A Complete OCR System for Gurmukhi Script

G. S. Lehal[1] and Chandan Singh[2]

[1] Department of Computer Science and Engineering
Thapar Institute of Engineering & Technology, Patiala, India
[2] Department of Computer Science and Engineering
Punjabi University, Patiala, India

Abstract. Recognition of Indian language scripts is a challenging problem. Work for the development of complete OCR systems for Indian language scripts is still in infancy. Complete OCR systems have recently been developed for Devanagri and Bangla scripts. Research in the field of recognition of Gurmukhi script faces major problems mainly related to the unique characteristics of the script like connectivity of characters on the headline, characters in a word present in both horizontal and vertical directions, two or more characters in a word having intersecting minimum bounding rectangles along horizontal direction, existence of a large set of visually similar character pairs, multi-component characters, touching characters which are present even in clean documents and horizontally overlapping text segments. This paper addresses the problems in the various stages of the development of a complete OCR for Gurmukhi script and discusses potential solutions.

1 Introduction

Research on Devanagri, Tamil and Telugu optical text recognition started around mid 70s[1-4] and recently complete OCR systems for Indian scripts such as Devanagri and Bangla[5-6] have been developed. The research work for Gurmukhi OCR has started only in mid 90s and is still in its infancy stage. To the best of our knowledge this is the first time that a complete OCR solution for Gurmukhi script has been developed and presented.

The word 'Gurmukhi' literally means from the mouth of the Guru. Gurmukhi script is used primarily for the Punjabi language, which is the world's 14th most widely spoken language. Gurmukhi script like most of other Indian language scripts is written in a nonlinear fashion. The width of the characters is also not constant. The vowels getting attached to the consonant are not in one (or horizontal) directions, they can be placed either on the top or the bottom of consonant. Some of the properties of the Gurmukhi script are:

- Gurmukhi script is cursive and the Gurmukhi script alphabet consists of 41 consonants and 12 vowels and 3 half characters, which lie at the feet of consonants (Fig 1).

T. Caelli et al. (Eds.): SSPR&SPR 2002, LNCS 2396, pp. 358-367, 2002.

- 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 and formation of merged characters is a norm rather than an aberration in Gurmukhi script
- 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.
- The bounding boxes of 2 or more characters in a word may intersect or overlap vertically.
- The characters in the lower zone frequently touch the characters in the middle zone.

ੳ	ਅ	ੲ	ਸ	ਹ	ਕ	ਖ	ਗ	ਘ	ਙ	
ਚ	ਛ	ਜ	ਝ	ਞ	ਟ	ਠ	ਡ	ਢ	ਣ	
ਤ	ਥ	ਦ	ਧ	ਨ	ਪ	ਫ	ਬ	ਭ	ਮ	
ਯ	ਰ	ਲ	ਵ	ੜ	ਸ਼	ਜ਼	ਖ਼	ਫ਼	ਗ਼	ਲ਼
ੌ	ੋ	ੇ	ੈ	ੱ	ੰ	ਂ	ਿ	ੀ	ਾ	
ੁ	ੂ	੍ਰ	੍ਹ	੍ਵ						

Fig. 1. Character set of Gurmukhi script

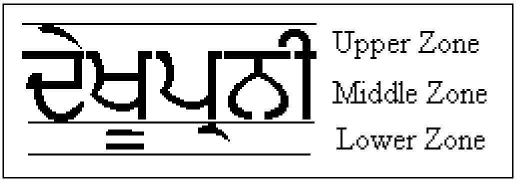


Fig. 2. Three zones of a word in Gurmukhi script

In our current work, after digitization of the text, the text image is subjected to pre-processing routines such as noise removal, thinning and skew correction. The thinned and cleaned text image is then sent to the text segmenter, which segments the text into connected components. Next these connected components are recognized and combined to form back characters. Finally post processing is carried out to refine the results.

2 Text Segmentation

Gurmukhi script is a two dimensional composition of consonants, vowels and half characters which require segmentation in vertical as well in horizontal directions. Thus the segmentation of Gurmukhi text calls for a 2D analysis instead of the commonly used one-dimensional analysis for Roman script. Besides the common segmentation problems faced in Indian language scripts, Gurmukhi script has other typical problems such as horizontally overlapping text segments and touching characters in various zonal positions in a word.

To simplify character segmentation, since it is difficult to separate a cursive word directly into characters, a smaller unit than a character is preferred. In our current work, we have taken an 8-connected component as the basic image representation throughout the recognition process and thus instead of character segmentation we have performed *connected component segmentation*. The segmentation stage breaks up a word and characters which lie above and below the headline into connected components and the classifier has been trained to recognize these connected components or sub-symbols (Table 1). It is to be noted that the headline is not considered the part of the connected component.

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

Symbol	Sub-symbols	Symbol	Sub-symbols	Symbol	Sub-symbols
ਉ	ੳ and ⌒	ਖ਼	ਖ and .	ੀ	| and ^
ਗ	ਰ and |	ਫ਼	ਫ and .	ਿ	| and ^
ਸ਼	ਸ and .	ਗ਼	ਰ , | and .	=	- and -
ਜ਼	ਜ and .	ਲ਼	ਲ and .	ਓ	ੳ and ⌐
Gurmukhi Characters in upper zone	Same shapes retained	Gurmukhi Characters in lower zone	Same shapes retained	Rest of Gurmukhi characters in middle zone	Gurmukhi characters with their headlines stripped off

A combination of statistical analysis of text height, horizontal projection and vertical projection and connected component analysis is performed to segment the text image into connected components. We have employed a 5 phased segmentation scheme. These phases, which are described in detail in [7] are:

1) Dissect the text image into text strips using valleys in the horizontal projection profiles. Each of these strip could represent either one text row or only the upper or lower zones of a text row or more than one text row (Fig. 3).

2) Perform statistical analysis to automatically label the text strips as multi strip, core strip, upper strip or lower strip, depending on if the text strip contains more than one text row, one text row, upper zone or lower zone of a text row respectively. As for example, in Fig. 3 strip nos. 2 and 3 are lower strips, strip no. 1 is core strip, strip no. 12 is upper strip and strip no. 15 is multi strip.
3) Decompose the text strips into smaller components such as words and connected components using vertical projection profile analysis. In case of multi strip, the strip is first split into individual text rows using the statistics based on the average height of a core strip and next each text row is split into words. In case of upper and lower strips we just have sub parts of upper and lower zone vowels respectively. A connected component analysis is carried out to obtain the connected components in these strips.
4) Split words into connected components in case of core strip and multi strip. For obtaining the connected components the headline is rubbed off and after segmentation it is restored back.
5) Detect and segment touching characters in connected components. This phase is explained briefly in the following subsection.

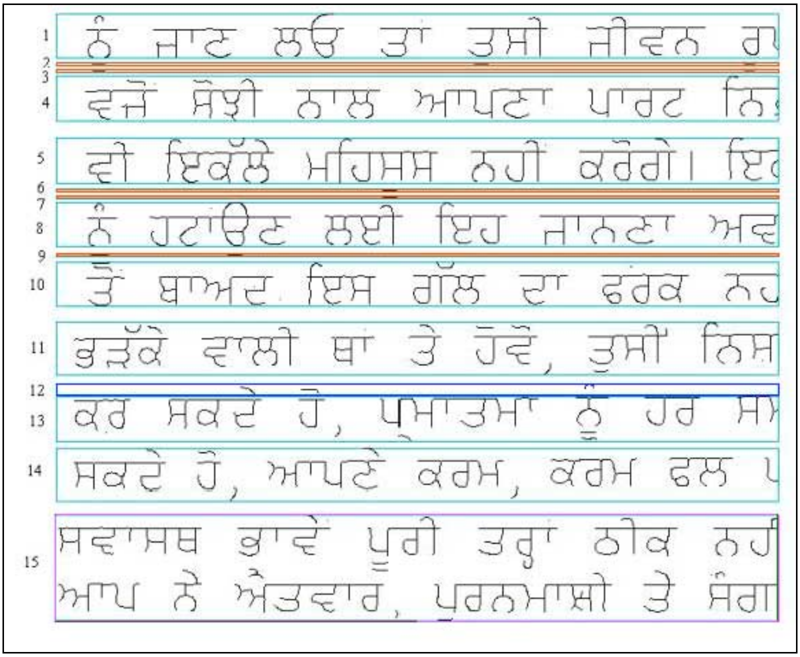


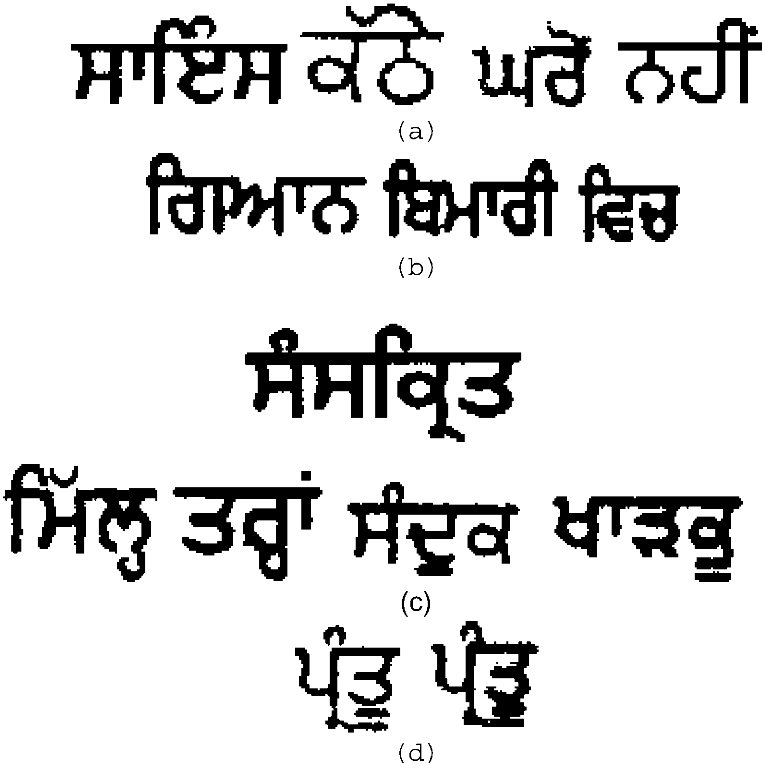
Fig. 3. A sample image split into text strips by horizontal projection

2.1 Touching Characters

It was observed that touching characters were frequently present even in clean machine printed texts. As already mentioned, segmentation process for Gurmukhi script proceeds in both x and y directions since two or more characters of a word may be sharing the same x coordinate. Therefore, for the segmentation of touching characters in Gurmukhi script, the merging points of the touching characters have to be determined both along the x and y axes. These touching characters can be categorized as follows:

(a) Touching characters in upper zone
(b) Touching characters in middle zone
(c) Lower zone characters touching with middle zone characters
(d) Lower zone characters touching with each other

Fig. 4 shows examples of touching characters for these categories. The existing techniques for detecting and segmenting touching characters were used and certain heuristics were developed to solve the segmentation problem for Gurmukhi characters. The details are discussed elsewhere[7]. Table 2 displays the percentage frequency of occurrence of the touching characters in the three zones and the accuracy rate of detection and correction.



(a)

(b)

(c)

(d)

Fig. 4. Examples of touching characters a) touching characters in upper zone, b)touching characters in middle zone, c) Lower zone characters touching with middle zone characters, d) Lower zone characters touching with each other

Table 2. Accuracy rate of detecting and segmenting touching characters

Type of touching characters	% of occurrence	% of correct detection and segmentation
Touching/merging upper zone vowels	6.90%	92.5%
Touching middle zone consonants	0.12%	72.3%
Touching middle zone and lower zone characters	19.11%	89.3%
Touching lower zone characters	0.03%	95.2%

3 Recognition Stage

The main phases of the recognition stage of OCR of Gurmukhi characters in our present work are:

1. Feature extraction.
2. Classification of connected components using extracted features and zonal information.
3. Combining and converting the connected components to form Gurmukhi symbols.

3.1 Feature Extraction

After a careful analysis of shape of Gurmukhi characters for different fonts and sizes, two sets 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 set of characters lying in middle zone into smaller subsets which can be easily managed. The cardinality of these subsets varies from 1 to 8. The Boolean valued features used in the primary feature set are:

1. Is the number of junctions with the headline one
2. Is a sidebar present
3. Is there a loop
4. Is a loop formed with headline

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 characters and efficiently distinguish and identify the character from a small subset of characters. The secondary feature set consists of following features:

1. Number of endpoints and their location (S_1)
2. Number of junctions and their location (S_2)

3. Horizontal Projection Count (S_3)
4. Right Profile depth (S_4)
5. Left Profile Upper Depth (S_5)
6. Left Profile Lower Depth (S_6)
7. Left and Right Profile Direction Code ($S_{7,}$ S_8)
8. Aspect Ratio (S_9) :
9. Distribution of black pixels about the horizontal mid line (S_{10})

3.2 Classification

In our present work, we have used a multi-stage classification in which the binary tree and nearest neighbour classifiers have been used in a hierarchical fashion. The classification scheme for the Gurmukhi characters proceeds in the following 3 stages:

(a) Using zonal information, we classify the symbol into one of the three sets, lying either in upper zone, middle zone or in lower zone. The upper zone and lower symbols are assigned to set nos. 11 and 12 of Table 3 respectively.

(b) If the symbol is in the middle zone, then we assign it to one of the first ten sets of Table 3 using primary features and binary classifier tree. At the end of this stage the symbol has been classified into one of 12 sets including the sets for characters in upper and lower zones.

(c) Lastly, the symbol classified to one of the 12 sets of Table 3 is recognized using nearest neighbour classifier and the feature set of secondary features assigned for that particular set.

Table 3. Secondary feature set for classification of character sets

Set #	Character Set	Primary Feature Vector	Classification 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 ੜ	[1, 0, 0, X]	S_1 S_2 S_3 S_4 S_5 S_6 S_7 S_8 S_{10}
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 ਝ ਞ ਲ	[0, 0, 0, X]	S_1 S_2 S_3 S_4 S_7 S_8
11	ਿ ੀ ੇ ੈ ੰ ੋ ਂ	[X, X, X, X]	S_1 S_7 S_8 S_{10}
12	ੁ ੍ ੂ	[X, X, X, X]	S_8 S_9

The complete feature set used for classification is tabulated in Table 3. 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 P_i primary feature is true or false for that character set. X denotes that the feature is not needed for classification. Thus for example, the primary feature vector for set number 1 is [1, 1, 1, X], which means that all the characters in this set have one junction with the headline, have a side bar and have a loop.

3.3 Merging 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 and convert the sub-symbols to Gurmukhi characters. It is to be noted that most of the sub-symbols can as such be converted to equivalent character (Table 1). It is only in some typical cases where a character may be broken into more than one sub-symbol that some rules have to be devised to merge these sub-symbols. For example, if the sub-symbol in middle zone is ਰ and the next sub-symbols in middle and upper zones are | and ⌒respectively, then if the upper sub-symbol is vertically overlapping with one or more middle zone sub-symbols, then these sub-symbols might represent one of the character combinations ਰੀ, ਰਿ or ਰੋਾ.The information regarding the overlapping of the upper and middle zone connected components (CCs) is used to identify the characters represented by the CCs. Thus, if ⌒ is overlapping with both ਰ and | then the CCs combine to form ਰੀ . If ⌒ is overlapping with only | then the CCs combine to form ਰਿ and if ⌒ is overlapping only with only ਰ then the CCs combine to form ਰੋਾ.

4 Post Processing

For the post processing we have used a Punjabi corpus, which serves the dual purpose of providing data for statistical analysis of Punjabi language and also checking the spelling of a word. Punjabi grammar rules are also incorporated to check for illegal character combinations such as presence of two consecutive vowels or a word starting with a forbidden consonant or vowel. The main steps in the post processing phase are:

1. Create word frequency list from the Punjabi corpus. The list stores the frequency of occurrence of all words present in the corpus.
2. Partition the word frequency list into smaller sub lists based on the word size. We have created 7 sub-lists corresponding to word sizes of two, three, four, five, six, seven and greater than seven characters.
3. Generate from each of the sub-list an array of structures which is based on visually similar characters. We say that two characters are visually similar if they belong to the same set of Table 3. Similarly two words are visually similar if each character is corresponding position is visually similar. Thus the words m^C and p>r are visually similar since the first character in both the words belongs to set 7, the second character belongs to set 11 and third character belongs to set 1. This array of visually similar words records the percentage frequency of occurrence of character in all the positions of these visually similar words. This list is combined with the confidence rate of recognition of the recognizer to correct the mistakes of the recognizer.

4. Store the twenty most commonly occurring words. Any word which is visually similar to any of these words and which is not recognized with high confidence is automatically converted to the nearest visually similar word.
5. Use Punjabi grammar rules to eliminate illegal character combinations.

These steps are explained in detail in[8].

5 Experimental Results and Conclusion

We tested our OCR on about 25 Gurmukhi text documents consisting of about 30000 characters. The documents were pages from good quality books and laser print outs in multiple sizes and fonts. We tested on font sizes in the range 10-24 point size and 8 fonts were used.

It was found that seven characters (ਹ ਖ ਘ ਙ ਞ ਢ ਯ) with a combined frequency of occurrences of 5.67% were recognized with almost 100% accuracy. Out of these the character ਹ has a high frequency of occurrence (4.2%) but in the subset 2 (Table 3), there are only two other characters for resolving the confusion and their shapes are quite different so ਹ is not confused with them. Twenty two characters with cumulative frequency of occurrences of 44.69% are recognized with more than 98% accuracy. On the lower end, eleven characters (ਲ਼, ਜ਼ ੋ ਂ ਸ਼ ਥ ਚ ੂ ਖ਼ ਫ਼ ਗ਼) with a cumulative frequency of occurrences of 10.08% have a low recognition rate of 80% or less. It is these characters which are the main bottlenecks in the performance of the OCR. It can be seen that majority of these characters are the characters with dot at their feet. The reason for this inaccuracy is that during the thinning either the dot is deleted or it gets merged with the character. Even among the characters with dot at their feet the characters ਗ਼ ਖ਼ and ਫ਼ have a far more poor recognition accuracy as compared to characters ਲ਼ ਸ਼ and ਜ਼. The reason for this is that the dot is positioned in centre for characters ਲ਼ ਸ਼ and ਜ਼ while for characters ਗ਼ ਖ਼ and ਫ਼ the dot is positioned very close to the character and so it gets easily merged on thinning. The characters ਚ and ਥ have low recognition accuracy as they are very closely resembling with charactersਰ and ਬ respectively and are often confused with them. The characters ੋ and ੂ , have their strokes often joined together or touching with other characters which makes it difficult to recognize them. The character, ਂ (*bindi*), which is similar to a dot and is present in the upper zone is also difficult to recognize. There were two type of errors produced: a) Deletion - The character *bindi* would be removed during the scanning and binarization process or by the thinning algorithm. In many cases the *bindi* character would be merged with other symbols in the upper zone and vanish. b)Insertion - The noise present in the upper zone would be confused with *bind*i. Sometimes an upper zone vowel would be broken into smaller components, which would generate extra *bindi* characters. The above statistics are obtained without the application of the post processor. The recognition accuracy of the OCR without post processing was 94.35%, which was increased to 97.34% on applying the post processor to the recognized text.

This is the first time that a complete multi-font and multi-size OCR system for Gurmukhi script has been developed. It has been tested on good quality images from books and laser print outs and has recognition accuracy of more than 97%. We are now working for testing and improving the performance of the OCR on newspapers and low quality text.

References

1. Govindan, V. K., Shivaprasad, A. P.: Character recognition-A review. Pattern Recognition. Vol. 23. (1990) 671-683.
2. S. N. S. Rajasekaran, S. N. S., Deekshatulu, B. L.: Recognition of printed Telugu characters. Computer Graphics and Image Processing. Vol. 6. (1977) 335-360.
3. G. Siromoney, G., Chandrasekaran, R., Chandrasekaran, M.: Machine recognition of printed Tamil characters. Pattern Recognition. Vol. 10. (1978) 243-247.
4. Sinha, R. M. K., Mahabala, H. N.: Machine recognition of Devanagari script. IEEE Trans on Systems, Man and Cybernetics. Vol. 9. (1979) 435-449.
5. Chaudhuri, B. B., Pal, U.: A complete printed Bangla OCR system. Pattern Recognition. Vol. 31. (1998) 531-549.
6. Bansal, V.: Integrating knowledge sources in Devanagri text recognition. Ph.D. thesis. IIT Kanpur (1999).
7. Lehal, G. S., Singh, C.: Text segmentation of machine printed Gurmukhi script. Document Recognition and Retrieval VIII. Paul B. Kantor, Daniel P. Lopresti, Jiangying Zhou (eds.), Proceedings SPIE, USA. Vol. 4307. (2001) 223-231.
8. Lehal, G. S., Singh, C.: A shape based post processor for Gurmukhi OCR. Proceedings 6th International Conference on Document Analysis and Recognition, Seattle, USA. (2001) 1105-1109.