International Journal of Computer Theory and Engineering, Vol. 1, No.2,June2009
1793-8201

Recognizing Online Handwritten Gurmukhi Characters using Comparison of Small Line Segments

Anuj Sharma, R. Kumar and R.K. Sharma

***Abstract*— This paper presents a method to recognize online handwritten Gurmukhi characters. Gurmukhi is the script of Punjabi language which is widely spoken across the globe. The proposed method is called small line segments and based on idea of chain code and elastic matching techniques. Using this method, we have obtained an overall recognition rate of 94.59% for a set of 2460 Gurmukhi characters collected from 60 writers. In recognizing a single stroke, the average speed using this method is approximately 0.156 seconds.**

***Index Terms*— Online handwriting recognition, preprocessing, Feature extraction, chain code, elastic matching.**

I. INTRODUCTION

The online handwriting recognition has great potential to improve user and computer communication. There are various devices used in online handwriting recognition such as PDA, Tablet PCs, cross pad etc. In order to use these devices, accuracy rate must be sufficiently high so that it is acceptable by the user. The established procedure to recognize online handwritten characters include phases as data collection, preprocessing and normalization, feature extraction, segmentation, recognition and post-processing [1]. Statistical, syntactical and structural, neural network and elastic matching are the common handwriting recognition methods [2, 3]. The output obtained from one phase becomes input for the next phase. In this paper, authors proposed a technique to recognize online handwritten Gurmukhi characters that uses chain code and elastic matching techniques. Nouboud and Plamondon [4] used chain codes to recognize hand printed characters. Our method is different from their method as we have used: different format of input handwritten stroke, no positions are used and different comparison formula to recognize strokes.

Gurmukhi is the script of Punjabi language which is widely spoken across the globe. Gurmukhi characters are shown in Table 1. This paper consists of 6 sections including Introduction. Second section discusses about preliminary stages before recognition stage. The preliminary stages include collecting pen movements, preprocessing and computation of features. Third section discusses the recognition technique. Section 4 discusses post-processing of strokes and character recognition. Experimental results and discussion are given in fifth section. Last section, conclusion concludes the findings of this paper.

Anuj Sharma is with the Center for Advanced Study in Mathematics, Panjab University, Chandigarh (INDIA).

Dr. R. Kumar is with the School of Mathematics and Computer Applications, Thapar University, Patiala (INDIA).

Prof. R.K. Sharma is with the School of Mathematics and Computer Applications,Thapar University, Patiala (INDIA).

TABLE 1. GURMUKHI CHARACTERS

a,	A,	e,	s,	h,	k,	K,
g,	G,	|,	c,	C,	j,	J,
t,	T,	f,	d,	x,	q,	Q,
D,	n,	p,	P,	b,	B,	m,
r,	l,	v,	V,	S,	^,	Z,
&,	z					

II. COLLECTING PEN MOVEMENTS, PREPROCESSING AND COMPUTATION OF FEATURES

The process flow of online handwritten Gurmukhi characters using Small Line Segments (SLS) method has been presented in Figure 2. The first step includes Input Handwritten Stroke (IHS) collection and its preprocessing and features computation. A typical format of online handwriting data is a sequence of coordinate points of the moving pen point. Connected parts of the pen trace, in which the pen point is touching the writing surface, are called strokes. Preprocessing and computation of features are performed to the stroke(s) obtained after collecting input handwritten stroke. In present study, the preprocessing stages have been used as size normalization and centering of stroke, interpolating missing points, smoothing, slant correction and resampling of points [5]. The features are computed after preprocessing of input handwritten stroke. The high level features are computed on the basis of low level features. The high level features include loop, crossings, straight line, headline and dots. Some of the common low level features are position of stroke, area, length, curliness and slope [5].

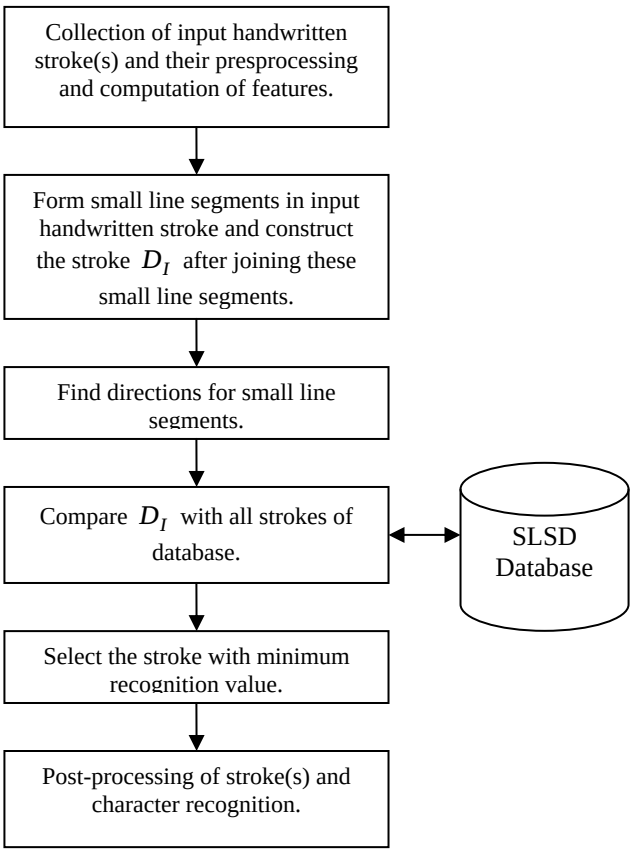

Figure 1. A character recognition process using small line segments method.

III. Recognition

A Gurmukhi character is mixture of one or more strokes. We have recognized the Gurmukhi characters in two stages: first stage recognizes the stroke and character is evaluated in next stage on the basis of recognized strokes. The detailed description of these two stages in online handwritten Gurmukhi character recognition process has been given in previous study [6]. In the proposed SLS method, IHS is compared against all the strokes in the database. The output of comparison is a numeric value referred as Recognition Value (*RV*). In the subsection *A*, a procedure is discussed that convert IHS to SLS format. Procedure for comparison of IHS and database strokes is explained in subsection *B*.

A. Small Line Segments

A.1 SLS in IHS

An IHS is a list consisting of points in sequential order. A point is having two attributes, namely, *x* and *y* co-ordinates. After preprocessing and computation of features, IHS contains fixed number of points (= 40) in our experiments. We have formed SLS from a stroke by joining first point to third point, third point to fifth point and so on, and thirty seventh point to fortieth point and thus have obtained nineteen small line segments for a given stroke. Figure 2 contains the IHS with 40 points and the SLS of this IHS.

A.2 Small line segments directions

Nineteen small line segments of IHS formed in Subsection A.1 are assigned names as per their directions. We have considered twelve directions in the range from 0 to 360 degrees. The names of these directions have been considered as alphabets from 'A' to 'L' and each direction having range of 30 degrees. It is worth mentioning here that if we consider range other than 30 degrees, then the recognition does not improve. We have experimented with a range of 15 degrees, 20 degrees and 45 degrees. Table 2 gives the information about names of twelve directions with their direction numbers and ranges, respectively.

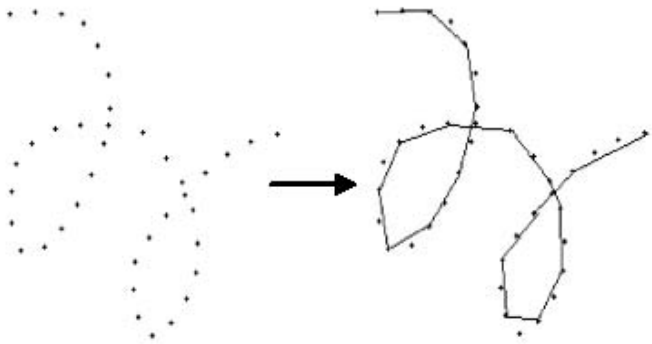

Figure 2. SLS for an IHS.

TABLE 2. DIRECTION NUMBERS, NAMES AND THEIR RANGES.

Direction Number	Direction Name	Direction Range
1	A	$A > 345^0$ *or* $A \leq 15^0$
2	B	$15^0 < B \leq 45^0$
3	C	$45^0 < C \leq 75^0$
4	D	$75^0 < D \leq 105^0$
5	E	$105^0 < E \leq 135^0$
6	F	$135^0 < F \leq 165^0$
7	G	$165^0 < G \leq 195^0$
8	H	$195^0 < H \leq 225^0$
9	I	$225^0 < I \leq 255^0$
10	J	$255^0 < J \leq 285^0$
11	K	$285^0 < K \leq 315^0$
12	L	$315^0 < L \leq 345^0$

A.3 Small line segments directions database

The small line segments approach includes the comparison of IHS and the strokes stored in database. The database stores the strokes in direction format as discussed in Subsection A.2. If the format of directions is not opted, then, each stroke of database is required to convert in direction format at the time of comparison. This will cost more processing time. Therefore, all the strokes collected from writers are converted to direction format at the time of database development. This database in our experiment is named as Small Line Segments Directions (SLSD) database. The first five records of SLSD database for the stroke id 20 are presented in Table 3. In Table 3, each record consists of a single stroke. 'StrokeID' tag represent unique stroke id. Stroke id is combination of three attributes as script number,

stroke number and stroke sample. The contents of first record in Table 3 are explained below.

In 410020001, first two digits 41 represent script number and next four digits 0020 are the stroke number. As such, we can have a total of 10000 strokes for a script. After stroke number, the last three digits 001 correspond to the sample number and we can thus have 1000 samples for a given stroke. These ranges can be altered with respect to different requirements in future. Directions for a stroke are shown under 'Direction' tag. Small line segments directions database is developed in XML format.

TABLE 3. FIRST FIVE RECORDS IN THE DATABASE FOR THE STROKE ID 20.

```
<Record>
<StrokeId>410020001</StrokeId>
<Direction>IIIIIHHGGEDBBALLKKK</Direction>
</Record>
<Record>
<StrokeId>410020002</StrokeId>
<Direction>JJJJJJIIHFDBBAALLLL</Direction>
</Record>
<Record>
<StrokeId>410020003</StrokeId>
<Direction>JIIIIHGGECBALKJJJJJ</Direction>
</Record>
<Record>
<StrokeId>410020004</StrokeId>
<Direction>JIIIIHHGGDBAALKKKKK</Direction>
</Record>
<Record>
<StrokeId>410020005</StrokeId>
<Direction>JIIIIIHGEDBAALKKKKJ</Direction>
</Record>
```

B. *Small line segments comparison*

In this subsection, we have proposed an algorithm to compute the recognized value for each database stroke when compared against input handwritten stroke. The stroke in database that gives minimum recognized value is the recognized stroke. The steps in order to compare input handwritten stroke with all strokes of database are given in Algorithm 1.1. In this algorithm, following variables are used:

N : Total number of strokes in small line segments directions database.

k : k is an integer and $1 \le k \le N$.

D_I : Input handwritten stroke.

D_k : k^{th} Stroke of small line segments directions database.

RS : Recognized stroke.

RV : Recognition value of RS .

RV_k : Recognition value of k^{th} stroke, where $1 \le k \le N$.

d, j : Integers.

Algorithm 1.1

1. Set $RV = 0$ and $k = 1$.
2. Repeat steps 3 to 15 until $k \le N$.
3. Set $j = 1$.
4. Repeat steps 5 to 10 until $j \le 19$.
5. Set $d = |D_{I_j} - D_{k_j}| + 1$.
6. If ($d > 7$) then
7. Set $d = |12 - d| + 1$.
8. End if.
9. Set $RV_k = RV_k \times |d|$.
10. Increment j by 1 and go to step 4.
11. If ($RV_k < RV$) then
12. Set $RV = RV_k$.
13. Set $RS = D_k$.
14. End if.
15. Increment k by 1 and go to step 2.

In this Algorithm, RS and RV are two outputs, RS is the recognized stroke and RV is the recognition value of recognized stroke. Difference of directions must be less than 7 because of angular behavior of directions. For example, 'H' and 'F' directions are at same angular difference from direction 'A' as illustrated in Table 2. The implementation of Algorithm 4.1 is explained below for an input handwritten stroke.

Let S_1 and S_2 be input handwritten stroke and a stroke from small line segments direction database, respectively. S_1 and S_2 contents are "LLAJGFCAAALJIGFCBAA" and "LLAIDFBAKKKJIGFDBBA".

Therefore, RV for S_2 is calculated as:

RV $= 1 \times 1 \times 1 \times 1 \times 2 \times 4 \times 1 \times 2 \times 1 \times 2 \times 2 \times 2 \times 1 \times 1 \times 1 \times 1 \times 2 \times 1 \times 2 \times 1 = 512$.

The value of RV can thus be calculated for any given input handwritten stroke with respect to SLSD database. The minimum value of RV gives the recognized stroke as stated earlier also.

IV. POST-PROCESSING AND CHARACTER RECOGNITION

Post-processing is applied after recognition process in order to refine the recognition results. The results of computed features, as discussed in Section 2, are used in post-processing. We have implemented post-processing phase based on features, namely, loop, straight line, headline, crossings, curliness and dots existing in the stroke. The detailed description for post-processing in online handwritten Gurmukhi character recognition has been explained by us in [7].

A character is recognized after inputting all handwritten strokes. The recognition of strokes is performed with the algorithm 3.1. The recognized stroke(s) obtained after using recognition, are stored in a dynamic list ' C ' called character key. All elements of dynamic list ' C ' are merged and sorted. Finally, value of ' C ' is searched in character database. The process of formation of this list and searching of corresponding character in character database is explained in the following example.

Let ' C ' consists of two nodes as 15 and 11. Merging and sorting give output as 1115. This final value is called character key. This character key is matched with all the character keys in the character database. If calculated character key exists in the character database, the corresponding character is displayed. The character database

developed in this study is in XML format. First five records of character database are given in Table 4.

V. Results and Discussion

This section describes the experiments we carried out and contains the recognition results. The application has been developed in VC++ that implement SLS method. The overall recognition rate using SLS method has been achieved as 94.59% when tested on 2460 characters. These 2460 online handwritten Gurmukhi characters have been collected from 60 writers and each writer has written all 41 Gurmukhi characters. These 41 Gurmukhi characters are given in Table 1.

We have noticed that 100% accuracy have been achieved for 3 writers. Also, we could achieve the accuracy 97.56%, 95.12%, 92.68%, 90.24%, 87.8%, 85.37% and 82.93% for 20, 14, 14, 5, 2, 1 and 1 writers, respectively. These results are presented in Table 5. We have also found that 24 characters out of total 41 characters have been recognized correctly for all writers. The recognition rates of all Gurmukhi characters are presented in Table 6. To know the stable nature of small line segments method, we studied the results for first 15, 30, 45 and 60 writers as shown in Figure 3. It has been noted that a change of 1.22% in recognition rates exists for the first 15 and 30 writers. Similarly, a change of 0.03% for the first 30 and 45 writers, change of 0.09% for the first 45 and 60 writers. The average recognition time to recognize a single stroke is approximately 0.156 seconds using small line segments method. The small size of small line segments directions database is the major reason behind less recognition time in case of small line segments method.

Table 4. The first five records of Character database.

```
<Record>
<CharacterKey>15</CharacterKey>
<CharacterSymbol>a</CharacterSymbol>
<CharacterName>ura</CharacterName>
</Record>
<Record>
<CharacterKey>1115</CharacterKey>
<CharacterSymbol>a</CharacterSymbol>
<CharacterName>ura</CharacterName>
</Record>
<Record>
<CharacterKey>16</CharacterKey>
<CharacterSymbol>A</CharacterSymbol>
<CharacterName>aira</CharacterName>
</Record>
<Record>
<CharacterKey>111216</CharacterKey>
<CharacterSymbol>A</CharacterSymbol>
<CharacterName>aira</CharacterName>
</Record>
<Record>
<CharacterKey>1216</CharacterKey>
<CharacterSymbol>A</CharacterSymbol>
<CharacterName>aira</CharacterName>
</Record>
```

TABLE 5. RESULTS OF SMALL LINE SEGMENTS BASED RECOGNITION METHOD.

Number of writers	Recognition rate (%)
3	100
20	97.56
14	95.12
14	92.68
5	90.24
2	87.80
1	85.37
1	82.93

VI. Conclusion

In this paper, we have presented an online handwritten Gurmukhi character recognition system using SLS method. An application is developed by authors that implements proposed SLS method. IHS is collected and preprocessed using size normalization and centering, interpolating missing points, smoothing, slant correction and resampling of points techniques. SLS method is applied and the stroke with minimum *RV* is selected. A character is recognized only after all the input handwritten stroke(s) are written. Using SLS method, we have obtained an overall recognition rate of 94.59% for a set of 2460 Gurmukhi characters collected from 60 writers and each writer has contributed 41 different characters. In recognizing a single stroke, the average speed using proposed method is 0.156 seconds.

TABLE 6. RECOGNITION RATE OF CHARACTERS USING SMALL LINE SEGMENTS METHOD.

Characters	Recognition rate (%)
a, A, s, k, g, G, c, C, j, |, T, f, d, q, D, n, P, m, V, S, ^, Z, z, L	100
t, F	96.67
B, &	95
l, h, r	93.33
J	91.67
b	90
e, v, Q	85
\	83.33
X	80
p	75
x	71.67
K	68.33

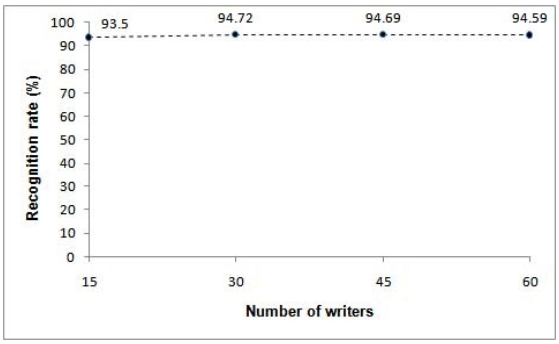

Figure 3. Stability of small line segment method for first 15, 30, 45 and 60 writers.

ACKNOWLEDGMENT

We are thankful to Prof. Rejean Plamondon, Department of Electrical Engineering, ECOLE Polytechnique Montreal, Canada to provide his some of the previous published papers.

REFERENCES

[1] C.Y. Suen, A.L. Koerich and R. Sabourin, "Lexicon¬driven HMM decoding for large vocabulary handwriting recognition with multiple character models", IJDAR, Vol 6, no. 2, pp. 126-144, 2003.
[2] E.J. Bellegarda, J.R. Bellegarda, D. Namahoo and K.S. Nathan, "A probabilistic framework for online handwriting recognition", Proc. IWFHR III, Buffalo, New York, pp. 225-234, 1993.
[3] A.K. Jain, Robert P.W. Duin and Jianchang Mao, "Statistical Pattern Recognition: A Review", IEEE transactions of Pattern Recognition and Machine Intelligence, Vol. 22, no. 1, pp. 4-37, 2000.
[4] F. Nouboud and Rejean Plamondon, "A structural approach to on-line character recognition: system design and applications", World scientific series in computer science, vol. 30, pp. 311-335.
[5] Anuj Sharma, R.K. Sharma and R. Kumar, "Online Handwritten Gurmukhi Strokes Preprocessing", Machine GRAPHICS and VISION, 18, 2009.
[6] Anuj Sharma, R. Kumar and R.K. Sharma, "Online Handwritten Gurmukhi Character Recognition using Elastic Matching", IEEE Proceedings on International Congress on Image and Signal Processing (CISP), Sanya, vol. 2, pp. 391-396.
[7] Anuj Sharma, R. Kumar and R.K. Sharma, "On the Recognition of Online Handwritten Gurmukhi Characters", Proceedings of 2nd National Conference on Recent Trends in Information Systems (ReTIS), Kolkata (INDIA), 2008, pp. 42-45.

Anuj Sharma is currently a PhD candidate in School of Mathematics and Computer Applications, Thapar University, Patiala, India. Since 2003, He has been with Center for Advanced Study in Mathematics, Panjab University, Chandigarh, India. His current research interests are online handwriting recognition in Indian languages and English.

Rajesh Sharma is assistant professor in School of Mathematics and Computer Applications, Thapar University, Patiala, India. His current research interests are frature mechanics and pattern recognition.

R.K. Sharma is professor in School of Mathematics and Computer Applications, Thapar University, Patiala, India. His current research interests are soft computingg, statistical computing and reliability analysis.