

Interpolation of Missing Points and Smoothing of Character for Online Handwriting Recognition of Gurmukhi Script

Thesis submitted in partial fulfillment of the requirement for

The award of the degree of

Masters of Science

in

Mathematics and Computing

Submitted by

Gagandeep Kaur

Roll No. – 300903002

Under

The guidance of

Dr. Rajesh Kumar



School of Mathematics and Computer Application

Thapar University

Patiala – 147004 (Punjab)

India

July 2011

Certificate

I hereby clarify that the work which is being presented in the thesis entitled "Interpolation of Missing Points and Smoothing of Character for Online Handwriting Recognition of Gurmukhi Script" in partial fulfillment of the requirements for the award of degree of Masters of Science, School of Mathematics and Computer Application, Thapar University, Patiala is an authentic record of my own work carried out under the supervision of Dr. Rajesh Kumar.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.


(Gagandeep Kaur)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.



Dr. Rajesh Kumar

(Supervisor)

SMCA, Thapar University

Patiala

Countersigned by:

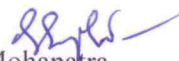


Dr. S. S. Bhatia

(Professor & Head)

School of Mathematics and Computing

Thapar University, Patiala.



Dr. S. K. Mohapatra

Dean of Academic Affairs

Thapar University

Patiala.

Abstract

Handwriting has continued to persist as a means of communication and recording information in day-to-day life even with the introduction of new technologies. The main goal of the work presented in this thesis has been the improvement of preprocessing phase of an online handwriting recognition system for Gurmukhi script. In this thesis two methods, namely Bezier interpolation method and B-spline interpolation method, are presented and compared for interpolation of missing points. After that smoothing has also been done using data points interpolated by the different methods.

Acknowledgement

The key elements concentration, dedication, hard work and application are not the only essential factors for achieving the desired goals but also guidance, assistance and co-operation of people is necessary.

I would like to express my deep and sincere gratitude to my supervisor Dr. Rajesh Kumar, Associate Professor, School of mathematics and computer Applications (SMCA). His wide knowledge and logical way of thinking have been of great value for me. His understanding and personal guidance have provided a good basis for the present thesis. Especially the strict and extensive comments and many discussions and the interactions with Dr. Rajesh Kumar had direct impact on the final form and quality of the thesis.

I owe my sincere thanks to Dr. S. S. Bhatia, Professor and Head of School of Mathematics and Computer Applications (SMCA), Thapar University, Patiala for providing facilities. I would also like to pay my special thanks to Dr. A. K. Lal, Associate Professor, SMCA and PG coordinator.

My thesis work would be incomplete without thanking my friends, who were always there in the hour of need.

I was very fortunate to have unconditional support of my family. I thank my Parents, who gave me the courage to get my education, supported in my all ups and downs though out my life. Without their encouragement and suggestions, this work would have been very difficult for me to tackle.

Above all, I pay my reverence to the almighty GOD. May Your name be exalted, honored and glorified.

Gagandeep Kaur
Gagandeep Kaur

(300903002)

List of Figures

Fig. No.	Name of Figure	Page No.
Fig.1.1	Writing styles of 4 different persons.	4
Fig.1.2	Boxed discrete handwriting.	5
Fig.1.3	Word ‘ਮੁੰਦਰ ਦਿਸ਼’ is written in different styles	5
Fig.1.3(1)	Spaced discrete	
Fig.1.3(2)	Run-on discrete	
Fig.1.3(3)	Cursive	
Fig.1.3(4)	Mixed cursive	
Fig.1.4	Similar Gurmukhi Characters.	8
Fig.1.5	Vowels and Vowel diacritics (Laga Matra).	9
Fig.1.6	Consonants (Vianjans).	9
Fig.1.7	Other symbols.	9
Fig.2.1	Phases of online handwriting recognition.	10
Fig.2.2	Commonly used hardware devices for capturing handwriting.	11
Fig.2.3	Main steps for preprocessing phase.	13
Fig.3.1	Character ‘ਗ’ written with three strokes.	15
Fig.3.2	Handwritten character before preprocessing.	16
Fig.3.3	Screen shot net bean output file containing collected data point of original character.	17
Fig.3.4	Normalization of character.	19
Fig.3.5	Handwritten character after size normalization and centering.	19

Fig.3.6	Bezier interpolation.	20
Fig.3.7	Handwritten character after Bezier interpolation.	22
Fig.3.8	B-spline interpolation.	23
Fig.3.9	Handwritten character after B-spline interpolation	24
Fig.3.10	Formation of angle α at point P_i .	25
Fig.3.11	Handwritten character after smoothing.	26
Fig.3.11(a)	smoothing after interpolation by B-spline interpolation method	
Fig.3.11(b)	smoothing after interpolation by Bezier interpolation method	

Contents

Certificate	i
Abstract	ii
Acknowledgement	iii
List of Figures	iv
Contents	vi
Chapter 1: Introduction	
1.1 Background of Study	1
1.2 Issues in Online Handwritten Recognition System	3
1.2.1 Handwriting Styles Variations	3
1.2.2 Constrained and Unconstrained Handwriting	4
1.2.3 Limited Resources in Small Devices	5
1.2.4 Stroke Number and Order Variation	6
1.2.5 Personal, Situational and Material Factors	6
1.2.6 Writer Dependent vs. Writer Independent Recognition Systems	7
1.2.7 The Similarity of Some Characters Gurmukhi Script	7
1.3 Overview of Gurmukhi Script	8
1.3.1 Origin of Gurmukhi Script	8
1.3.2 Brief Introduction of Gurmukhi Script	8
Chapter 2: Literature Review	
2.1 Data Collection	11
2.2 Preprocessing	11

Chapater 3: Interpolation of Missing Points and Smoothing for Online Handwriting Recognition of Gurmukhi Script

3.1 Data Collection	14
3.2 Preprocessing Phase	16
3.2.1 Size Normalization and Centering of Stroke	18
3.2.2 Interpolating Missing Points	20
3.2.2.1 Bezier Interpolation Method	20
3.2.2.2 B-spline Interpolation Method	22
3.3 Smoothing of Character	24
3.4 Conclusion	26
References	28

Chapter 1

Introduction

1.1 Background of Study

The development of technology is happening how people live and work. Situations and things which were unimaginable earlier are now becoming possible and practical. Today's computers are very powerful and capable of processing very large amount of data in few moments, which was mere imagination a century back. In the 1980's the computer was like a great giant machine that had been a miracle. Out of imagination, computers became more and more powerful and at the same time more and more compact.

Today, Computers have become as compact and possesses attractive applications as well. Historically, the whole thing was started when a company called Palm produced PDAs (Personal Digital Assistant), whose main function was to replace personal organizers. Those PDAs had applications like address book, calendar, notepad etc. These devices were called palmtop since they were one handed and fit in one's palm. More companies also involved in the production of such devices. On the other hand, Microsoft produced minimized laptop-style computers with screen of size 20 cm wide and only 8 cm high and with an attached keyboard. These computers used a version of Windows, named Windows CE and called handheld rather than palmtops to indicate that they don't fit in palm. Pocket PCs were the other devices using the modified version of Windows CE which are sometimes called handheld. This time, these terms that emerged at different times have been used interchangeably. In general, handheld computers are defined to be any small devices that provide computing and information storage and retrieval and that can be easily carried and used [D. Perry, 2003].

Though these devices turn out to be common in an increasing number of countries around the World. But these are not very common in India because these are available mainly with English language not in Indian languages. Localizing the applications of these devices could play an important role in computerization of different departments in India. We have envisaged that usage of devices like handheld computers and smart phones will highly increase in the future as mobile device. If this happens, then it would be compulsory to have

localized applications that run on such devices for the purpose of exploiting the opportunity of mobile computing.

When these devices are used, data and commands are to be entered with pen like device called stylus. So, we can say that pen computing will become emerging trend of computing. Pen computing is field broadly includes computers and applications in which a pen is the main input device [[“www.research.ibm.com/electricInk/”](http://www.research.ibm.com/electricInk/)]. Pen-based input incorporated with online handwriting recognition feature allows people to write in a natural way to input data, and provide a pen-paper like interface [W. Mulugeta, 2003].

It has already been asserted that the pen-based input is an advantage if handwriting recognition is implemented. Handwriting recognition is the task of transforming a language represented in its spatial form graphical marks into symbolic representation [R. Plamondon and S. N. Srihari, 2000]. Handwriting recognition is mainly of two types: online handwriting recognition and offline handwriting recognition. In offline handwriting recognition systems, the whole data will be collected and provided for the recognizer as a bitmap. On the other hand, online handwriting recognition systems run and receive the data as the user writes and they are expected to process the data and recognize in a real time [R. Plamondon and S. N. Srihari, 2000; C. C. Tappert *et al.*, 1990].

The offline and online handwriting recognition systems are already very much devolved for English language. These systems are also available for other non-Latin languages such as Chinese, Japanese, Thai and Arabic. As handwriting recognizers are language specific so this calls the need for development of such systems for Indian languages.

In spite of the fact that a recognition system with 100% recognition rate is not still developed, these systems are improved time to time with different researches in this field. The presence of online handwriting recognizer for Devanagiri, Gurmukhi, Bangla and other Asian scripts shall provide a natural way of communication between users and computers and it will increase the usage of personal digital assistant or tablet PCs in Indian languages. Also, some of the Asian scripts such as Devanagiri, Gurmukhi, Bangla and Tamil share many similarities and therefore advances made for one script with respect to online handwriting recognition could be useful for other such similar scripts. The present study has been carried out for Gurmukhi script.

1.2 Issues in Online Handwriting Recognition System

As already mentioned the technique by which a computer system can recognize handwritten characters and other symbols is called handwriting recognition system. Online handwriting recognition captures a character as a set of strokes that are represented by a sequence of coordinate points. This way of capturing characters becomes conspicuous when dealing with strongly distorted characters written in the cursive style [T. Wakahara, 1992]. Other main advantages of online handwriting recognition are interactivity and adaptation. The rewriting of an editing symbol can cause the display to change appropriately. Also, recognition errors can be corrected immediately. When the user sees that some of his characters are not being accurately recognized, he can alter his drawing to improve recognition. Thus, the user adapts to the recognition system. Thus, both writer adaptation and machine adaptation is possible. Moreover, editing, annotating, and other applications that use direct pointing and manipulation are well suited to online handwriting recognition [C. C. Tappert *et al.*, 1990 and T. Wakahara *et al.*, 1992]. The online handwriting recognition has great potential to improve user and computer communication. Due to variability in handwriting styles and distortions caused by the digitizing process, even the best handwritten character recognizer is unreliable.

1.2.1 Handwriting Styles Variations

Writing in one's own style brings unevenness in writing units which is most difficult part to classify. Writing units reveals number, shape, size, order of stroke and speed of writing. Variations in number of strokes, their order, shapes and sizes, tilting angles and similarities among characters from one another are important factors, which are to be considered. The shape of a character is also influenced by the word in which it is appearing. Characters can look similar although their number of strokes, and the drawing order and direction of the strokes may vary considerably [C. C. Tappert *et al.*, 1990]. Fig.1.1 illustrates the samples of different writing styles of 4 different persons. Variation of writing styles can be noticed from this.

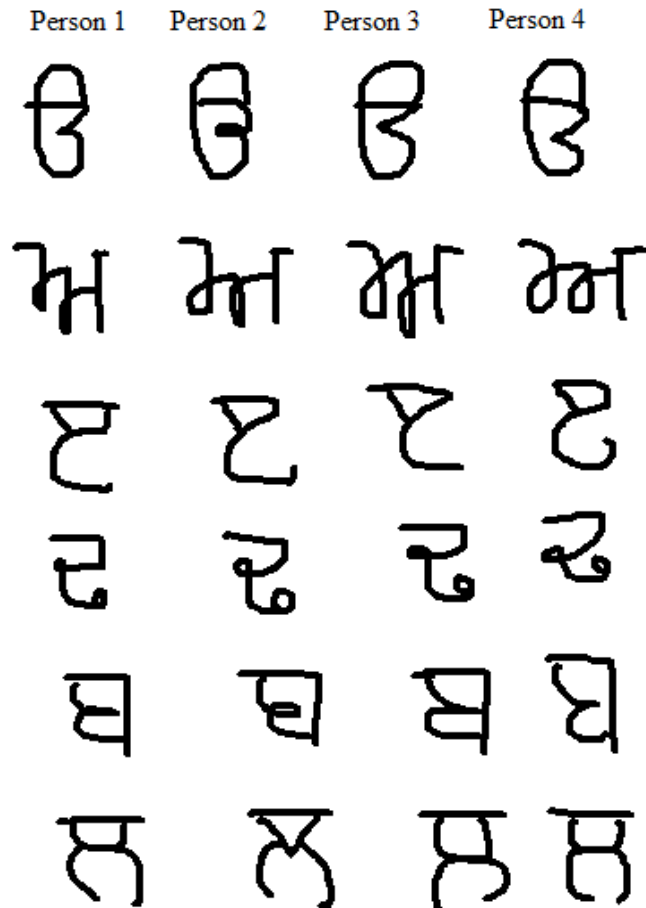


Fig. 1.1: Writing styles of 4 different persons.

1.2.2 Constrained and Unconstrained Handwriting

Handwriting styles could be constrained or unconstrained [C. C. Tappert, 1984]. Constrained handwriting is boxed discrete and spaced discrete in nature. Unconstrained handwriting means that no restrictions on the writing style are imposed. Therefore, data might contain hand printed or cursively written words, or a mixture of those two basic writing styles [G. A. Fink and T. Plotz, 2006]. When each character is written separately with spaces and no character touches other character is called spaced discrete handwriting shown in Fig.1.2. If each character is written separately and touches other characters, it is referred as run-on discrete handwriting. When characters in one word are connected and strokes are used more than once in individual character, it is referred to cursive handwriting. It is observed that most of the people write in mixed cursive styles that include mixture of spaced, run-on discrete and cursive styles handwriting shown in Fig.1.3. It is a difficult task to recognize cursive

handwriting due to great amount of variability. Each writer is having one's own speed of writing and uses different shapes to represent characters. Also, in cursive handwriting no clear boundaries are specified between characters to distinguish them.

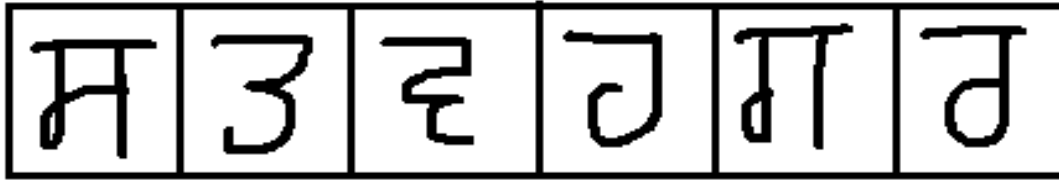


Fig 1.2: Boxed discrete handwriting.

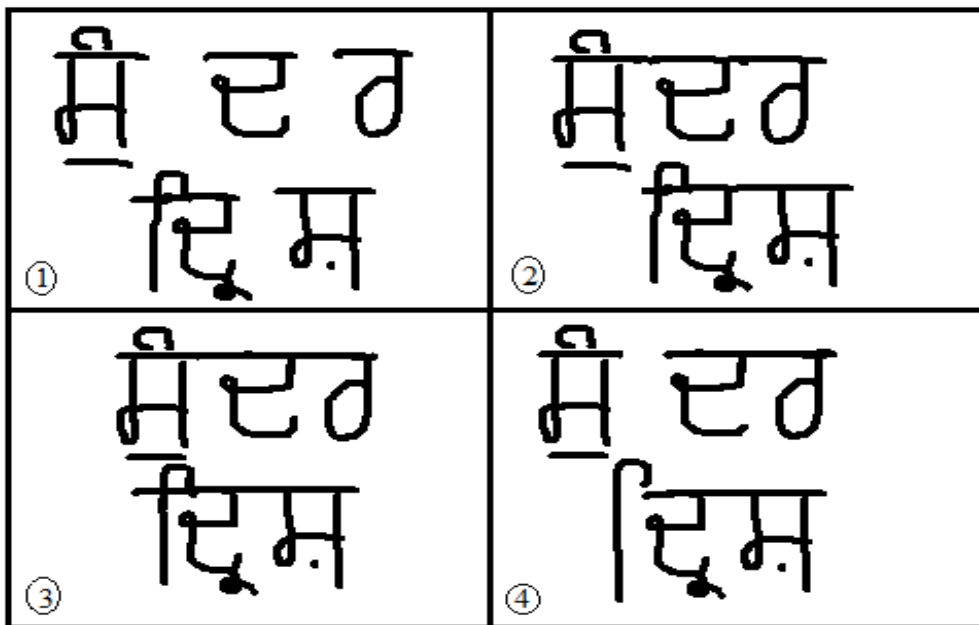


Fig. 1.3: Word 'ਸੁੰਦਰ ਦਿਸ਼' is written in different styles (1) Spaced discrete (2) Run-on discrete (3) Cursive (4) Mixed cursive.

1.2.3 Limited Resources in Small Devices

It sounds good to remind that online handwriting recognition systems help in avoiding keyboard-based data input method in small handheld devices. It is not optional to let users of such devices input data in a pen-paper like environment by aiding the process with hardware

digitizers along with pen-like devices and incorporating online handwriting recognition feature.

It is good news that the storage and processing capability of handheld devices is improving from time to time. Limitation in resources has been a problematic phenomenon of these devices for recognition engine developers. The headache would be severe for the languages consisting large character sets such as the Chinese/Japanese [S. Jaeger *et al.*, 2000; C. L. Liu *et al.*, 2004]. Carefully designed algorithms to optimize the amount of data and the processing requirements are supposed to be attained for online handwriting recognition systems. A related design goal could be to make the systems platform independent and capable of dealing with different pen capture technologies that may reveal different characteristics such as varying sampling rate.

1.2.4 Stroke Number and Order Variation

A stroke is defined as the sequence of sample points occurring between consecutive pen-down and pen-up transitions [S. D. Connell *et al.*, 2000]. A character could possibly be a uni-stroke character or formed from two or more strokes. The number and order of strokes is available for online systems, which is always considered as an advantage. In contrast, variation in stroke number and stroke order in a single character is a source of complication in online handwriting recognition system. Stroke order and stroke number variations are even terribly severe in particular systems such as Chinese/Japanese systems [S. Jaeger *et al.*, 2003]. As a result, the issue of handling stroke order and number variations is one of the design goals when designing online handwriting recognition systems particularly the writer independent ones.

1.2.5 Personal, Situational and Material Factors

A good recognition requires neat and clean handwriting. In most of the cases, it has been noted that neat and clean handwriting do not take place as handwriting of people also depends on their profession [T. T. Kuklinski, 1984; J. R. Ward and T. Kuklinski, 1988]. Another main personal factor with which handwriting of person is greatly influenced is whether person is right handed or left handed.

The situational factors depend on the way of presentation of writing. The way of presentation could be stressful or in haste or distraction while writing [T.T. Kuklinski., 1984; A.M. Wing, 1979]. Material factors depend upon the material which is used for writing by a person. The material used in writing may provide comfort or discomfort to writer that result

into variations in handwriting. This includes the position and size of writing board. The length of the writing line or the size of the writing boxes for characters can have effect on the handwriting style [T. T. Kuklinski, 1984].

1.2.6 Writer Dependent vs. Writer Independent Recognition Systems

The system that is based on known writing styles is called writer dependent system. The writer dependent systems are expert in certain handwriting styles and include recognition constraints with respect to stored handwriting styles. A writer independent recognition system recognizes wide range of possible writing styles, while a writer dependent recognition system is trained to recognize only from specific users. Therefore, a writer dependent recognition system works on data with a smaller variability and therefore a chance of having higher reliability is achieved in contrast to writer independent recognition system. Writer independent systems are meant for the unknown handwriting styles. Writer independent system is more difficult to develop in comparison with writer dependent system. It is because writer independent system needs to study all common aspects of handwriting. Also, writer independent system demands all possible options to store handwriting variations in the database. Writer dependent recognition systems have achieved better recognition accuracy in comparison to writer independent recognition systems [J. Subrahmonia and T. Zimmerman, 2000]. The difficulty of development of writer independent systems arises from the fact that the system is expected to handle much greater varieties of handwriting styles [S. D. Connell, 2000].

1.2.7 The Similarity of Some Characters in Gurmukhi Script

There are many characters in Gurmukhi script which have almost same shapes, some of those shown in Fig.1.4. Because of same shapes it becomes difficult to recognize these characters accurately.

ਖ and ਖ਼	ਵ and ਵ਼	ਜ and ਜ਼
ਲ and ਲ਼	ਤ and ਤ਼	ਬ and ਥ
ਨ and ਨ਼	ਤ and ਤ਼	ਥ and ਥ
ਫ and ਫ਼	ਟ and ਟ਼	ਪ and ਪ਼
ਅ and ਘ	ਹ and ਹ਼	ਪ and ਪ਼
ਸ and ਸ਼	ਏ and ਏ਼	ਗ and ਗ਼
ਸ and ਸ਼	ਵ and ਵ਼	ਚ and ਚ਼

Fig. 1.4: Similar Gurmukhi characters

1.3. Overview of Gurmukhi Script

1.3.1 Origin of Gurmukhi Script

The Gurmukhi alphabet was devised during the 16th century by Guru Nanak Dev Ji, the first Sikh guru, and popularized by Guru Angad Dev Ji, the second Sikh guru. Gurmukhi script is derived from “landa” script. The name Gurmukhi means "from the mouth of the Guru".

1.3.2 Brief Introduction of Gurmukhi Script

Gurmukhi is the script of Punjabi language which is widely spoken across the world. Gurmukhi script is written in left-to-right direction and in top-down approach. Most of the characters of Gurmukhi have a horizontal line at upper part. The characters of words are connected mostly by this line called head line. Gurmukhi has 41 consonants (*vianjan*) shown in Fig.1.6 and 12 vowel symbols (*lāga mātrā*) shown in Fig.1.5, two symbols for nasal sounds (*bindi* (ँ) and (*tippi* (ँ)), and one symbol which duplicates the sound of any consonant (*addak*(ँ)) shown in Fig.1.7. In addition, four conjuncts are used: three subjoined forms of the consonants ਚ, ਚ and ਵ, and. There is no concept of upper or lowercase

characters.

Vowels and Vowel Diacritics (Laga Matra)

ਅ	ਆ	ਇ	ਈ	ਉ	ਊ	ਏ	ਐ	ਓ	ਔ
a	ā	i	ī	u	ū	e	ai	o	au
	ਕੰਨਾ	ਸਿਹਾਰੀ	ਬਿਹਾਰੀ	ਅੰਕੜ	ਦੁਲੈਂਕੜ	ਲਾਂਵਾਂ	ਦੁਲਾਂਵਾਂ	ਹੇੜਾ	ਕਨੇੜਾ
	kannā	sihārī	bihārī	auñkaṛ	dulainkaṛ	lānvāñ	dulānvāñ	hōṛā	kanaurā

Fig. 1.5

Consonants (Vianjans)

ੳ ਊੜਾ (ūrā)	ਅ ਅੰੜਾ (airā)	ੲ ਈੜੀ (īrī)	ਸ ਸੱਸਾ (sas'sā)	ਹ ਹਾਹਾ (hāhā)
ਕ ਕੱਕਾ (kakkā)	ਖ ਖੱਖਾ (khakhkhā)	ਗ ਗੱਗਾ (gaggā)	ਘ ਘੱਗਾ (ghaggā)	ਙ ਙੰਙਾ (ñāññā)
ਚ ਚੱਚਾ (caccā)	ਛ ਛੱਛਾ (chachchā)	ਜ ਜੱਜਾ (jajjā)	ਝ ਝੱਜਾ (jhajjā)	ਞ ਞੰਞਾ (ñāññā)
ਟ ਟੈਂਕਾ (taiñkā)	ਠ ਠੱਠਾ (thaththā)	ਡ ਡੱਡਾ (daddā)	ਢ ਢੱਡਾ (dhaddā)	ਣ ਣਾਣਾ (ṇāṇā)
ਤ ਤੱਤਾ (tattā)	ਥ ਥੱਥਾ (thaththā)	ਦ ਦੱਦਾ (daddā)	ਧ ਧੱਧਾ (dhaddā)	ਨ ਨੱਨਾ (nannā)
ਪ ਪੱਪਾ (pappā)	ਫ ਫੱਫਾ (phaphphā)	ਬ ਬੱਬਾ (babbā)	ਭ ਭੱਬਾ (bhabbā)	ਮ ਮੱਮਾ (mam'mā)
ਯ ਯੱਯਾ (yayyā)	ਰ ਰਾਰਾ (rārā)	ਲ ਲੱਲਾ (lallā)	ਵ ਵੱਵਾ (vavvā)	ੜ ਝਾੜਾ (rārā)
ਸ਼ ਸੱਸਾ (śasśā)	ਖ਼ ਖੱਖਾ (khakhkhā)	ਗ਼ ਗੱਗਾ (gaggā)		
ਜ਼ ਜੱਜਾ (zazzā)	ਫ਼ ਫੱਫਾ (faffā)	ਲ਼ ਲੱਲਾ (lallā)		

Fig. 1.6

Other Symbols

ਂ ਅਧਕ (adhak)	ੰ ਬਿੰਦੀ (bindī)	ੌ ਵਿਸਰਗ (visarg)
ੰ ਟਿੱਪੀ (tippī)	੍ ਹਲਨਤ (halant)	

Fig. 1.7

Chapter 2

Literature Review

In the mid-seventies, digitizer tablets were available in which resistive technique and analog to digital conversion technique were used. It was possible to measure the pen tip using these tablets. A number of technologies were available for tablets or writing pads. These technologies were based on electronic or electromagnetic or electrostatic or pressure sensitive techniques and the tablets with combination of input and output digitizer or display on same surface were most common in handwriting recognition.

The established procedure to recognize online handwritten characters includes following phases or components: data collection, preprocessing, feature extraction or computation of features, segmentation, recognition and post-processing [S. Jaeger *et al.*, 2001; Y. Suen *et al.*, 2003]. It has been noted that segmentation can be performed before or after preprocessing [A. Sharma, 2009]. The output obtained from one phase becomes input for the next phase.

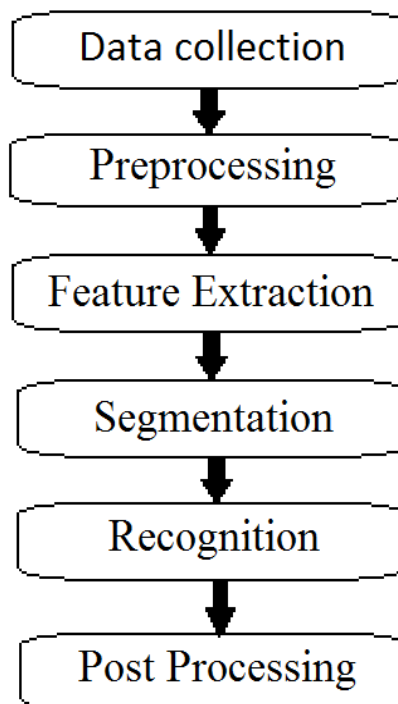


Fig. 2.1: Phases of online handwriting recognition.

2.1 Data Collection

Handwriting data is usually collected using either electromagnetic/electrostatic or pressure-sensitive tablets onto which the characters are written using a stylus. The most common of these devices is the electronic tablet or digitizer. These devices use a pen that is digital in nature. Data collection is the first phase in online handwriting recognition that collects the sequence of coordinate points of the moving pen. A typical pen includes two actions, namely, PenDown and PenUp. The trajectory of the pen between a PenDown event and a PenUp event is called Stroke. All the unique strokes of a script are manually identified and given unique labels. It is the smallest physically identifiable unit in online handwriting. These pen traces are sampled at constant rate, therefore these pen traces are evenly distributed in time and not in space. The common names of electronic tablet or digitizer are personal digital assistant, cross pad (or pen tablet) and tablet PC. The appearances of personal digital assistant, cross pad and tablet PC are shown in Fig.2.2.



Fig. 2.2: Commonly used hardware devices for capturing handwriting.

2.2 Preprocessing

The main objective of preprocessing phase in handwriting recognition is to remove noise or distortions present in input text due to hardware and software limitations, that would otherwise complicate recognition and reduce the recognition rate. [V. Govindaraju and S. N. Srihari, 1997; J. Subrahmonia and T. Zimmerman, 2000; S. A. Husain *et al.*2007]. This noise exists in the input text in the form of sharp edges, non-centered text, uneven sizes of text and missing points in text trajectories due to high speed of handwriting and slants in characters

[H. Beigi *et al.*, 1994; S. Jeager *et al.*, 2003] In online handwriting recognition, preprocessing includes five common steps, namely, size normalization and centering, interpolating missing points, smoothing, slant correction and resampling of points [Jeager *et al.*, 2001]. These steps are described in below.

Size normalization depends on how user moves the pen on writing pad. Centering is required when pen is moved along the border of writing pad. The use of size normalization techniques in online handwriting recognition have been discussed by H. Beigi *et al.*,(1994).According to him recognition is better for only in case of nominal size of writing as well as standard orientation and nominal slant. The system's ability to recognize a character should not depend on the size of the input pattern or on the location of the pattern on the surface. Therefore size normalization is often used to create pattern of a special size and location which are easy to compare to other patterns [P.Scattolin, 1995; F.Alimoglu, 1994;M. K. Brown and S. Ganapathy, 1983]. High speed of handwriting results into missing points. These missing points can be interpolated using various techniques such as Bezier interpolation method and B-spline interpolation method [M. Unser *et al.*, 1993]. B-spline interpolation method is more flexible and pleasing to work with, which is the reason why it has become more part of almost every serious graphics development environment [F. Andersson, 2003]. Smoothing of input handwriting is required to remove jitter in handwriting [Kavallieratou *et al.*, 2002]. Smoothing usually averages a point with its neighbors [P. Scattolin, 1995; F. Alimoglu, 1994;]. Slant correction and normalizing slant is required to correct the shape of input handwritten character as most of the writers handwriting is bend to left or right directions. The slant correction and normalizing is important for handwriting recognition [S. Madhanath *et al.*, 1999; P. Slavik and Govindaraju, 2001; Yimei *et al.*, 2000]. Slant correction is employed for shape normalization of the component characters of input words [S. Uchida *et al.*, 2001]. Handwritten words are usually slanted or italicized due to the mechanism of handwriting and the personality. The main techniques for slant estimation and correction are run length based technique, projection method, extrema method and generalized chain code slant correction estimator [R. M. Bozinovic and S. N. Srihari, 1989; D. Guillevic and C. Y. Suen, 1994; A. Negi *et al.*, 1995; L. Simoncini and Zs. M. K. Kovacs, 1995]. Resampling of points refers to the points in the list to be equidistant from neighbouring points as far as feasible. It means that new data points are calculated on the basis of the original points of list. The resampling techniques have been discussed in literature with respect to retain information about corner points [J. J. Brault and R.

Plamondon, 1993; M. Kobayashi *et al.*, 2001; I. Pavlidis *et al.*, 1997; K. Zhang *et al.*, 2000; W. Guefali and W. Plamondon 1993] . H. Beigi *et al.* (1996) have applied preprocessing to dynamic information in online handwriting recognition.

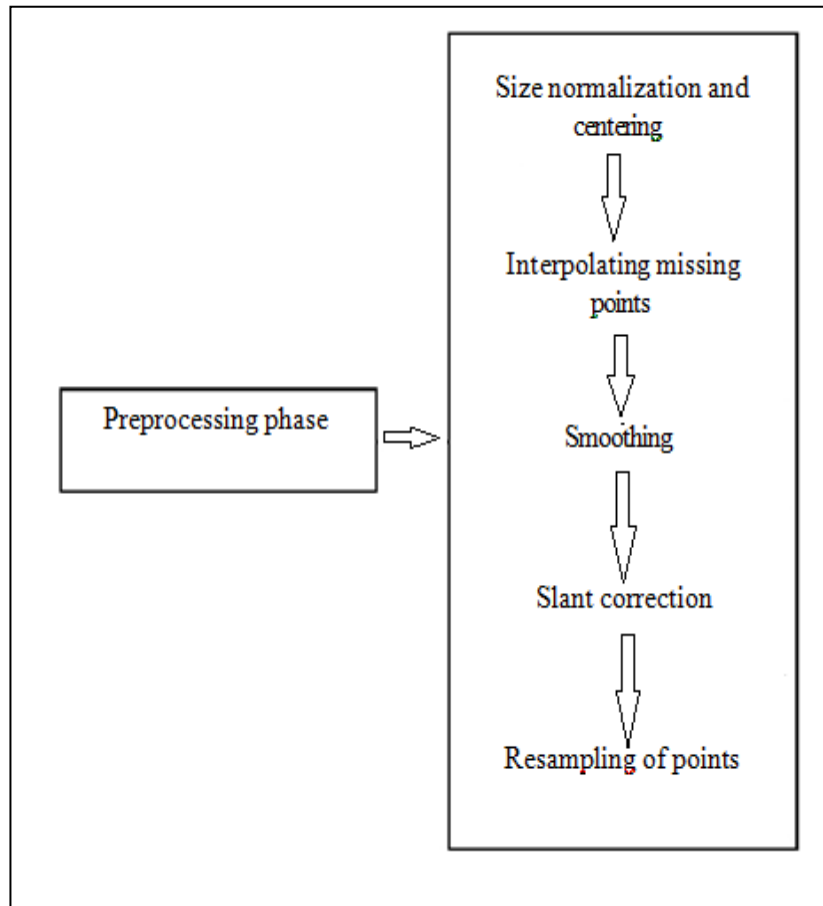


Fig. 2.3: Main steps for preprocessing phase.

Chapter 3

Interpolation of Missing Points and Smoothing for Online Handwriting Recognition of Gurmukhi Script

Data collection And Preprocessing of characters are two important phases required for recognition phase in online handwriting recognition process. In this chapter firstly we will discuss data collection process and then different stages of preprocessing phase.

First of all size normalization and centering of stroke has been carried out. After that interpolation of missing points is done by Bezier interpolation and B-spline interpolation method. Smoothing of character has also been done.

3.1 Data Collection

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 it is 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. Other devices such as PDA's and CrossPad use touch screens for collecting data. Advantage of such devices is that writing pen is not a part of sensor and pad which works as paper and makes the writing experience more natural.

A stroke is defined as the sequence of sample points occurring between consecutive pen-down and pen-up transitions [S. D. Connell, 2000]. A character could possibly be a uni-stroke character or formed from two or more strokes. In Gurmukhi Script also single stroke as well as multi-stroke both type of characters are present. In the Fig. 3.1 we have Gurmukhi Character 'ੴ'. We can easily observe that 'ੴ' is 3-stroke character.

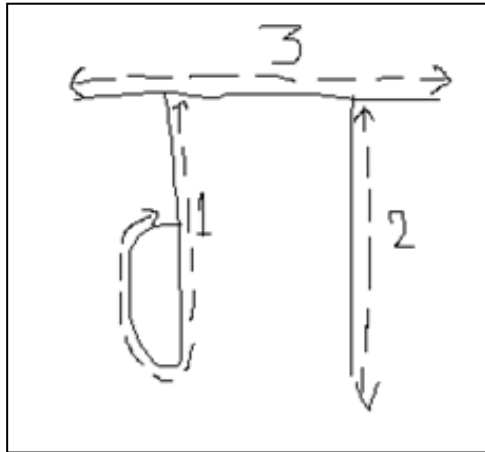


Fig. 3.1: Character 'क' written with three strokes.

Speed of writing directly affects recognition process. When the speed of writing is slow, the sample points are located densely on the true pen trace, whereas quick writing produces sparsely located points. One can note that the speed of writing typically slows down on sharp corners, in the beginning of the stroke and at the end of stroke. It also slows down if writer is feeling hesitation in writing or taking a pause. Sampling rate and resolution should be so high that the sampled data points represent the true pen trace correctly. Naturally, the selection of suitable level of sampling rate and resolution depends on the writing speed and the scale of the meaningful pen trace features. If sampling rate is too low, odd corners will be introduced on the sampled pen trace and some of the real corners and miniscule trace features can be missed.

3.2 Preprocessing Phase

The main objective of the preprocessing steps is to normalize words and remove variations that would otherwise complicate recognition and reduce the recognition rate. [S. A. Husain *et al.*, 2007]. Preprocessing of handwriting data is done prior to the application of shape recognition algorithms. It is observed that Data collected by digitizing devices in the form of captured stroke is usually noisy. In order to reduce noise some form of smoothing is often applied to the captured strokes. Most of the preprocessing techniques also resample the data so that points are equidistant in (x, y) space rather than time [S. D. Connell and A. K. Jain, 2002]. This noise exists in the input text in the form of sharp edges, non-centered text, uneven sizes of text and missing points in text trajectories due to high speed of handwriting and slants in characters [H. Beigi *et al.*, 1994; S. Jeager *et al.*, 2003].

Preprocessing consists of noise elimination, data reduction, and shape normalizations. The commonly used noise reduction techniques are smoothing, filtering, wild point correction, and stroke connection [C. C. Tappert, *et al.*, 1990]. As the quality of input devices is getting better, trajectory noise becomes less influential and simple smoothing operations will suffice [C. L. Liu *et al.*, 2004].

The preprocessing phase to which we are improving here consists of five steps *Size normalization and centering of stroke, Interpolation of missing points, Smoothing of stroke, Slant correction of stroke, Resampling of points of stroke*. In the present work, we have concentrated on first three steps namely, size normalization and centering of stroke, interpolation of missing points and smoothing of stroke. Visual comparison of two different methods for interpolating missing points has also been done. We have implemented first three preprocessing steps in *Java programming language*. Fig. 3.2 shows original data image which is inputted to program. Fig.3.3 shows Screen shot of Net Bean output file containing collected data of handwritten character shown in Fig.3.2

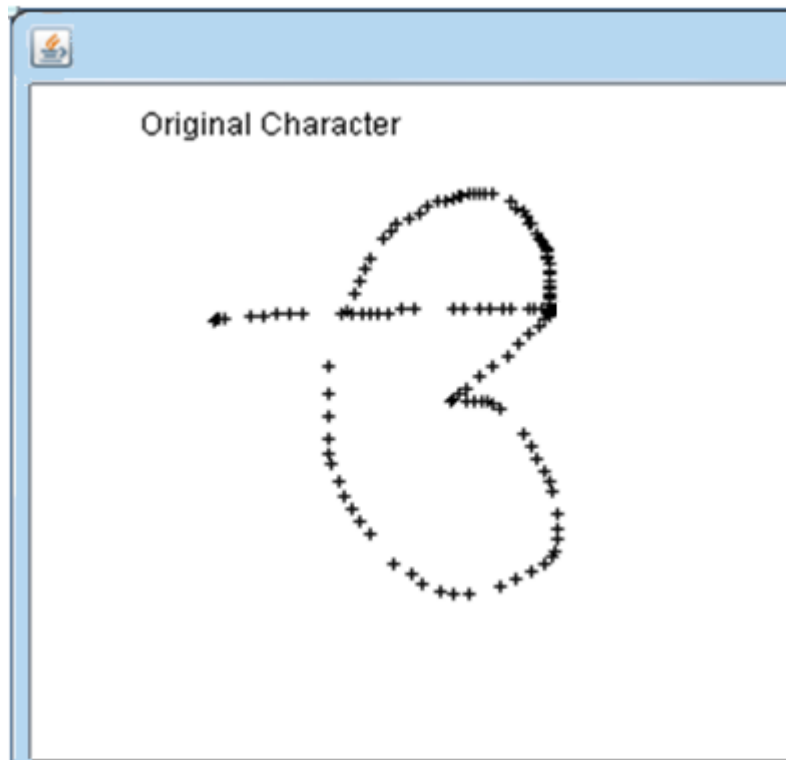


Fig. 3.2: Handwritten character before preprocessing.

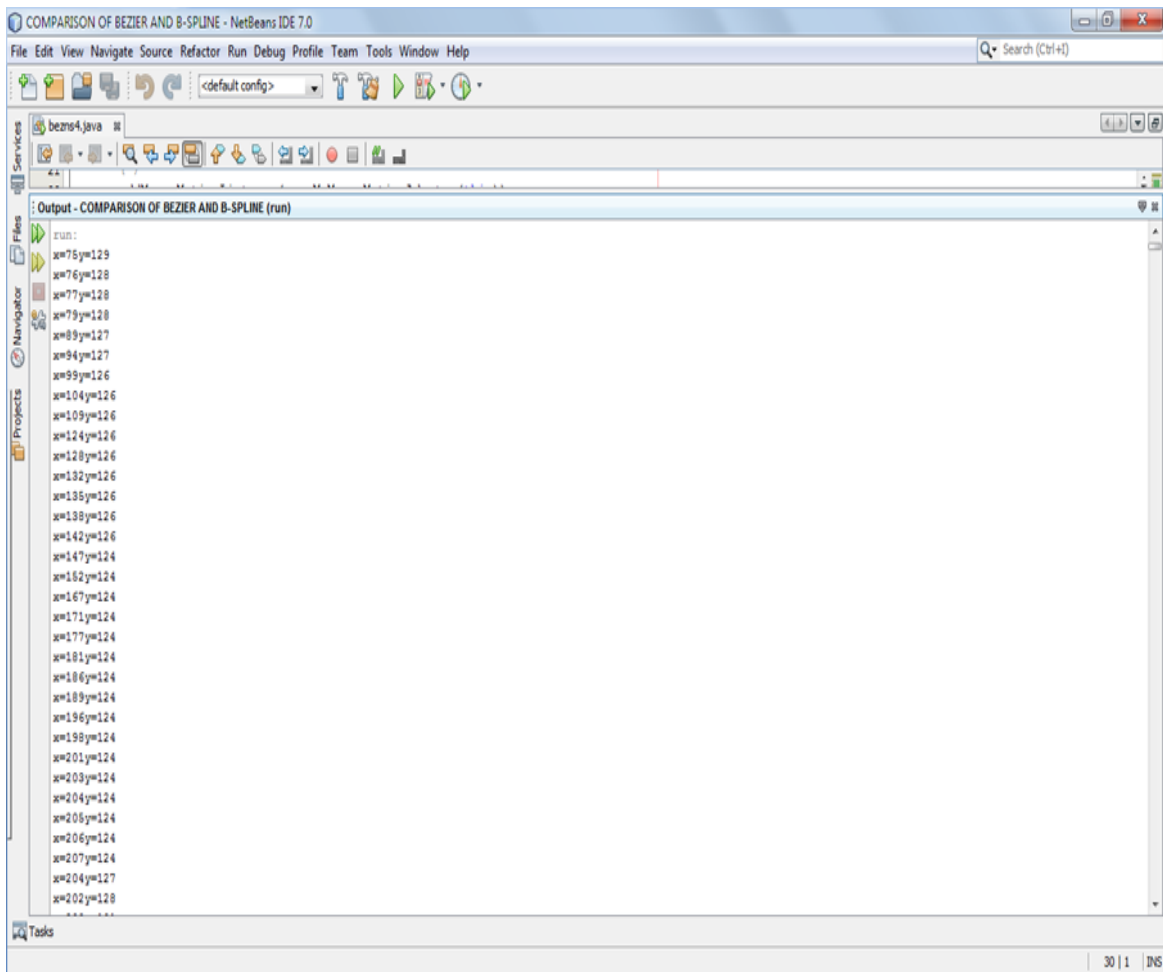


Fig. 3.3: Screen shot net bean output file containing collected data point of original character.

As, we have already discussed in section 3.1, that collected data is storage of pen movements in online handwriting recognition. These movements appear at various positions on viewport and joining these positions in first-cum-first-serve basis shows the appearance of drawn text. A character may consist of single or multiple strokes. The list formed in data collection includes nodes, where each node includes two fields, namely, point and stroke number. Here, the point represents x and y coordinates of view port and stroke number represents identity and sequential order of stroke. Also, stroke number helps in identifying similar points, gaps and crossings. In Fig. 3.1, for character ‘ Θ ’ if stroke 1 and stroke 2 include S_1 and S_2 points respectively, then the size ' n ' of the list will be S_1+S_2 .

As mentioned in section 3.1, the pen movement consists of three functions, namely, PenDown, PenMove and PenUp. When one presses, moves, lifts the pen up consecutively, and more than one point collected, the stroke number is incremented. PenMove function stores movements of pen on writing pad. The points of the list are denoted

by $P_i(x, y)$, $i = 1, 2, 3, \dots, n$; where n is total number of points in the list. For sake of brevity, we have used P_i for the point $P_i(x, y)$ in various algorithms in this Chapter. We also denote the x -coordinate of $P_i(x, y)$ by P_{ix} and its y -coordinate by P_{iy} . PenUp indicates end of stroke and this process of storing the points is repeated till the last stroke. The data collected in this way is segmented at stroke level.

3.2.1 Size Normalization and Centering of Stroke

Size normalization depends on how user moves the pen on writing pad. Centering is required when pen is moved along the border of writing pad. Each user has different writing size but for recognition system some standardized size is required. Firstly, we change the characters to some standard size by size normalizing. Size normalization adjusts the character size to a fix standard [C. C. Tappert *et al.*, 1990], size normalization is done by translating each data point of a character to a box of a standard size. This is accomplished by first finding the box that encloses the character with its original size and then computing the width and length of the box. Later these values are used to find (x, y) coordinate values of newly produced data points in the standard sized box, that correspond to each data point from the original box. The complete algorithm for size normalization and centering of stroke is presented below.

In this algorithm, origin of the frame of reference is taken as (x_0, y_0) and the set of pixels in which a Gurmukhi character is drawn is given by $\{(x, y): 0 \leq x \leq l_x, l_y \leq y \leq 0\}$, where l_x and l_y are the lengths in x and y directions. It may be noted that there are n pixels in a Gurmukhi character.

Algorithm 3.1

1. Set $l_x = 200(\text{pixels})$, $l_y = 200(\text{pixels})$.
2. $P_{ix} = P_{ix} \times (l_x/L_x)$, $P_{iy} = P_{iy} \times (l_y/L_y) \quad \forall$ points P_i in list $i = 1, 2, \dots, n$.
3. $P_{ix} = P_{ix} \pm x_0$, $P_{iy} = P_{iy} \pm y_0 \quad \forall$ points P_i in list $i = 1, 2, \dots, n$.

This algorithm normalizes the stroke in size and places it in the centre of fixed frame as depicted in Fig. 3.4. With this algorithm, we retain original aspect ratio of the character. Fig. 3.5 contains the character of Fig. 3.2 after size normalization and centering.

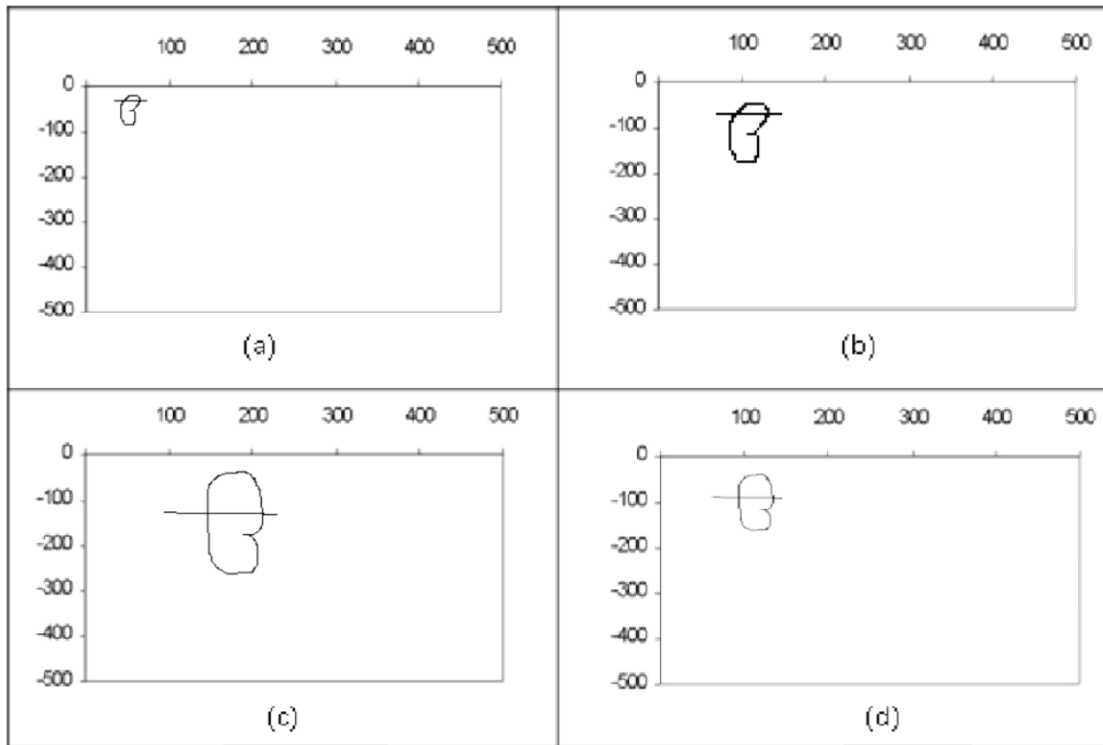


Fig. 3.4(a): Input character of size smaller than 200×200 pixels.

Fig. 3.4(b): Transformation of character (given in Fig. 3.4(a)) after size normalization and centering.

Fig. 3.4(c): Input character of size larger than 200×200 pixels.

Fig. 3.4(d): Transformation of character (given in Fig. 3.4(c)) after size normalization and centering.

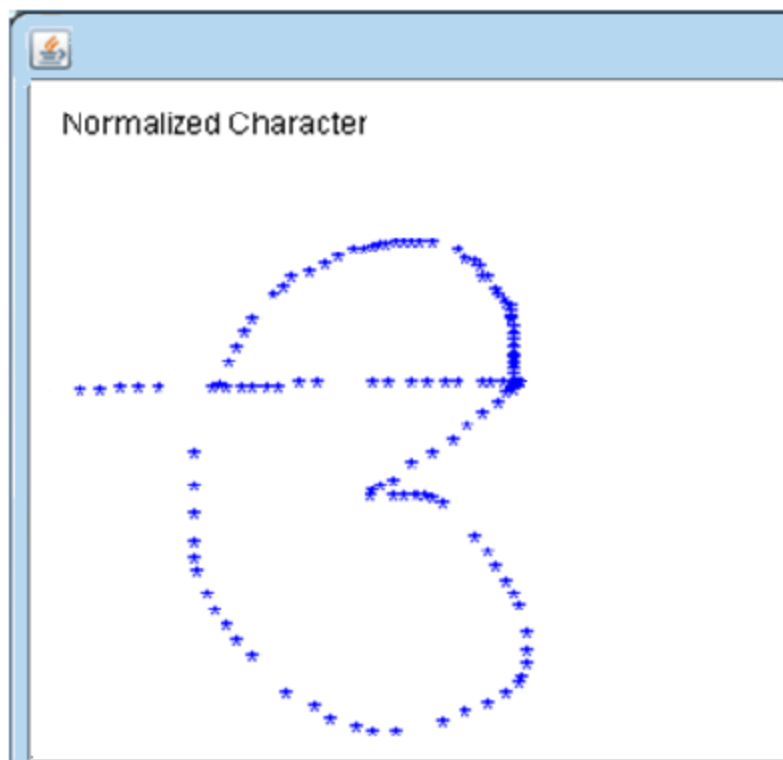


Fig. 3.5: Handwritten character after size normalization and centering.

3.2.2 Interpolation of Missing Points

As, already discussed in section 3.1 when user writes with high speed then the stroke has many missing points which makes recognition difficult. So we interpolate the missing points before recognition process. These missing points can be calculated using various techniques such as Bezier interpolation and B-spline interpolation [M. Unser *et al.*, 1993]. In the present study, we have compared the two methods, i.e. Bezier interpolation method and B-spline interpolation method.

3.2.2.1 Bezier Interpolation Method

In piecewise Bezier interpolation technique, a set of consecutive four points is considered for obtaining the Bezier curve. The next set of four points gives the next Bezier curve. By continuing this process, we get curve which has many new interpolated points.

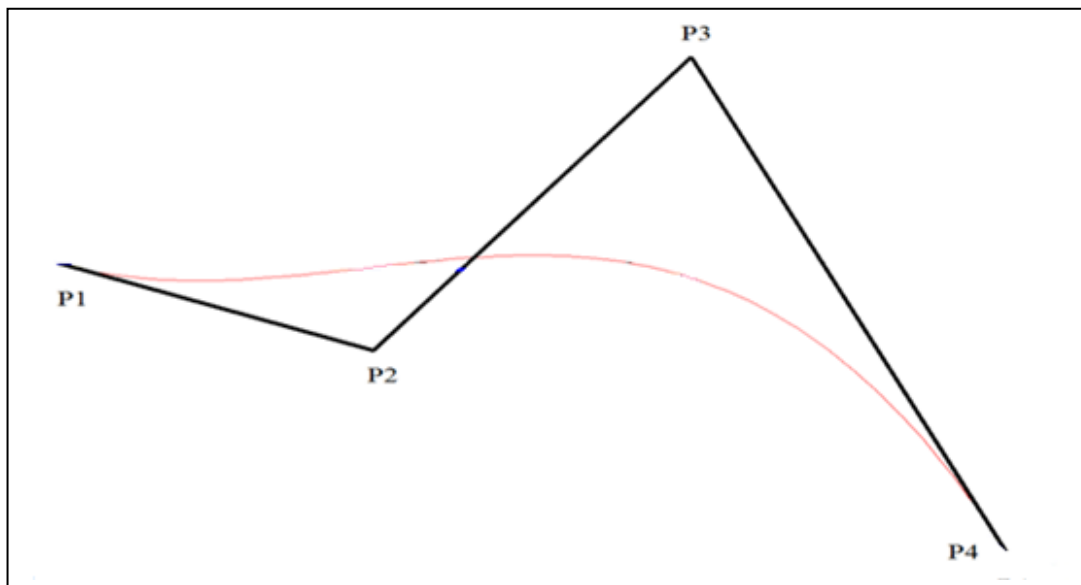


Fig. 3.6: Bezier interpolation.

In Fig. 3.6 the curve connecting the original data points P1, P2, P3 and P4 is shown in black. The approximating Bezier curve is shown in red. Bezier algorithm used for interpolation of missing points is given below.

Algorithm 3.2

1. Create an empty list L for storing the points generated from the *Bezier* function.
2. Set t = number of strokes in the list and set $k = 1$.
3. Repeat step 4 for each stroke k , until $k \leq t$.
4.
 - (a) Calculate m as the total number of points in the current stroke.
 - (b) If $(m \geq 4)$ then
 - CALL *Bezier* ($P_i, P_{i+1}, P_{i+2}, P_{i+3}$) \forall points $P_i, i = 1, 2, \dots, m-3$
 - Else
 - Set $k = k+1$.
 - End if
 - (c) Update list L by incorporating the new points as the consecutive points obtained through *Bezier* function.
 - (d) Set $k = k+1$.
 5. Exit.

function Bezier ($P_i, P_{i+1}, P_{i+2}, P_{i+3}$)

1. u is a variable such that $0 \leq u \leq 1$.
2. Set $u = 0.1$ and $\Delta u = .01$.
3. Repeat steps 4 and 5 until $u \leq 1$.
4. Calculate x coordinate of new point as

$$(1-u)^3 P_{ix} + 3u(1-u)^2 P_{(i+1)x} + 3(1-u)u^2 P_{(i+2)x} + u^3 P_{(i+3)x} ,$$
 And calculate y coordinate of new point as

$$(1-u)^3 P_{iy} + 3u(1-u)^2 P_{(i+1)y} + 3(1-u)u^2 P_{(i+2)y} + u^3 P_{(i+3)y} ,$$
5. Set $u = u + \Delta u$.
6. Return.

Fig. 3.6 contains the character of Fig. 3.2 after interpolating missing points.

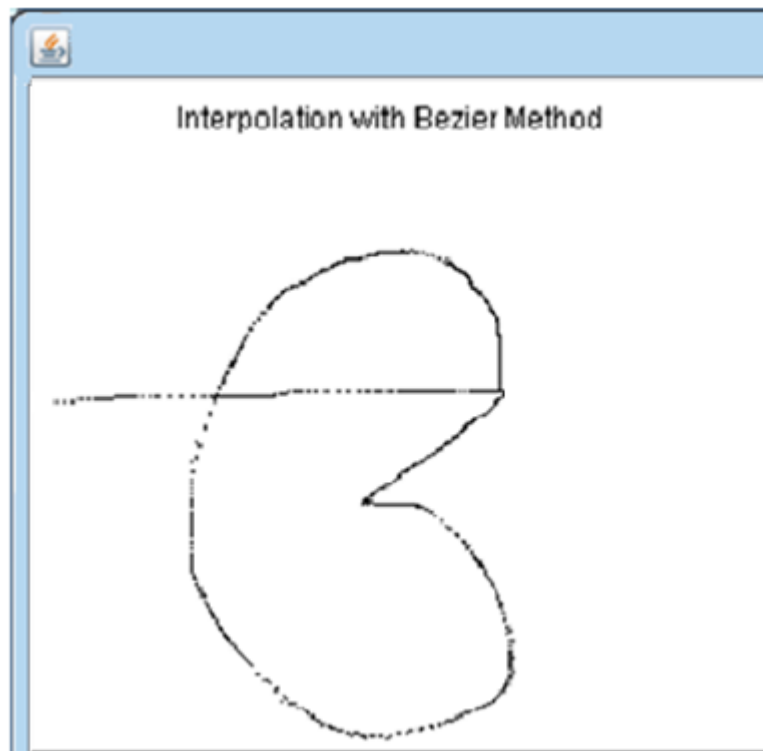


Fig. 3.7: Handwritten character after Bezier interpolation.

3.2.2.2 B-spline Interpolation Method

B-spline curves share many important properties with Bezier curves, because the former is a generalization of the later. Moreover, B-spline curves have more desired properties than Bezier curves. B-spline interpolation method is more flexible and pleasing to work with, which is the reason why it has become more part of almost every serious graphics development environment [F. Andersson, 2003]. B-spline curve of degree m with n control points consist of $n - m$ Bezier curve segments. Unlike Bezier interpolation method where algorithm is applied repeatedly on different sets of n points, in B-spline we apply algorithm to all $n+1$ control points simultaneously.

Algorithm 3.3

B-spline interpolation method for $n+1$ points of the curve

1. Set $P_{-1} = P_{-2} = P_0$
2. Set $P_{n+1} = P_{n+2} = P_n$
3. Do for $i = 0$ to $n-1$

Do for $u = 0$ to 1 STEP 0.1

Compute new x co-ordinate as

$$\frac{(1-u)^3}{6}P_{(i-1)x} + \frac{(3u^3-6u^2+4)}{6}P_{ix} + \frac{(-3u^3+3u^2+3u+1)}{6}P_{(i+1)x} + \frac{u^3}{6}P_{(i+2)x},$$

Compute y co-ordinate as

$$\frac{(1-u)^3}{6}P_{(i-1)y} + \frac{(3u^3-6u^2+4)}{6}P_{iy} + \frac{(-3u^3+3u^2+3u+1)}{6}P_{(i+1)y} + \frac{u^3}{6}P_{(i+2)y},$$

PLOT (new x , new y).

END DO (u).

END DO (i).

4. Exit.

In the Fig. 3.8 P1, P2, P3 and P4 are the given data points which are connected by black curve. Red curve shows B-spline curve.

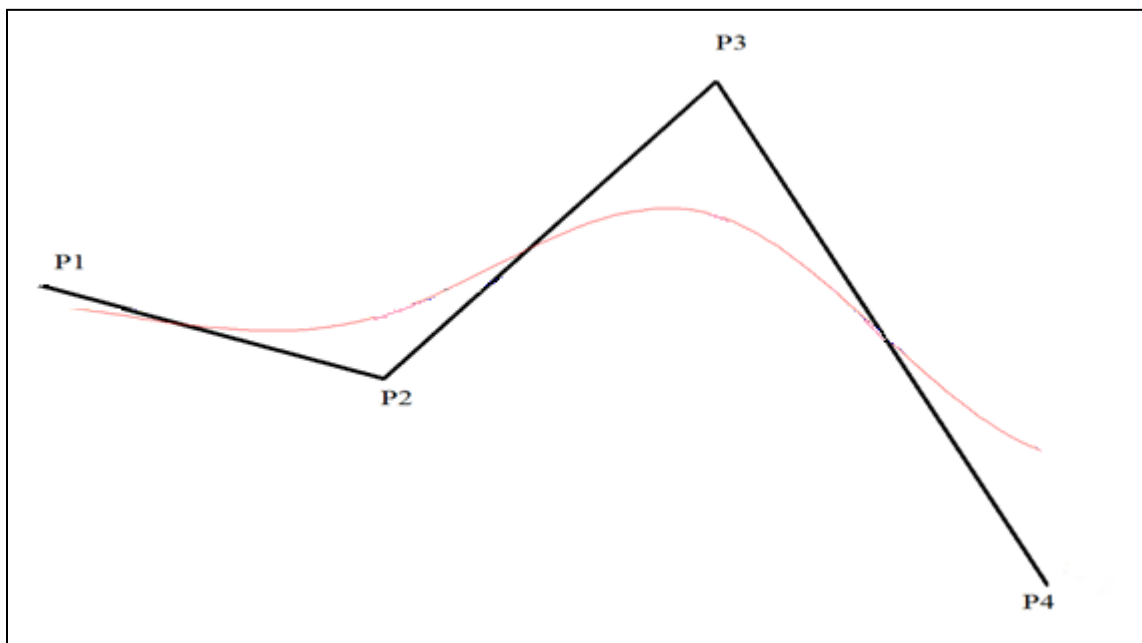


Fig. 3.8: B-spline interpolation

Fig. 3.9 depicts the character in Fig.3.2 after interpolation of missing points using B-spline interpolation method. By comparing this Fig.3.9 with Fig.3.7, we can easily observe that the number of interpolation points are more in case of B-spline interpolation method than Bezier interpolation method.

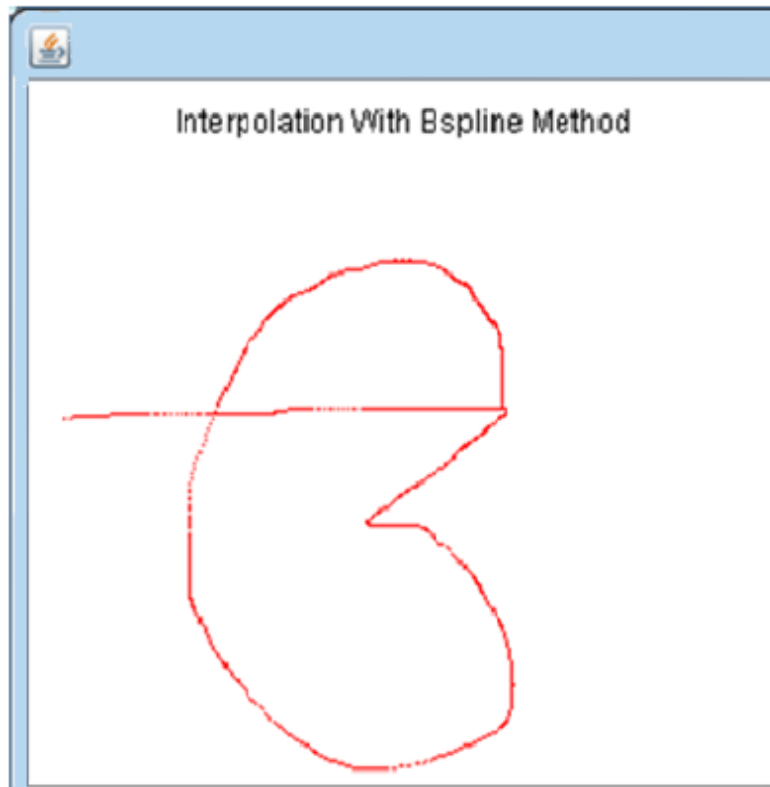


Fig. 3.9: Handwritten character after B-spline interpolation.

3.2.3 Smoothing of Character

Flickers exist in handwriting because of individual handwriting style and the hardware used. These flickers can be removed by modifying each point of the list with mean value of k - neighbors and the angle subtended at k^{th} position from each end [Kavallieratou *et al.*, 2002]. Fig. 3.10 depicts how 2-neighbors from each side can be considered for this purpose. In this figure five points of the list, generated in the previous step, have been used for smoothing of the stroke. The point P_i has been modified with the help of points P_{i-2} , P_{i-1} , P_{i+1} and P_{i+2} . It is worth mentioning here that if we consider three points then it will not affect the nature of stroke and if we consider more than five points then we tend to lose the nature of stroke in terms of sharp edges. Algorithm 3.4 contains the steps that have been used for smoothing of a stroke.

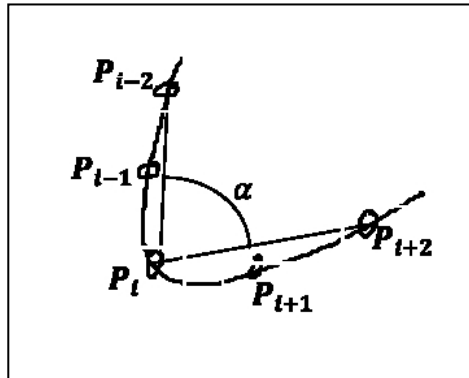


Fig. 3.10: Formation of angle α at point P_i .

Algorithm 3.4

1. Set $t =$ number of strokes in the list and set $k = 1$.
2. Repeat step 3 for each stroke k , until $k \leq t$.
3.
 - (a) Calculate m as the total number of points in the current stroke k .
 - (b) Repeat steps (c) and (d) \forall points $P_i, i = 3, 4, \dots, m-2$.
 - (c) Calculate $\alpha = \text{Angle} P_{i-2} P_i P_{i+2}$.
 - (d) Set $P_{ix} = (P_{(i-2)x} + P_{(i-1)x} + \alpha P_{ix} + P_{(i+1)x} + P_{(i+2)x}) / (2 \times 2 + \alpha)$.

$$\text{Set } P_{iy} = (P_{(i-2)y} + P_{(i-1)y} + \alpha P_{iy} + P_{(i+1)y} + P_{(i+2)y}) / (2 \times 2 + \alpha).$$

4. Set $k = k + 1$.
5. Exit.

Fig.3.11 shows the character in Fig.3.11 after smoothing it. Fig.3.11 (a) shows smoothing after interpolation by B-spline interpolation method and Fig. 3.11(b) shows smoothing after interpolation by Bezier interpolation method.

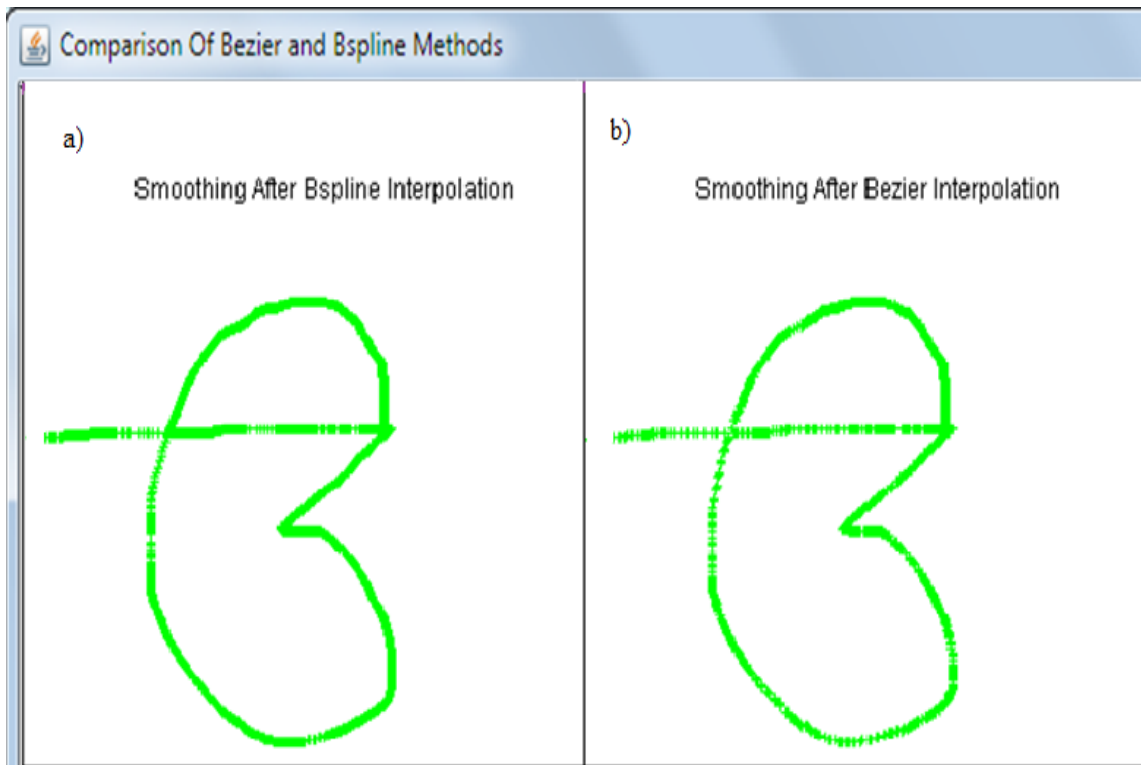


Fig 3.11: Handwritten character after smoothing.

Fig.3.11 (a) shows smoothing after interpolation by B-spline interpolation method and

Fig.3.11 (b) shows smoothing after interpolation by Bezier interpolation method.

Here, we can conclude after observing Fig.3.11 (a) and Fig.3.11 (b) that the smoothing results after B-spline interpolation are better than smoothing after Bezier interpolation.

3.4 Conclusion

Bezier interpolation method has long been used for interpolation of missing points. It has a drawback that it is non-flexible in nature, which has led to development of newer and better algorithm. This drawback is overcome by B-spline interpolation method.

It was found that B-spline interpolation method gives better results for interpolation than Bezier as the number of interpolation points are found to be more and the next consequent dependent step of smoothing was thus improved.

However, further advances in this field will be highly revolutionizing and can lead to development of better online handwriting recognition system. From the mathematical

understanding of B-spline algorithm, we can conclude that even the resampling, which is another preprocessing step, will also give better results.

References

1. A. M. Wing, "Variability in Handwritten Characters", Visible Language Vol. 13, No. 3, pp. 283-298, 1979.
2. A. Negi, K. S. Swaroop, Arun Agarwal, "A Correspondence Based Approach to Segmentation of Cursive Words", Proc. of the 3th ICDAR, Vol. 2, pp. 1034-1037, August 1995.
3. A. Sharma, "Online Handwritten Gurmukhi Character Recognition", PhD. Thesis, Thapar University, 2009.
4. C.C. Tappert, C.Y. Suen, T. Wakahara, "The State of the Art in Online Handwriting Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 8, August 1990.
5. C. C. Tappert, "Adaptive On-line Handwriting Recognition", In Proceedings of 7th International Conference on Pattern Recognition, IEEE, pp. 1004-1007, 1984.
6. C. L. Liu, S. Jaeger and M. Nakasaki, " Online Recognition of Chinese Characters: The State-of-the-Art", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, No. 2, 2004.
7. D. Guillevic and C. Y. Suen, "Cursive Script Recognition- a Sentence Level Recognition Scheme", Proceedings of IWFHR, pp. 216-223, 1994.
8. D. Perry, "Handheld Computers (PDAs) in Schools", Becta ICT Reseach, March 2003.

9. F. Alimoglu,” Combining Multiple Classifiers for Pen-based Handwritten Digit Recognition”, Master’s Thesis, Bogaziei University, 1994.
10. F. Andersson,” Bezier and B-spline Technology”, Umea university Sweden, 2003.
11. G. A. Fink, T. Plotz,” Unsupervised Estimation of Writing Style Models for Improved Unconstrained Off-line Handwriting Recognition”, Proceedings of 10th Int. Workshop on Frontiers in Handwriting Recognition held at La Baule, France, 2006.
12. H. Beigi, K. Nathan, G. J. Clary and J. Subhramonia, “Size Normalization in Unconstrained Online Handwritng Recognition”, Proceedings ICIP, pp. 169-173, 1994.
13. H. Beigi, “Pre-processing and Dynamics of Online Handwriting Data, Feature Extraction and Recognition”, Proceedings of fifth IWFHR, pp. 255-258, 1996.
14. I. Pavlidis, R. Singh and N. P. Papanikolopoulos, “An Online Handwritten Note Recognition Method Using Shape Metamorphosis” Proceedings of fifth International Conference on Document Analysis and Recognition, Vol. 2, pp. 914- 918, 1997.
15. J. J. Brault, and R. Plamondon, “Segmenting Handwritten Signatures at Their Perceptually Important Points” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No. 9, pp. 953-957,1993.
16. J. R. Ward and T. Kuklinski, “A Model for Variability Effects in Hand Printing With Implications for Design of Handwriting Character Recognition Systems”, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 18, No. 3, pp. 438-451, 1988.
17. J. Subrahmonia, T. Zimmerman, “Pen Computing: Challenges and Applications”, Proceedings of 15th International Conference on Pattern Recognition, Vol. 2, pp. 60-66, 2000.

18. Kavallieratou, N. Fakatakis, G. Kolkkinakis, “An Unconstrained Handwriting Recognition System”, International Journal of Document Analysis and Recognition, Vol. 4, No. 4, pp. 226-242, 2002.
19. K. Zhang, I. Pratikakis, J. Cornelis, and E. Nyssen, “Using Landmarks to Establish a Point-to-point Correspondence Between Signatures Pattern Analysis and Applications”, Vol. 3, No. 1, pp. 69-75, 2000.
20. L. Simoncini, Zs.M.Kovacs-V, “A System for Reading USA Census’90 Hand-Written Fields”, Proc. of the 3th ICDAR, Vol. 2, pp. 86-91, Aug. 1995.
21. M. K. Brown and S. Ganapathy, “Preprocessing Technique for Cursive Script Word Recognition”, Vol. 16, pp. 447-456, 1983.
22. M. Kobayashi, S. Masaki, O. Miyamoto, Y. Nakagawa, Y. Komiya, and T. Matsumoto, “RAV (Reparametrized Angle Variations) Algorithm for Online Handwriting Recognition”, International Journal on Document Analysis and Recognition Vol. 3, No.1, pp. 181-191, 2001.
23. M. Unser, A. Aldroubi, M. Eden, “B-Spline Signal Processing: part II - Efficient Design and Applications”, IEEE Transactions on Signal Processing, Vol. 41, No. 2, pp. 834-848, 1993.
24. “Pen Technologies”, IBM Pen Technologies, Available from www.research.ibm.com/electricInk/.
25. P. Seattolin, “Recognitin of Handwritten Numerals using Elastic Matching”, Master’s Thesis, Concordia University, 1995.

26. P. Slavik, V. Govindaraju, "Equivalence of Different Methods for Slant and Skew Corrections in Word Recognition Applications", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 3, pp. 323-326, 2001.
27. R. M. Bozinovic, S. N. Srihari, "Offline Cursive Script Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 2, No. 1, pp. 68-83, 1989.
28. R. Palmondon, S.N. Srihari: "Online and Offline Handwriting Recognition: A Comprehensive Survey", IEEE Transactions on pattern Analysis and Machine Intelligence, Vol. 22, No.1, 2000.
29. S. A. Husain, Asma Sajjad, Fareeha Anwar, "Online Urdu Character Recognition System", MVA2007 IAPR Conference on Machine Vision Applications, 2007.
30. S. D. Connell, and A. K. Jain, "Writer Adaptation for Online Handwriting Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 3, pp. 329-346, 2002.
31. Scott D. Connell, "Online Handwritten Recognition Using Multiple Pattern Class Models", (Doctor of Philosophy Dissertation), Department of Computer Science and Engineering, Michigan State University, 2000.
32. S. D. Connell, R. M. K. Sinha, and A. K Jain, "Recognition of Unconstrained Online Devanagari Characters", Proceedings of International Conference on Pattern Recognition, Vol. 2, pp. 368-371, 2000.
33. S. Jaeger, L. C. Liu, M. Nakagawa, "The State of Art in Japanese Online Handwriting Recognition Compared to Techniques in Western Handwriting Recognition", International Journal of Document Analysis and Recognition, Vol. 6, No. 2, pp. 75-88, 2003.

34. S. Jaeger, S. Manke, J. Reichert, and A. Waibel, "Online Handwriting Recognition: The NPen++ Recognizer", International Journal on Document Analysis and Recognition Vol.3, No. 3, pp. 169-180, March 2001.
35. S. Madhvanath, G. Kim and V. Govindaraju, "Chain Code Contour Processing for Handwritten Word Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, No. 9, pp. 928-932, 1999.
36. S. Uchida, E. Taira, H. Sakoe, "Nonuniform Slant Correction using Dynamic Programming", Proceedings of International Conference on Document Analysis and Recognition, pp. 434-438, 2001.
37. T. T. Kuklinski, "Components of Handprint Style Variability", In Proceedings of the 7th International Conference on Pattern Recognition, pp. 924-926, 1984.
38. T. Wakahara, H. Murase, K. Odaka, "Online Handwriting Recognition", Proceedings of IEEE, Vol. 80, No. 7, pp. 1181-1194, 1992.
39. V. Govindaraju, S. N. Srihari, "Paradigms in Handwriting Recognition", Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, pp. 1498 – 1503, 1997.
40. W. Guerfali and R. Plamondon, "Normalizing and Restoring Online Handwriting", Pattern Recognition, Vol. 26, No. 3, pp. 419, 1993..
41. W. Mulugeta, "OCR for Special Type of Handwritten Amharic Text ("Yekum Tsifet")", Neural Network Approach (Master's Thesis), School of Information Studies for, Africa, Addis Ababa University, 2003.

42. Yimei, K. Fumitika, M. Yasuji, S. Malayappan, "Slant Estimation for Handwritten Words by Directionally Refined Chain Code", Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition, pp. 53-62, 2000.
43. Y. Suen, A. L. Koerich and R. Sabourin, "Lexicon-Driven HMM Decoding for Large Vocabulary Handwriting Recognition with Multiple Character Models", International Journal of Document Analysis and Recognition, Vol. 6, No. 2, pp. 126-144, 2003.

