Writer Independent Online Handwriting Recognition for Ethiopic Characters

By

Daniel Negussie

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES OF ADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTERS OF SCIENCE IN COMPUTER SCIENCE

JUNE 2006

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
FACULTY OF INFORMATICS
DEPARTMENT OF COMPUTER SCIENCE

Writer Independent Online Handwriting Recognition for Ethiopic Characters

By

Daniel Negussie

Name and Signature of Members of the Examining Board

1. Dr. Solomon Atnafu, Advisor _____

2. _____

3. _____

ACKNOWLEDGEMENTS

This thesis would not have bee possible (quite literally) without the contribution of many individuals. I extend my gratitude to my advisor Dr. Solomon Atnafu for his helpful feedback and openness to new ideas which was an enabling factor in initiating this project. I would also like to thank Estifanos Gashaw and Abnet Shimelis for their expertise and encouragement with the development of the experiments.

I wish to thank my parents for their unwavering moral support throughout my university experience. A special thank you goes to all the candidates whose handwritten samples appear in the experiments in chapter five.

Table of Contents

List of figures

List of Tables

CHAPTER ONE: Introduction

Since the conception of the first alphabet, handwriting has been a medium of communication. As the literacy rate in most societies improved, handwriting has played a major role in technological advancement, keeping historical records and also as a persistent means of communication [8]. With the advancement of technology more and more technical barriers have been broken. The advent of computers was a great enhancement to mankind's everyday life which also revolutionized writing systems. In addition to the automated writing systems, various technologies like foldable keyboards, virtual keyboards and speech recognition are some of the methods implemented so far. However these methods have encountered challenges that have made them ineffective at times. Both virtual and real-life keyboards have introduced stress related ailments like Carpal Tunnel Syndrome [4]. Additionally, these keyboard technologies are difficult to make use of when implemented in small cramped spaces. Speech recognition is plagued by environmental noise pollution. In order for this technology to function one needs a reasonably quite arena.

Through time, state of the art innovations led to the miniaturization of computing devices. The integration of communication technology and computing has opened the door to everyday use gadgets like the smart phone and PDAs. The pervasive nature of small handheld computing devices is spear heading a new movement in information technology. Small devices like handheld computers; smart phones and PDAs are a few of the gadgets that are making this phenomenon become a reality. In spite of this, handwriting has still prevailed in this day and age of modern technology.

Interaction between human beings and most computing devices employed keyboards and pointing devices like the mouse. However, these input methods are inappropriate when it comes to the application of small devices; mostly because of their size [11]. This necessitates the need for innovative input methods. Handheld computing devices required easier methods of interaction for use. Researchers have come up with yet another means of interaction, handwriting recognition [8, 10, 14].

Smart Phones, Palmtop computers and PDAs utilize a stylus as one of their main input devices. The stylus is used as both a pointing device and also for text entry [17]. Handwriting Recognition systems (HWR) with PDAs, comprises of the software component that facilitates data entry, recognition and interpretation [10].

Handwriting recognition can be broadly classified into two groups: online recognition and offline recognition. Online handwriting recognition makes use of pressure put upon an electrostatic-sensitive writing surface upon which the user forms handwriting with the stylus. Online recognition system considers samples of the movement of the pen-tip, the coordinates of the sampled points, and information on pen-up and pen-down states [8, 10, and 14]. On the other hand offline handwriting recognition utilizes the handwriting image after completion of the handwriting process [15, 12]. This type of handwriting recognition utilizes a scanner as an input to get the handwriting image. As a result it lacks the temporal input sequence information provided directly by the user. On-line data, in general, is more compact compared to off-line data because of the different dimensionalities in representation. The difference in the data size results in substantial difference in the processing time [12].

Another taxonomy in handwriting recognition is the classes of writer-independent and writer-dependent systems. Writer-independence means that the system can handle the idiosyncrasies of multiple individual writing styles, and a writer-dependent system is trained and optimized to recognize an individual's writing [8].

Handwriting recognitions systems are language specific. Both online and offline handwriting recognition system accuracy rates have been progressively improving for Latin based and other scripts. However, when it comes to the case of Ethiopic scripts very few researches have been conducted in this field. We will address these few researches that have shed some light for our work especially ('Online Handwriting Recognition for Ethiopic Characters by Abenet Shimeles [14]).

This thesis will explore various approaches and technologies, to design and develop an online *writer independent* handwriting recognition system for Ethiopic characters.

1.1 Motivation

Handwriting is a natural way of putting information in legible form to be shared with readers. The scope and importance of handwriting is not all together out-shined with the creation of very sophisticated digital computers with facilitated input methods. In addition, for the new trend of small form factor computers and devices used for mobile computing, carrying a keyboard, even in miniaturized form, is becoming less and less of an option. It is particularly inconvenient to have keyboards in situations where one only has the need to jot down short notes. Another application is as a more natural and easier-to-use interface to the tasks involving complex formatting, like entering and editing equations, and drawing sketches and diagrams [8].

In Ethiopia, the creation of this system will enable individuals with poor English language and typing skills to have access to information technology regardless of their limited knowledge. Moreover, individuals that are computer literate still note flaws in conventional data entry methods that utilize keyboards and keypads. Therefore, individuals that use this system will be able to exercise the convenience of a much more facilitated data entry method in their native language. In addition, this will also be highly beneficial for the circulation of information amongst individuals enabling knowledge and information transfer an easier task.

The current use of PDAs and other hand held devices in Ethiopia are not that common even though they are becoming widely available to most people in other countries. One of the technical reasons is that they are not suited for local languages. The goal of this research is to facilitate the localization of the online handwriting recognition system feature of handheld devices so that Ethiopians can benefit from this technology.

Applications of Writer Independent Systems

The application of writer dependent systems has been the favorite among handwriting recognition systems mainly due to their high accuracy rate [17]. Nevertheless, their applications are limited for personal use.

On the other hand, writer independent systems are not restrained with the requirement of data training. Through time, the applications of writer independent systems have grown in many fields three of which are described in the points that follow:

- Pen based computers – This famous area of application refers to the recognition of handwritten messages to interact with pen computing platforms [12].
- Signature verifiers – This application deals with authenticating a well-learned handwriting to identity of a person's signature. Signature verifiers require the extraction of writer-specific information from a signature signal irrespective of the handwritten content [12].
- Developmental tools – At the educational level writer independent handwriting recognition can be used to teach handwriting to children to help them achieve speed and flexibility. It can also be used as a rehabilitation tool to revive patients that have suffered from hypertensive strokes. During therapy writer independent online handwriting recognition systems can be used to motivate patients to revive their motor neuron skills. It is used to help them communicate with minimum handwriting abilities [12].

In Ethiopia, there is an increasing requirement among Ethiopian professionals for an alternative input device that doesn't require the cumbersome keyboard entry method for Ethiopic script. It is known that the Ethiopic script is used as the official mode of written communication in many parts of Ethiopia. It has been one of the working scripts throughout modern times. Hence the importance of writer independent systems as an alternative input method among institutions may help to share systems and information.

A writer independent handwriting recognition system requires the least learning curve compared to other systems provided that a reasonable accuracy rate is achieved. Writer independent systems have also been identified as the basis to build writer dependent systems [10]. Furthermore, adaptive character recognition systems have been implemented [10]. A writer independent system differs from a writer dependent one in that it caters to a variety of handwriting styles. A writer independent system designed for Latin based characters has to recognize multiple styles of handwriting such as cursive, printed and a mix of both upper and lower case characters. In the case of a writer independent system for the Ethiopic script, the absence of cursive, printed and a mix of both upper and lower case characters proves beneficial regarding the construction of the system. The only obstacle of handwriting variety this system may face is the usual difference noted in individual writing styles.

1.2 Objective

The main objective of this work is to investigate analyze and design a writer-independent online Ethiopic handwriting recognition system for Ethiopic characters.

Specific Objectives:

To meet the general objective, the following specific objectives will be accomplished in this research.

- Study the characteristics of the Ethiopian alphabet and the various types of handwriting styles with respect to online handwriting recognition.
- Assess the different techniques for preprocessing, segmentation, feature extraction and classification for writer-independent online handwritten recognition.
- Study the techniques of online handwriting recognition algorithms in relation to their appropriateness for Ethiopic online handwriting recognition and propose a technique.
- Develop or adopt an algorithm for writer-independent online handwriting recognition systems for Ethiopic script.
- Develop a prototype to demonstrate that the proposed algorithm works for Ethiopic online writer-independent handwriting recognition.

1.3 Scope and Limitation

A handwriting recognition system is quite complex and involves addressing a lot of issues and problems. Due to the time limitation imposed on this research we are forced to put some boundaries.

- Every attempt to cover a wide range of handwriting styles will be made in this research.
- This research work deals with the development of a style-tolerant handwriting recognition system. Since the number of writing styles and the number of Ethiopic characters is very high, it will be difficult to implement a complete writer-independent recognition system.
- Only character level recognition of the 1st order Ethiopic characters will be considered.

1.4 Organization of the thesis

This document contains a total of seven chapters. Chapter two, deals with general concepts and categories of online handwriting recognition. In the third chapter, related works and various methods implemented in the studies are assessed. In addition, these studies are assessed for possible similarity and applicability for the Ethiopic language script. The fourth chapter presents a brief perspective on Ethiopic characters for the purpose of designing the recognition engine. The design of the writer independent online handwriting recognition system for the first order Ethiopic characters will be presented in the fifth chapter. In this chapter a range of steps and the developed algorithms for the various activities that are involved in the recognition system are detailed. Chapter six presents the experimental results by incorporating discussion about the results attained. Finally, the conclusion and future works are presented in the last chapter.

CHAPTER TWO: Online Handwriting Recognition

Handwriting Recognition is the task of transcribing a language message represented in a spatial form of graphical marks, into a computer text [8]. Studies in this field of pattern recognition have been on going for more than four decades. Nevertheless, various applications exist that necessitate this ever continuing research in search of better, more robust and reliable recognition systems. One such application, handwriting interpretation, deals with the task of determining the most likely meaning of a sample of handwriting [8]. This can be observed in sorting mailing addresses from an envelope, and sorting cheques in the bank. Handwriting verification is another application that determines whether a particular handwriting belongs to a specific writer or not.

2.1 Categories of handwriting recognition

Handwriting recognition can be classified into various categories. At a broader level, handwriting recognition can be broken into *offline* and *online*. These two categories arise from the method of input and the information that is made available to the handwriting recognition system.

2.1.1 Offline Handwriting Recognition Systems

Offline handwriting recognition is the automatic transcription of handwriting, where only the image of the handwriting is available [12]. This hand writing needs to be scanned to the computer for the handwriting recognition system to access it and analyze it consequently. A host of applications of offline handwriting can be envisaged, including document transcription, automatic mail routing, and machine processing of forms, checks, and faxes [12]. A few numbered studies have been conducted in this category of handwriting recognition for the Ethiopic text [16, 5 and 9]. One advantage of offline systems over online systems is that they are immune to the various stroke orders among writers. The

scanned representation of the handwriting stays the same without regard to the sequence of strokes, which is not the case with online systems. This imperviousness helps offline systems handle various different handwriting styles, though not without a cost. In order to handle the variety of handwriting styles offline systems need to employ an extensive range of preprocessing tasks to the input strokes of hand writing.

2.1.2 Online Handwriting Recognition Systems

Online handwriting recognition implements the use of a digital pen or stylus in conjunction with a pressure sensitive writing surface which is also called a tablet digitizer. The tablet detects the writer's movement of the stylus and records discrete X, Y coordinates. Furthermore, it records the state of the pen tip, when the pen is touching the surface and when lifted from the surface. A 'stroke' in online data is defined as a sequence of sampled points from the pen down state to the pen up state of the pen [8]. Application of online handwriting recognition systems consists of a more natural and easier to use interface, as well as a tool for diagnosing and teaching handwriting skills [12]. A minimal effort in the learning curve is observed with this mode of data entry. It can also be observed that the online handwriting signal contains more information on the writing process than the offline signal, especially regarding the temporal order and the dynamic information of the writing process, which has encouraged researchers to come up with higher accuracies compared to offline systems [12].

2.1.2.1 Constrained vs Unconstrained systems

Constrained hand writing systems are those that incorporate restrictions. On the other hand, unconstrained hand writing systems allow writers to use their own individual writing styles. Constrained systems have achieved higher accuracy levels because character separation is greatly simplified and the stroke segmentation issue is non-existent. Furthermore, systems like Graffiti have assigned specific individual strokes for each character of the Latin alphabet to avoid problems in recognition and post processing [25]. Example single strokes used in Graffiti are shown in Figure 2.1.

Figure 2.1 Single stroked samples

Even though, Graffiti does not cater to a wide range of writing styles and requires the writer to adapt to the restrictions imposed by the system to recognize the samples provided, its high accuracy rate has made it dominant in the market.

Nevertheless, with the advent of constrained writing systems a question might be asked: Is conventional handwriting facing extinction?

The theory that people will learn a new way to write the letters of the alphabet to achieve fast, consistent recognition may be true, but as the algorithms and networks to recognize normal handwriting improve, then the need for Graffiti decreases, as happened with the Newton with the transition to version 2.0. [3].

Technology has lead to the detailed study and reinterpretation of handwriting which in turn has lead to the further classification of online handwriting systems into, writer-dependent and writer-independent writing systems. Figure 2.2 further shows the hierarchical classification of handwriting recognition.

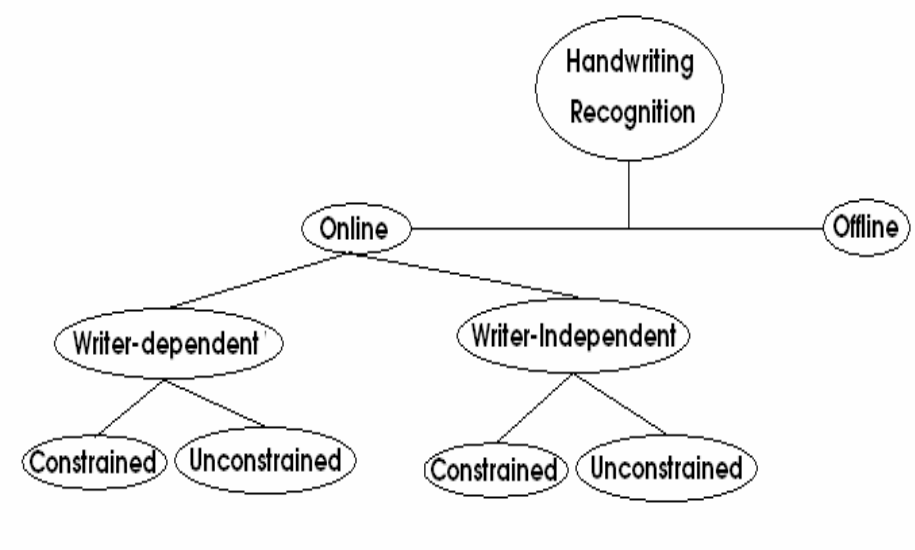

Figure 2.2 Various categories of HR system

2.1.2.2 Writer Independent vs Writer Dependent Handwriting

Another classification of online handwriting recognition systems is based on the amount of data, and the number of users that the system is targeted for. Writer-independence means that the system can handle the variations in multiple people's writing styles, and a writer-dependent system is trained and optimized to recognize a single person's writing [8].

Some handwritten characters have substantial difference in their visual shape due to the different writing styles that exist. For instance, in Figure 2.3 and Figure 2.4 one can see that there are various ways/allographs that represent the same Latin character. Likewise, the same variations hold true for Ethiopic characters as shown in Figures 2.5 and 2.7. This variety in allographs coupled with different writers makes the task of designing a writer independent handwriting recognition system more challenging.

Figure 2.3 Two allographs for the letter "a"

Figure 2.4 Three allographs for the letter "x"

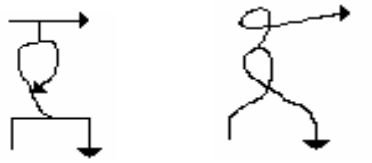

Figure 2.5 Two allographs for the letter "ℓ"

Figure 2.7 Two allographs for the letter "ω"

On the other hand, writer dependent systems deal with relatively lower handwriting variability. This leads to a higher accuracy in the developed writer dependent recognition systems. Nevertheless, a shortcoming of a writer dependent system is that such a system may encounter difficulties in handling variations of handwriting from the single individual. Hence, this may infer that a writer dependent system may present certain amount of constraints which may make it similar to a constrained handwriting recognition system. Alternatively, there is a considerable reduction of constraints in a writer independent system because writers are afforded flexibility with handwriting style variations.

2.1.2.3 Online Handwriting Recognition Systems with dictionaries/lexicon

The recognition of individual handwritten characters can very easily be ambiguous to the human eye. Most handwriting recognition researches have evolved from the study of isolated character recognition towards the recognition of words and sentences. Handwritten word recognition is quite challenging as characters may overlap and some characters within a word may be vague. Neighboring characters may shade some light onto the identity of these ambiguous characters, due to the context or meaning of the word that is formed as a whole. An over-reliance on the potential contribution from the discriminative power of isolated level character recognizer is a contributing factor to this problem. Nevertheless, it is now being realized that the ambiguities encountered during the recognition process are better and more naturally resolved by drawing relevant information from the context rather than trying to put the discriminative capacity of the character

recognizer to the limit. Underestimating the complexity of the string level recognition is responsible for hindering in-depth efforts to merge the research of word and character recognition [8].

No doubt, the character recognizer indeed plays an important role in the process, but more orchestrated and higher level integration of diverse information from the rest of the system is in strong demand to accomplish higher performance[8].

2.2 Conclusion

In this chapter various categories of online handwriting recognition systems have been identified, such as writer dependent and writer independent online handwriting recognition systems. The ability to cater to variations in writing styles makes a system more versatile and readily usable without requiring the user to learn or adapt to a new style of writing.

Based on these various classifications and their corresponding advantages and characteristics, in the study of this paper, it is the aim to develop a prototype of a writer independent, unconstrained online handwriting recognition system for isolated first order Ethiopic characters.

CHAPTER THREE: Related works

Several years ago, people who used computers did so with the understanding that their freedom regarding language medium and information input alternatives were restricted to keyboards, mouse and the like. Today, alternative input methods catering to a multitude of characters other than just those of Latin-character-based languages are surfacing. This in turn is helping to bridge the gaps between societies that speak a variety of languages formed from different character sets.

This chapter will cover a concise review on some of the handwriting recognition systems being developed for various languages and will also try to relate their applicability for Ethiopic script. This being noted, this chapter has primarily ventured into dynamic time warping (DTW) systems that have been applied on Latin and Tamil (Asian/India) characters. In the first two reviews presented the application of DTW has been tested on both writer dependent and writer independent scenarios. Subsequent to this, a review covering the attempts made for Japanese and Chinese languages is addressed. Finally, the last section of this chapter is comprised of the pioneer research conducted by Abenet Shimeles for a writer dependent online handwriting recognition system that shows a new approach for recognizing Ethiopic characters online.

3.1 Dynamic Time Warping (DTW) handwriting recognition systems for non-Ethiopic characters

3.1.1 Dynamic Time Warping for Latin based characters

The objectives of this work were set based on Latin characters and Arabic numbers. This work tries to address the usability of DTW in a Multiple Classifier System. Furthermore, the issue of DTW's intuitiveness to humans when compared with other systems is dealt with. This paper asserts that the system developed compares handwritten samples and classifies them in a similar fashion to human beings.

This work was most interesting because it set the direction of this research to find the answer to the following question.

How well does DTW (Dynamic Time Warping) classify handwritten characters & how does this compare to other systems?

Data Collection and Preprocessing

The UNIPEN database was used to create the prototype sets for the system. The character classifier was limited to 72 Latin characters namely isolated digits, lowercase letters & uppercase letters. These three separate categories were used independently for training and testing. The dataset had been cleaned to eliminate labeling errors, segmentation errors and case errors (mixing up lowercase with upper case characters).Various tools for visualization and a human expert were used to finalize this step.

Classification algorithm

DTW can distort the time axis by compressing & expanding it at places. DTW has proved to be a useful technique in speech recognition, gesture recognition, robotics, and data mining as well as in handwriting recognition.

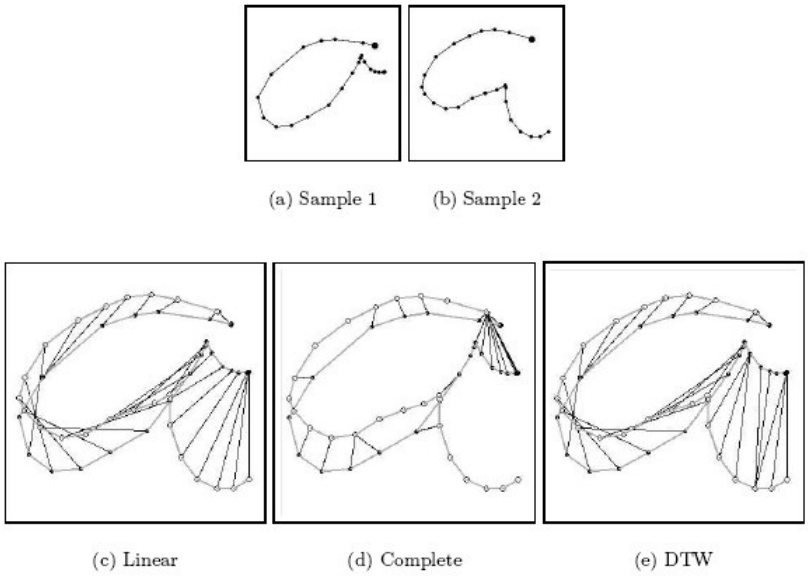

Figure 3.1 Depiction of Sample 1 and Sample 2 in different matching algorithms [10]

Figure 3.1 gives a visual perspective of three matching algorithms namely, linear, complete and DTW matching. DTW has four conditions that give it a significant difference from the other matching algorithms of which three of them are optional. The one condition that is

obligatory is called continuity condition. This condition states that the i^{th} points of the first curve and the j^{th} points of the second curve can be matched if

$$\frac{N_2}{N_1}i - cN_2 \leq j \leq \frac{N_2}{N_1}i + cN_2$$

where N_1 is the number of points on first curve, N_2 is the number of points on the second curve, "i" is a point on curve 1 and "j" is a point on curve 2. The constant c determines the amount that the matching is allowed to differ from linear matching ($0 \leq c \leq 1$). The other three conditions are boundary condition, monotonicity condition, and pen-up / pen-down condition. The boundary condition obliges that the first points of the curves match and that the same holds for the last points of the curves, while the monotonic condition prevents the matching from "going back in time". The last condition ensures that pen up points of the first curve only match with pen up points of the second curve, while pen down points of the first curve only match with pen down point s of the second curve. Euclidean distance was used to measure the distance between points.

In addition to these conditions a decision algorithm was implemented for the last phase of the classification. Construction of an optimal classifier was achieved and various combinations of options had to be attempted. The best classifier was achieved when C=0.13, boundary condition = on, monotonic condition = off, and pen-up / pen-down condition = on.

A further summary of the description of the characteristics of three matching algorithms is listed in Table 3.1.

Properties exhibited	Linear Matching	Complete Matching	DTW Matching
Requires the two curves being matched to be of equal length	Yes	No	No
Every i^{th} point of the first curve matches with the i^{th} point of the second curve	Yes	No	No
One to One correspondence between points being matched in the two curves	Yes	Yes	No
Can cope with writer speed variation	No	No	Yes
Utilizes the temporal order of the points in the curve	Yes	No	Yes
Each point on the first curve can match with more than one point of the second curve	No	No	Yes
Computationally Expensive (relatively)	No	Yes	No
The Matching of the points is intuitive	Yes	No	Yes

Table 3.1 Characteristics of DTW compared to linear and complete matching algorithms

Prototype Generation

The train set was used for prototype generation, which was subsequently subdivided to create 7 sets of prototypes. Once a selection of similar allographs was made, a hierarchical clustering based on DTW-distance was created. The hierarchical clustering is the grouping of samples of a single character to produce a representative character. Visual inspection by a human handwriting expert confirmed that the results were believable. The clustering algorithm used had a number of options which produced different prototype sets. Two different prototype sets were implemented using two different averaging algorithms (called Resample and Average and "Merge samples").

Results

A validation set of the data set was used to test the performance of the classifier for both prototype sets. In the case of lower case characters the performance of the classifier with the prototype set "Merge samples" was found to be higher than with the prototype "Resample and average" for all characters except the letter "d". Moreover this same trend continued when dealing with uppercase characters, where the superior performance of the "Merge samples" prototype set was observed through the entire range of characters. Results of the performance test on digits also indicate a similar pattern with the exception of number "8". The observed difference of results between the two prototype sets may be due to the prototype generation or the post processing which removed bad prototypes. A comparison of this classifiers performance with others is difficult and unfair due to the different datasets used and the variation of techniques applied. However, the general range of recognition rate of other classifiers lies between 85 and 95 percent, making the classifier implemented in this system comparable.

Rejection

If the classifier had the possibility of rejecting a sample after attempting to classify it, the number of errors made during classification could be reduced. This concept had been implemented for this classifier. One possible drawback of 'rejection' could be the amount of false rejections it might produce. Two certainty values called 'agreeness' and 'rejection list' were implemented in the classifier. "Agreeness" is the number of the remaining prototypes that agree with the nearest prototype, where a high value indicated a correct classification. The second certainty value, 'rejection list', is prototype dependent and more complex than the former.

Data Verification

Verification of data is one application of this classifier. A practical case has been noted in the cleaning of the UNIPEN dataset, where segmentation errors, labeling errors and ambiguous samples have been identified. A combination of classifiers was found to increase the robustness of the system.

Summary

In summary, DTW classified handwritten lowercase and uppercase letters including digits with an accuracy rate of 90.32%, 94.28%, and 97.17% respectively with the 'Merge samples" data set. A practical use of the DTW classifier in an MCS has shown that it can be useful by adding a different angle to the recognition problem. Promising results have been shown for the intuitiveness of DTW.

In this review it has been noted that one of the characteristics of handwriting variation is the difference of speed among writers in which a resolution is found in the DTW algorithm. Based on the promising documented results and the characteristics of DTW, it was the choice for classification in the writer independent online handwriting recognition system for Ethiopic characters. In so doing, various combinations of options and settings will be tested to localize this robust algorithm.

3.1.2 Dynamic Time Warping Applied to the Tamil Characters

This system was developed to create an alternative input device with an accurate handwriting recognition system for the Tamil alphabet.

The Dynamic Time Warping classifier system was used because of its ability to match two curves of unequal length which meant that the need for resampling could be avoided. This was most beneficial as resampling Tamil characters with differences in length either created a loss of vital information or created a computational complexity.
Tamil, which is a south Indian language, is one of the oldest languages in the world.
The complete Tamil alphabet and composite character formations are 247.

Data Collection & Preprocessing

All the data was recorded and stored online. The order in which the points were produced along the x, y and z coordinates were stored accordingly. The z coordinate was needed to indicate whether the pen was on (pen-down) or the pen was off (pen-up). All characters

were normalized by equally translating their center to (0,0) and scaling their RMS (root mean square) radius to one.

For this research each of forty writers wrote ten instances of one hundred and fifty six different Tamil characters that were written in separate boxes, so that no segmentation was necessary.

Classification

Dynamic Time Warping is a technique that matches two Trajectories of hand written characters and calculates a distance from this matching. Given two trajectories $P=(p_1,p_2,\ldots,p_n)$ and $Q=(q_1,q_2,\ldots,q_m)$, two points p_i and q_j that will be classified as points that match if the following three conditions are met.

i. Boundary condition p_i and q_j are both the first of the last points of corresponding trajectories P and Q.
ii. Pen-up/Pen-down → p_i and q_j must either be written with the pen-up or the pen-down.
iii. Continuity condition → Equation 1, where constant 'c' is between 0 and 1 indicating the strictness of the condition, must be satisfied.

$$\frac{M}{N}i - cM \leq j \leq \frac{M}{N}i + cM \qquad \ldots \textit{Equation 1}$$

The algorithm computes the distance between the trajectories of P and Q by finding a path that minimizes the average cumulative distance. The distance from P to Q is defined by the average Euclidean distance between all matches (p_i, q_j).

On this system, the classification of an average sample, using a prototype set containing 1847 samples takes about 1.9 seconds. Thus for the envisaged interactive applications, Dynamic Time Warping is a relatively time consuming technique. However, 70% of the samples with less than 50 coordinates only required less than a second to perform the recognition. In addition to the Dynamic Time Warping algorithm, the classifier was provided with the possibility to reject a classification based upon the following methods.

Two variables were used by the system to judge the certainty of a classification. If this certainty was below a set threshold, the classification would be rejected.

The two variables that were implemented were:

I. Agreement: The certainty value is calculated using a list of the five prototypes that are nearest to the sample. First, the label of the nearest prototype is decided. Then the certainty value is the amount of the other four prototypes that agree with this label. If this certainty is high, it means that the chance that the label of the nearest prototype decided is correct is high. This also means that the chance that an incorrect prototype has ended up being the nearest prototype is small.

II. Rejection distance: For each prototype, a rejection distance is calculated. These rejection distances are calculated by classifying a set of unseen samples and recording the distances for incorrect classifications. This rejection distance is set to the distance between the prototype and the nearest sample for which it caused an incorrect classification which resulted in eventual rejection.

The rejection of the classifier also included the following performance measures:

i. % of unknown samples accepted of which the classification was below the rejection threshold.
ii. % of unknown samples falsely accepted while the classification was incorrect.
iii. % of unknown samples wrongly rejected while the classification was correct.

The writer dependent test

The prototypes and the classifier options were based only on data that was produced by a writer that also produced the train data. The recognition performance for the random selection of ten different writers was recorded. The ten numbers were then averaged resulting in a general recognition performance of the system. The data was divided into three sets. The average performance of the system on these 10 writers was found to be 87.77%.

The writer independent test

The prototypes and the classifier options were based only on data that was produced by writers other than the ones that produced the train data. In addition, the effects and performance of the rejection option were examined. The complete data set was divided into four sets. The Test set, containing 1570 allographs, was offered to the system to test the performance of the system using the optimal prototype set and c-value. The system correctly classified 72.11% of the samples.

Automatic prototype creation

Train set 1, part of the data set, was used for the creation of the prototypes. First the set was divided into one hundred fifty six subsets. For each of the one hundred fifty six subsets the next steps were taken:

a. Distance calculation: A matrix of the DTW-distances between all samples in the subset was calculated. This made sure that the positioning of the prototypes in the feature space would be optional for the testing.
b. Clustering: Using the distance matrix produced in the previous step, all samples from Train set 1 were clustered through hierarchical clustering.
c. Cluster selection: In this step, a number of clusters from the complete cluster structure created in the previous step, was selected.
d. Merging: The members of each cluster that was selected in the previous step were merged into one prototype. An Algorithm based on Learning Vector Quantization and Dynamic Time Warping was used for this.

Rejection test

To test the behavior of the system using the optimal prototype set and c-value, a rejection list was created, and a different rejection threshold was tried to classify the Test set. A list of rejection distances was created. The samples in the Rejection set were offered to the classifier using the optimal prototype set and optimal c-value. For each prototype that was at least once responsible for an incorrect classification, a rejection threshold was set. The created list was used in the next step. The strictness of the rejection was varied by changing the Agreement threshold and the multiplication factor of the rejection distance.

Summary

The outcomes from the experiments described show that Dynamic Time Warping implementation is suitable for the automatic recognition for Tamil handwriting and that when using rejection strategies, the reliability of the classifier can be improved.
Subsequent to the previous paper reviewed, the applicability of DTW in different languages shows it's adaptability in handwriting recognition of different character sets. It was also shown that this algorithm was applicable in both a writer dependent and independent setting. These observations and results further strengthen the applicability of DTW for Ethiopic character recognition.

3.2 Online Recognition of Chinese Characters

This work concentrates on relaxation of the constraints of online handwriting recognition systems for Chinese characters [6].

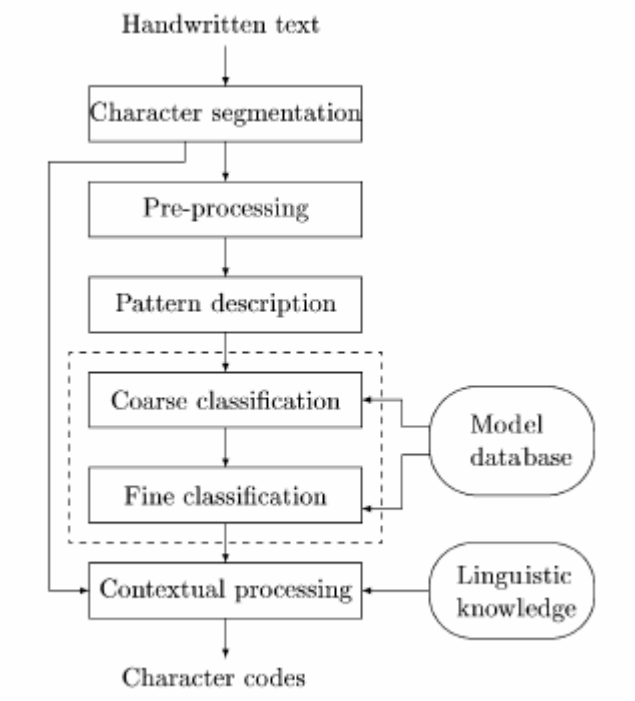

Figure 3.2 Diagram of a practical online Chinese character recognition system (OLCCR)

Chinese characters are quite large in number (in the range of 3755 up to 7773). In addition to this, a Chinese character is an ideograph with mostly straight-line strokes. An ideograph is a character symbolizing a thing without indicating the sounds in its name like a number. Many characters contain relatively independent substructures, called radicals, and some common radicals are shared by different characters. This property can be utilized in recognition to largely reduce the size of reference model database and speed up recognition.

Chinese handwritten scripts are classified into three typical styles: regular script, fluent script, and cursive script. Online Chinese character recognition (OLCCR) can be divided into two categories, namely, structural and statistical methods. Structural methods are based on stroke analysis, whereas statistical methods mainly utilize the holistic shape information.

Preprocessing

The preprocessing tasks of online characters include

- Noise elimination - Noise elimination techniques used in most recognition systems are smoothing, filtering, wild point correction, stroke connection, etc…In our work the data collected for Ethiopic Script incorporated the elimination of pen up points. Since Ethiopic scripts have virtually no cursive form and since the pen up points were so different with in writers, we did not feel that eliminating pen up points would affect the integrity of the shape of the characters.
- Data reduction - Data reduction can be met through the following approaches: equidistance sampling and line approximation. A higher data reduction rate can be achieved by detecting corner points and ends of a stroke trajectory (feature points). The type of data reduction we have tried to incorporate in our research is filtering where points that are normalized end up having the same x, y coordinates were redundant and thus were eliminated.
- Shape normalization - Normalization of character trajectories to a standard size is adopted in almost every HWR system. It is probably the most important of all the preprocessing steps. The options of normalization that are most common are
    - Linear normalization involves the shifting and scaling of stroke point coordinates to be enclosed in a standard box.

- Moment normalization shifts the centroid of the input pattern to the center of the standard box.

In the case of Chinese characters nonlinear normalization has been successfully applied. It reassigns the coordinates of stroke points according to the line density distribution with the aim of equalizing the stroke spacing.

Pattern Representation / Feature Extraction

The representation schemes of the input pattern can be divided into three groups explained below:

1. In the statistical recognition approach we are mainly concerned with the representation of input patterns (feature vectors). The feature vector representation of character patterns enables stroke-order and stroke-number free recognition by mapping the pattern trajectory into a 2D image and extracting so-called offline features.
2. The structural representation is divided into five levels: sampling points, feature points (line segments), stroke codes (HMMs), relational, and hierarchical.
3. In a statistical-structural representation scheme, a character model is described in a string, tree, or graph structure, with the primitives and/or relationships measured probabilistically to better model the shape variations of input patterns. The character-based HMM is stroke-order dependent but multiple models are often generated for the characters with stroke-order and shape variations.

Classification

Chinese characters classification is decomposed into coarse and fine classification, which helps to speed up the recognition. Coarse classification can be achieved by class set partitioning or dynamic candidate selection. Fine classification techniques can be categorized into three groups: structural, probabilistic, and statistical matching.

Structural matching involves matching the input pattern to the structural model of each (candidate) class and the class with the minimum matching distance is taken as the recognition result. Dynamic Programming (DP) matching, stroke correspondence,

relational matching and knowledge based matching are the various structural matching techniques.

Probabilistic matching entails using probabilistic attributes in representing structural models and computing matching distance. This helps improve the tolerance of shape deformations.

Statistical matching methods include the multiple similarity method, the subspace method, and the modified quadratic discriminate function (MQDF). These quadratic classifiers yield high accuracies, but are expensive regarding storage and computation.

Model Learning

The quality of the model database influences the recognition performance. It is noteworthy that many previous works avoided the problem of model learning. Instead, they built the structural models manually using prior knowledge or used carefully written character patterns as prototypes. Mean prototype learning, HMM learning, multi-prototype learning, and structured learning are some of the methods employed for model learning in OLCCR.

Summary

In conclusion, OLCCR systems have shown the applicability of preprocessing tasks that were considered in Latin systems with some modifications, such as non-linear normalization. This indicates that similar technologies could be suggested for the case of Ethiopic character recognition systems. More over, similar structural representation schemes were also implemented and shown a promising result both in Latin and OLCCR. A tradeoff between the number of learning samples and recognition accuracy with statistical matching algorithms has been observed. However, a high accuracy rate with a relatively small storage has also shown very good results.

3.3 Developments in Japanese online handwriting recognition compared to techniques in Latin Based handwriting recognition

In this work the important developments in western handwriting recognition were compared to those that done for Japanese character recognition [7].

The Japanese writing system consists of a mixture of three types of characters: Kanji, Hiragana and Katakana. Kanji characters are pictographic-ideographic characters adopted from the Chinese language. Hiragana and Katakana are phonetic alphabets similar to the Latin alphabet. For the purpose of computer processing, all these characters are further divided into two classes. These classes are the Japanese Industrial Standard (JIS) first and second level. The first JIS level contains the most common characters and the second JIS level contains less common characters. Each alphabet contains 47 officially recognized symbols.
Handwriting recognition is divided into three processing steps – preprocessing, classification, and post processing.

Preprocessing & Feature Extraction

The size normalization is most important normalization in Japanese handwriting recognition. Nonlinear normalization uses a method called line density equalization. This method expands dense sectors of the writing box, while sparsely occupied sectors are contracted. This is the type normalization is prevalent in Japanese recognition engines.

Recognition of Japanese characters resembles recognition of Latin based words in terms of complexity. Unlike Latin based western handwriting recognition engines the resampling method is not always implemented in Japanese handwriting recognition, because it can deteriorate Japanese characters.

Classification

The nearest neighbor classifiers are the classifiers in Japanese character recognition. These are classifiers that compute the distance from an unknown input pattern to all reference patterns stored in a database. These types of classifiers are widespread in Japanese character recognition because of the high number of characters. Various distances ranging from simple Euclidean distances to more complex elastic matching techniques have been applied. Elastic matching is usually based on dynamic programming techniques (DP-matching). This was most helpful for our research as dynamic time warping is very similar to the above mentioned procedures.

Dynamic time warping was adopted from the speech domain in Latin based handwriting recognition before it was replaced by hidden Markov models (HMM) technique. Introducing the Hidden Markov Models is not a common technique in Japanese character recognition. One reason is that HMMs are a technique for implicit segmentation, which is integrating segmentation and classification into one process. But segmentation is not the main problem in Japanese character recognition. Another reason is that modeling Japanese characters and training models is not straightforward due to the high complexity of Japanese characters. Therefore an HMM approach of Japanese character recognition based on sub stroke modeling which employed models describing each character as a whole proved to be a solution. In this way a number of HMMs can model a large number of characters.

Considering techniques from artificial Intelligence (AI) had double sided effects because these techniques were often characterized as symbolic, structural, and syntactic or knowledge based methods. Nevertheless, recognition units called detectors are arranged into a recognition network to detect and identify line segments and their relationships, which reported good recognition results, despite the now omnipresent prejudice against AI methods.

Pattern Representation

In Latin based handwriting recognition, application-specific dictionaries facilitate recognition by confining the number of words that the recognizer needs to recognize using

prefixes or suffixes in the word. The search for the best HMM model is based on an efficient tree structure. But due to the complexity of Kanji characters a similar tree structure is utilized in Japanese handwriting recognition to represent single characters, not words. Kanji characters are composed of radicals, which are elementary building blocks that can be characters themselves. Radicals are shared by many different Kanji characters, thus they can be represented by means of a tree structure.
In general, the writing of a radical is consistent for one writer in the sense that the writing of the radical does not change among different kanji characters containing this radical.

Feature Computation

Features are high-level attributes derived from normalized raw data that are used in most handwriting recognition systems and result in generally higher recognition rates compared to recognizers based on normalized raw data. An often-applied technique to handle stroke order variations in Asian characters is the use of histogram features. Histogram features are statistics describing absolute or relative frequencies of feature values in hand written trajectories. The main idea is to count the number of occurrences of a specific online feature value, such as directional change, and disregard the time of its occurrences. Directional features are a specific type of histogram feature used by many online and offline systems today and they play an important role in most modern eastern character recognizers.

Recognition of Japanese characters

Unlike offline recognition Japanese online recognition used to lack appropriate benchmark data to relate to until the advent of widely accepted databases. These databases have since been used for benchmarking Japanese online handwriting character recognizers. The features and results of the two databases worth mentioning are shown below.
The Kuchibue database – This is a database on online Japanese characters that covers 3357 kanji categories from the JIS first and second level. 120 writers donated 11,927 characters to the overall sum of 1,435,440 handwritten Japanese characters. The current state of the art for the Kuchibue database is around 90%.
Nakayosi – This is the second database similar to Kuchibue, containing the data of more than 170 writers and is now also available for benchmark tests.

Altogether Kuchibue and Nakayosi contain more than three million characters written by 283 writers.

Post processing

This is generally understood as improving recognizer output by means of additional information sources, mostly syntactical knowledge (AI) including information about spelling, grammatical structure, vocabulary etc…

The large number of Japanese character set is no major obstacle for spelling correction based on string-matching techniques. These techniques are applied to western and eastern character strings without significant modifications.

Summary

To sum up the most striking difference between online Japanese character recognition and Latin based handwriting recognition lies in the recognition engines themselves. While the mainstream in Latin based handwriting recognition has shifted to HMM classifiers, most Japanese character recognizers adhere to the nearest neighbor classifier as shown above. The drawback of resampling has been addressed due to the observed deterioration of Japanese characters. Moreover, histogram features have been utilized to tackle the stroke order and stroke number variation issue, despite its limited use in Latin recognition systems. Nevertheless, this could be a good feature to incorporate in a writer independent online handwriting recognition system for Ethiopic characters.

3.4 Online Handwriting Recognition for Ethiopic Characters

This system was designed to model a writer-dependent online Ethiopic handwriting recognition system [14].

In this research the characters in the Ethiopic character set was divided into two groups: the basic characters and the non-basic characters. The number of the basic characters was

thirty-three plus one and they were all located in the first column called the 1st order character set.

All the other six columns were considered as non-basic characters and each of them was derived from their basic relative. These characters are referred to as 2nd order, 3rd order and 4th order characters etc….

Data collection

Similar to the work in this research, the data was acquired using software called MovAlyzer. The data was collected using a mouse that was meant to simulate a light pen. This presented inconveniences to writers because it did not allow them the actual flexibility of a pen paper scenario. The collected data was divided into training and testing data sets.

A deviation from this in our work was the use of an actual electronic pen in our data collection to see if the inconvenience mentioned in this review could be avoided.

Preprocessing

The preprocessing steps were as follows:

- noise elimination,
- size normalization,
- filtering and
- resampling

After this a feature extraction module that produced the observation code sequences from the given handwriting pattern was implemented.

Feature Extraction

The feature representation method used in this work is called observation coding. The assignment of the observation code was done to the x and y coordinates. The observation code sequence extraction algorithm assigned the codes listed in Table 3.2 based on the conditions listed.

Condition	Number Code
Beginning of a new stroke	**1**
Increasing	**2**
Decreasing	**3**
Constant (Nearly constant)	**4**
Stroke separator	**5**

Table 3.2 Codification of observation [14]

Different sequences produced different observation codes (see Table 3.2), mainly due to character shape variations. With modifications the algorithm was made to reject observation codes which last for quite a short time. The integration of the length of an observation code helped refine the characteristics that determine the shape of a character. Furthermore, consecutive identical observation codes were identified to simplify the observed sequence by summing up the length of the code and keeping only one observation code. However intolerable variations between sequences have been observed, mainly due to the observation code 4. Despite the importance of this particular observation code number in representing the shape of characters the weight of its contribution to wrong classification made it a candidate to be discarded. However the observation code number 4 would later be reinstituted for detailed matching.

Training

In this research the trainer takes observation code sequences as input and produces a set of reference observation code sequences. After training, two reference files are generated for each character that corresponds to the x and y observation sequences. A character may have more than one pair of reference files, based on the different number of strokes used to write the character. The trainer identifies characters that have similar sequences and outputs an averaged reference code sequence. On the other hand, some characters don't exhibit similarities that could be averaged, hence necessitating individual reference code sequences.

Classification

The coarse classification seen in Figure 3.3 layer produced the five most likely characters, of which the character in the first order is suggested to be the character recognized.

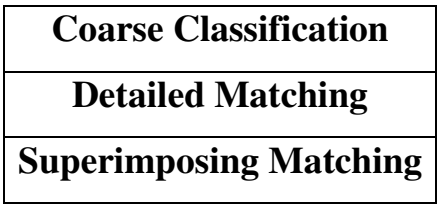

Figure 3.3 Steps of layered classification

This claim is counterchecked and these five characters are passed on to the second layer of the classifier for detailed matching.

Different techniques are employed at the various layers for the recognition process. The speed of the recognizer won't be compromised, because only a small number of characters pass through from one layer to the next whenever there is indecision during recognition, which makes the system efficient. In the coarse classification stage, inter-stroke distances, inter-sequence distances and inter-character distances are calculated between the input character and the trained data. The first task for the recognizer is to determine the number of strokes of the input character, so that it could be compared with a corresponding number of strokes in the reference sequence. Distance will be computed if and only if the number of strokes is equal in both the unknown input and the reference sequences. If the computed inter-character distance between the unknown and the reference sequences is above the accepted amount then refined classification is further required of the top five characters for detailed matching. Otherwise the distance is calculated if there is one to one correspondence in the length of the sequence of the unknown character and the reference sequence. In the case when the sequences are of different length, the shorter one is expanded to create the one to one correspondence.

Detailed matching incorporates occurrences of the observation code 4 in the five candidate reference codes sent to it by the coarse classification module. The distance computation is done in the same way as the coarse classification. Characters that couldn't be classified in the above two modules will be sent to the final classification layer, the superimposing

module. Here, matching is done via inter-point distance calculation. A translation and distance computation process are components of this module.

Experiments and Discussion

The first three instances of each of the 33+1 characters was used for the training data. The rest six instances were used for test data. Two different types of experiment were conducted for this research. The first of these entailed using the training data as test data, while the second one was done by using the allocated test data set (see Table 3.3 and Table 3.4).

Experiment No.	Writer	Data Used	Accuracy
1	Writer A	Training data	99.7%
2	Writer A	Testing Data	99.4%
		Average	99.55%

Table 3.3 Results of Writer A

Experiment No.	Writer	Data Used	Accuracy
1	Writer B	Training data	99.5%
2	Writer B	Testing Data	99%
		Average	99.25%

Table 3.4 Results of Writer B

Summary

Review of this paper has shaded some light into the characteristics of Ethiopic characters, and the various steps implemented to create a writer dependent system. Despite the fact that this paper doesn't show results for a writer independent system, some steps like the preprocessing phase has common algorithms for both writer dependent and independent systems.

In conclusion algorithms for preprocessing activities are designed and implemented in the recognition engine for Ethiopic characters that include extra point noise elimination, size normalization, filtering and resampling algorithms

Furthermore, a three layered classification phase was used to improve the accuracy of the writer dependent system implemented. Deviant to this the matching algorithms applied in our recognition engine have not considered stroke number, which are utilized in the first two layers of the classifier in this review.

3.5 Conclusion

It is fair to say that when approaching the design of the writer independent online handwriting recognition in this work should be the transformation of the raw handwritten data to become recognizable, i.e. the preprocessing phase. The various preprocessing techniques described in this chapter have allowed us to approach these steps carefully.

As shown in section 3.1 of this chapter DTW provides a lot of flexibility for character matching in both Latin based and Tamil scripts. Thus the preprocessing steps in the recognition engine developed in this work [10] have been reduced. The implementations of these steps are further explained in Chapter 5. Since no work has been done on writer-independent online handwriting recognition for Ethiopic characters so far, incorporating the relevant aspects of online handwriting recognitions systems designed for other languages was the key step in the design of the writer independent online handwriting recognition system for Ethiopic script.

CHAPTER FOUR: The Ethiopic Writing System:- An Overview

The Ethiopic writing system originates in Semitic ancestral writing systems as those of European alphabets. Consonantal script developed among Semitic people on the Eastern shore of the Mediterranean some time between 1800-1300 BC [19].
The basic Ethiopic system can be analyzed as thirty-three plus one basic consonant forms with relatively systematic variations to indicate vowels [19]. The Ethiopic writing system is now used on a large scale in the representation of Semitic languages such as Ge'ez, Amharic and Tigrinya [19].

The Ge'ez script is a writing system composed of signs or graphemes denoting consonants with an inherent following vowel [19]. Each symbol represents a consonant + vowel combination, and the symbols are organized in groups of similar symbols on the basis of both the consonant and the vowel [18].

For each consonant, there is a basic or unmarked symbol which represents that consonant followed by a default vowel, called the inherent vowel. For the Ge'ez script, the inherent vowel is /ä/, located in the first column of the Table 4.3. For the other vowels, the basic consonant symbol is modified in relatively consistent ways [22].

Today, Ge'ez is no longer the mother tongue of any living person in Ethiopia. Ge'ez is classified as a sacred language that is still used in the culture of highland Ethiopia as the traditional language of literature and religion. Today, people speak Amharic in their daily life. Amharic is born from original Ge'ez script and has further evolved to include more characters in the character set. Amharic is the second most spoken Semitic language in the world, after Arabic [21].

4.1 The writing system

Unlike other Semitic scripts such as Arabic and Hebrew, Amharic is written from left to right, there are also no systematic variations in the form of the symbol according to its position in the word [20]. The Ethiopic writing system is not classified into upper and lower case letters and has no conventional cursive form which is advantageous to this work. Nevertheless, an unpredictable cursiveness which may produce a lack of clear distinctions in the usage of these characters often arises with hand written Ethiopic characters. In this work the focus is on the basis of thirty-three plus one base symbols of the first order shown in Table 4.1 below.

ሀ	ለ	ሐ	መ	ሠ	ረ	ሰ	ሸ	ቀ	በ	ተ	ቸ	ነ	ኘ	ኀ	አ	ከ
ኸ	ወ	ዐ	ዘ	ዠ	የ	ደ	ጀ	ገ	ጠ	ጨ	ጰ	ጸ	ፀ	ፈ	ፐ	ቨ

Table 4.1 First order characters in the Ethiopic Character Set (1st order)

The thirty-three plus one characters of the 1st order consist of twenty six characters that have mainly been adapted from the Ethiopic script and seven characters that were derived later on from already existing Ethiopic script. These derivations are shown in Table 4.2.

Derived Characters	ሸ	ቸ	ኘ	ኸ	ዠ	ጀ	ጨ	ቨ
Ge'ez Script	ሰ	ተ	ነ	ከ	ዘ	ደ	ጠ	በ

Table 4.2 Derived Characters in the Ethiopic Character Set

These thirty-three plus one symbols have all together formed the first order of the Ethiopic script.

The following 2nd,3rd, 4th, 5th, 6th and 7th orders have been adapted with the use of diacritical marks sometimes called accent marks which are marks added to a letter to alter the pronunciation [20].

For example, Figure 4.1 is an example of the formation of order sequences adapted to incorporate the necessary use of vowels in the Ethiopic script. Nevertheless, the adaptation through time has incorporated structural modifications that are not all together consistent through out the seven orders.

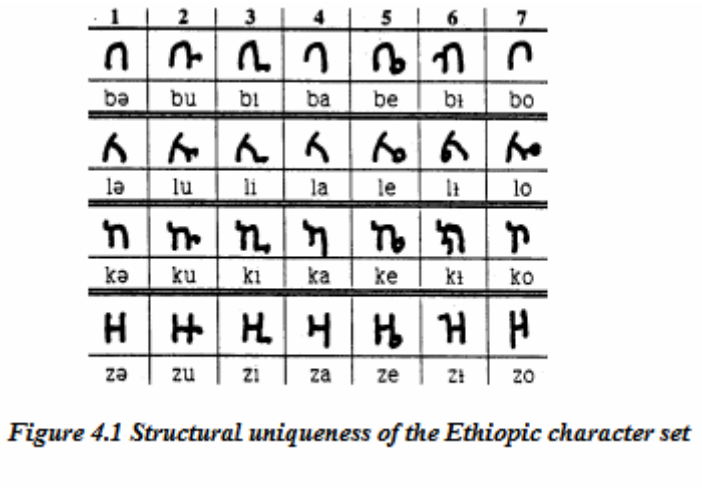

Figure 4.1 Structural uniqueness of the Ethiopic character set

From the observation of the Ethiopic character set otherwise known as the '*Feedel Gebeta',* in Table 4.3*,* it is easy to see that there is a consistent pattern in the shapes given such that: there is considerable regularity of letter shapes, but as already shown some orders are more regular than others. The shapes are most consistent in the 2nd and 3rd order.

1st	2nd	3rd	4th	5th	6th	7th
ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ
ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ
ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ
ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ
ቸ	ቹ	ቺ	ቻ	ቼ	ች	ቾ
ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ
ነ	ኑ	ኒ	ና	ኔ	ን	ኖ
ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ
አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ
ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ
ኸ	ኹ	ኺ	ኻ	ኼ	ኽ	ኾ
ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ
ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ
ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ
ዠ	ዡ	ዢ	ዣ	ዤ	ዥ	ዦ
የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ
ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ
ጀ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ
ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ
ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ
ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ
ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ
ቨ	ቩ	ቪ	ቫ	ቬ	ቭ	ቮ

Table 4.3 The Ethiopic character set 'Feedel Gebeta'

In the 2nd order the structural modification applied is a diacritic horizontal stroke usually attached to the middle of the right side of the first orders character. The exceptions in the 2nd order that do not follow this type of structural modification are ሩ, ፉ and ጠ.

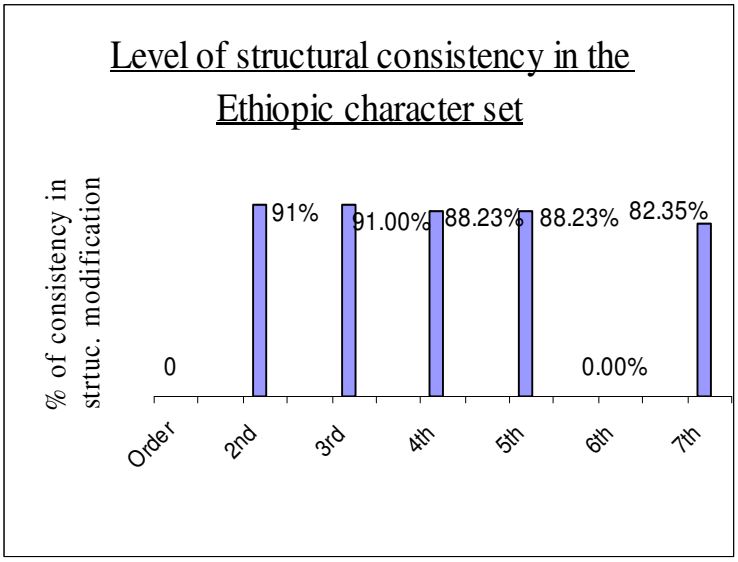

Figure 4.2 Level of Structural consistency in the Ethiopic character set

In the 3rd order, the structural modification applied is a diacritic horizontal stroke usually attached to the bottom of the right side leg of the first orders character. The exceptions to this rule of structural modification in the 3rd order are ሪ, ዩ and ፉ.

The 4th and 5th orders are runners up regarding consistency and this is because they contain the same number of structural modification inconsistencies. In the 4th order the structural modification entails the shortening of the left leg(s). Single legged characters in the 4th order are modified by extending their single legs to the left. The exceptions to this rule in the 4th order are ራ, ኘ, ዦ and ሩ.

Alternatively, the 5th order follows structural modifications where a small circular loop is attached to the right leg. The exceptions to this rule in the 5th order are ሬ, ኔ, ፌ, ቼ and ዬ.

The 6th and 7th order is least consistent, with the greatest number of structural modification patterns, so the form is largely unpredictable although in the entire system some clusters of similar 7th order patterns can be found.

Unfortunately, the system is not quite as regular as the examples in Figure 4.1. Proof of this is shown in Figure 4.3 giving the set of syllables with ገ starting off regularly enough but becoming unpredictable in the 4th, 6th and 7th orders. The ረ set in Figure 4.3 is even more irregular. ወ is most unpredictable with regard to the 2nd and 7th orders as can be seen in Figure 4.3[18].

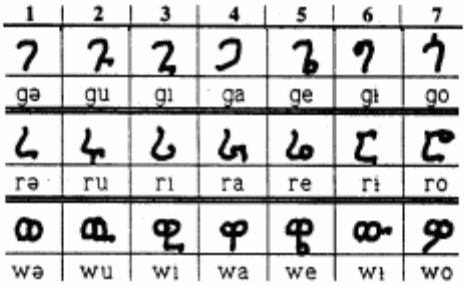

Figure 4.3 Samples in the Ethiopic character set showing unpredictability in structural modifications [18]

4.2 Numerals and Punctuation in the Ethiopic writing system

Unlike the Latin character set alphabet, word boundaries were marked by two vertically placed dots that resembled the colon (**:**). Later letter spaces were conveniently adapted instead of the traditional symbol resembling a colon [18, 19]. Sentence boundaries have always been indicated by using dual colons (**::**). The comma symbol to the present day is still indicated by this symbol (&). The three vertically placed dots which previously indicated the question mark, has been replaced by the modern question mark (**?**). Quotes have followed the French style symbols (**<< … >>**) and parentheses and exclamation marks are the same as those used in the Roman system like ((**…**)) and (**!**) [18, 19].

Although Ethiopic numerals won't be covered in this work the depiction in Table 4.4 below shows how complex they are. Nowadays Latin numerals have been adapted to everyday use and the Ethiopic numerals are only used in religious scriptures. It is our hope that with the coming of a writer independent online handwriting system it may be possible to revive this all but lost Ethiopic number set.

፩	፪	፫	፬	፭	፮	፯	፰	፱	፲
1	2	3	4	5	6	7	8	9	10
፳	፴	፵	፶	፷	፸	፹	፺	፻	፼
20	30	40	50	60	70	80	90	100	1000

Table 4.4 Ethiopic Numerals

4.3 Other approaches to Ethiopic characters

A lot of other avenues have been explored, during the research for this work, to reduce the burden of the total number of Ethiopic characters that the recognition system was expected to recognize. The obvious reason was to achieve higher recognition rates and an interesting opportunity lied in the prospect of reducing *superfluous* characters.

In the process of the adaptation of Ge'ez characters to Amharic and the incorporation of eight new characters, one can see that systematic *superfluous* characters have occurred in the Amharic character set [18, 19]. Namely, three sets of '**ሀ**' '**ሐ**' '**ኀ**', two sets of '**አ**' '**ዐ**', two sets of '**ጸ**' '**ፀ**', and two sets of '**ሠ**' '**ሰ**' are observed. For the sake of clarity, these systematic redundancies are better referred to as *superfluous* characters because their participation and fate in the Amharic Character set used in the Amharic language has still not gotten the proper address in terms of their application in technological advances.

Technology and history come to a cross roads at points like these where scholars from both streams of education have argued on behalf of simplifying the Amharic character set by eliminating or maintaining these *superfluous* characters to make the applicability of technology easier.

Even the literary giant Dr. Hadis Alemayehu, on the preface of his book '**ፍቅር እስከ መቃብር**', forwards the suggestion that society should eliminate these *superfluous* character sets and take only one character set from each redundant group [1]. In the book '**ፍቅር እስከ መቃብር**', Dr. Hadis Alemayehu has chosen to take '**ሀ**' from the group of three sets of '**ሀ**' '**ሐ**' '**ኀ**'. This was done to get the end result which is a more condensed character set that may facilitate efficiency in communication today [1].

Nevertheless these *superfluous* characters have been included in our research because doing otherwise may hinder the future development of this work as some of these *superfluous* characters are more frequently used than we know. Another reason for including them was the fact that doing otherwise would include the elimination of all their orders as well and that would put a level of constraints on the writer because they would not be able to use characters that they may be used to.

4.4 Conclusion

At the onset of this work, designing a writer independent online handwriting recognition system for such a large number of the Ethiopic character set seemed such a cumbersome task. A close study of the Ethiopic character set was necessary to decide how to apply recognition strategies that faced so many hindrances. Obviously, the first difficulty was the total number of Ethiopic characters that the recognition system was expected to recognize. The total number of characters in the Ethiopic character set is two hundred and thirty eight (34 *1st order* x 7 *orders* = 238).

After inspecting the Ethiopic character set from a structural point of view, as shown in Table 4.3 it was apparent that the modifications applied to the 1st order characters to write all other orders were minor structural adjustments known as diacritics [20]. Hence to facilitate the undertaking of this project and reduce the burden of characters needing recognition from the system, the consideration of the first order characters seemed beneficial. Strong justifications for this are;

- The diacritics of the 2nd, 3rd, 4th, 5th, 6th and 7th orders are minor structural adjustments to the major 1st order characters and thus adapting the Dynamic Time Warping algorithm to accommodate for this structural modification will be an easy task.

- Hence, the ground work for the future study of this work will have been done by the time the writer independent online handwriting recognizer is faced with the task of recognizing all other orders.

From the close study of DTW it is clear that it can be classified as being more of a structural approach to handwriting recognition. We are aware that structural approaches have been highly criticized for the difficulties they introduce to the process of recognition. One difficulty is that manually formulating a dependable set of classification rules that can account for a rich range of shape variability is a daunting challenge. Another is a possible brittleness introduced by the summary nature of the representation [8].

This research is some what of a divergence from the above classification of a structural approach because DTW affords flexibility in handwriting styles. The next chapter describes the design for the recognizer of this work was carried out in detail.

CHAPTER FIVE: Writer Independent Online Handwriting Recognition for Ethiopic Characters - The design

The driving force behind the inception of this work was the reality that the use of online handwriting recognition for Ethiopic characters in daily life as an alternative input method is faint if not all together absent. While contributing factors to rely on the Latin alphabet when using computers may be, the availability of Latin based keyboards and the medium of teaching throughout the educational system. It is our strong belief that the use of Ethiopic characters for communicating in daily life can be encouraged greatly by the establishment of a writer independent online handwriting system that can cater to a wide range of users.

There is a growing demand to bridge the gap of technology that exists in the world. Speedy information exchange and advancements in any language medium, contribute to meet this demand and the implementation of a writer independent handwriting recognition engine is one of the tools that contribute to this goal. The possible alternative contributions of a complete writer independent online handwriting system are far more than those discussed in chapter one.

Therefore with these aspects in mind, we set out to study closely the work done by our local and international predecessors. On the Latin character based recognition front the advancements were more than we knew of at the onset of this work. From the review of their work it became apparent that the isolated character recognition entails word and then text recognition as its obvious steps forward. This in turn armed us with the foresight that the future of our writer independent online handwriting recognition system still has difficulties to overcome.

Nevertheless as a model of consideration for future amelioration we have put forward the following design of the first writer independent online handwriting recognition system.

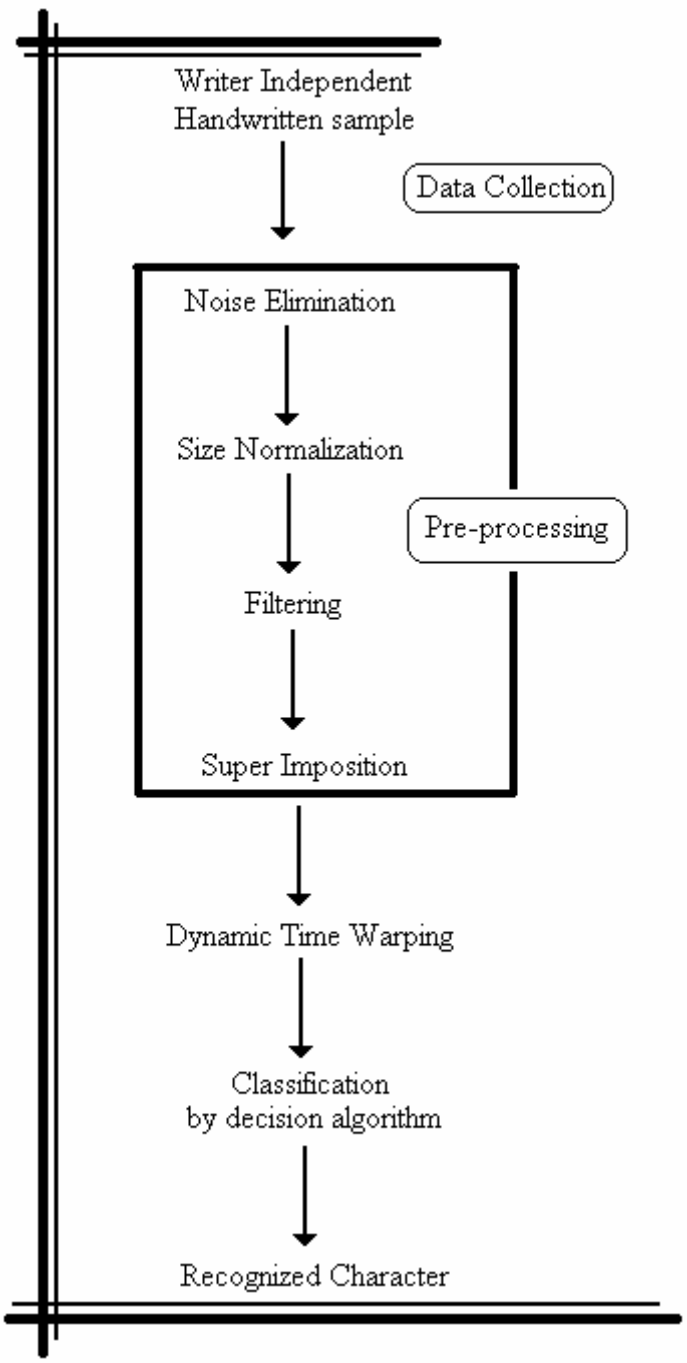

Figure 5.1 Design of Writer Independent Handwriting Recognition for Ethiopic Characters

As illustrated in Figure 5.1 the recognition engine consists of two major modules that handle preprocessing and classification. The various algorithms used to develop the recognition engine are written in C/C++. The rest of this chapter will discuss the choice of various tools and algorithms that make up the writer independent handwriting recognition.

5.1 Design of Online Handwriting Recognition System for Ethiopic Characters

5.1.1 Data Collection

Most researches have utilized a readily accessible data corpus like the UNIPEN project and the TUAT (Tokyo University of Agriculture and Technology) databases. The handwriting samples for these databases also utilized digitizers. However the lack of such data corpuses has been noted as one of the drawbacks for this and previous researches for Ethiopic character recognition. [14]

On the other hand most online handwriting recognition researches conducted have either utilized electronic tablet or digitizers and in some cases small handheld devices for the collection of data. Electronic tablets are further subdivided into two categories: electromagnetic/electrostatic and pressure sensitive. Recent development of digitizer technology has been towards a combination of input and output-digitizer and display-on the same surface [17]. Tablet digitizers are characterized by resolution, accuracy, and sampling rate. A resolution of around 200 points/in and a sampling rate of 100 sample/s are the common requirements that are met by tablets.

Figure 5.2 The Wacom digitizer tablet used for the data collection (model CTE-440 / Graphire 4 A7)

Nevertheless, this research has utilized a WACOM digitizer tablet as seen in Figure 5.2. The Graphire 4 A7 tablet was used in a Windows XP environment. The software component for the collection of handwriting samples was served by NeuroScript MovAlyzer (see Figure 5.3) which consisted of various windows. Consistent with windows explorer the left window was used for navigation. The right window was used for

recording handwriting samples. In this work three values were recorded. Namely the x, y and z coordinate of the sampled point. The x and y coordinates are the rectangular coordinate values set by MovAlyzer, while the z coordinate is the pen pressure points of the digital pen on the Graphire tablet. Inline with researches in online handwriting recognition, this coordinate data is kept in a persistent manner for analysis in the classification phase. A character's sampled points are saved in a text file consisting of the x, y and z coordinates which is also known as the raw data.

The writers were chosen randomly so as no to make the prototype developed restricted to the use of subjects of a certain sector in society. The first four subjects were library members of the Alliance-Ethio Français language learning institute. Among the first four candidates only one was very familiar with a computer environment. The next five candidates were all students attending the computer science masters programs in the Addis Ababa University; hence their familiarity with the computer environment was high. The last candidate was an engineering major at the Addis Ababa University. The rest of the candidates were restaurant customers with a mix of various educational backgrounds and different levels of familiarity to the computer environment. The candidates were comprised of 10 females and 20 males all in the age group range of 21-40.

The data collection was conducted so that each writer could write the 34 first order characters consecutively (**υ - Τ + ñ**). A total of thirty different randomly chosen writers were involved in this experiment. Traditional data entry peripherals like the keyboard and mouse require some training before a user could comfortably start using them. Even though in theory, simulating a pen paper environment with a digital pen and tablet digitizer may not seem to need training, the writers in this research were given a few minutes to familiarize themselves with this novel data entry method. During the data collection process some writers have been noted to write carefully while others realized that the character input using this interface does not resemble the desired entry. Hence each writer was given the choice of accepting the data displayed on the screen before moving on to the next character. This option helps the accuracy of the recognition system as each writer confirmed their satisfaction with each character input.

Writers have commented that the data collection environment presented did simulate the pen paper environment but lacked the visual confirmation that one usually experiences

while writing. Due to this reason the strongest comment among writers has been that with the addition of visual confirmation on the tablet this would be an ideal pen paper simulation.

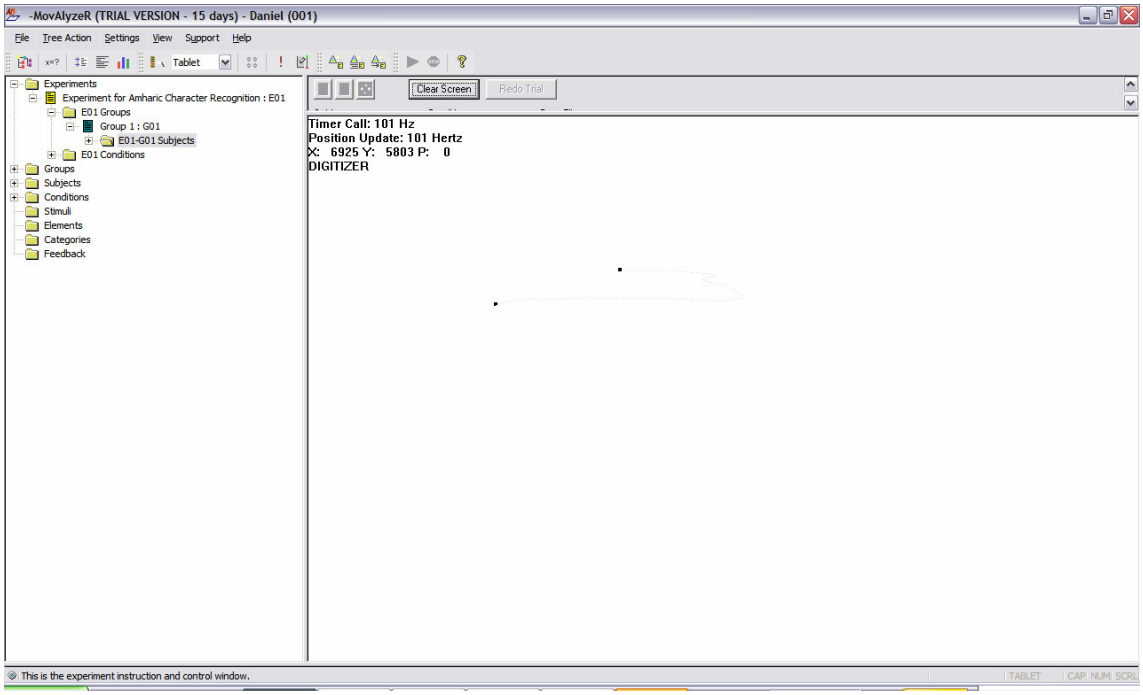

Figure 5.3 Neuroscript Movalyzer

Upon inspection of the collected raw data it was noted that the first candidate had skipped character "ñ". Therefore, the characters written by subject one were only used as prototypes. The character set used in this work is a total of 1019 characters.
The data collection phase was limited only to thirty subjects due to time constraint.

5.1.2 Preprocessing

Much like most online handwriting recognition systems our design also entailed the process of presenting the unknown patterns of points that made up the character written by a random writer to the recognizer. As expected the preprocessing phase enabled the reduction of variations in handwriting styles. No doubt, the same characters written by different users can vary greatly in size, shape and distortion. We also encountered the same writer that wrote in substantially different ways. Therefore it becomes very clear that the first task of our recognition system was to suppress noise and reduce the variability in the raw data for easier and more accurate recognition. Noise is classified as the pen up points

that are recorded after the subject has finished writing a sample character. Various preprocessing algorithms implemented in the development of the writer independent handwriting recognition engine for Ethiopic characters will be discussed next.

Extra pen-up point elimination

The collected raw data was observed to incorporate the three coordinates; x, y and z of which the latter represented the pen pressure points. Pen up points were indicated by a '0' while pen down points were observed to have pen pressure points that varied temporally. A sample character from one of the subjects is shown in Figure 5.4 consisting of 176 sampled points to represent the indicated character '**ñ**'.

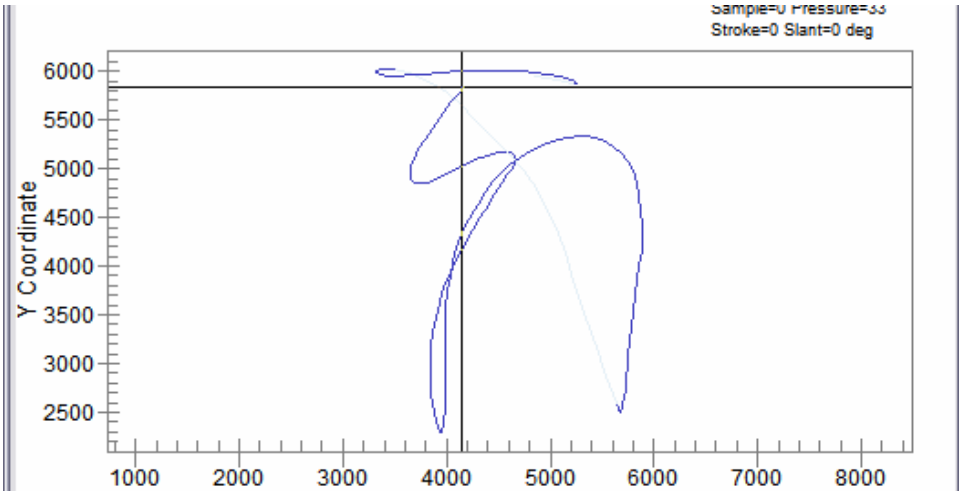

Figure 5.4 Sample character "ñ" from subject 10.

Out of the 176 sampled data points 30 points represented pen-up points that indicated that the character was written by two strokes. These pen-up points were also indicative of the starting and finishing points of the second stroke. These points are unnecessary at the classification stage of this recognition engine and hence were eliminated at the pre-processing phase. This preprocessing task identifies consecutive pen-up points within a character that indicate separation of strokes and replaces them with a single triple (0,0,0).

The algorithm used for this preprocessing task was adapted from Abenet's extra pen-up point elimination. This adapted algorithm not only eliminates noise (extra pen-up point after the character is drawn) but also represents pen-up points encountered between strokes by a single triple (0,0,0).

```
Input: File that contains the raw set of data points sampled for a character
        //index: points to each data point
        //NoOfPts: number of data points collected for a given character
        //OrigX, OrigY, OrigZ: set of x, y and z components of data points

        index ←0
        //read the initial point of the character
              Read x, y, z of a data point from file
              OrigX [index] ← x
              OrigY [index] ← y
              OrigZ [index] ← z

              Do
               if z=0
                 OrigX[index]=0 and OrigY[index]=0
                 index ← index + 1

                 While z=0 and end of file is not reached
                       Read x, y, z of a data point from file
               Else
                 Read x, y, z of a data point from file
                 OrigX [index] ← x
                 OrigY [index] ← y
                 OrigZ [index] ← z
              Until (end of file is reached)
 NoOfPts = i
 index = 0
 While (index < NoOfPts)
        Write OrigX [index] to file
        Write OrigY [index] to file
Write OrigZ [index] to file
```

Figure 5.5 Extra Pen-up point elimination algorithm.[14]

When the algorithm in Figure 5.5 was applied to the character '**ที**' in Figure 5.4 the number of data points was decreased from 176 to 148.

Size Normalization and filtering

The size of a character in this work is defined as the bounding box of the character. Considerable size variation was noted amongst the characters collected. In order to avoid this disparity, a size normalization algorithm was implemented to standardize the size of all characters both in the test set and prototype set. All characters were mapped to a standard 40x60 box [14]. This size normalization algorithm has proven to be successful in the previous research of online handwriting recognition for Ethiopic characters [14]. Due to the process of size normalization identical pairs of pen-down x, y coordinates were observed. These redundant points do not contribute to the classification process and hence were represented by only a single pair of coordinates. This task was integrated into the size normalization algorithm. The algorithm [14] is illustrated in Figure 5.7.

The character '**ñ**' in Figure 5.4 had been reduced to 148 points after eliminating extra pen-up points. After undergoing size normalization and filtering of redundant points this same character ended up having 54 points as illustrated in Figure 5.6 a and Figure 5.6 b. A sizable reduction of the number of points is observed.

```
4153    5808    33
4160    5817    55
4160    5817    77
4154    5807    113
4138    5786    153
4108    5747    219
4063    5687    245
4002    5605    275
3933    5504    297
3860    5394    317
3790    5284    333
3732    5177    361
3688    5079    361
3661    4992    371
3651    4921    367
3661    4870    385
3690    4836    389
3736    4824    389
3803    4835    393
3893    4870    389
        .
        .
```

Figure 5.6 a Datapoints of character 'ñ'

```
65      93      33
64      92      153
63      91      245
62      89      275
61      88      297
60      86      317
59      84      333
58      82      361
57      81      361
60      78      389
64      80      389
66      81      393
70      82      395
72      79      399
71      78      399
        .
        .
        .
```

Figure 5.6 b Datapoints of character 'ñ' after size normalization and filtering

Input: File that contains the set of pen-down points sampled for a character (noise eliminated)

//*MaxX, MinX*: Maximum and minimum x values among the data points sampled for a character

//*MaxY, MinY*: Maximum and minimum y values among the data points sampled for a character

```
//Find the block that encloses the character.
    Compute MaxX,  MinX
    Compute MaxY, MinY
    Width ← MaxX-MinX       //width of the box that encloses the character
    Length ← MaxY-MinY      //Length of the box that encloses the character

//Normalize character and filter redundant point
        Read x, y, z of the first data point from file
            //Translate the data point to a box of size 40 × 60
            newx ← (40 × x)/width
            newy ← (60 × y)/length
            PreviousX ← newx
            PreviousY ← newy

    Do
        Read x, y, z of a data point from file
        if z=0
          write x,y,z to the new file
        else
            newx ← (40 × x)/width
            newy ← (60 × y)/length
            if (newx=PreviousX and newy=PreviousY)
                do nothing
            else
               Write newx, newy,z to file
            PreviousX ← newx
            PreviousY ← newy

            Write newx, newy, z  in a new file
    Until (end of file is reached)
```

Figure 5.7 Size Normalization and filtering algorithm [14]

5.1.3 Classification

In this work the classification process implemented in the handwriting recognition engine is Dynamic Time Warping (DTW). The same characters written by various subjects were observed to consist of a varied number of sampled points.

As stated in the previous section, the character '**ಗೆ**' written by various subjects were observed to remain with 54, 53, 51, etc points after normalization. One of the reasons for this difference in the number of sampled points maybe the variation in the speed of handwriting. It has been noted earlier that the tablet enables sampling at 100 points per second (100Hz). Hence writers that write relatively fast have fewer sample points recorded while those that write slower end up having a lot of sampled points recorded. Another reason for the difference in the number of sampled points of the character may be the shape variation of the character. This difference in the number of sampled points of different samples of the same character is handled by the Dynamic Time Warping (DTW) algorithm which entails the following four conditions.

These conditions differentiate DTW from other matching algorithms.

Continuity condition

The continuity condition of DTW allows flexibility in matching two curves. This flexibility is determined by the constant 'c' (see chapter 3.1.1, 3.1.2 and Figure 5.8).

Boundary condition

The boundary condition restricts matching between the corresponding first and last points of the two curves under comparison. In Figure 5.8 the beginning and ending of curve 1 is matched with that of curve 2.

Monotonicity condition

This condition prevents a point from the first curve from matching with points in the second curve that have already been matched in a previous iteration. In Figure 5.8 '*pi'* of curve 2 is restricted from match with. '*P0'* of curve 1.

Pen-up/Pen-down condition

Pen-down points from the first curve are allowed to match only with other pen-down points of the second curve while pen-up points of the first curve only match with pen-up points of the second curve.

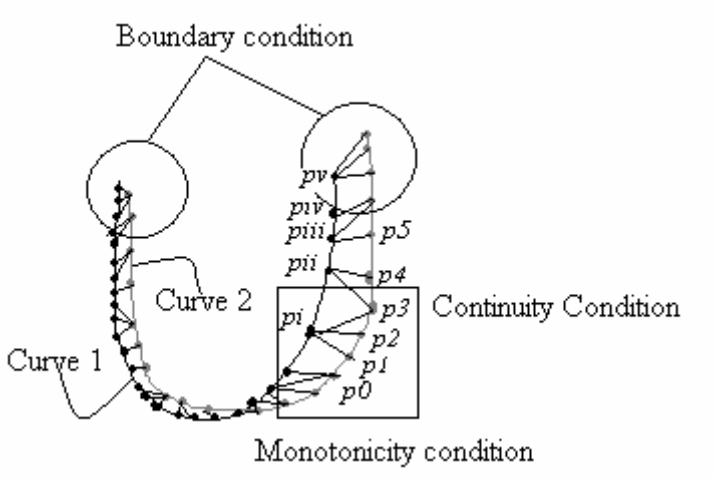

Figure 5.8 Depiction of DTW matching incorporating the three conditions upon the letter 'ሀ'

The conditions of DTW for recognizing first order Ethiopic characters were set as follows:

- Continuity condition ('c' parameter) was set to 0.13.
- Boundary condition was set on.
- Monotonicity condition was set on.
- Pen-up / Pen-down condition was set on.

The best choice for the 'c' parameter was found through a trial and error process. Higher accuracy rates were reported when c was set at 0.13. The boundary condition was set on because Ethiopic characters are written from left to right and hence are better recognized when starting and finishing points are correspondingly matched.
This being noted, monotonicity was unavoidable due to the choice made in the boundary condition. It was necessary that the Euclidean distance calculated between the two characters only concentrated on the pen-down points. Hence, all pen-up points were excluded from the distance calculations. This step makes the classification stroke number independent.

Visualization of the preprocessed characters pointed out the necessity of superimposition for the matching process. Therefore the matching process took place once the character to be recognized was translated to the bounding box of the prototype character. The algorithm [14] for the superimposition is given in Figure 5.9 .

***Input*: Preprocessed data of a reference character and an unknown character**
//Each data point of the unknown character is a triple (x, y, z)

Get *UnknownMaxX, UnknownMinX, UnknownMaxY, UnknownMinY*
//to get the box in which the unknown character is drawn
Get *RefMaxX, RefMinX, RefMaxY, RefMinY*
//to get the box in which the reference character is drawn
ChangeInX ← [(*RefMaxX – UnknownMaxX)+(RefMinX- UnknownMinX*)]/2
ChangeInY ← [(*RefMaxY – UnknownMaxY)+(RefMinY- UnknownMinY*)]/2
i ← 1
Do
    if ($x_i \neq 0$)
        $newx_i \leftarrow x_i$ + ChangeInX
        $newy_i \leftarrow y_i$ + ChangeInY
        $newz_i \leftarrow 0$
    else
        $newx_i \leftarrow 0$
        $newy_i \leftarrow 0$
        $newz_i \leftarrow 0$
    i ← i + 1
Until (i > n) //n is the number of data points that make up the unknown character

Figure 5.9 Superimposition Algorithm [14]

Once the character has been superimposed it is ready to be matched via the DTW matching algorithm. This algorithm takes the superimposed character and computes the distance with characters in the prototype set. As each character is compared with the unknown superimposed character, the distance is computed and sorted for further analysis. This algorithm [10] is given in Figure 5.10.

Input: **Superimposed data of an unknown character and a list of characters in the prototype set**

```
//Each data point of the unknown character is a triple (x, y, z)

        Read the reference character name from the list of prototype set for
         comparison with the query
        Open the unknown character which is known as the query and the
         reference character for comparison
//the matching path is useful for detail analysis of which sampled points are being
         matched and also to ensure accuracy

        Create an empty matching path   //a text file is used
        Total distance =0
        N1 is the no of points for the query character while N2 is the no of points
         in the reference character under comparison

        For every i in N1
              Smallest distance = infinite
              For every j in N2
//the points under comparison are tested whether they are boundary points
                    If (boundary condition=on) AND
                    ((i=0 AND j=0) OR (i=N1 AND j=N2)
                    // here the matching path is updated
                      Add ("i-j" combination to matching path
                    //Note that the distance calculated is the Euclidean distance
                    between the two points under consideration
                      Add distance between points i and j to total distance
                    Else
//if the points under comparison are not boundary points but satisfy the
continuity condition and are also pen down points
                      If (continuity condition between i and j is TRUE) AND (points i
                    and j are pen down points)
                              If (distance between i and j < smallest distance)
                              smallest distance = distance between i and j
         if ("i-j" combination with smallest path is not in the path yet)
                    Add ("i-j" combination to matching path
                    Add distance between i and j to total distance
                    Add smallest distance to total distance

Total distance = total distance/ no of matching path
```

Figure 5.10 Dynamic Time Warping Algorithm [10]

Note that the algorithm keeps track of the matching sampled points between two characters under consideration. Furthermore, the Euclidean distance between the two points considers only the ‘x’ and ‘y’ coordinate values.

5.2 Conclusion

In this system, different components that make up a writer independent online handwriting recognition system have been proposed. Two preprocessing tasks namely extra pen-up point elimination and size normalization have been implemented. Resampling has been used in various online handwriting recognition systems. Nevertheless, resampling has not been incorporated in this work because it eliminates vital information that may be useful for classification. Based on the preprocessed data, a classification algorithm that handles two important aspects has been applied.

These aspects are:

- stroke number variety and
- the difference in the number of sampled points exhibited due to writer speed and shape variation.

This work has addressed drawbacks in a previously conducted research in Ethiopic online handwriting recognition. The recognition engine in [14] heavily depends on the number of strokes used to write a character as a feature for recognition. A writer that has trained the system writing a certain character in a single stroke will have to continue writing that character in a single stroke for the system to recognize it. If this writer changes the number of strokes to write a certain character, then the system will not recognize the character unless the system is trained with the character that required a different number of strokes. Though, stroke number dependence may be considered as a minor limitation in a writer dependent system where such variations in stroke number may be rare because of the number of writers involved, the case is different in a writer independent system. This is because a writer independent system must have stroke independence in order to recognize various written characters from different subjects. Similar characters are written with different number of strokes. The system developed in this work is stroke number independent. Moreover, the distance computation between two curves tolerates the difference in the number of sampled points due to the matching DTW algorithm applied.

Independence to stroke number helps the expansion of the system developed to be expanded to incorporate the second, third, fourth, fifth, sixth and seventh order characters of the Ethiopic character set. Furthermore the Ethiopic numerals, labialization characters and additional characters from Tigrinya and other languages can also be included because DTW is not a character set dependent algorithm. It should be noted that DTW is a structural matching algorithm that only considers the structure of a given character. This justifies the possible expansion to include all other orders, numerals and labialization characters in the Ethiopic language script.

It has not been forgotten that the Ethiopic script contains a rather wide set of characters. This work tests the DTW matching algorithm on the first order Ethiopic characters to show the results obtained from these fundamental characters of the Ethiopic character set. It is hoped that this in turn will shade light on future challenges to achieve a higher accuracy rate on a data corpus that encompasses all the characters in the Ethiopic character set.

The various options and possible directions that may be undertaken in the future are addressed in chapter seven.

CHAPTER SIX: Experiment and Discussion

After collecting the data of ten individuals, the recognition engine described in detail in Chapter five was tested for its accuracy rate. It performance was further fine tuned by modifying the 'c' parameter to produce the results that are presented in this chapter.

6.1 Experimentation & Results

Twenty nine major experiments were carried out. Samples of thirty four first order Ethiopic characters were collected from each candidate that were stored in corresponding text files. A total of nine hundred and eighty six Ethiopic character samples were tested for recognition. Once the collected data had gone through the entire preprocessing phase thirty four first order characters from each subject were checked for recognition against all other samples from the remaining twenty nine subjects. For example, subject two wrote thirty four characters that were stored in sequential order. Then, these characters were tested for recognition against nine hundred eighty five {(34X29)-1} (see chapter 5 *'data collection' page 48*) characters written by all other subjects. The nine hundred eighty five characters used as prototypes did not include any of the thirty four characters written by subject two. Experiments for other subjects were carried out in a similar manner excluding the subject's characters (characters being tested for classification) from the prototype set.

Results of the experiments have been summarized in the tables shown in this chapter. The first column in the tables indicates the Ethiopic character to be recognized in the experiments. The subsequent columns indicate whether the character in question was recognized (based on the shortest distance reported in the experiment).

The distances recorded after the DTW matching process for each character tested against the prototype set is displayed in the form of a text file with distances listed in ascending order. The character being tested is matched with all the characters in the prototype set. An example of a DTW distance list can be seen in Figure 6.1.

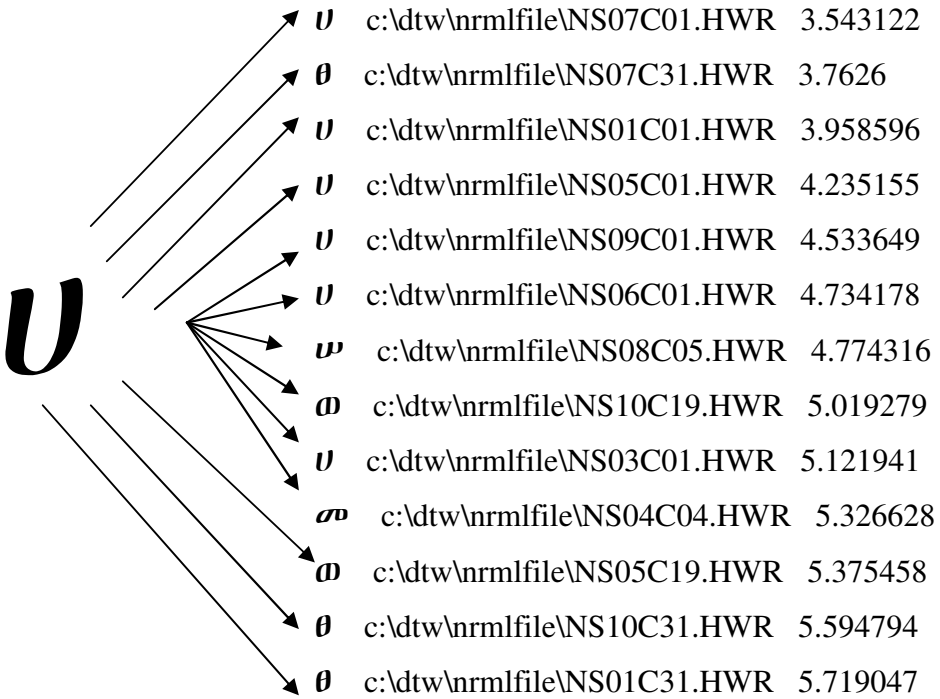

Figure 6.1 Shortest DTW distance list stored in a text file for the character 'U' written by subject 2.

In Figure 6.1 the first line shows that when testing for the shortest distance for character **U** (C01) written by Subject 2, the shortest DTW distance was found to be with character **U** (C01) from subject 7 with a distance (*Euclidean*) of 3.543122. Like wise character **θ** (C31) written by subject 7 produced the second shortest distance recorded to be 3.7626. And thus the list goes on for the matched distances against each character in the data corpus. As it can be noted from Figure 6.1 the DTW distance measurement between character **U** of subject two and the rest of the other subjects characters are listed in ascending order. Since the main aim of this experiment is character recognition, information regarding which subject wrote the character will not be retained. Based on these objectives we summarize the results of our experiments in Table 6.1 and Table 6.2.

In Table 6.1 a character is said to be recognized if the character to be recognized gives a shortest distance result with the same character (written by a different subject) from the prototype set. This recognition is indicated as a check mark '√' in Table 6.1 and 6.2. On

the other hand if a character in question has the shortest DTW distance with a different character than itself (written by a different subject) then the misrecognized character is listed in the tables.

Table 6.1 shows the shortest distance results after the DTW matching was completed for the 33+1 characters written by subjects 2 to 15. Similarly Table 6.2 shows similar results for the remaining subjects 16 to 30.

The reasons behind analyzing the shortest distances were:

- to identify how many incorrect recognition results had occurred in the shortest distances recorded after the DTW matching phase,
- to search for logical reasons for incorrect recognition results and suggest corrective measures that may improve accuracy.

.

Feedel	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11	S12	S13	S14	S15
ሀ	✓	✓	✓	✓	✓	✓	ፀ	✓	✓	✓	✓	✓	✓	✓
ለ	✓	✓	ጨ	✓	✓	✓	ጠ	✓	ዘ	✓	✓	✓	✓	ኦ
ሐ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
መ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	ወ	✓	✓	✓
ሠ	✓	✓	✓	✓	✓	✓	ወ	✓	✓	✓	✓	✓	✓	✓
ረ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ሰ	ሐ	ሐ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ሸ	✓	✓	✓	✓	✓	✓	✓	✓	ሸ	✓	✓	✓	✓	✓
ቀ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
በ	✓	✓	✓	✓	✓	✓	ጠ	✓	✓	✓	ወ	✓	✓	✓
ተ	✓	✓	✓	ሐ	✓	✓	✓	✓	✓	ቀ	ቀ	✓	ቀ	✓
ቸ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	ኸ
ነ	✓	ዘ	✓	✓	✓	✓	✓	✓	✓	ለ	✓	✓	✓	✓
ኘ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ኀ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
አ	ጸ	ጨ	✓	✓	ኸ	✓	ከ	ደ	✓	✓	ጸ	✓	✓	✓
ከ	✓	ጠ	✓	✓	ኦ	✓	መ	ጨ	✓	✓	ኸ	✓	ጠ	ጠ
ኸ	✓	ጀ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ወ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ዐ	✓	✓	ፀ	✓	✓	✓	✓	ወ	✓	✓	✓	ወ	✓	✓
ዘ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ዠ	ከ	✓	✓	✓	✓	መ	✓	✓	✓	✓	✓	✓	✓	✓
የ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ደ	✓	✓	✓	✓	✓	✓	ኸ	✓	ሸ	✓	✓	✓	✓	ጸ
ጀ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ገ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ጠ	ጨ	✓	✓	✓	✓	✓	ጨ	✓	✓	✓	ወ	ጨ	✓	✓
ጨ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ጸ	✓	✓	✓	✓	✓	✓	✓	ጀ	✓	✓	✓	✓	ጀ	✓
ጸ	ጨ	✓	ጨ	✓	✓	✓	ኸ	ደ	✓	✓	✓	✓	✓	ደ
ፀ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ፈ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ፐ	ቾ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ቨ	✓	ሸ	✓	✓	ሸ	✓	✓	✓	ሸ	✓	✓	✓	✓	✓
Errors	6	6	3	1	3	1	9	4	4	2	6	2	3	5
86.917	*82.35*	*82.35*	*91.18*	*97.06*	*91.18*	*97.06*	*73.53*	*88.24*	*88.24*	*94.12*	*82.35*	*94.12*	*91.18*	*85.29*

Table 6.1 Shortest distance measures for 34 Ethiopic characters from subject 2 to 15

Feedel	S16	S17	S18	S19	S20	S21	S22	S23	S24	S25	S26	S27	S28	S29	S30
ሀ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ለ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ሐ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
መ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ሠ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ረ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ሰ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ሸ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ቀ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
በ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ተ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ቸ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ነ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ኘ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ኀ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
አ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ከ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ኸ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ወ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ዐ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ዘ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ዠ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
የ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ደ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ጀ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ገ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ጠ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ጨ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ጸ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ጸ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ፀ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ፈ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ፐ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ቨ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Errors	3	3	5	5	4	5	2	7	6	8	6	7	4	6	3
	91.18	91.18	85.29	85.29	88.24	85.29	94.12	79.41	82.35	76.47	82.35	79.41	88.24	82.35	91.18

Table 6.2 Shortest distance measures for 34 Ethiopic characters from subject 16 to 30

It can be noted that recognition rates per subject ranged from a low of 73.53% to a high of 97.06%. The average recognition rate for this experiment is 86.917%.

6.2 Discussion

In an ideal scenario the shortest distances would be from characters that are identical to the character being classified. In table 6.1 the second shortest distance for character '**ሀ**' written by subject 2 resulted in a match for '**ϑ**' (see figure 6.2). A detailed analysis of these two characters in comparison helps pin point the error and help achieve higher accuracy. Nevertheless, of the first three shortest distances reported two of them matched the character and hence the character '**ሀ**' is said to be recognized. A detailed study of the second mismatched character shows that the character '**ϑ**' (see figure 6.2) has a similar stroke order to '**ሀ**' that has made it a possible match.

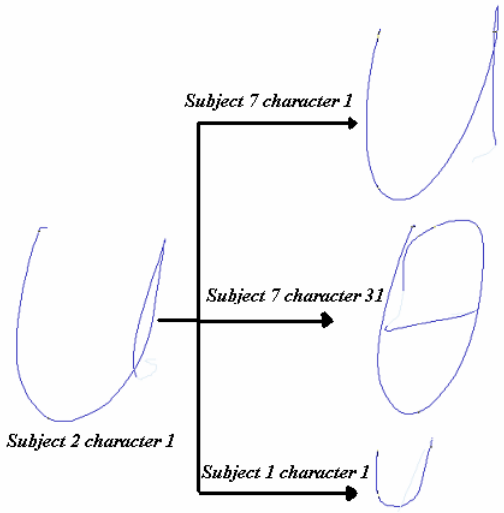

Figure 6.2 Depiction of raw data comparison for Subject 2 character 'ሀ' that recorded the first three shortest distance

Similarly, the recognition result written by subject two for the character '**ሰ**' brought back a result of wrong recognition where the first shortest distance was '**ሐ**' from subject four. Further analysis showed that subject 3 and subject 4 all brought back 100% recognition for the Ethiopic character '**ሐ**'. In Table 6.1, and 6.2 the recognition of a character was based on the shortest DTW distance computation as noted earlier. Further analysis has been done with the number of occurrences of the characters with the first three shortest distances reported by the recognizer. The overall average recognition rate for the first three occurrences has come up to 85.903% which is less than when only the shortest distance was being considered (86.917%).

It is important to note that a wide variation in handwriting styles could be the reason for such inconsistent results. These misrecognitions arose from numerous reasons such as, stroke order, overlapping strokes, disconnected loops, loss of significant points due to normalization, etc… Further analysis is required to improve the accuracy of the recognition engine.

Some subjects in these experiments noted a problem with their hand-eye coordination while using the pen tablet digitizer. It was inconvenient for them to write the characters while simultaneously looking at the tablet with no display and looking at the screen. Most of the writers suggested that a tablet with simultaneous functions for both display and input would be a remedy to this inconvenience and would allow them to right much better.

In this work, our focus has been on the first shortest distance reported by the recognizer.

This approach has given us a better recognition rate compared to the number of occurrences of the characters with the first three shortest distances reported by the recognizer.

It has to be noted that these results were recorded with a data corpus of only nine hundred eighty five (see chapter 5 *'data collection' page 48*) characters where all Ethiopic characters except the character ('ኸ' from subject 1) had nine instances written by different subjects. It should hold true that extending this study with a bigger data corpus will show a higher recognition result as indicated in the literature.

Not only does this entire procedure promise greater recognitions rates but its applicability for the complete Ethiopic Script that includes numerals and labialized characters is now apparent.

CHAPTER SEVEN: Conclusion and Future Works

Research in on-line handwriting recognition started in the early sixties, as the first generation of tablet digitizers became available [8]. Today solutions for online handwriting recognition are highly robust.
Even though the frontiers for online handwriting recognition for Ethiopic characters are only just being traversed, the results attained in this and earlier [14] works are highly promising for future applications of online handwriting recognition for Ethiopic characters. However, there is still a lot to be done before these recognition systems are usable.

This is the first writer independent online handwriting recognition prototype developed for Ethiopic characters.
After studying the strategies undertaken by online handwriting recognition systems developed for other languages, Dynamic Time Warping stood out from the group because of its applicability regardless of character sets.
This feature alone made it clear that Dynamic Time Warping was the best way to go about developing a recognition prototype for the Ethiopic Script.
After the data was collected, the developed DTW algorithms were applied and the results were recorded.
Future research will achieve higher recognition results when analysis and fine tuning continues if the future of this thesis.
Interesting ways to move forward for additional research in the future can be:

i. Due to time limitations additional preprocessing algorithms were not investigated. To enhance the accuracy rate of the DTW recognizer by implementing additional preprocessing algorithms such as slant correction, moment normalization etc…
ii. To apply DTW to a word based handwriting recognition engine that is integrated with an Ethiopic lexicon for languages that use Ethiopic characters.
iii. To test the accuracy rate of DTW with a bigger data corpus.
iv. To optimize the training prototype set by implementing various clustering algorithms

v. To develop an adaptive writer independent handwriting recognition engine based on the existing system.
vi. Integrating a rejection test to the system to make it more robust in handling erroneous character input

References

1. Alemayehu H. Dr. (1996), ፍቅር እስከ መቃብር 9th Edition, Mega Printers, Addis Ababa, pp 3-4
2. Claus Bahlmann and Hans Burkhardt, ***"The Writer Independent Online Handwriting Recognition System frog on hand and Cluster Generative Statistical Dynamic Time Warping",*** IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 3, March 2004.
3. Drissman A. ***"Handwriting Recognition Systems: An Overview"*** Dr. Sethi, CSC 496 February 26, 1997
4. Encarta Encyclopedia Standard Reference Library **"Carpel Tunnel Syndrome"** 2004, Microsoft Corporation.
5. Hailemariam M. "***Handwritten Amharic Character Recognition: The Case of Postal Addresses"*** (Masters Thesis). Addis Ababa University, School of Information Studies for Africa, Addis Ababa University, 2003.
6. Jaeger S., Liu C.-L., Nakagawa M., "***Online Recognition of Chinese Characters, The State-of-the-Art***", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, No. 26, 2004.
7. Jaeger S., Liu C.-L., Nakagawa M., **"*The state of the art in Japanese Online Handwriting Recognition Compared to Techniques in Western Handwriting Recognition*"**, International Journal on Document Analysis and Recognition, 2003.
8. Jong, Oh ***"An On-Line Handwriting Recognizer with Fisher Matching, Hypotheses Propagation Network and Context Constraint Models"*** (Doctor of Philosophy Dissertation), Department of Computer Science, New York University, May 2001.
9. Mulugeta W. "***OCR for Special Type of Handwritten Amharic Text ("Yekum Tsifet"), Neural Network Approach*** (Masters Thesis). Addis Ababa University, School of Information Studies for Africa, Addis Ababa University, 2003.
10. Niels R. ***"Dynamic Time Warping: An intuitive way of handwriting recognition?"*** (Masters Thesis) Radboud University Nijmegen, Faculty of Social Sciences, Department of Artificial Intelligence / Cognitive Science, Nijmegen, The Netherlands, November/December 2004.

11. Niels R. and Vuurpijl, L. *"Dynamic Time Warping Applied to Tamil Character Recognition"* Nijmegen Institute for Cognition and Information, fr.niels,vuurpijlg@nici.ru.nl.
12. Plamondon R., Srihari S. N. ***"On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey"*** 1EEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. VOL. 22, NO. 1. JANUARY 2000.
13. Senior, A. and Robinson A. ***"An off line cursive handwriting recognition system"*** IEEE TRANSACTIONS ON PATTERN ANALYSIS & MACHINE INTELLIGENCE VOL 20, NO. 3 MARCH 1998
14. Shimeles A., ***"Online Handwriting Recognition for Ethiopic Characters"*** (Masters Thesis), Department of Computer Science, Addis Ababa University, June 2005.
15. Sornlertlamvanich V., et al,. ***"The State of the Art in Thai Language Processing"*** Proceedings of the 38th annual Meeting of the Association for Computational Linguistics (ACL 2000), Hong Kong, pp 597-598, October 2000.
16. Tadesse N. "***Handwritten Amharic Text Recognition Applied to the Processing of Bank checks***", (Masters Thesis). Addis Ababa University, School of Information Studies for Africa, Addis Ababa University, 2000.
17. Tappert C. et al, ***"The state of the art in online handwriting recognition"*** (1990), IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. VOL. 12. NO 8. AUGUST 1990
18. Thomas Bloor, the simplified spelling society, (March 2006), "***The Ethiopic Writing System: a Profile***", http://www.spellingsociety.org/journals/j19/ethiopic.php
19. Wikipedia The Free Encyclopedia (July 2006), "***Amharic language***", http://en.wikipedia.org/wiki/Amharic_language
20. Wikipedia The Free Encyclopedia (July 2006), "***Diacritic***", http://en.wikipedia.org/wiki/Diacritic
21. Wikipedia The Free Encyclopedia (July 2006), "***Ge'ez alphabet***", http://en.wikipedia.org/wiki/Ge%27ez_script
22. Wikipedia The Free Encyclopedia (June 2006), "***Abugida***", http://en.wikipedia.org/wiki/Abugida
23. Wikipedia The Free Encyclopedia (June 2006), "***Handwriting Recognition***", http://en.wikipedia.org/wiki/Handwriting_recognition
24. Wikipedia The Free Encyclopedia (June 2006), "***Lexicon***", http://en.wikipedia.org/wiki/Lexicon

25. Wikipedia The Free Encyclopedia (May 2006), "***Graffiti (Palm OS)***", http://en.wikipedia.org/wiki/Graffiti_%28Palm_OS%29

Declaration

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university, and that all source of materials used for the thesis have been duly acknowledged.

Declared by:

Name:_____

Signature:_____

Date:_____

Confirmed by advisor:

Name:_____

Signature:_____

Date:_____

Place and date of submission: Addis Ababa, June 2006