Stroke Analysis of Devnagari Handwritten Characters

PRACHI MUKHERJI,* PRITI P. REGE **
Electronics and Telecommunication Department,
* Sinhgad Technical Education Society's SKNCOE, Pune 411041.
** College of Engineering Pune,
Shivajinagar, Pune, 411005.
INDIA

Abstract: - Devnagari script is the major script of India and is widely used for various languages. In this work we propose a stroke based technique for analyzing handwritten Devnagari characters. After preprocessing the character is segmented in various strokes using our thinning and segmentation algorithm. We propose average compressed direction codes for segmented strokes to classify the strokes as left curve, right curve, horizontal stroke, vertical stroke and slanted lines etc. The knowledge of script grammar is applied to identify the character using shapes of strokes, mean, relative strength, straightness, circularity and area. The system tolerates slant of about 10º left and right and a skew of 5º up and down. The system gives high discrimination between similar characters and gives a recognition accuracy of 85%.

Keywords: Devnagari Script, Character Segmentation, Thinning, Strokes, Average Compressed Direction Code, Unordered Stroke Matching.

1 Introduction

Over 500 million people all over the world use Devnagari script. It provides written form to over forty languages [1] including Hindi, Konkani and Marathi. It is a logical composition of its constituent symbols in two dimensions [2]. A marked distinction in Devnagari script from the scripts of Roman genre is the fact that a character represents a syllabic sound, complete in itself. There has been intense research work done on the English, Latin, Chinese, Persian, Tamil, and Bangla scripts on both handwritten and machine printed texts. While most work has been published for printed Devnagari text, very little is reported for handwritten Devnagari script. One of the first attempts for handprinted characters has been by Sethi [3] and for typed Devnagari script by Sinha and Mahabala [4]. V. Bansal and Sinha in [6] divided the typed word in three strips and separated it in top strip, core strip and bottom strip and achieved 93% performance on individual characters. Pal and Chaudhuri have attempted OCR for two scripts, Bangla and Devnagari in [5]. In [7], Binary Wavelet transform is used for feature extraction of handwritten Devnagari characters. In [8], a survey of structural techniques used for feature extraction in OCR of different scripts is given. Recently in [9], Quadratic classifier based method is proposed with 81% accuracy.

In this proposed work, we analyze 44 isolated handwritten characters in Devnagari script. The handwriting should be legible and adhering to structural syntax of Devnagari script. The processing steps of our OCR system can be summarized concisely. The documents are scanned and the digitized images are subjected to preprocessing techniques like filtering, binarization, skeletonization and pruning. The feature extraction module segments the character in strokes. Various features of these segmented strokes like the mean row and column coordinate, circularity, area and modified direction codes are extracted. Cognitive scientists report that humans base their thinking on conceptual patterns and mental images rather than on any numerical quantities [10]. Keeping this view in mind and the construction of Devnagari script, we analyze the similarity and dissimilarity of strokes for classification. The classification is achieved using unordered stroke matching. The applications of this system are many as in postal automation, reading aid for the blind and unconstrained Devnagari script recognition.

2 The Devnagari Character Set

The basic character set is of 48 characters Each of the consonant (*Vyanjan)* and the conjuncts (*Yuktakshar)* can be further modified by vowel (*Matra*) modifier. In a manner similar to that of the formation of conjuncts, combinations of vowels, with the nasal sounds, gives rise to combines in vowels also. There are as many characters in the Devnagari script as there are syllables in the spoken language. The basic character set for experimentation is based on their present-day usage and is shown in Fig. 1. Every character has a

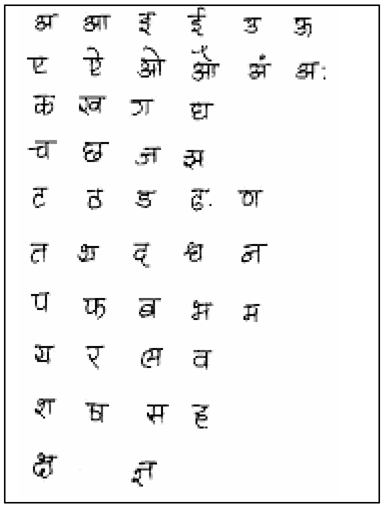


Fig. 1: Basic Character Set

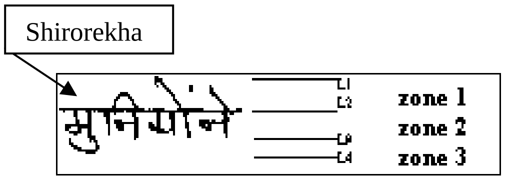


Fig. 2: Devnagari word with modifiers and zones

horizontal header line on the top, called the 'Shirorekha' as is evident from Fig. 2. This line serves as a reference to divide the character into two distinct portions: Head and Body or zone 1 and zone 2, if the top modifier is present. The lower modifier occupies zone 3.

3 Preprocessing

Each character is preprocessed before it can be recognized. Preprocessing includes filtering by Gaussian filter, binarization, and thinning.

Each handwritten document is scanned and converted into a digitized image using a desktop scanner. Noise removal and image smoothing is done using the Gaussian filter of size 5X5. Binarization is achieved by using Otsu's algorithm [11]. Next step, skeletonization [12] is an important aspect of feature extraction in our scheme.The spurs in the image are removed by the *spur* removal algorithm [12]. The process of cleaning up (or removing) these spurs is called pruning. Slant of a character is detected using a method of slices and applied successfully to Devnagari words in [13]. Keeping the topline as reference horizontal shear transform [14] is applied only if the slant is more than the set threshold of 10°.

4 Feature Extraction

4.1 Top Modifier Features

As shown, in Fig 1(b), the distinct feature of Devnagari script as well as individual characters is the topline (Shirorekha) that is detected using Hough Transform [12]. If top modifier is present, it is recognized on the basis of transitions [6] and shadows [13].

4.2 Segmentation of Character in Strokes

An adaptive thinning algorithm has been developed for separating an image in its constituent strokes. In the first step, the complete neighborhood pattern is mapped by finding and summing the contribution of all eight neighbors of a black pixel, Sum_of_Neighbours, $s(i, j)$ is given in (1). Since it has already passed through skeletonization once, expectedly the maximum neighborhood can be of four pixels signifying a crosspoint as shown in Fig. 3(a). This point is given value zero and then the map is checked for values of three. Blindly changing these to zero gives rise to over segmentation.

Sum_of_Neighbours =

$$s(i, j) = \sum_{i-1}^{i+1}\sum_{j-1}^{j+1} p(i, j) - 1 \qquad (1)$$

3	3	3
3	4	3
3	3	3

(a)Cross-point s(i,j)

2	0	2
0	0	0
2	0	2

(b)Two Neighbor Cross-point

0	3	0
0	2	3
0	0	0

c) Lower Left Corner corner

0	2	0
0	0	2
0	0	0

d) Two Neighbor

2	2	2
2	3	2
3	2	2

(e) Hole s(i,j)

1	1	1
1	0	1
0	1	1

(f) Two-Neighbor Hole

Fig.3: Steps in Adaptive Thinning Algorithm.

Instead, the combinations shown in Fig. 3(c) and Fig. 3(d) are used to detect corners and smoothen them. The pruned map is then thresholded, as in (2).Few other combinations are also stored for detecting T connections in all the directions.

$$\text{If} \quad \sum_{i-1}^{i+1}\sum_{j-1}^{j+1} s(i, j) - 1 \geq 3 \qquad (2)$$

$$s(i, j) = 0;$$

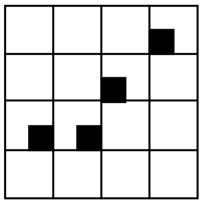
else

$$s(i, j) = 1;$$

end

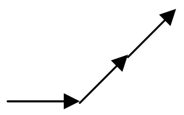
4.3 Average compressed Direction Codes

After segmentation of the character in strokes, all segments are labeled from left to right in the image frame. As each stroke (segment) is labeled, its row and column indices are also stored. Then each black pixel is coded by following the code directions as given in Fig.4 (a). This code is averaged to reduce the vector space. Depending upon the length of the stroke, code is averaged. Here care has been taken when combinations of (8,1,2) and (7,1,2) are obtained, as direct averaging gives wrong average. The code per section of the stroke is averaged as a combination of 1, 2, or 3 codes as shown in Fig. 4(b) using anticlockwise angle representation. These angles are then quantized as per code direction 1-8.



Code is (8,1,1)

Fig. 4(a): Stroke



arctan (2/3) = 33. 9°
Code =2

Fig. 4(b): angle

A vertical line will have all codes as 6. This code is run length coded to obtain a compressed code. This Average Compressed Direction Code (ACDC) is then stored as a part of the feature vector. Code 9 is used to represent a change from (8,1) in the averaged code. Code 10 indicates isolated single pixels. Some basic ACDC codes are shown in Fig.5.

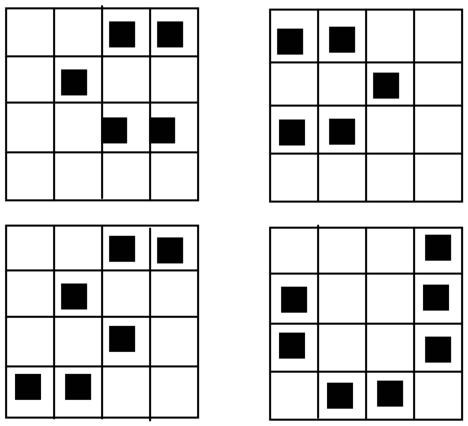


Fig. 5: Examples of ACDC.

Apart from these directly obtained codes, insertion and deletions are done so as to accommodate the variations in handwritten strokes. This is from the perceptual study of how people read Devnagari script. For example the shape in Fig. 5(a) is equivalent to (4,5,6,7,8) and the basic shape defining code is (5,7) or (5,6,7) which indicates a left curve.

4.4 Stroke Features

Six features extracted on all k strokes are listed below where k = 1 to Num, Num is the total number of strokes obtained in a character.

1. $L_k = \sum_{i,j} S(i,j)$, total no of black pixels in the stroke k.

2. $R_k = (\sum_i i)/M$, mean of the row indices of the segment stroke k.

3. $C_k = (\sum_j j)/N$, mean average of column indices of the stroke k.

4. $REL_k = L_k / Max(L_k)$, Length of stroke L_k, divided by the maximum length stroke.

5. $CIR_k = sqrt\,[(R1_k - R2_k)^2 + (C1_k - C2_k)^2] / L_k$, indicating a straight line or a curve, where $R1_k$ and $R2_k$, $C1_k$ and $C2_k$ are the row and column indices of endpoints of the stroke k.

6. $AREA_k = |(RE2_k - RE1_k)| * |(CE2_k - CE1_k)|$, area occupied in pixels by the stroke k where $RE2_k$ and $RE1_k$ are row indices of top most row and lower most row, $CE2_k$ and $CE1_k$ are leftmost and rightmost column indices of the stroke. Here (M,N) is the size of the image.

5 Character Classifier

First the direction codes classify the strokes as left curve, right curve,u-curve, s-curve and straight lines. After this, we examine their location features in the image after normalizing the image to a size of 1 by 1.Unordered stroke matching is used to determine the matching strokes.

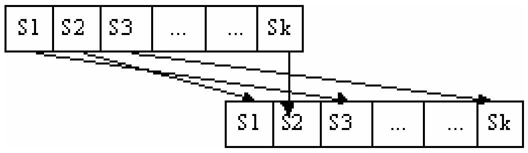


Fig. 6: Unordered stroke matching.

6 Experimentation and Results

As no standard database is available for handwritten Devnagari characters, 3 samples of 44 characters were collected from 250 persons belonging to different age groups and social backgrounds. 60 % of the database is used for feature extraction. Feature vectors of three prototypes of each character are stored. The remaining database is used for validation. The results for features extracted for two samples of character 'Ksha' in Fig. 7 are given in Table 1. The number of strokes extracted may slightly differ but the average location and direction codes are similar. This simulates human vision where recognition of characters is by judging their shapes. The accuracy of recognition is

Table 2: Results of different features for image in Fig.7

k	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
L_k *	18	10	4	3	1	1	4	4	11
L_2 *	15	17	4	4	1	6	9		10
R_k	0.6588	0.2759	0.7241	0.9138	0.3966	0.2241	0.3448	0.2414	0.5690
R_2	0.6250	0.1508	0.7143	0.9143	0.3214	0.3929	0.2179	-----	0.6071
C_k	0.2249	0.2727	0.4545	0.4886	0.5455	0.5909	0.7727	0.9545	0.9545
C_2	0.2202	0.3466	0.4476	0.4381	0.4762	0.6667	0.9000	-----	0.9134
REL_k	1.0000	0.5556	0.2222	0.1667	0.0556	0.1667	0.2222	0.2222	0.6111
REL_2	0.8824	1.0000	0.2353	0.2353	0.0588	0.3529	0.5294	-----	0.5882
$CIRk_k$	0.6136	0.8602	1.0000	1.2019	1.0000	1.0000	1.0000	1.0000	1.0000
$CIRk_2$	0.8246	0.1860	1.1180	1.0607	0	1.0000	0.7454		1.0050
$AREA_1$	96	56	5	12	2	4	5	5	12
$AREA_2$	104	48	15	16	1	7	35	-----	22
$ACDC_1$	545678	6787	6	7	6	6	8	6	6
$ACDC_2$	568	24768	65	7	10	8	86	-----	6

* Values are in Pixels

85%.The system rejects 1.1% of the characters that are broken or do not match with any of the stored features of strokes. Errors are observed when the character is not segmented accurately and due to similarity in some characters. A few characters are written in structurally different ways depending on the educational and regional background of the writer.

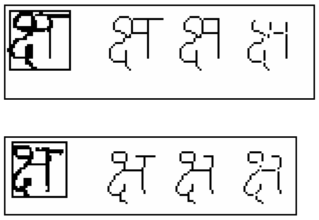


Fig. 7: Results of Stroke Extraction of character 'Ksha'

7 Conclusion

The proposed work presents recognition of Devnagari characters free from normalization thereby giving flexibility and allowing size variation. Database was collected from writers of varied background. Modified thinning algorithm needed for separating a character in its constituent strokes was developed and tested successfully. Modified direction codes (ACDC) algorithm, though simple, works efficiently to detect curves and turning points. The overall recognition accuracy is 85%. Rejection also plays an important role in system evaluation. This work is a step towards unconstrained Devnagari script recognition, as in unconstrained text; words have to be segmented into its basic characters and modifiers in the order of top modifier, body (characters) and lower modifier. The Devnagari script is used extensively at the grass root levels in many states of India. This Devnagari character recognition system can be used in developing Indian postal automation system, bank check processing system, as a reading aid to blind and also in forensic science.

References:

[1] S. Kompalli, S. Nayak, S. Setlur and V. Govindaraju, *Challenges in OCR of Devnagari Documents*, Proceedings of Eighth International Conference on Document Analysis and Recognition, Seoul, South Korea, pp. 327- 331, 2005.

[2] N. Joshi, G.Sita, A.G. Ramakrishnan, Deepu V., S. Madhavnath, *Machine Recognition of Online Handwritten Devnagari Characters*, Proceedings of Eighth International Conference on Document Analysis and Recognition, Seoul, South Korea, pp. 1156- 1160, 2005.

[3] I.K. Sethi, *Machine Recognition of Constrained Handprinted Devnagari,*

Pattern Recognition, Vol. 9, pp. 69-75, 1977.

[4] R. M. K. Sinha and H. N. Mahabala, *Machine Recognition of Devnagari Script*, IEEE Transactions on Systems, Man and Cybernetics, Vol. 9 (8), pp. 435- 441,1979.

[5] B.B. Chaudhuri and U. Pal, *An OCR system to read two Indian Language Scripts: Bangla and Devnagari (Hindi*), Proceedings of Fourth International Conference on Document Analysis and Recognition, Ulm, Germany, pp. 1011-1015, 1997.

[6] R. M. K. Sinha and Veena Bansal, *A complete OCR for Printed Hindi Text in Devnagari Script,* Proceedings of Fifth International Conference on Document Analysis and Recognition, Seattle, US, pp. 800-804, 2001.

[7] Prachi Mukherji, Vaishali B Gapchup and Priti P. Rege, *Feature Extraction of Devnagari Characters using sub bands of Binary Wavelet Transform*, Proceedings of RETIS-06, Kolkata, pp. 176 – 179, 2006.

[8] Prachi Mukherji and Priti. P. Rege, *A Survey of Techniques for Optical Character Recognition of Handwritten Documents with reference to Devnagari Script*, Proceedings of First International Conference on Signal and Image Processing, Hubli, India, pp. 178 –184, 2006.

[9] N. Sharma, U. Pal, F. Kimura and S. Pal, *Recognition of Off-Line Handwritten Devnagari Characters Using Quadratic Classifier*, N. Sharma, U. Pal, F. Kimura and S. Pal, Proc. of Indian Conference on Computer Vision Graphics and Image Processing (ICVGIP), Madurai, India, pp-805-816, 2006.

[10] Timothy J. Ross, *Fuzzy Logic with Engineering Applications*, MCGraw Hill International Editions, New York, 1997.

[11] N.Otsu, *A threshold selection method from gray level histograms*, IEEE Transactions on Systems, Man and Cybernetics, *9*(1),pp. 62-66, 1979.

[12] Rafel C Gonzalez, Richard E.Woods, *Digital Image Processing*, Second Edition, Pearson Education, 2003.

[13] Prachi Mukherji, Priti P Rege and Leena K. Pradhan, *Analytical Verification System for Handwritten Devnagari Script*, Proceedings of the Sixth IASTED VIIP, Palma De Mallorca, Spain, pp. 237-242, 2006.

[14] B.B. Chaudhari and D. Datta Majumder, *Two Tone Image Processing and Recognition*, Wiley Eastern Limited, 1993.