3
CURSIVE WORD RECOGNITION AND FUTURE SUGGESTIONS

Muhammad Faisal Zafar
Dzulkifli Mohamad

1 INTRODUCTION

Virtually, cursive script recognition systems preprocess the input data. Many systems perform lexical verification of the obtained results. However, the segmentation can be bypassed or merged with the recognition. Three general classes of approaches can be identified using the approach to segmentation as the criterion:

- holistic recognition. Entire words are matched, without any attempt to segment or locate individual letters.
- recognition based segmentation. Again, entire words are processed. An attempt is made to locate some smaller entities (e.g. letters or graphemes) within the word. The combination of such entities which matches the processed word best constitutes the solution. Segmentation in this approach is secondary and becomes available only after the recognition is performed.
- segment and recognize approach. Possible segmentation points are identified. Recognition is attempted for various entities located by the identified possible segmentation points. Various selections of the segmentation points are used. The process

results in a large number of alternative solutions, the best of which constitute the answer.

This chapter presents a number of cursive script recognition systems reported in the literature and their results using holistic recognition.

1.1 Holistic Recognition

Shape recognition algorithms can be applied directly to whole words. Recognition procedures are usually similar to those applied to characters. Whole word recognition omits one potential source of errors: the letter segmentation. It is also expected to cope with illegible writing, where not all the letters can be found. Unfortunately, the possible variability of the way whole words are written is much higher than in the case of single letters. To ensure accurate recognition the number of words needs to be relatively small.

Farag used Freeman encoding and Markov chain for whole word recognition (Farag, 1979). A recognition rate of 100% is reported for ten cursive words and one writer. A whole word recognition approach based on Hidden Markov Model is also applied to off-line cursive postal code recognition (Bertille93 and Yacoubi, 1993). Upper and lower outlines are used for the recognition. Recognition rates between 23% and 63% are reported for different configurations of the system and data sets of 3839 and 1121 images of postal codes taken from the real mail.

Brown and Ganapathy used feature vectors and an estimate of the length of the word to represent the word characteristics in a global way (Brown and Ganapathy, 1980). Classification is then done on the extracted feature vectors using the K-nearest neighbour method. The recognition domain was 43 words. Three authors wrote ten samples, each containing 22 words. The recognizer was trained on data of one of the writers and tested on data of the two others. Recognition rates between 63.2% and 80.3% are reported.

Simon presents an off-line system for the recognition of bank cheque amounts (Simon, 1992). The system locates anchor points which form the basis for the recognition. Other features are located and a large set of rules applied to distinguish between words. The recognition domain contains 25 words. Average recognition rate for eight writers (875 words in total) is 76% for top five word alternatives. The average position of the correct word on the list of alternatives is 1.8. This indicates that the described system has some disambiguation problems.

Powalka introduces a wholistic recognizer as an auxiliary expert in an on-line hybrid recognition system (Powalka, 1994). The recognizer uses a very limited set of features consisting of sequence of ascenders and descenders and an estimate of the word length. A fuzzy logic based matching algorithm is used. Average recognition rates obtained for a 200 words lexicon are 40.8% and 60.6% for top one and top five alternatives, respectively. Handwriting of 18 writers was used, each data set containing 200 words.

Severe disambiguation problems can be observed for this recognizer. This is the result of a limited set of features used. Applying more complex features decreases the gap between the recognition rates for top one and top five alternatives (57% and 63.4% respectively).

Whole word recognition can also be used as a filtering process to reduce a large dictionary to a small number of candidates that would be further processed by more powerful recognition algorithms. The system presented in (Hull *et al.*, 1991; Ho *et al.*, 1991), works on static images of postal addresses. The use of the word shape allows limiting a 33850 words lexicon to 500 words with 93% accuracy (Ho *et al.*, 1991).

The word recognition system proposed in this work is also based on the wholistic recognition approach.

2 SYSTEM OVERVIEW

The task of cursive handwriting recognition, generally, involves three major modules, namely, the preprocessing, where noise reduction and normalization take place, the feature extraction, where the data of a handwritten word is transformed into a sequence of numerical feature vectors, and the recognizer, which converts these sequences of feature vectors into a word class. The first step in the processing chain, the preprocessing, is mainly concerned with input data normalization. The goal of the different normalization steps is to produce a uniform data of the writing with less variation of the same character or word across different writers. The aim of feature extraction is to derive a sequence of feature vectors which describe the writing in such a way that different characters and words can be distinguished, but which do not contain too much redundant information. At the core of the recognition procedure is a recognizer i.e NN or HMM. It receives a sequence of feature vectors as input and outputs a word class.

An overview of the whole system is presented in Figure 1. The flow of data during training is shown by the dashed line arrows, while the data flow during recognition is shown by solid line arrows. In following sections, the techniques and algorithms to be involved for each processing block are briefly introduced.

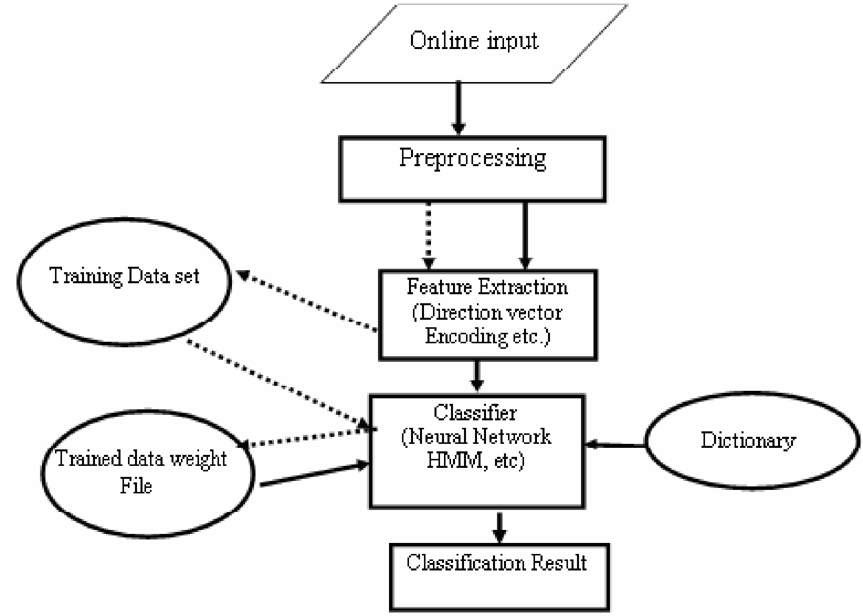

Figure 1 Block diagram of the system

2.1 Data Collection

Online handwritten data must be collected using a special device. Typically, a digitizing tablet is used that samples the location of a stylus on the tablet at the rate of approximately 73 - 200 times per second. This generates a sequence of (*x; y*) coordinates which define the trace of the pen over time. The stylus will typically have a switch to detect pen-down (when the pen is touching the tablet) and pen-up (the pen is not touching the tablet) status.

2.2 Preprocessing

The goal of preprocessing is to reduce or eliminate some of the variations in handwriting that may exist that are not useful for pattern class discrimination. It is possible to assume that a good representation of carefully written handwriting is available and to consider only such data. However, the preprocessing becomes essential when dealing with unconstrained or loosely constrained writing is required. Two different tasks are usually performed (Berthod, 1982):

- reduction of the noise;
- reduction of the volume of information to be processed by eliminating those points which are not necessary for the recognition algorithms used.

Other aspects of the preprocessing may include efforts to reduce the variability of the input data, that is transform it into some "normal" form. This includes size normalisation, deskewing, deslanting, etc (Guerfali and Plamondon, 1993). Preprocessing alters the input data. Great care must be taken in order not to alter the data in such a way that useful information is lost. Many supportive and counter arguments for preprocessing can be presented. However, preprocessing is generally understood to normalise the data and thus reduce its variability. The use of preprocessing improves recognition rates (Brown and Ganapathy, 1983; Burr, 1982). If preprocessing is not performed, recognition algorithms face a potentially more difficult task (Powalka, 1995).

2.3 Feature Extraction

As mentioned previously, the feature extraction plays an important role in the overall process of handwriting recognition. Here for the word recognition system, the whole word is converted to 8-directional vector sequence. This sequence behaves as observation sequence for HMM thus it is directly fed to the HMM for training and recognition purposed after smoothing.

3 EXPERIMENTS

3.1 Data Set and Model Parameters

The data used in this work was collected using tablet SummaSketch III. It has an electric pen with sensing writing board. An interface was developed to get the data from tablet.

First ten digits written in cursive word form i.e. one, two, three and so on (see Figure 2), were considered as case study. In the data set, the total number of handwritten words is about 150 words. Hidden Markov Models (HMMs) were used as classifier. We used just one HMM per word class. The number of training samples per class was eleven, and the number of observation symbols in all HMMs were fixed at $M = 8$, due to the 8-direction vector encoding method. The length T of the observation sequence was variable depending upon the length of handwritten cursive word.

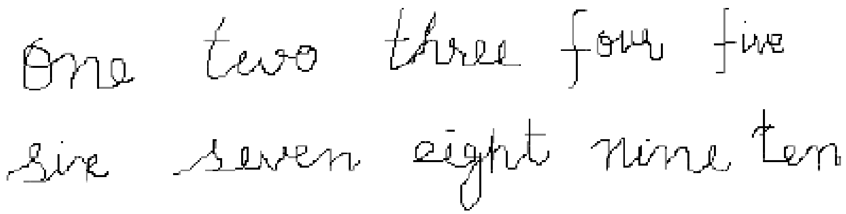

Figure 2 Examples of handwritten cursive words

3.2 Model Training

Four different sets of HMMs were experimented with model states N = 5, 6, 7 and 8. *Baum–Welch* algorithm has been used. The main advantage of HMM bases approaches is the existence of a *Baum–Welch* procedure (Rabiner and Juang,1993) that adjusts iteratively and automatically HMM parameters given a training set of observation sequences. This algorithm guarantees that the model converges to a local maximum of the probability of observation of the training set according to the maximum likelihood estimation criterion. The local maximum depends strongly on the initial HMM parameters (Koerich, 2002).The convergence of the training process was judged by a small threshold 0.001.

3.3 Recognition

Word recognition was performed using the following steps and rules:

1. Given an observation O from the testing data, $P(O|\lambda_i)$ $i = 1, 2, \ldots, 10$ are evaluated, where λ_i are well trained HMMs corresponding to word classes.

2. If $P_i \leq P(O|\lambda_i)$ where P_i is a threshold, then λ_i will enter the competition; otherwise, it will be eliminated. All $P(O|\lambda_i)$ that pass their thresholds then are sorted in the order of highest probability first.

3. Suppose O is from class j. If $P(O|\lambda_i) < P_i$, $i = 1,2, \ldots, 10$, then we say O is rejected. In other cases, if $P(O|\lambda_j) = \max i\{ P(O|\lambda_i) \}$, then we say O is recognized; otherwise, we say it is substituted (confused).

The word recognition results using different number of symbol categories are summarized in Table 1. The total number of testing words was more than 200. From Table 2, one can see that using different number of state categories, the recognition rate varied from 60% to 77.0%, the substitution rate varied from 36% to 21%, and the rejection rate varied from 4% to 2%.

Table 1 Cursive Word Recognition with different # of states of HMM

# of states	Recognition	Substitution	Rejection
N=5	60%	36%	4%
N=6	63.5%	33%	3.5%
N=7	67.2%	30.1%	2.7%
N=8	77%	21%	2%

3.4 Results and Discussion

So far we have assumed a unique word model (HMM) for each word class. In word recognition systems with a small vocabulary, it is possible to build an individual HMM for each word (Simon, 2004). However, this assumption does not hold in the case of large vocabularies because it is very difficult for a single model to capture the high variability and ambiguity of a large number of writing styles and writers. It has been shown that when a large amount of training data is available, the performance of a word recognizer generally can be improved by creating more than one model for each of the recognition units because it provides more accurate representation of the variants of handwriting (Scott, 2000; Yacoubi, 1999; Rabiner et al., 1989). On the other hand, while multiple word models may improve the recognition accuracy, they also may increase the computational complexity.

To achieve high recognition rates, the character HMMs have to be adapted to the problem. In particular the number of states, the possible transitions and the type of output probability distributions has to be selected. To set the free parameters of the HMMs, the Baum-Welch training (Rabiner and Juang, 1993) is used. Baum-Welch training is a version of the Expectation-Maximization technique (Dempster, 1977) and works with labeled training data. The product of the likelihood values for the correct word HMMs of the training patterns is guaranteed to increase in each iteration of the algorithm, yet the recognition rate of the classifier may decrease when using too many iterations. This is because the HMMs are overfit to the training data (Simon, 2004).

4 FUTURE SUGGESTIONS

The undertaken research work that is presented here, resulted in a number of achievements. Looking back, several possible improvements, alterations and shortcuts come to mind. This is also an important outcome of the work. Suggestions presented in this section are believed to be universal and applicable to other handwriting recognition systems as well. It is believed that results can be further improved by the introduction of different, multiple algorithms at each processing stage, like 16-direction encoding, slant removal, and middle zone estimation (for word recognition) etc. Following section presents the brief summary of these algorithms which are developed but yet to apply. Author suggests their application for future work.

4.1 16 - Direction Vector Encoding

Although the current recognition system uses 8-direction vector encoding, however, algorithm for 16-direction vector encoding has also been developed and tested. But, due to time constrains, experiments could not be conducted with this scheme. Figure 3 shows a comparison between the results of the 8-direction vector encoding and 16-direction vector encoding. It is clear that increase in the number of coding vectors resulted in change in variability and fidelity of the encoding. Varying the directions of the coding vectors may allow compensating for some writing style phenomena, like writing slant (Powelka, 1995).

For sixteen directions encoding the following measures have been taken: Horizontal direction along x-axis (at angle 0) is considered as 1. Directions at angles 22,45,67,90, 112, 135, 157,180, 202, 225, 247, 270, 292, 315, 337 are coded as 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 and 16 respectively. Table 2 shows the direction code with corresponding values of angles in degree'θ' and in radian.

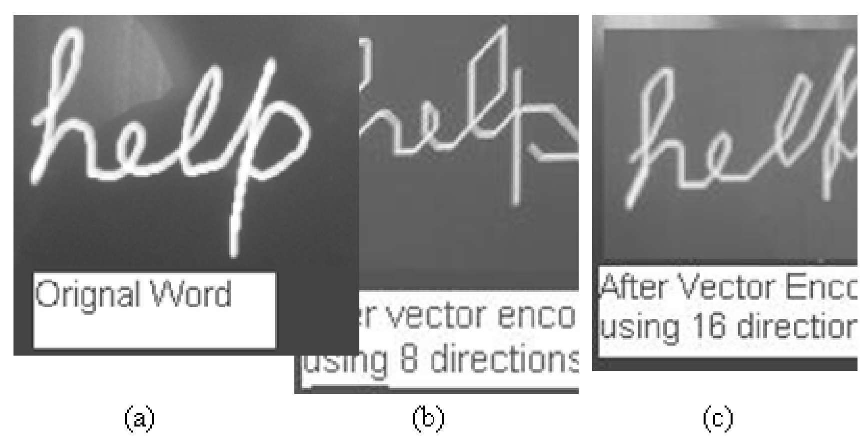

Figure 3 Direction Vector Encoding

Table 2 Direction code with corresponding values of angles in degree'θ' and in radian

DIRECTION CODE	VALUE IN DEGREE 'Θ'	VALUE IN RADIAN	DIRECTION CODE	VALUE IN DEGREE 'Θ'	VALUE IN RADIAN
1	0	0	9	180	0
	11	0.192		191	0192
2	22	0.384	10	202	0.384
	34	0.593		213	0.593
3	45	0.785	11	225	0.785
	56	0.977		236	0.977
4	67	1.169	12	247	1.169
	79	1.379		258	1.379
5	90	1.571	13	270	1.571
	101	1.379		281	1.379
6	112	1.169	14	292	1.169
	123	0.977		303	0.977
7	135	0.785	15	315	0.785
	146	0.593		326	0.593
8	157	0.384	16	337	0.384
	169	0192		349	0192

4.2 Deslanting

Deviation between the principle axis of characters and the vertical axis is known as the handwriting slant (Guerfali and Plamondon, 93). The slant is a very common phenomenon in human handwriting. Humans deal with it without much problem, however it makes the machine cursive script recognition considerably more difficult. Parts of letters which are expected to be vertical become diagonal. This can make their detection prone to error and sometimes impossible. The direction and degree of the writing slant can also vary greatly. For a system using the vector direction encoding it means that any part of a letter that is expected to be vertical (and written from top to bottom) can in fact be encoded by three different directions. This results in increased ambiguity in the recognition process and hence worse results.

The aim of the deslanting process is to detect and compensate for the writing slant in order to reduce the writing variability and improve the recognition. The algorithm given, compensates the writing slant by altering the way it "sees" the input data. This has the advantage of leaving the original data (and all the writing features that can be derived from it) intact. The process consists of two stages:

- slant estimation; where the system evaluates how big the writing slant is. This is performed separately for each word.
- transformation of the coding vector directions; where the system adjusts the way it "sees" the data to perceive it as non-slanted.

Transforming the encoding instead of the data makes it possible to apply the slant compensation transparently to the existing recognizer (Powelka, 1995).

4.2.1 Slant Estimation By Vector Encoding

Average slant of an English word is easily estimated using the chain code histogram of entire border pixels. The estimator is given by (Ding et al., 2000):

$$\theta = \tan^{-1} \frac{(n_1 - n_3)}{(n_1 + n_2 + n_3)} \quad \text{.............................}(1)$$

where n_i is the number of chain elements at an angle of i x 45. (/ or | or \). Shear transformation (2) is then applied to correct the slant, where (x, y) and (x', y') are the coordinates of before and after the transformation respectively.

$$x' = x + y \tan \theta \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$
$$y' = y \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

Figure 4 shows that the slant of a vector code segment is calculated by (1). In this example, $n_1 = 3$, $n_2 = 3$ and $n_3 = 1$. Similarly, the average slant of a whole word is also estimated by (1).

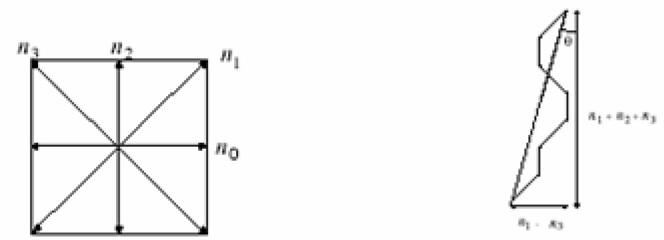

Figure 4 Average slant of a vector code sequence

Figure 5 shows the result of the slant removal algorithm given above.

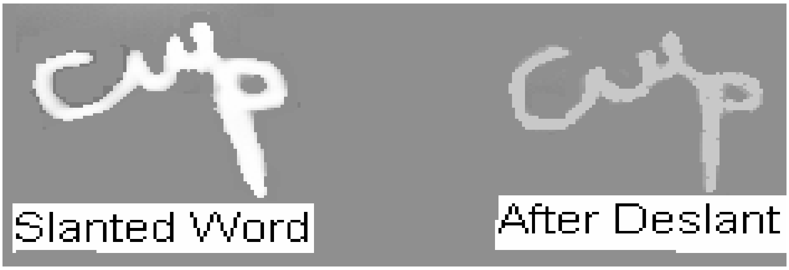

Figure 5 Deslanting

4.3 Middle Zone Estimation

Zoning information is defined as the clear boundaries between tops of the middle zone letters (letters without ascenders or descenders, e.g. "a", "c", "e.") and tops of the upper zone letters (letters with ascenders, e.g. "b", "d", "f.") and between bottoms of the middle zone letters and bottoms of the lower zone letters (letters with descenders, e.g. "g", "j", "p."). Figure 6 presents an example of zone boundaries. The boundaries do not have to be close to letters in any zone. Their purpose is to delimit parts of letters positioned in different zones. Once the zone boundaries are decided, more detailed information can be extracted. In general, letters in particular zones will never ideally line up. Some differences in vertical letter position and size are to be expected. A minimum value of the middle zone width is used to prevent such situation. The minimum value of the middle zone width is set to a quarter of the word height. This is considered a safe assumption, as ideally the middle zone should not be narrower than one third of the word height (Powalka, 1995).

Figure 6 Figure boundaries

If the zoning information is available, it is possible to exclude some of the letter alternatives. This is expected to prevent the segment/recognize approach from considering obviously wrong alternatives and thus improve the recognition. Zoning information is also important for the wholistic word recognition (Powalka et al., 1995).

4.3.1 Histogram Method

This zoning method uses the histogram1 of horizontal word density. Each bar of the histogram represents the number of times an imaginary horizontal line intersects the trace of the pen at a different height within the word. If the trace of the pen is horizontal it is treated as a single crossing. Such a procedure prevents anomalies produced by horizontal bars (e.g. “t” crossings). The obtained histogram is additionally smoothed by averaging the height of each histogram bar with those of its neighbours. Figure 7 presents examples of original and smoothed horizontal word density histograms obtained for various words. The average height of the smoothed histogram is calculated and histogram areas above it are detected. If more than one such area is found the algorithm abandons the zoning estimation process. Otherwise the histogram shape is followed starting from the bars of height equal to the average and towards the appropriate ends of the histogram. The histogram is followed until a region of relatively flat areas is encountered. Such points are, potentially, the middle zone boundaries. If the flat areas are too small, it is assumed that no upper or lower zone is present.

Figure 8 presents example results of the zone location using the described method. Note the error in the word "guessing" induced by the dot. The dot, which belongs to the upper zone, has been included in the middle zone, thus increasing its height. It can be observed that long words produce easier to interpret histograms. Histograms for words "of" and "the" in Figure 8 are quite flat and the zone estimation is prone to error.

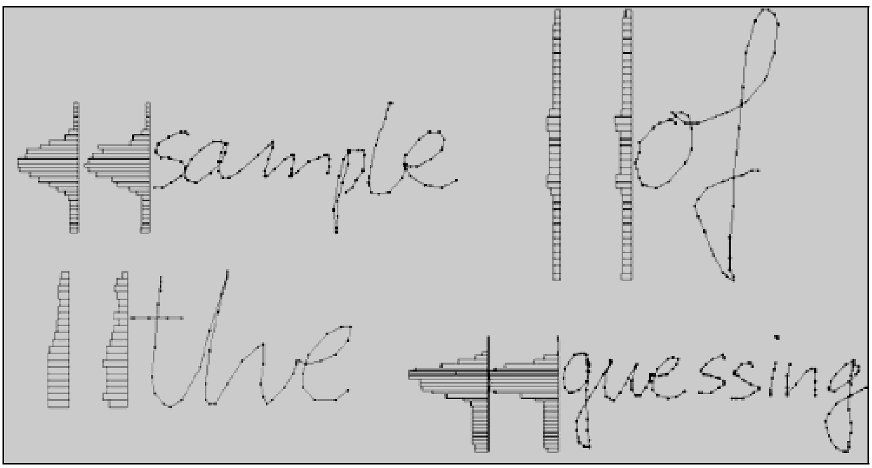

Figure 7 Examples of the horizontal word density histograms. Histograms closer to the words are originals, the further ones are smoothed

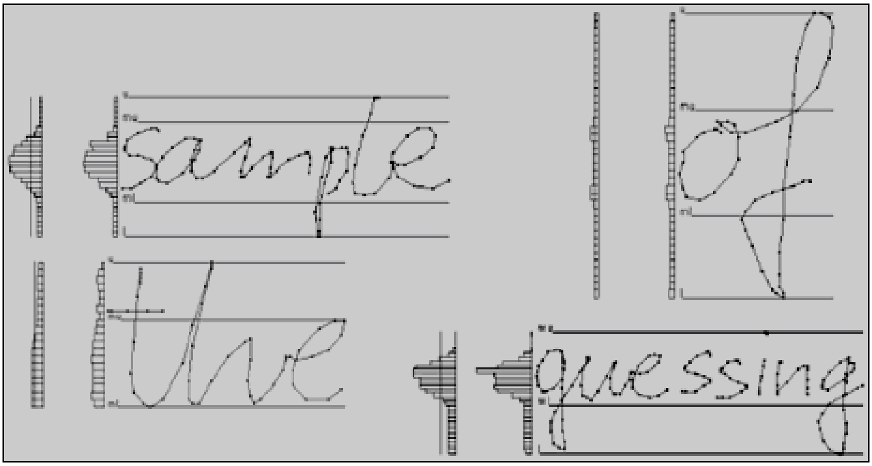

Figure 8 Examples of locating zones using the histograms. Note the error in the word "guessing"

Middle zone estimation based on the horizontal word density histogram is not robust enough, particularly for short words. Hence it appears useful only as a rough measure which can serve for the calibration of the cross product filter.

4.4 More Suggestions

Better classification and matching algorithms can be introduced. These would produce fewer alternatives, thus lowering the disambiguation difficulties. The number of extracted features and robustness of the process can be increased. New features can be used for recognition, providing new recognition experts. Less discriminative features can be used for results verification. Certain features can be used to guide the recognition process by limiting the recognition domain. These include, but are not limited to, zoning information, word length estimation and detection of diacritical marks within a word. Algorithms for detection of such features need to be very robust in order to be suitable for limiting the recognition domain.

While individual letters can be recognized reasonably well, the process of their location within words is very ambiguous and prone to mistakes. Yet correct letter segmentation is vital for recognition approaches based on the segmentation. The segmentation could be performed by using proposed features. Segmentation based recognition approach inherently fails when the segmentation is not possible, for instance for very sloppy or illegible handwriting. However, humans can often cope with such handwriting. Wholistic recognition is free of this limitation and can potentially cope with illegible handwriting. Results obtained in this work are encouraging. Wholistic recognizers introduced in this work were intended as auxiliary recognition mechanisms. Using larger numbers of features could improve the disambiguation and result in stand-alone wholistic recognizers capable of good performance and dealing with sloppy handwriting.

The objective of this work is a user independent recognition system. The inter-subject variability of handwriting is usually higher than the intra-subject variability. This makes the user independent recognition more difficult. On the other hand, handwritten input seems mostly applicable to personal systems with one or few users. User adaptation is expected to greatly improve the recognition performance. However, such a process must be performed transparently, so that it is unobtrusive. One of the possible solutions could use generic information about handwriting and use its own results to adjust to a particular writer's handwriting. A cache of such user specific information could be built. Results obtained using this information could be assigned higher importance than those obtained using general information. Methods for unobtrusive confirmation of recognition results from the user need to be investigated.

A small part of this research concentrated on a wholistic recognition approach. Wholistic word recognition has been observed capable of outperforming the segment and recognize approach, despite the simplicity of the algorithms used. Moreover, the segment and recognize approach appears to have a limited potential. Recognition rates do not increase significantly when more than a certain number of alternatives is taken into consideration (Powalka, 1995). On the other hand, recognition rates provided by wholistic recognizers keep growing with the number of alternatives taken into consideration (Powalka, 1995). Therefore further work on the wholistic recognition approach is suggested. Methods using larger number of features and better classification algorithms need to be investigated.

Naturally, the wholistic recognition is not the ultimate solution. There appears to be no such solution. Wholistic recognizers can have severe disambiguation problems, which grow with the size of the recognition domain. A compromise between the lexicon size and its comprehensiveness can be sought. Limiting of the lexicon size appears restrictive, however humans frequently use a relatively small vocabulary. Work on partitioning vocabulary into various domains and detecting them using a limited amount of input can be pursued.

Another improvement can be obtained through hybrid approaches, integrating wholistic and segmentation based recognition methods. Presented in this research ideas of the use of letter verification for the wholistic recognition could be expanded upon. Analysis of the word shape could identify areas of the word which are particularly important, for instance ascenders and descenders. These areas could become recognition anchors, from which the recognition process spreads around.

Lastly, despite the ambitions for user independent recognition, work on unobtrusive user adaptation is recommended. Humans learn new handwriting styles. This capability could provide a significant improvement in the recognition accuracy for consistent use.

REFERENCE

BERTHOD M., 1982. On-line analysis of cursive writing. Computer Analysis and Perception,v ol. 1. Visual Signals; Ch. IV; pp. 55-81; CRC Press Inc.

BERTILLE, J.-M., EL YACOUBI, M.1993. Global cursive postal code recognition using hidden Markov models, *Proc. of the 1st European Conf. on Postal Technologies (JetPoste'93),* pp.129 138.

BROWN M.K., GANAPATHY S., 1980.Cursive script recognition. *Fifth Int. Conf. on Cybern. and Society*, Boston, pp. 47-51.

BROWN M.K., GANAPATHY S., 1983. Preprocessing techniques for cursive script word recognition. Pattern Recognition, vol. 16, no. 5, pp. 447-458.

DEMPSTER, A. P., N. LAIRD, AND D. B. RUBIN, 1977. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *J. Roy. Stat. Soc., Series B,* vol.39, pp.1–38.

FARAG R.F.H., 1979.Word level recognition of cursive script. *IEEE Transactions on Computers*, vol. 28, pp.172-175.

GUERFALI W., PLAMONDON R., 1993. Normalizing and restoring on-line handwriting. Pattern Recognition, vol. 26, no. 3, pp. 419-431.

GUERFALI W., PLAMONDON R., 1993. Normalizing and restoring on-line handwriting. Pattern Recognition, vol. 26, no. 3, pp. 419-431.

HO T.K., HULL J.J., SRIHARI S.N., 1991. Word recognition with multi-level contextual knowledge. *Proc. First Int. Conf. on Document Analysis and Recognition*, pp. 905-915.

HULL J.J., HO T.K., FAVATA J., GOVINDRAJU V., SRIHARI S.N., 1991. Combination of segmentation-based and wholistic handwritten word recognition algorithms. *Second Int. Workshop on Frontiers in Handwriting Recognition*, Bonas, France, pp. 229-240.

KOERICH, L., 2002. *Large vocabulary off-line handwritten word recognition*, Ph.D thesis, Ecole de Technologie Superieure, Montreal-Canada.

POWALKA R.K., SHERKAT N., EVETT L.J., WHITROW R.J., 1994. Dynamic cursive script recognition. A hybrid approach. Advances in Handwriting and Drawing: A multidisciplinary approach, Faure C., Keuss P., Lorette G., Vinter Eds. A., pp. 137-154, Europia, 1994, ISBN 2-909285-02-2.

POWALKA R.K., SHERKAT N., WHITROW R.J., 1994. Feature extraction: on the importance of zoning information in cursive script recognition. *Progress in Image Analysis and Processing III*, S. Impedovo Ed., pp. 342-349, World Scientific, ISBN 981-02-1552-5.

POWALKA R.K., SHERKAT N., WHITROW R.J., 1994. The use of word shape information for cursive script recognition. *Fourth Int. Workshop.*

POWELKA, R.K., 1995. *An algorithm toolbox for online cursive script recognition*. Ph.DThesis, The Nottingham Trent University

RABINER, L. R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE,* vol. 77, no. 2, pp.257-286.

RABINER, L., JUANG, B., 1993. Fundamentals of Speech Recognition, *Prentice Hall Signal Processing Series*, Englewood Cliffs.

SCOTT D. CONNEL, 2000.*Online handwriting recognition using multiple pattern class models*, Ph.D thesis, Michigan State University, USA.

SIMON GUNTER, HORST BUNKE, 2004. Optimization of Weights in a Multiple Classifier Handwritten Word Recognition System Using Genetic Algorithm, *Electronic Letters on Computer Vision and Image Analysis* vol 3, no.1, pp.25-41.

SIMON J.-C., 1992. Off-line cursive word recognition. *Proc. of the IEEE*, vol. 80, no. 7, pp. 1150-1161.