, we adopted the framework of Hidden Markow Models (HMM). That is we use thin vertical sliding windows; information between neighbouring windows is encoded by the state transition probabilities. The advantages of adopting this framework are 1) there exists a well etablished mathematical theory, 2) the training can be performed without manually segmenting each numeral string into individual numerals and 3) many software implementations are available. Our continuous recognition system is built on the software package HTK provided by Entropic Cambridge Research Laboratory [29]. We first describe the HMM framework, then explain the feature extraction.

3.1 HMM Framework

The idea of HMM is the decomposition of a complex stochastic process into two simpler but nevertheless coupled processes. The first process accounts for the transition between states whereas the second process models the state-dependent emission of symbols or signals (images). Ideally, the emission process should be dependent on both the state and the transition, but for mathematical tractability the latter is often neglected. Moreover, the transition is usually simplified to be of first order Markovian. Despite these simplifications, HMMs have proved very powerful in various applications, e.g. speech and character recognition. In the following, we first describe the modelling of a single numeral by HMM. Then, the modelling of a numeral string which is a sequence of numerals is explained.

Formally, a HMM is specified by the following parameters; see Fig. 4.

1. Number of states.
2. Initial probability of being in a state.
3. Transition probabilities between states.
4. Emission probabilities of each state.

The number of states is a design parameter, whereas the second and third groups of parameters are obtained in a training phase. The parameters of the fourth group are partially designed and partially trained. For instance, as in our system, the emission probabilities are modelled by a mixture of Gaussian distributions, where the number of mixture components is chosen by the designer and the parameters of each component (mean, variance, and component weigh) are obtained by training. HMM modelling can thus be performed in two steps: chosing the design parameters

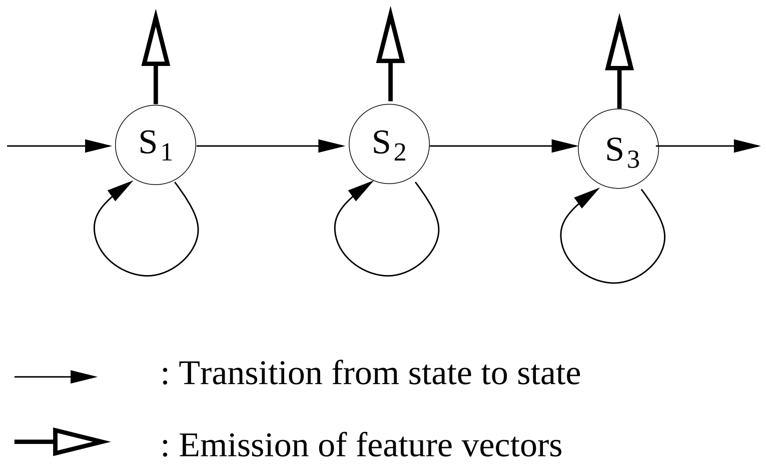

Figure 4: A left-to-right three-state hidden Markov model.

and training the remaining parameters. The first is mostly based on experience and heuristics whereas the second is based on statistical estimation theory, e.g. the Baum-Welch algorithm.

In our work, each numeral is modelled by one HMM. The numeral "1" written in American style is usually short and is thus modelled by a small number of states (7); the other numerals have a larger number of states (8). The difference in the number of states of various models is small because each model has to take into account its left and right empty parts.

In order to model a numeral string, we have to connect many HMMs together. In the training phase, the connection is straighforward. For instance, the HMM of string '213' is obtained by concatenating the HMMs of '2', '1', and '3'. The training can therefore be performed on string level: No manual segmentation is required. In the recognition phase, the connection is of course unknown. One solution would be to compare the input image with all possible sequences of numerals and choose the best matching one. Unfortunately, this approach is impractical for our problem because there are an infinite number of possible sequences (any numeral can follow any other and the number of numerals is unknown). The solution to this problem is provided by a dynamic programming technique, called Viterbi algorithm, together with a backtracking mechanism; see [29] for more details.

In sum, applying HMM to numeral string recognition consists of the following steps:

1. Select a feature extraction method.
2. Choose the HMM design parameters for each numeral.
3. Train the remaining HMMs parameters.
4. Test the models.

In our works, the first step received much attention due to our past experience that feature selection is critical. Therefore, its description is given in the Section 3.2. The second step is mostly heuristic, i.e., based on trial and error. The last two steps correspond to standard functionalities of the HTK package for which full details can be found in [29].

3.2 Feature Extraction

Although HMM can be applied directly to the raw pixels within a thin vertical sliding window, our past experience in the field of isolated numeral recognition suggests that more sophisticated features can significantly enhance the discrimination power of the system. Therefore, we selected a set of features based on image contour histograms. The feature extraction works as follows.

The string image is normalised to a fixed height (30 pixels) while keeping the aspect ratio constant. The width of the normalised image is thus variable. The normalised image is scanned by a thin vertical window of size 30x10. Each position of the scanning window defines a sub-image of the same size (30x10), the raw pixels. Each sub-image is in turn divided into three zones: upper, middle, and lower. In each zone, we compute the directional histogram of the contour pixels. The histograms of the three zones constitute the feature vector. In our system, eight directions are used for the histogram. The feature vector is of size 24 (=8x3).

4 Combination

Combination refers to the idea of using more than one recognition system in an appropriate way to enhance the performance of the combined system with respect to the individual systems. There are basically two ways of combining systems, namely, serial and parallel.

The serial scheme first applies a fast but low-precision system. Easy cases are directly recognised at this stage while difficult cases, according to some rejection criterion, are submitted to a slower but high-precision system for recognition. In this way, the precision of the combined system is nearly as high as the second system while increasing the speed to something between the two systems. The serial scheme is of great importance to practical applications and is already routinely used [25].

In our work, we are interested in the parallel scheme, which has the potential to improve the precision over those of the individual systems, at the price of a higher computation time. In particular, we are interested in combining the continuous recognition system (CRS) with the discrete recognition system (DRS) in an attempt to improve the recognition rate of the latter. Our DRS, tested on a worldwide standard database from NIST, has proven to be amongst the best systems. It turns out that a very simple scheme combining DRS and CRS can actually reduce the error rate of the DRS by roughly 10%, in relative value.

We first describe the NIST database and then the combination scheme. Next, we present the experimental results and finally further research directions are proposed.

4.1 NIST Database

The database was provided by the American *National Institute of Standards and Technology* (NIST) in 1992 as part of a conference to assess the state-of-the-art in isolated handwritten character recognition [28]. The database contains isolated numerals, upper- and lower-case letters and 2100 scanned images of specially designed forms containing fields for numeral string images; see Fig. 1 for some examples.

As test set we have selected the files f1800-f1899 and f2000-f2099, which have also been used as a test set by [18]. The file f1823 was not used due to a form registration failure. Another 48 fields were not included in the test. Among these fields were empty fields (3 cases), fields with field extraction errors (32 cases) and fields containing wrong numbers, or numbers which are not recognizable even for humans (13 cases). The total number of strings used in our experiments is 4920.

4.2 Combination Scheme

Many combination schemes, such as voting and neural networks, have been proposed in literature. The scheme used in our experiments is called *score summation.* To use this scheme, it is assumed that each recognition system yields as outputs a list of candidates together with their score. A candidate is a string of numerals and the score is that of the whole string. The score value lies between 0.0 (the worst) and 1.0 (the best). The combination scheme consists in merging the two lists of candidates from the two systems to produce a new list of candidates. If a candidate is present in both lists, its new score is simply the sum of the two previous scores; otherwise its new score is equal the old one. After merging, the new list of candidates is re-sorted in decreasing order of score and the candidate on the top of the new list is chosen as the best candidate of the combined system.

The scheme is simple and has the advantage of using not only the best candidate from each system but also the second, third, ... best candidates. Moreover, not only their rank has an influence but also their score, i.e. a second best candidate with high score is intuitively more important than a second best candidate with lower score.

Notice that before using the score summation scheme, the scores of each recognition system must be normalised so that they sum up to one. This normalisation step makes the scores of the two systems comparable. Additionally, we could also assign a different weight to each system to take into account their relative performance.

4.3 Experimental Results

The three systems DRS, CRS, and ComRS (combined recognition system) were tested on the 4920 string images from the NIST database. A result is considered

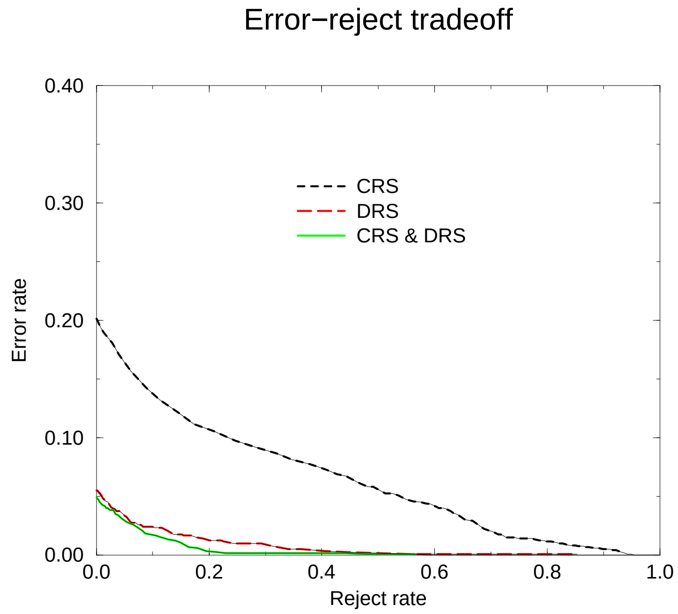

Figure 5: Error-reject tradeoff on string level.

correct only if all numerals composing the string are correctly recognised. The numbers of errors from DRS, CRS, and ComRS were 361, 1050, and 322, respectively. Although CRS produced about three times more errors than DRS, it could nevertheless reduce the errors of the latter in the combined system. In percentage, the error rates of DRS, CRS, and ComRS were 7.34%, 21.34%, and 6.54%. In relative value, CRS contributed to reduce the error rate of DRS by 10.90% ((7.34 - 6.54)/7.34).

We also experimented with the rejection capabilities of the three systems. For this purpose, we plotted the error-reject tradeoff curves of the three systems by varying a threshold between zero and one. The rejection rule consisted in rejecting a result if the score is below the threshold and accepting it otherwise. Figure 5 shows the tradeoffs of the three systems CRS, DRS, and ComRS. The same results are presented in Fig. 6 where the error rates are plotted in a logarithmic scale. For very low error rates, the curves are not stable anymore due to the limited size of the test set.

4.4 Further Research

There is no doubt that we can design more sophisticated combination schemes than the score summation. However, it is remarkable that even this simple scheme can already improve the recognition rate of the DRS. Note also that the above results were obtained without fine-tuning. One simple way to improve the result could be obtained by weighting the scores of the DRS more than the scores of CRS (DRS is much better that CRS). Another possibility is to improve the recognition rate of CRS by optimising its parameters and use the same combination scheme.

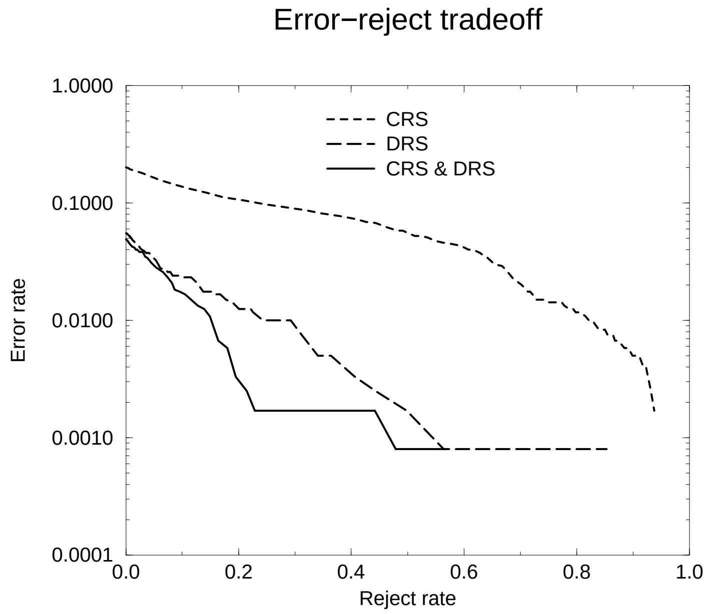

Figure 6: Error-reject tradeoff on string level.

Designing more sophisticated combination schemes would require as a first step the analysis of error cases on a large number of samples. In particular, it would be useful to distinguish between segmentation and recognition errors, since they may require different treatments.

5 Conclusion

This report describes work on handwritten numeral string recognition. In particular, we address the building of a continuous recognition system based on hidden Markov models and the combination of this system with a previously developed discrete recognition system. We experimented with a simple combination scheme, called score summation, that turned out to be capable of improving the recognition rates of both individual systems. Further research directions were also proposed. The main conclusion of our works is that combining variable length results does not necessarily require sophisticated dynamic matching techniques, such as string matching. Whether these more sophisticated techniques can further improve the simple combination scheme is the subject of our ongoing research.

References

[1] H. Baird, H. Bunke, and K. Yamamoto (Eds.), *Structured Document Image Analysis.* Springer-Verlag, 1992.

[2] H. Bunke, M. Roth, and E.G. Schukat-Talamazzini, "Off-line recognition of cursive script produced by a cooperative writer," Proc. 12th Int. Conf. on Pattern Recognition, Jerusalem, 1994, 383 - 386.

[3] H. Bunke, P. Wang, and H. Baird (Eds.), *Document Image Analysis.* World Scientific Publ. Co., 1994. (Also available as special issue of Int. Journal of Pattern Recognition and Art. Intelligence.)

[4] H. Bunke and R. Liviero, "Classification and Postprocessing of Documents Using an Error-correcting Parser," Proceedings of the third International Conference on Document Analysis and Recognition, Aug. 14-16, 1995, Montréal, Canada, pp. 222-226.

[5] H. Bunke, M. Roth, E.G. Schukat-Talamazzini, "Off-line cursive handwriting recognition using hidden Markov models," to appear in Pattern Recognition, 1995.

[6] R.G. Casey and E. Lecolinet, "Strategies in Character Segmentation: A Survey," Proceedings of the third International Conference on Document Analysis and Recognition, Aug. 14-16, 1995, Montréal, Canada, pp. 1028-1033.

[7] G. Congedo, G. Dimauro, S. Impedovo, and G. Pirlo, "Segmentation of Numeric Strings," Proceedings of the third International Conference on Document Analysis and Recognition, Aug. 14-16, 1995, Montréal, Canada, pp. 1038-1041.

[8] B.V. Dasarathy (Ed.), *Decision Fusion,* IEEE Computer Society Press, 1994.

[9] A. Filatov, A. Gitis, and I. Kil, "Graph-based Handwritten Digit String Recognition," Proceedings of the third International Conference on Document Analysis and Recognition, Aug. 14-16, 1995, Montréal, Canada, pp. 845-848.

[10] H. Fujisawa, Y. Nakano, and K. Kurino, "Segmentation Methods for Character Recognition: From Segmentation to Document Structure Analysis", *Proceedings of the IEEE,* Special Issue on Optical Character Recognition, Vol. 80, No. 7, pp. 1079-1092, July 1992.

[11] Thien M. Ha and H. Bunke, "Very Fast Recognition of Giro check Forms", in D.P. D'Amato (ed.): *Character Recognition Technologies,* SPIE Proc. Vol 1906, Feb. 1-2, San José, California, pp. 100-111, (1993).

[12] Thien M. Ha, H. Bunke, and J. Clarinval, "A Real-Time System for Processing Forms," *Actes des Journées sur le Traitement d'Images, la Reconnaissance des Formes et leurs Applications,* R. Ingold (Ed.), Laboratoire d'Informatique de l'Université de Fribourg, Switzerland, pp. 28-31, (1993).

[13] Thien M. Ha and H. Bunke, "Model-Based Analysis and Understanding of Check Forms," *International Journal of Pattern Recognition and Artifial Intelligence,* Vol. 8, No. 5, 1994, pp. 1053-1080.

[14] Thien M. Ha and H. Bunke, "Handwritten Numeral Recognition by Perturbation Method", *Proc. of the Fourth Int. Workshop on Frontiers of Handwriting Recognition,* Taipei, Taiwan, Dec. 7-9, pp. 97-106, 1994.

[15] Thien M. Ha, D. Niggeler, and H. Bunke, "A System for Segmenting and Recognising Totally Unconstrained Handwritten Numeral Strings," Proceedings of the third International Conference on Document Analysis and Recognition, Aug. 14-16, 1995, Montréal, Canada, pp. 1003-1009.

[16] Thien M. Ha, D. Niggeler, H. Bunke, and J. Clarinval, "Giro Form Reading Machine," *Optical Engineering,* Vol. 34, No. 8, pp. 2277-2288, 1995.

[17] Thien M. Ha, M. Zimmermann, and H. Bunke, "Off-line Handwritten Numeral String Recognition by Combining Segmentation-based and Segmentation-free Methods," *Pattern Recognition.* To appear.

[18] J. Keeler and D.E. Rumelhart, "A Self-Organizing Integrated Segmentation and Recognition Neural Network," in *Advances in Neural Information Processing Systems,* Vol. 4, J.E. Moody, S.J. Hanson, and R.P. Lippmann (Eds.), San Mateo, CA : Morgan Kaufmann, 1992, pp. 496-503.

[19] F. Kimura, Y. Miyake, and M. Shridhar, "Handwritten ZIP Code Recognition Using Lexicon Free Word Recognition Algorithm," Proceedings of the third International Conference on Document Analysis and Recognition, Aug. 14-16, 1995, Montréal, Canada, pp. 906-910.

[20] S.W. Lee and E.J. Lee, "Integrated Segmentation and Recognition of Connected Handwritten Characters with Recurrent Neural Network," Proceedings of the third International Conference on Document Analysis and Recognition, Aug. 14-16, 1995, Montréal, Canada, pp. 413-416.

[21] E. Lethelier, M. Leroux, and M. Gilloux, "An Automatic Reading System for Handwritten Numeral Amounts on French Checks," Proceedings of the third International Conference on Document Analysis and Recognition, Aug. 14-16, 1995, Montréal, Canada, pp. 92-97.

[22] G.L. Martin, M. Rashid, and J.A. Pittman, "Integrated Segmentation and Recognition Through Exhaustive Scans or Learned Saccadic Jumps," *International Journal of Pattern Recognition and Artifial Intelligence,* Vol. 7, No. 4, 1993, pp. 831-847.

[23] O. Matan, J.C. Burges, Y. Le Cun, and J.S. Denker, "Multi-Digit Recognition Using a Space Displacement Neural Network," in *Advances in Neural Information Processing Systems,* Vol. 4, J.E. Moody, S.J. Hanson, and R.P. Lippmann (Eds.), San Mateo, CA : Morgan Kaufmann, 1992, pp. 488-495.

[24] H. Nishida and S. Mori, "A Model-Based Split-and-Merge Method for Recognition and Segmentation of Character Strings", in *Advances in Structural and Syntactic Pattern Recognition*, H. Bunke (Ed.), World Scientific, pp. 300-309, 1992.

[25] J. Schürmann, "Pattern Classification", *John Wiley & Sons, Inc,* 1996.

[26] N.W. Strathy and C.Y. Suen, "A New System for Reading Handwritten ZIP Codes," Proceedings of the third International Conference on Document Analysis and Recognition, Aug. 14-16, 1995, Montréal, Canada, pp. 74-77.

[27] C.Y. Suen, C. Nadal, R. Legault, T.A. Mai, and L. Lam, "Computer Recognition of Unconstrained Handwritten Numerals," in *Proceedings of the IEEE, Special Issue on Optical Character Recognition,* T. Pavlidis and S. Mori (Eds.), 1992, pp. 1162-1180.

[28] R.A. Wilkinson, J. Geist, S. Janet, P.J. Grother, C.J.C. Burges, R. Creecy, B. Hammond, J.J. Hull, N.W. Larsen, T.P. Vogl, and C.L. Wilson, *The First Census Optical Character Recognition Systems Conference,* The U.S. Bureau of Census and the National Institute of Standards and Technology, Technical Report #NISTIR 4912, Gaithersburg, MD, Aug. 1992.

[29] S. Young, J. Jansen, J. Odell, D. Ollasen, and P. Woodland, *The HTK Book,* Cambridge University, 1996.