A Gurmukhi Script Recognition System

G S Lehal and Chandan Singh
Department of Computer Science & Engineering, Punjabi University, Patiala, INDIA.
gslehal@mailcity.com, *chandan@pbi.ernet.in*

Abstract

In this paper, a system for recognition of machine printed Gurmukhi script is presented. Research in the field of character recognition of Gurmukhi script faces major problems mainly related to the unique characteristics of the script like connectivity of characters on the headline, a large number of similar characters and two or more characters in a word having intersecting minimum bounding rectangles. The recognition system presented in this paper operates at sub-character level. The segmentation process breaks a word into sub-characters and the recognition phase consists of classifying these sub-characters and combining them to form Gurmukhi characters. A set of very simple and easy to compute features is used and a hybrid classification scheme consisting of binary decision trees and nearest neighbours is employed. A recognition rate of 96.6% at the processing speed of 175 characters/second was achieved on clean images of text without employing any post-processing technique.

1. Introduction

Optical Character Recognition (OCR) is the process of converting scanned images of machine printed or hand-written text into a computer processable format. The practical importance of OCR applications, as well as the interesting nature of the OCR problem, have led to great research interest and measurable advances in this field. But these advances are limited to English, Chinese and Arabic languages[1-3] and there has been very limited reported research on OCR of the scripts of Indian languages[2]. Some of the papers dealing with machine recognition of Indian language scripts have been presented in[4-7]. To the best of our knowledge there has been no reported published research on OCR of Gurmukhi script though some research papers on pre-processing and classification techniques for OCR of Gurmukhi script are reported [8-10] and perhaps this is the first paper on Gurmukhi script recognition. Gurmukhi script is used primarily for the Punjabi language, which is the world's 14th most widely spoken language.

2. Characteristics of Gurmukhi Script

The inadequate research on OCR of Gurmukhi script can be attributed in part to the special characteristic of the script. Some of the properties of the Gurmukhi script are:

i. Gurmukhi script is cursive and the alphabet consists of 41 consonants, 12 vowels and 2 half characters, which lie at the feet of consonants (Fig 1).

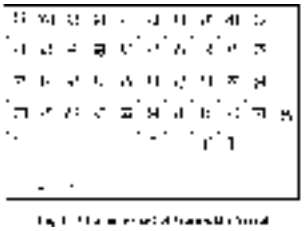
ii. Most of the characters have a horizontal line at the upper part. The characters of words are connected mostly by this line called head line and so there is no vertical inter-character gap in the letters of a word

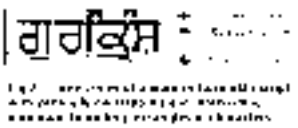
iii. A word in Gurmukhi script can be partitioned into three horizontal zones (Fig 2). The upper zone denotes the region above the head line, where vowels reside, while the middle zone represents the area below the head line where the consonants and some sub-parts of vowels are present. The middle zone is the busiest zone. The lower zone represents the area below middle zone where some vowels and certain half characters lie in the foot of consonants.

iv. The bounding boxes of 2 or more characters in a word may intersect or overlap vertically. As for example in Fig 2 the bounding boxes of ਿ and ਕ interesct and the bounding boxes of ਕ and ੰ overlap vertically.

v. There are lot of topologically similar character pairs in Gurmukhi script. Some of the similar pairs are (ਟ and ਦ, ਤ and ੜ, ਬ and ਥ, ਤ and ੜ, ਾ and ਾ , ਸ and ਸ਼, ਖ and ਖ਼, ਜ and ਜ਼, ਢ and ਫ਼, ਗ and ਗ਼)

All these above properties complicate the segmentation and recognition of Gurmukhi text.





3. Proposed Recognition System

As in other scripts, the main phases of the proposed character recognition system for Gurmukhi script are: (1)Pre-processing (2)Segmentation (3)Recognition

Proceedings of the International Conference on Pattern Recognition (ICPR'00)
1051-4651/00 $10.00 @ 2000 IEEE

3.1 Pre-processing

In the preprocessing stage skew correction, thinning and rectifying the shape of headline is performed.

3.1.1 Skew detection and correction The scanned image is often skewed due to mis-alignment of the document with the scanner axis. The method proposed by Lehal and Dhir[9], which is based on projection profile, has been used in our current work for deskewing the input image

3.1.2 Thinning Thinning is an essential pre-processing step whose main task is reducing patterns to their skeletons. For our present work we have used the thinning algorithm as given by Datta and Parui [11].

3.1.3 **Smoothening the headline** The headline is usually distorted after thinning. It could be broken or some of the black pixels may get shifted up or down or it could have rough edges (Fig 3b). The headline plays a very important role in segmentation and recognition and so its shape has to be improved so that it appears as a smooth single straight line. The broken parts of the headline are joined, the dislocated pixels of headline are shifted to the headline and the pixels, which are not part of any character are deleted.

3.2 Segmentation

The unique physical structure of Gurmukhi word such as connectedness of most of the characters of a word at headline and vertically overlapping characters make the segmentation process more complicated as compared to other scripts.

In our present work, the segmentation process is performed in three successive stages : line segmentation, word segmentation and character segmentation. For line and word segmentation horizontal and vertical projection profiles are respectively used.

For character segmentation, first of all the position of the headline is identified in the word by looking in the upper half of the word for the horizontal row with maximum pixel density. Instead of segmenting the word into characters it is segmented into connected components or sub-symbols, where each sub-symbol corresponds to the connected portion of the character lying in one of the three zones. For example, the symbol ੀ is broken into two sub-symbols | and ˆ, where | is present in middle zone and ˆ is present in upper zone, similarly the symbol ਗ is partitioned into connected sub-symbols ਰ and | . Since all the consonants are glued along the headline, so for checking connectedness, the headline is not considered as part of the symbol and two sub-symbols connected by headline are considered as unconnected as in case of ਗ. Table 1 lists all the sub-symbols of Gurmukhi character set. The connected components or sub-symbols are generated by grouping together black pixels which are 8 connected to one another. The information related to each sub-symbol about its shape i.e. the pixels that make up the sub-symbol, the zone in which the sub-symbol lies and its position in the 2-dimensional space is stored in an array of structures. This information is used in the later stage for classification and combining the sub-symbols to form Gurmukhi characters. The word image of Fig3c is segmented into sub-symbols(Fig 3d) using the above discussed segmentation technique

Table 1 : Sub-symbols of Gurmukhi script used for segmentation and recognition

Symbol	Sub-symbols	Symbol	Sub-symbols
ੳ	ਖ and ⌒	ਖ਼	ਖ and .
ਗ	ਰ and ।	ਫ਼	ਫ and .
ਸ਼	ਮ and .	ਗ਼	ਰ , । and .
ਜ਼	ਜ and .	ਲ਼	ਲ and .
ਿ	। and ˆ	=	- and -
ੀ	। and ˆ	Rest of Gurmukhi symbols	Gurmukhi symbols with their headlines stripped off

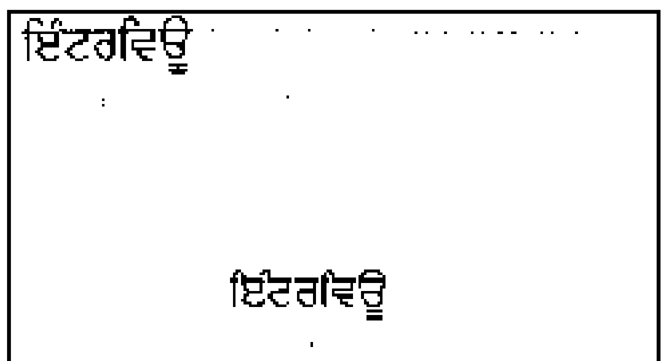


Fig 3 : Recognition of a Gurmukhi word. (a) Scanned image of word (b) Skeletonized image of word (c) Image of word after improvement in shape of headline (d) Segmented sub-symbols (e) Sub-symbols recognized and combined to form characters and word

3.3 Recognition

The recognition of Gurmukhi characters is carried out in following stages:

1. Feature extraction
2. Classification of sub-symbol using extracted features and zonal information.

3. Combining and converting the sub-symbols to form Gurmukhi symbols.

3.3.1 Feature Extraction After a careful analysis of shape of Gurmukhi characters for different fonts and sizes, two set of features were developed. The first feature set, called *primary feature set,* is made up of robust and font and size invariant features. The purpose of *primary feature set* is to precisely divide the character set into smaller subsets which can be easily managed. The cardinality of these subsets varies from 1 to 8. The *secondary feature set* is then used on these subsets.

The features used in *primary feature Set* are :

1. **Number of junctions with the headline equals 1 (P_1)** : It can be noted that each sub-symbol merges with the headline at one or more than one point. As e.g. the sub-symbol of ਫ has one junction while the sub-symbol of ਪ has two junctions with the headline. This feature has been used to divide the complete Gurmukhi sub-symbol set into almost 2 equal sized subsets.
2. **Presence of Sidebar (P_2)**: The presence or absence of sidebar is another very robust feature for classifying the sub-symbols. This feature is true if a vertical line is present on the rightmost side of the sub-symbol else it is false. As e.g. this feature is true for sub-symbol of ਮ, while it is false for sub-symbol of ਕ.
3. **Presence of a loop (P_3):** The presence of a loop in the sub-symbol is another important classification feature. This feature is true for sub-symbol of ਰ but is false for sub-symbol of ਧ , since the sub-symbol of ਧ is obtained after stripping off the headline from ਧ and so the loop of ਧ formed along the headline is lost in the sub-symbol.
4. **Loop along the headline (P_4)**: This feature is true if the sub-symbol forms a loop with the headline. Examples of sub-symbols with this feature are sub-symbols of ਲ and ਧ.

The second feature set, called *secondary feature set,* is a combination of local and global features which are aimed to capture the geometrical and topological features of the sub-symbols.

The features used in *Secondary Feature Set* are :

1. **Number of endpoints and their location (S_1)** : A black pixel is considered to be an end point if there is only one black pixel in its 3 x 3 neighbourhood. The number of endpoints as well as their positions in terms of 9(3*3) quadrants are considered.
2. **Number of junctions and their location (S_2)** : A black pixel is considered to be a junctions if there are more than two black pixels in its 3 x 3 neighbourhood. The number of junctions as well as their positions in terms of 9(3*3) quadrants are considered.
3. **Horizontal Projection Count (S_3)** : Horizontal Projection Count represented as $HPC(i) = \sum_i F(i, j)$, where F(i,j) is a pixel value (0 for background and 1 for foreground) of a document image, and i and j denote vertical and horizontal coordinates of the pixel respectively, when the image's top left corner is set to F(0,0).
4. **Right Profile depth (S_4)** : The maximum depth of the right projection profile of sub-symbol image is stored as percentage with respect to total width of the box enclosing the sub-symbol image.
5. **Left Profile Depth (S_5 and S_6)** : The maximum depth of the upper half (S_5) and lower half (S_6)of the left projection profile is stored as percentage with respect to total width of the box enclosing the sub-symbol image.
6. **Right and Left Profile Direction Code (S_7, S_8)** : The profiles are scanned from top to bottom and local directions of the profile at each pixel are noted. Starting from current pixel, the pixel distance of the next pixel in left, downward or right direction is noted. The cumulative count of movement in the three directions is represented by the percentage occurrences with respect to the total number of pixel movement and stored as a 3 component vector with the three components representing the distance covered in left, downward and right directions respectively.
7. **Aspect Ratio (S_9) :** Aspect ratio is obtained by dividing the sub-symbol height by its width.

3.3.2 Classification The classification stage is the main decision making stage of an OCR system. The classification stage uses the features extracted in the previous stage to identify the text segment according to preset rules. Binary classifier trees and nearest neighbour classifiers are the two most commonly used classifiers. The binary tree classifier has the advantage of speed but is sensitive to noise. The nearest neighbour is less sensitive to noise and can easily be trained for more fonts and sizes but is computationally space and time intensive. The *primary features* are noise, font and size invarient and so the binary classification tree was used for them. The *secondary* features were found to be sensitive to font in some cases and so the nearest neighbour classifier with a variant sized vector was used. The disadvantage of speed for nearest neighbour classifier is taken care of by operating on small sized subsets which contained at most 8 distinct sub-symbols. Thus a hybrid classification scheme, which has combined the relative advantages of binary classifier and nearest neighbour classification methods and taken care of their disadvantages, has been used in our present work. This has resulted in a very fast classification scheme.

The classification of sub-symbols proceeds in the following 3 stages:

1. Use zonal information to classify the sub-symbol into one of 3 sets containing sub-symbols lying in upper (set 11), middle(super set of sets 1-10) and lower zones (set 12) respectively (Table 2).

2. If the sub-symbol is in middle zone then assign it to one of the sets 1-10 of table 2 using binary classifier tree.
3. Recognize the sub-symbol classified to one of sets of using nearest neighbour classifier and feature set assigned for that particular set.

The complete feature set used for classification is tabulated in table 2. The primary feature vector is obtained from binary classifier tree and the i_{th} component of the vector is 1 or 0 depending on if the $\mathbf{P_i}$ primary feature is true or false for that character set. X denotes don't care condition.

Table 2 : Feature sets for Classification

Character Sub-Set	Primary Feature Vector	Secondary Features
1. ਚ ਰ	[1, 1, 1, X]	S_1 S_2 S_3
2. ਹ ਜ ।	[1, 1, 0, X]	S_1 S_2 S_3
3. ਕ ਙ ਛ ਠ ਡ ਢ ਫ ਭ	[1, 0, 1, X]	S_1 S_2 S_3 S_4 S_5 S_6 S_7 S_8
4. ਟ ਣ ਤ ਦ ਨ ਵ ਝ	[1, 0, 0, X]	S_1 S_2 S_3 S_4 S_5 S_6 S_7 S_8
5. ਖ	[0, 1, 1, 1]	-
6. ਥ ਬ	[0, 1, 1, 0]	S_5 S_8
7. ਅ ਘ ਪ ਮ	[0, 1, 0, 1]	S_1 S_2 S_3 S_5
8. ਸ ਧ ਯ	[0, 1, 0, 0]	S_1 S_2 S_3 S_5
9. ੳ	[0, 0, 1, X]	-
10. ੲ ਞ ਵ ਲ	[0, 0, 0, X]	S_1 S_2 S_3 S_4 S_7 S_8
11. [illegible]	[X, X, X, X]	S_1 S_7 S_8
12. [illegible]	[X, X, X, X]	S_9

3.3.3 Combining Sub-symbols In this last stage of recognition of characters, the information about coordinates of bounding box of sub-symbols and context is used to merge some of the sub-symbols. The sub-symbols are then converted to Gurmukhi characters. For combining the sub-symbols, three queues are maintained for storing the sub-symbols lying in upper, middle and lower zone. The recognized sub-symbols are sorted in ascending order on their position on the x-axis and then pushed into their respective queues. Next a sub-symbol is removed from each of the queue and then using some decision rules, the vertically overlapping sub-symbols are combined and converted into character(s).

4. Experiments and Results

All the algorithms were coded in C++ and run under WINDOWS 98 Operating system on Pentium Celeron 333 Mhz system. Four fonts of Gurmukhi script were used for training : Punjabi, Amrit-Lipi, GurmukhiLys and PN-TTAmar. Four point sizes were used 12, 14, 18 and 24. About 100 pages of documents from laser print outs, books and forms were scanned using an HP Scanjet P5 scanner at 300 dpi. A recognition rate of 96.6% was achieved on clean images. The total processing time for pre-processing (excluding skew detection), segmentation and recognition was 175 characters/second.

5. Conclusion

An attempt has been for the development of a system for recognition of Gurmukhi script. The segmentation process breaks a word image into 3 zones and the character image in each of these zones is segmented into sub-character or sub-symbol images. A multi-stage classifier is used to classify the sub-symbols and they are then combined using heuristics and finally converted to characters. A recognition rate of 96.6% at a processing speed of 175 c/s was obtained in experimental results.

6. References

[1] J. Mantas, "An overviw of character recognition methodologies", *Pattern Recognition,* Vol. 19, pp 425-430 (1986).

[2] V. K. Govindan and A. P. Shivaprasad, "Character recognition – A survey ",*Pattern Recognition,* Vol. 23, pp 671-683 (1990).

[3] B. Al-Badr and S.A. Mahmoud, "Survey and bibliography of Arabic optical text recognition", *Signal Processing,* Vol. 41, pp. 49-77(1995).

[4] K. Sethi and B. Chatterjee, "Machine recognition of constrained hand printed Devanagari", *Pattern Recognition,* Vol. 9, pp. 69-75(1977).

[5] R. Chandrasekaran, M. Chandrasekaran and G. Siromony, "Recognition of Tamil, Malayalam and Devanagri characters", *J. Inst. Electron. Telecom. Engg. (India),* Vol. 30, pp. 150-154 (1984).

[6] R. M. K. Sinha, "Rule based contextual post processing for Devnagari text recognition", *Pattern Recognition,* Vol. 20, pp. 475-485(1985).

[7] B. B. Chaudhuri and U. Pal, "A complete printed Bangla OCR system", *Pattern Recognition,* Vol. 31, pp 531-549 (1998).

[8] Ajay Goyal, G S Lehal and S S Deol, "Segmentation of Machine Printed Gurmukhi Script" , Proceedings 9th International Graphonomics Society Conference, Singapore, pp. 293-297 (1999).

[9] G S Lehal and Renu Dhir, "A Range Free Skew Detection Technique for Digitized Gurmukhi Script Documents" , Proceedings 5th International Conference of Document Analysis and Recognition, Bangalore, pp. 147-152, (1999).

[10] G. S. Lehal and C. Singh, "Feature extraction and classification for OCR of Gurmukhi script", *Vivek,* Vol. 12, No. 2, pp. 2-12 (1999).

[11] A. Dutta and S.K. Parui , "A robust parallel thinning algorithm for binary images", *Pattern Recognition,* Vol. 27, pp. 1181-1192 (1994).

1051-4651/00 $10.00 @ 2000 IEEE