Segmentation of Touching Characters in Upper Zone in Printed Gurmukhi Script

M. K. Jindal
Department of Computer Science and Applications
Panjab University Regional Centre
Muktsar, Punjab, India
+919814637188, +911642214849
manishphd@rediffmail.com

R. K. Sharma
School of Mathematics and Computer Applications
Thapar University
Patiala, Punjab, India
+919872202705
rksharma@thapar.edu

G. S. Lehal
Department of Computer Science
Punjabi University
Patiala, Punjab, India
+919815473767
gslehal@gmail.com

ABSTRACT
A new technique for segmenting touching characters in upper zone of printed Gurmukhi script has been presented in this paper. The technique is based on the structural properties of the Gurmukhi script characters. Concavity and convexity of the characters has been studied and using top profile projections, the touching characters in upper zone have been segmented. Recognition rate of 91% has been achieved for segmenting the touching characters in upper zone.

Keywords
Touching characters, upper zone, projection profiles, Gurmukhi script.

1. INTRODUCTION
As part of the Optical Character Recognition (OCR) system, character segmentation techniques are applied to word images before individual characters are recognized. The simplest way to segment the characters is to use inter-character gap as segmentation points. This technique does not work well if the text to be segmented contains touching characters [11].

Many algorithms have been proposed in the past [2, 8, 12] for segmenting touching characters in roman script. Kahan *et al.* [8] have proposed a very useful double differential function to segment the touching characters. Tsujimoto and Asada [12] constructed a decision tree for resolving ambiguity in segmenting touching characters. Casey and Nagy [2] proposed a recursive segmentation algorithm for segmenting touching characters. Hong [6] has utilized visual inter-word constraint available in a text image to split word images into pieces for segmenting degraded English language text.

Some work has also been done on segmenting the touching characters in Indian languages [1, 4, 5]. Bansal and Sinha [1] have segmented the conjuncts (one kind of touching patterns) in Devanagari script using the structural properties of the script. Garain and Chaudhuri [4, 5] have used a technique based on Fuzzy Multifactorial Analysis to segment the touching characters in Devanagari and Bangla scripts. Chaudhuri *et al.* [3] have used the principle of water overflow, from a reservoir, to segment the touching characters in Oriya script. Jindal *et al.* [7] have used the structural properties for segmenting the touching characters in middle zone of printed Gurmukhi script. Lehal and Singh [9, 10] have also proposed algorithms to segment the touching characters in upper zone of Gurmukhi script.

In this paper, we have proposed an algorithm to segment touching characters in the upper zone of printed Gurmukhi script, using top profile projections. The structural properties of the characters present in upper zone have also been used for deciding the cut columns for segmenting touching characters.

2. GURMUKHI SCRIPT
Gurmukhi syllabary initially consisted of thirty two consonants, three vowel bearers, ten vowel modifiers (including *muktā* having no sign) and three auxiliary signs. Later on, six more consonants have been added to this script. These six consonants are multi-component characters that can be decomposed into isolated parts. Besides these, some characters modify the consonants once they are appended just below to them. These are called half characters or subjoined characters. The consonants, vowel bearers, additional consonants, vowel modifiers, auxiliary signs and half characters of Gurmukhi script (jointly called sub-symbols) are given in Figure 1. In other words, a connected component after removing the headline is called a sub-symbol.

The Consonants

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

The Vowel Bearers

ੳ ਅ ੲ

Compute 2009, Jan 9, 10, Bangalore, Karnataka, India.

The Additional Consonants (Multi Component Characters)

ਸ਼ ਜ਼ ਖ਼ ਫ਼ ਗ਼ ਲ਼

The Vowel Modifiers

ੰ ੇ ੇ ੈ ੀ ਿ ਾ ੁ ੂ

Auxiliary Signs

ੱ ੰ ਂ

The Half Characters

ੵ ੍ਰ ੍ਵ

Figure 1. Characters and symbols of Gurmukhi script.

A line of Gurmukhi script can be partitioned into three horizontal zones, namely, upper, middle and lower zone. These three zones are described in Figure 2, with the help of one example word. The middle zone generally consists of the consonants. The upper and lower zones may contain parts of vowel modifiers, auxiliary sings and half characters (collectively termed as vowels or characters in this paper).

Figure 2. Gurmukhi script Word, (a) upper zone from line number 1 to 2, (b) middle zone from line number 3 to 4, (c) lower zone from line number 4 to 5.

Figures 2(a), 2(b) and 2(c) show the elements of the three zones, i.e., upper, middle and lower zones respectively. With reference to Figure 2, line number 1 is called the start line, line number 2 defines start of the headline and line number 3 defines end of the headline. Also, line number 4 is called the base line and line number 5 is called the end line.

We can divide the vowels of Gurmukhi script into following four classes.

Class 1: vowels present in upper zone only.

Class 2: vowels present both in upper and middle zone.

Class 3: vowels in middle zone only.

Class 4: vowels present in lower zone only.

Table 1 shows the actual number of vowels falling in different classes in Gurmukhi script.

Table 1. Number of vowels falling in each class

Class of vowel	Number of vowels
1	7
2	2
3	1
4	2

Figure 3. Pronunciation of name, actual shape and example words containing vowels falling in class 1.

The pronunciation of name, actual shape and example words containing the vowels falling in class 1 are given in Figure 3 and that of falling in class 2 are given in Figure 4.

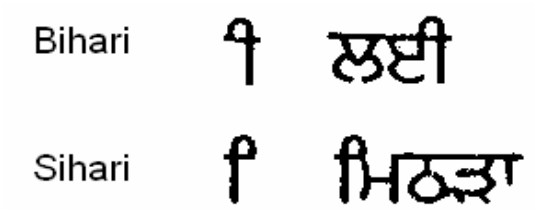

Figure 4. Pronunciation of name, actual shape and example words containing vowels falling in class 2.

Besides the above mentioned vowels falling in upper zone, there are some characters whose one stroke falls in upper zone. The pronunciation of name, the stroke of the character falling in upper zone and example word containing these characters are shown in Figure 5.

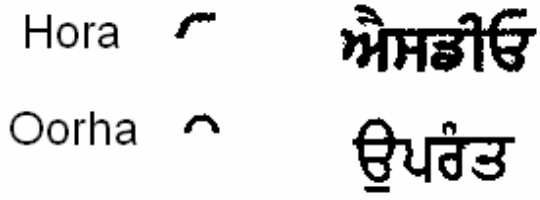

Figure 5. Pronunciation of name, actual shape and example words containing strokes of some characters in upper zone.

3. PROBLEM OF TOUCHING CHARACTERS IN UPPER ZONE

Before identifying the problem of touching characters in upper zone and proposing its solutions, we hereby give some definitions:

Definition 1. *(Horizontal projection)*: For a given binary image of size $L \times M$ where L is the height and M is the width of the image, the horizontal projection is defined by [1] as:

$HP(i), i = 1, 2, 3, \ldots, L$

where $HP(i)$ is the total number of black pixels in i^{th} horizontal row.

Definition 2. *(Vertical projection)*: For a given binary image of size $L \times M$ where L is the height and M is the width of the image, the vertical projection is defined as:

$VP(j), j = 1, 2, 3, \ldots, M$

where $VP(j)$ is the total number of black pixels in j^{th} vertical column.

3.1 Data Collection

Data collection for the purpose of proposing the segmentation algorithm was a time consuming and complex task. We selected degraded documents containing touching characters from various books and magazines as well as normal documents, faxed them, zeroxed them and scanned them at 300 dpi resolutions. About 85 such documents were scanned which contain almost 2800 touching characters, thus a sufficiently large database of touching characters has been created. The problem of touching characters in upper zone has also been identified in fine printed documents of Gurmukhi script.

3.2 Categories of the Touching Characters in Upper Zone

After carefully analyzing the database of touching characters in upper zone, it is found that on the basis of structural properties of the Gurmukhi script, various touching characters can be classified among few categories. Following three categories are being proposed for touching characters in the upper zone.

3.2.1 Category 1: bindi (ਂ) touching with other characters

By carefully analyzing the collected data, it is found that 55% of the total pairs of touching characters in upper zone fall in this category. In this category, vowel *bindi* (dot shaped) touches with other characters present in upper zone either form left or right side. Figure 6(a) contains words from Gurmukhi script in which *bindi* touches with other characters in upper zone.

3.2.2 Category 2: adhak (ੱ) touching with other characters

Approximately 35% touching characters of the total touching characters in upper zone fall in this category. In this category, *adhak* vowel touches with other characters present in upper zone. Figure 6(b) contains some example words of Gurmukhi script containing problem of *adhak* touching with other characters in upper zone.

3.2.3 Category 3: tippi (ੰ) touching with other characters

It has been seen that 10% touching characters of the total touching characters in upper zone fall in this category. In this category, *tippi* vowel touches with other characters present in upper zone. Further, it has been revealed from the analysis that the vowel *tippi* always touches with upper zone segment of the vowel ੋ . Figure 6(c) contains examples of *tippi* touching with upper zone segment of ੋ in upper zone.

Figure 6. Gurmukhi words containing touching characters in upper zone (touching characters have been marked with circles), (a) *bindi* touching with other characters, (b) *adhak* touching with other characters, (c) *tippi* touching with other characters.

4. SEGMENTATION OF TOUCHING CHARACTERS IN UPPER ZONE

For segmenting the touching characters in upper zone, we have proposed an algorithm based on the structural properties of Gurmukhi characters. It has been revealed from the structural properties of Gurmukhi characters that every character in upper zone consists of single concavity or convexity in its structure. This concept of single concavity or convexity has been used to segment the touching characters in upper zone. For detection of touching position, if a character in upper zone has a concavity followed a convexity or vice-versa; it is supposed to be the touching or merged characters, *i.e.*, candidate of segmentation. In the merged characters, whenever the first concavity (or convexity) terminates, we have put a segmentation column to segment the touching characters in upper zone. Algorithm **Segment _Upper_Outline** contains outline of the algorithm**,** while Algorithm **Segment_Upper_Detailed** contains the detailed algorithm.

Algorithm: Segment_Upper_Outline (binary matrix of character in upper zone)

begin

get binary matrix of the character in upper zone;
compute top profile of character;
for (all columns of character)
if (top profile moves upward)
concavity confirmed;

while (profile moves upward)
move to next column;
end-while
while (profile moves downward)
move to next column;
end-while
current column is segmentation column;
else
convexity confirmed;
while (profile moves downward)
move to next column;
end-while
while (profile moves upward)
move to next column;
end-while
current column is segmentation column;
end-if
end-for
end-algorithm

Algorithm: Segment_Upper_Detailed (binary matrix of character in upper zone)

BEGIN

Step 1: Using the vertical projection in upper zone identify the boundary of each character. For that whenever $VP(i) = 0$ for $i = 1, 2, 3, \ldots, L$, it is marked as the boundary of character. Let us denote the different characters as $C_1, C_2, \ldots, C_n$. Denote first column of each character as $FC_1, FC_2, \ldots, FC_n$ and last column of each character as $LC_1, LC_2, \ldots, LC_n$.

Step 2: for $k = 1$ to n performs the following operations (for each character in upper zone)**:**

Step 2.1: find the top profile of the character. For that, for $j=FC_k$ to LC_k perform the following:

Step 2.1.1: mark the row as X, in which first black pixel in j^{th} column is encountered. Now calculate $TP(j) = LR - X + 1$, where TP is top profile and LR is last row of upper zone.

Step 2.2: for $j = FC_k$ to LC_k perform the following:

Step 2.2.1: if $TP(j+1) >= TP(j)$ go to step 2.2.3(concavity) else goto step 2.2.5(convexity)

Step 2.2.2: while $TP(j+1) >= TP(j)$ & $j < LC_k$ increment j and repeat step 2.2.2

Step 2.2.3: while $TP(j+1) <= TP(j)$ & $j < LC_k$ increment j and repeat step 2.2.3.

Step 2.2.4: if $j <= LC_k$, j marks the segmentation column. Go to step 2 for next character.

Step 2.2.5: while $TP(j+1) <= TP(j)$ & $j < LC_k$ increment j and repeat step 2.2.5

Step 2.2.6: while $TP(j+1) >= TP(j)$ & $j < LC_k$ increment j and repeat step 2.2.6.

Step 2.2.7: if $j <= LC_k$, j marks the segmentation column. Go to step 2 for next character.

END.

In the proposed algorithm, the top profile of each character in upper zone as shown in Figure 7(d) is considered. While moving from left side to right side of the top profile, we look for any concavity or convexity. For example in Figure 7(d), one can see that as the column number increases while moving from left to right the number of pixels in top profile also increases. After few iterations number of pixels starts decreasing. This increase and subsequently decrease of the number of pixels represents the concave shape of the character. Now whenever the downward trend of the pixels changes its direction to upwards, that marks the segmentation column.

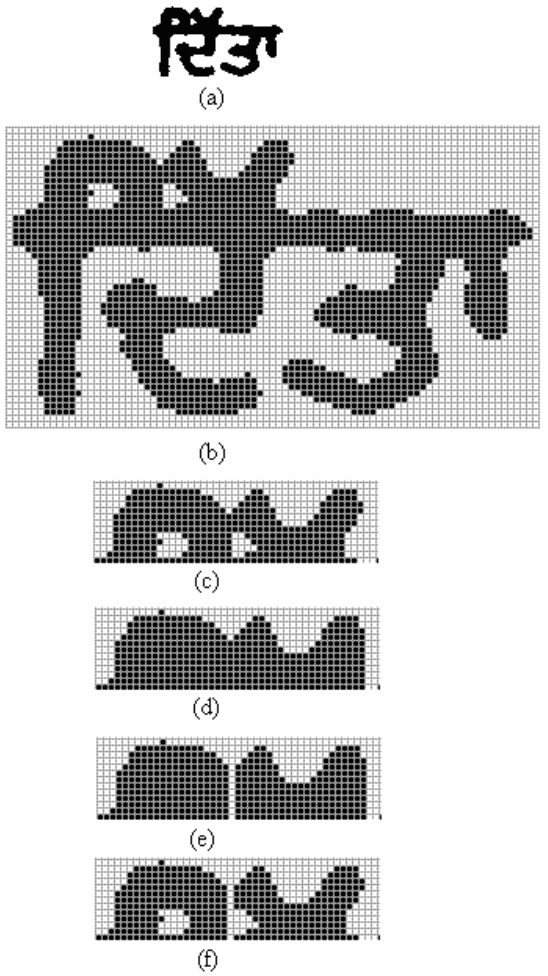

Figure 7. Segmentation of touching characters in upper zone, (a) example words, (b) extended view of example word, (c) extended view of problem area, (d) top profile of problem area, (e) segmenting columns in top profiles, (f) actual segmented characters.

The reverse happens when first character has convexity. Initially, number of pixels decreases and after few columns it starts increasing. After this increase, whenever any downward trend appears it is marked as segmentation column. Second example word in Figure 8 explains this concept.

Figure 8. Segmentation of touching characters in upper zone, (a) example words, (b) problem areas, (c) top profile of problem areas, (d) segmenting columns in top profiles, (e) actual segmented characters.

The explanation given above does not work well in the situation when *Kanoda* touches with other characters. As the shape of this character contains one little concavity followed by one little convexity, it produces incorrect segmentation. But the chances of occurring touching characters involving this character are very less.

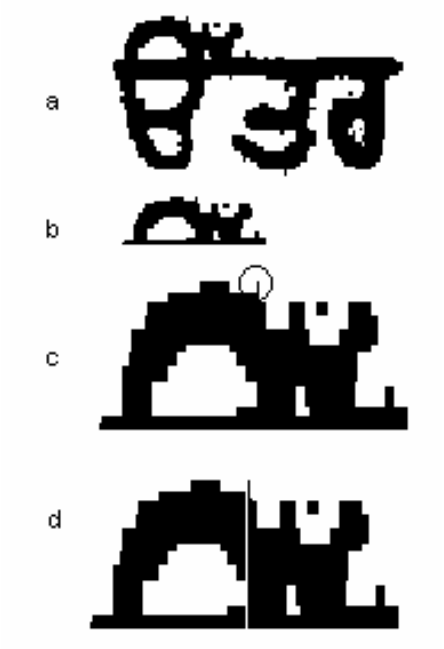

Figure 9. Effect of noise on algorithm, (a) example word, (b) problem area, (c) extended view of problem area and noise pixel encircled, (d) incorrect segmentation column.

Noise may also affect the accuracy. During finding the concavity or convexity if some noise pixels are present in such a way that it disturbs the concavity or convexity of the touching characters, it may also result in incorrect segmentation as shown in Figure 9(d).

5. RESULTS AND DISCUSSIONS

It has been found that by applying the algorithm for segmenting the touching characters in upper zone for Gurmukhi script, we have achieved about 91% accuracy. These results have been obtained by applying the proposed algorithm on 85 documents of Gurmukhi script, containing touching characters in upper zone. The touching characters in upper zone have been found even in fine printed Gurmukhi newspapers and books. So this algorithm is very important for developing OCR for both degraded text as well as fine printed text. Since the problem of touching characters in upper zone also appears in some other Indian languages, the algorithm proposed in this paper can be used to segment them as well.

6. REFERENCES

[1] Bansal, V., and Sinha, R. M. K. 2002. Segmentation of touching and fused Devanagari characters. Pattern Recognition. 35, 4, 875-893.

[2] Casey, R. G., and Nagy, G. 1982. Recursive Segmentation and Classification of Composite Character Patterns. In Proceedings of the 6th International Conference on Pattern Recognition. Munich, Germany. 1023-1026.

[3] Chaudhuri, B. B., Pal, U., and Mitra, M. 2001. Automatic Recognition of Printed Oriya Scrip. In Proceedings of the 6th International Conference on Document Analysis and Recognition. 795-799.

[4] Garain, U., and Chaudhuri, B. B. 1997. On recognition of touching characters in printed Bangla Documents. In Proceedings of the 4th International Conference on Document Analysis and Recognition. Germany. 1011-1016.

[5] Garain, U., and Chaudhuri, B. B. 2002. Segmentation of touching characters in printed Devanagari and Bangla scripts using fuzzy multifactorial analysis. IEEE Trans. on Systems, Man and Cybern. Part C. 32, 449-459.

[6] Hong, T. 1995 Degraded text recognition using visual and linguistic context. Doctoral Thesis, Computer Science Department of SUNY at Buffalo.

[7] Jindal, M. K., Lehal, G. S., and Sharma, R. K. 2005. A Study of Touching Characters in degraded Gurmukhi text. In Proceeding of World Academy of Science, Engineering and Technology. 4, 121-124.

[8] Kahan, S., Pavlidis, T., and Baird, H. S. 1987. On the recognition of printed characters of any fonts and sizes, IEEE Transactions on Pattern Analysis and Machine Intelligence. 9, 2, 274-288.

[9] Lehal, G. S., and Singh, C. 2001. Text segmentation of machine-printed Gurmukhi script. Document Recognition and Retrieval VIII, proceedings SPIE. USA. 4307, 223-231.

[10] Lehal, G. S., and Singh, C. 2001. A technique for segmentation of Gurmukhi text. Computer Analysis of Images and Patterns, Proceedings CAIP 2001, W. Skarbek

(Ed.), Lecture Notes in Computer Science. 2124, Springer-Verlag, Germany. 191-200.

[11] Lu, Y. 1995. Machine Printed Character Segmentation – An Overview. Pattern Recognition. 28, 1, 67-80.

[12] Tsujimoto, S., and Asada, H., 1991. Resolving Ambiguity in Segmenting Touching Characters. In Proceedings of the 1st International Conference on Document Analysis and Recognition. Saint-Malo, France. 701-709.