A Novel Method to Segment Online Gurmukhi Script

Manoj K.Sachan[1], G.S.Lehal[2], V.K.Jain[1]
[1]Sant Longowal Institute of Engineering & Technology, Longowal, India
[2]Punjabi University Patiala, India
Manoj K.Sachan, manojsachan@gmail.com

Abstract. Segmentation of handwritten script in general and Gurmukhi Script in particular is a critical task due to the type of shapes of character and large variation in writing style of different users. The data captured for online Gurmukhi Script consists of x,y coordinates of pen position on the tablet, pressure of the pen on the tablet, time of each point and pen down status of the pen. A novel method to segment the Gurmukhi script based on pressure, pen down status and time together is presented in the paper. In case of some characters getting over segmented due to the shape of character and user's style of writing, a method to merge the substrokes is presented. The proposed algorithms have been applied on a set of 2150 words and have given very good results.

Keywords: Stroke,Substroke,Merging,Segmentation,Gurmukhi,Devanagari

1 Introduction

The process of online handwriting recognition consists of steps such as preprocessing of the user handwriting, segmentation of script into meaningful units or shapes, recognition of shapes, and post processing to refine the results of recognition [1, 2]. Gurmukhi Script like Devanagari script consists of large number of strokes with high variation. The number and type of strokes constituting the character may vary from writer to writer. For example as shown in fig 1a and 1b, the character ਲ (lalla) is written with three strokes and two strokes by two different writers respectively.

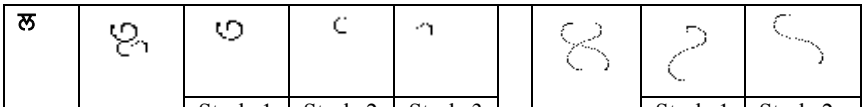


Fig1 a) ਲ(lalla) = stroke1+stroke2+stroke3 b) ਲ(lalla) = stroke1+stroke2

Similarly, the headline (shirorekha) which is an important feature of Gurmukhi/Devanagari script is generally drawn from left to right as one stroke as shown in fig 2b , but in some cases it may be drawn in parts. For example , the word ਯੱਕਾ (yakka) is written using two headline stroke as shown in fig 2a and with one headline stroke as shown in fig 2b.

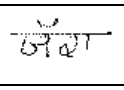
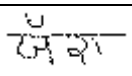
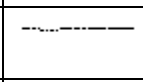
a							
	First Writer	Headline Stroke	Stroke1	Stroke2	Stroke3	Stroke4	Headline Stroke
b							
	Second Writer	Stroke1	Stroke2	Stroke3	Stroke4	Headline Stroke	

Fig 2 Words with single or multiple headlines

Thus, there is high variation in the number and type of strokes constituting the Gurmukhi word. In the past, Niranjan Joshi et.al [10] have used syntactic and structural approaches to recognize and segment shirorekha and vowels from isolated Devanagari characters. Anuj Sharma et.al [14] have used point based method to segment isolated Gurmukhi character. In this method, each stroke is observed for number of points and if number of points exceed 300 (an empirically observed value) and the direction of stroke at that point is less than 90 degree , the stroke is segmented at that point. We have taken complete word instead of isolated character in online Gurmukhi Script for segmentation. In subsequent sections, the techniques of segmenting online Gurmukhi Script are described. [3, 4, 12].

2 Characteristics of Gurmukhi

The script of Gurmukhi is cursive and has 41 consonants, 12 vowels and 2 half characters which lie at the feet of consonants .The consonants, vowels and half characters of Gurmukhi are shown in Fig 3.

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	ਖ਼	ਗ਼	ਲ਼	ਫ਼
ਜ਼	◌ਾ	◌ਿ	◌ੀ	◌ੁ	◌ੂ	◌ੇ	◌ੈ	◌ੋ	◌ੌ
◌ੰ	◌ੱ	◌ੲ	◌ੵ	◌ਂ					

Fig 3 Gurmukhi character set

Most of the characters have a horizontal line (shirorekha) at the upper part. The characters of word are connected mostly by this line called the head line (shirorekha). Hence there is no vertical inter-character gap in the letters of the word. A word in the Gurmukhi can be partitioned into three horizontal zones. 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 parts of vowels are present. The middle zone is the busiest zone. The lower zone represents the area below the middle zone where some vowels and some half characters lie at the foot of consonants. The three zones are shown in Fig 4. [3, 4, 12].

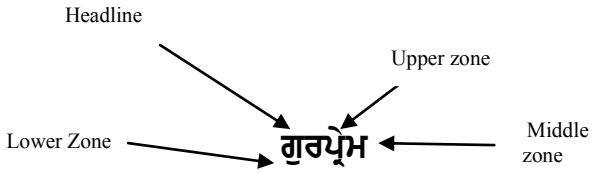


Fig 4 Three zones in Gurmukhi Script

3 Data Capture and Preprocessing

The data is collected through a Wacom Tablet and Digitizer Pen. As the pen touches the tablet and moves on the surface it sends data packets. The data captured in each packet includes pressure of the pen, position of the pen in terms of x, y coordinates, button which captures the touch of pentip on the surface and time which captures system time of capturing each packet. Fig 5 shows some of the samples of the data captured. In general, the Gurmukhi script is written left to right. Each word is composed of a sequence of strokes. Each stroke consists of the trace of pentip from pendown to penup. From segmentation point of view, the pre-processing steps followed are rescaling and duplicate point removal.

Rescaling : In rescaling, the x,y coordinates are rescaled to origin. The pressure and time are also recalculated w.r.t to their minimum values. The button value remains unchanged.

Duplicate Point Removal: Each stroke contains a large number of duplicate points that occur either in the beginning of the stroke or when the stylus or digitizer pen is stationary on the surface of the tablet. In duplicate point removal, the point (x_i,y_i) is compared with point (x_{i+1},y_{i+1}). If $(x_i == x_{i+1})$ and $(y_i == y_{i+1})$ then point (x_{i+1},y_{i+1}) is removed from the stroke.[11]. The pressure and button component at (x_i,y_i) remains unchanged. The time component is averaged and is stored at point (x_i,y_i).

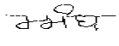
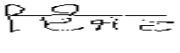
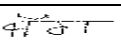
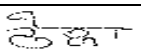
a		b	
c		d	

Fig 5 Samples of words captured

4 Proposed Segmentation Algorithm

The segmentation technique is based on the idea to separate the strokes based on the way user write them on the tablet. The segmetation algorithm consists of two phases namely Extraction of Strokes and Merging of Strokes.

4.1 Extraction of Strokes

In this step, the strokes in the word as written by the user are extracted. The packet captured by the digitizer tablet consists of information such as button,pressure and time apart from x-y position.When the user places the digitizer pen on the tablet surface, the pressure increases and when the pen is taken off the tablet surface it approaches to zero. Thus, a pressure gap is created in every stroke. Also a time gap is

created when the digitizer pen is takenoff the tablet and put on the tablet. Each tip of the digitizer pen have some identification number.Thus, when the tip of digitizer pen touches the surface its identification number is recorded with every packet. The information in button field is the identification number of the pentip. Therefore the strokes in the word can be extratcted on the basis of pressure, time and button. In our algorithm, a combination of all parameters are used for extracting the strokes. The algorithm for extraction of strokes named as ESPPBT (Extraction of strokes based on position,pressure,button and time) is described below.

```
Algorithm ESPPBT
    varibles
    x,y // x,y coordinates of pen
    button   // pentip identification
    pressure // pressure of pentip
    time_diff_cur // current time - prev time
    time_diff_next // next time - current time
    pressure_diff  //diff in pressure
    pressure_th // pressure threshold
    time_i //time at i^th point,
    time_i+1 //time at i+1^th point,
    number_of_points // total number of points,
    input[] //array of all input packets,
    strokes[][]  //stroke data,
    packet_data//packet of all input parameters from tablet
    point_id // index into stroke data,
    stroke_id //index of strokes
1.Repeat for  i = 0 to number_of_points
       if button == 0
        {  x← -1; y← -1;
          pressure_diff← pressure_i – pressure_i-1;
          time_diff_cur←time_i–time_i-1;
          time_diff_next←time_i+1-time_i   }
2. Repeat for  i = 0 to number_of_points
        If(x! = -1)and(y! = -1)and(button! = 0)
        and(pressure_diff>pressure_th)and
        (time_diff_current<time_th)and
        (time_diff_next < time_th)
         {
         strokes[stroke_id][point_id].packet_data
         = input[point_id].packet_data;
         point_id←point_id+1;
         }
         else  stroke_id ← stroke_id + 1;
```

The words shown in fig 5a - 5d have been segmented by algorithm ESPPBT and the results are shown in fig 6a – 6d respectively. The results of segmentation show

that the strokes are extracted as they are written by the writer on the tablet. All the strokes show left to right order. The fig 6a shows that the strokes forming the ਅੰਬ (pronounced as 'amb') is clearly segmented into characters ਅ , ਬ and ੰ respectively. The fig 6b to fig 6d show that the characters of words are segmented into substrokes or subparts because the writer has written the characters in that way. For example, characters ਏ , ਜ and ਣ as shown in fig 6b and character ਜ as shown in fig 6c is segmented into two substrokes. However, character ਲ as shown in fig 6d, is segmented into three substrokes.

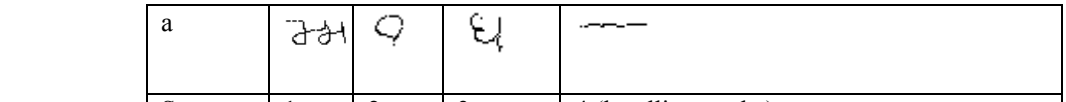
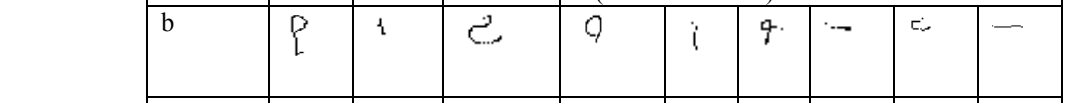
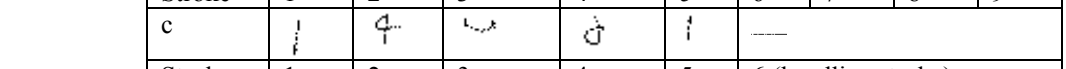
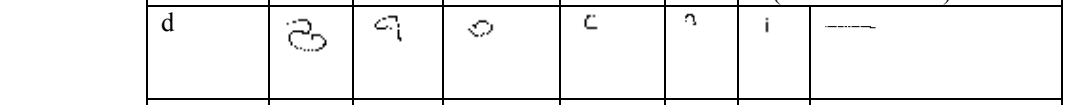
a									
Stroke	1	2	3	4 (headline stroke)					
b									
Stroke	1	2	3	4	5	6	7	8	9
c									
Stroke	1	2	3	4	5	6 (headline stroke)			
d									
Stroke	1	2	3	4	5	6	7(headline stroke)		

Fig 6 Extraction of Strokes

The results of segmentation also show that headline is extracted as a different stroke as can be seen from fig 6a to fig 6d. Thus, the characters which were segmented into substrokes as shown in fig 6b to fig 6d requires merging of substrokes in order to form proper character shapes .

4.2 Merging of Substrokes

In merging of substrokes, the two substrokes are compared to find whether the substrokes lie below the headline and overlap horizontally. If the substrokes overlap then these can be merged. The algorithm for merging of strokes (MOS) is described below.

```
Algorithm MOS
  varibles
     x,y //position coordinates of each point in a stroke
    strxmin,strxmax //xmin and xmax for each stroke
    strymin,strymax //ymin and ymax for each stroke
    strqueue //queue of all the strokes of a word
      1. for each stroke find strxmin,strxmax,
         strymin, strymax ;
      2. Store all the strokes in strqueue
      3. Find the headline stroke
```

```
4. Repeat while (strqueue !=empty) thru 4a-4b
            4a. Pick stroke_i and stroke_i+1;
            4b. if stroke_i and stroke_i+1 are below
                Headline stroke and overlap
                horizontally then
                merge stroke_i+1 and stroke_i
```

The results of mergence are shown in fig 7. Comparison of fig 7a and fig 7b shows that subparts or substrokes which form the characters ਏ, ਜ and ਣ are merged. The merging of substrokes or subparts are illustrated by line drawing among different parts in fig 7. Similarly, comparison of fig 7c and fig 7d shows mergence of substrokes of character ਜ, comparison of fig 7e and fig 7f shows shows mergence of substrokes forming character ਲ. The fig 7b,7d and 7f shows the clear segmentation of Gurmukhi words shown in fig 5b to fig 5d into proper shapes for recognition.

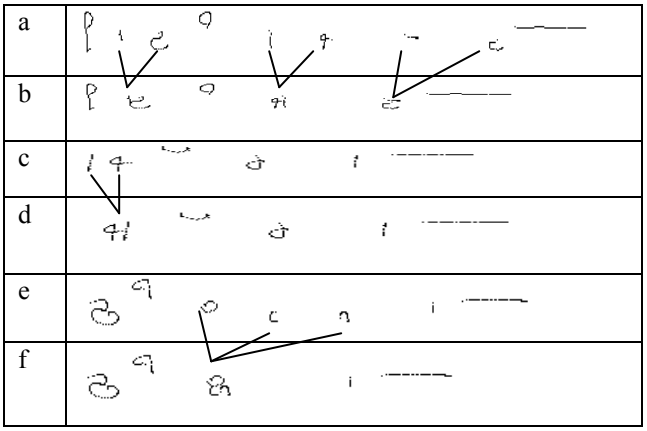


Fig 7 Mergence of sub-strokes

5 Results , Discussions and Future Scope

The results of ESPPBT produces substrokes constituting the individual Gurmukhi character. The parameters such as pressure, time and button are used only to segment the strokes from Gurmukhi word. The number of substrokes constituting individual Gurmukhi character varies from writer to writer. It is difficult to fix the substrokes for individual Gurmukhi character. Therefore, the substrokes constituting the Gurmukhi character are merged using MOS to form the proper Gurmukhi character shape. The correctness of the MOS algorithm is checked manually by comparing the outcome with the desired shape of Gurmukhi character. The final outcome from MOS algorithm is size normalized to 16x16. The sized normalized character is used for feature extraction and classification. The proposed algorithm for extraction of strokes and merging of substrokes were applied on 2150 words collected from 50 users. The extraction of strokes algorithm (ESPPBT) gives 100% accuracy. However the

merging of substrokes algorithm (MOS) gives accuracy varying between 80% to 100% with the average accuracy being 86.4%. The MOS algorithm gives inaccurate results in the following cases :

- If the words are written slanted or the headline drawn is slanted then it results in incorrect finding of headline stroke which results in incorrect merging of substrokes. For example, in fig 8a, subcharacter '-' of character ਥ (thatha) is detected as headline stroke due to which the subcharacter '-' is not considered for merging with stroke ਪ.
- In some cases, the stroke forming the vowels such as aunkar 'ੁ' or dulainkar 'ੂ' is detected as headline, resulting in incorrect merging of substrokes. For example in fig 8b vowel aunkar is detected as headline as the desired headline stroke is slanted.
- If the stroke to be merged is cutting above the headline at some point then it results in incorrect merging. For example in fig 8c the substroke or subcharacter of character ਝ is cutting the headline.

Thus, MOS algorithm gives inaccurate results mainly due to the incorrect finding of headline stroke or improper alignment of headline stroke. In case, the user writes the strokes in proper left to right direction then the performance of both algorithms touches 100%. The performance of MOS algorithm can be improved by adding smoothing operations on strokes and by incorporating fuzzy rules for merging.

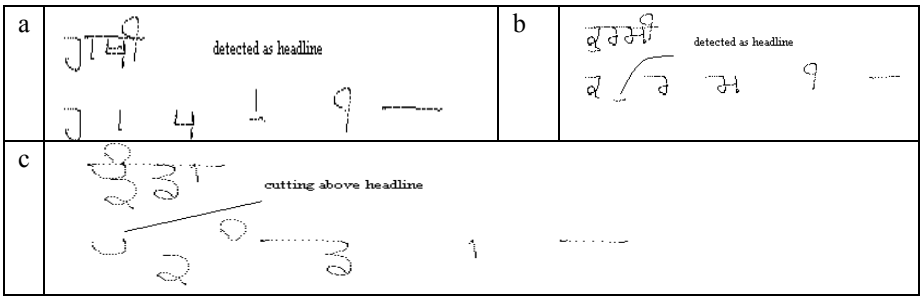


Fig 8 Cases of failure of MOS algorithm

References

1. Tappert,Charles.C., Suen, C.Y.,Wakahara, Toru. : The State of the Art in Online Handwriting Recognition. IEEE. Trans. On. Pattern. Analysis. and. .Machine. Intelligence. 12, 8, 787--808 (1990)
2. Plamondon, Rejean., Srihari, Sargur. N. : On-Line and Off-Line Handwriting Recognition A Comprehensive Survey. IEEE. Trans. On. Pattern. Analysis. And. Machine. Intelligence. 22, 1, 63-84 (2000)
3. Lehal, G.S.,Singh, Chandan. :A Gurmukhi Script Recognition System. In: International Conference on Pattern Recognition, pp. 557-- 560, IEEE Press (2000)
4. Connell, Scott. D., Sinha, R. M. K. , Jain, Anil. K.: Recognition of Unconstrained On-Line Devanagari Characters. In: International Conference on Pattern Recognition, pp. 368-- 371, IEEE Press (2000)

5. Connell, Scott. D.,Jain, Anil. K.:Template- based online character recognition. Pattern. Recognition. 34, 1-14, (2001)
6. Deepu, V.,Sriganesh, M. ,Ramakrishnan, A.G. :Principal Component Analysis for Online Handwritten Character Recognition. In: 17th International Conference on Pattern Recognition, IEEE Press, Cambridge(2004)
7. Aparna, K. H., Subramanian, Vidya., Kasrajan, M., Prakash, G.Vijay., Chakravarthy, V.S. :Online Handwritten Recognition for Tamil, In: 9th International Workshop on Frontiers in Handwriting Recognition, pp. 438-443 , IEEE Press (2004)
8. Joshi, Niranjan., Sita, G., Ramakrishnan, A.G., Madhavanath, Sriganesh.:Comparison of Elastic Matching Algorithm for Online Tamil Handwritten Character Recognition, In: 9th International Workshop on Frontiers in Handwriting Recognition, pp. 444-449, IEEE Press (2004)
9. Namboodiri, M., Jain, Anil. K.:Online Handwritten Script Recognition. IEEE. Trans. On. Pattern. Analysis .and. Machine. Intelligence. 26, 1, 124-13 (2004)
10. Joshi, Niranjan., Sita, G., Ramakrishnan, A.G., Madhavanath, Sriganesh., Deepu V.: Machine Recognition of Online Handwritten Devanagari Characters In: 8th International Conference on Document Analysis and Research,2,pp. 1156-1160 , IEEE Press (2005)
11. Huang,B.Q.,Zhang,Y.B.,Kechadi,M.T.:Preprocessing Techniques for Online Handwriting Recognition. In:7th International Conference on Intelligent Systems Design and Applications, pp. 793-798, IEEE Press (2007)
12. Sharma,Anuj.,Kumar,Rajesh.,Sharma,R.K. :Online Handwritten Gurmukhi Character Recognition Using Elastic Matching. In: International Congress on Image and Signal Processing, pp.391-396, IEEE Press (2008)
13. Sharma,Rajiv.K.,Singh,A.:Segmentation of Handwritten Text in Gurmukhi Script. International. Journal. Of. Image. Processing. 2, 3 , 12-17(2008)
14. Sharma, Anuj. :Rearrangement of Recognized Strokes in Online Handwritten Gurmukhi Word Recognition. In: 10th International Conference on Document Analysis and Recognition, pp. 1241-1245 (2009)