Fuzzy Directional Features for Unconstrained On-line Devanagari Handwriting Recognition

Lajish V.L.
TCS Innovation Labs - Mumbai
Tata Consultancy Services Ltd., Yantra Park,
Thane (West), Maharashtra, India - 400601
Email: lajish.vl@tcs.com

Sunil Kumar Kopparapu
TCS Innovation Labs - Mumbai
Tata Consultancy Services Ltd.,Yantra Park,
Thane (West), Maharashtra, India - 400601
Email: sunilkumar.kopparapu@tcs.com

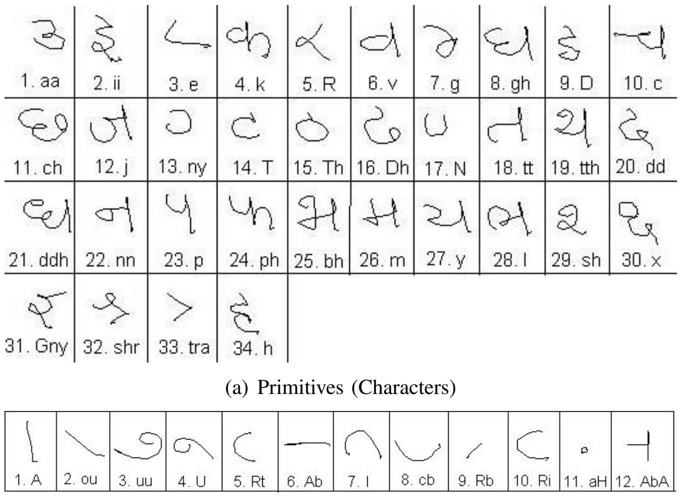
***Abstract*—This paper describes a novel feature set for recognition of unconstrained on-line handwritten Devanagari script. Experiments are conducted for the automatic recognition of handwritten character primitives (sub-character units) collected without any constraints from different writers. Initially we describe the Fuzzy Directional Feature (FDF) extraction method and then show how these features can be effectively utilized for writer independent Devanagari character recognition. The recognition algorithm uses second order statistics to construct different stroke models. Experimental results show that FDF set out performs commonly used Directional Features (DF) for writer independent data set at stroke level recognition.**

***Index Terms*—On-line Handwriting Recognition, Directional Features, Fuzzy Directional Features**

I. Introduction

While a large amount of literature is available for on-line handwriting recognition of English, Chinese and Japanese languages, relatively very less work has been reported for the recognition of Indian languages. Interest in on-line handwritten script recognition has been active because of large number of practical applications. In the case of Indian languages, research work has been active for Devanagari [1], [2], Bangala [3], [4], Telugu [5] and Tamil [6], [7]. Devanagari script is the most widely used Indian script (used by more than 500 million people) and consists of 14 vowels and 34 consonants. It is used as the writing system for over 28 languages including Sanskrit, Hindi, Kashmiri, Marathi and Nepali. Devanagari is written from left to right in horizontal lines and the writing system is alphasyllabary. In English script barring a few alphabets, all the alphabets can be written in a single stroke. But in most of the Indian languages, characters are made up of two or more strokes. This makes it necessary to analyze a sequence of strokes to identify a character. The main challenge in on-line handwritten character recognition in Indian language is the large character set size, variation in writing style (when the same stroke is written by different writers or the same writer at different times) and the similarity between different characters in the script and complexity in character shape.

We identified, through visual inspection, a basis like set of 46 strokes called primitives (34 character primitives and 12 vovel modifiers) as in Figure 1 for Devanagari script. Note that these set of primitives can be concatenated so as to form almost all characters [1] in Devanagari. Figure 2 shows an example where primitives ($m, U, R, A, Ab, \dots$) are combined together to form characters and hence words. In an unconstrained handwritten script these primitives exhibit large variability in shape, direction and order of writing. Variations within the primitives even for the same writer exist and the variation due to different writers is even larger; making the task of recognizing characters difficult. In this paper we use these primitives as the units for recognition taking parallel from the *phone set* used in speech recognition literature. In this paper, we propose a new feature set called Fuzzy Directional Features (FDF) for the recognition of primitives. The rest of the paper is organized as follows. We introduce the Fuzzy Directional Features set in Section II. Experimental Results are outlined in Section III, and conclusions are drawn in Section IV.



(a) Primitives (Characters)

(b) Primitives (Vovel modifiers)

Fig. 1. Devanagari primitives(basic strokes)

II. Fuzzy Directional Features

In this section we describe the FDF extraction procedure. We also discuss the pre-processing steps and the procedure

[1]We have excluded some special primitives that exist in the form of *half characters* which are normally used to from compound characters.

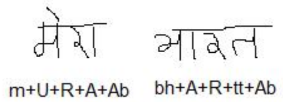


Fig. 2. Character formation using primitives

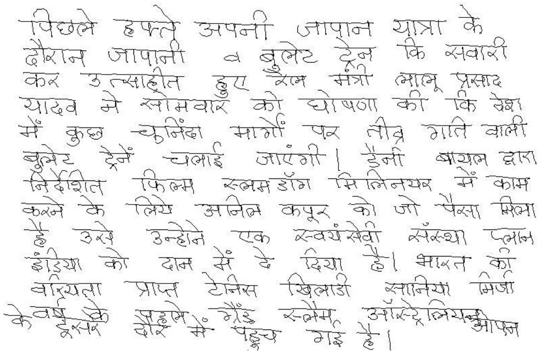

पिछले हफ्ते अपनी जापान यात्रा के दौरान जापानी व बुलेट ट्रेन कि सवारी कर उत्साहीत हुए रेल मंत्री लालू प्रसाद यादव ने सोमवार को घोषणा की कि देश में कुछ चुनिंदा मार्गों पर तीव्र गति वाली बुलेट ट्रेनें चलाई जाएंगी। डैनी बायल द्वारा निर्देशित फिल्म स्लमडॉग मिलिनेयर में काम करने के लिये अनिल कपूर को जो पैसा मिला है उसे उन्होंने एक स्वयंसेवी संस्था प्लान इंडिया को दान में दे दिया है। भारत की वरियता प्राप्त टेनिस खिलाडी सानिया मिर्जा के वर्ष के पहले ग्रैंड स्लैम ऑस्ट्रेलियन ओपन के दूसरे दौर में पहुंच गई है।

Fig. 3. Sample on-line handwritten paragraph data

for identifying the curvature points that are required prior to feature extraction.

Normally on-line data is represented by a variable number of 2D points which are in a time sequence. For example an on-line script would be represented as

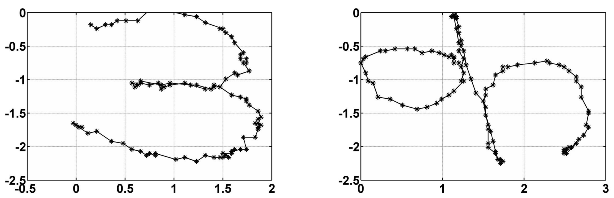
$$\{(x_{t_1}, y_{t_1}), (x_{t_2}, y_{t_2}), \cdots, (x_{t_n}, y_{t_n})\}$$

where, t denotes the time and assume that $t_1 < t_2 < \cdots < t_n$ and n represents the total number of points. Equivalently we can represent the on-line data (see Fig 4(a)) as

$$\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$$

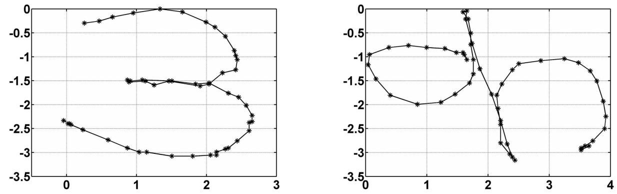
by dropping the variable t. The number of points denoted by n vary depending on the size of the character and also the speed with which it was written. Most script digitizing devices (popularly called electronic pen) sample the script uniformly in time. For this reason, the number of sampling points is large when the writing speed is slow which is especially true at curvatures (see Figure 4).

As a pre-processing we perform smoothing on the raw data based on Discrete Wavelet Transform (DWT) based decompo-



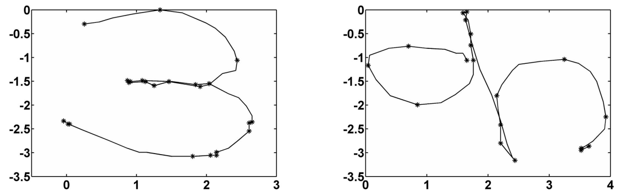
(a) (b)

Fig. 4. (a)-(b) Sample on-line Devanagari characters *"aa" and "k"* . The (x, y) points have been joined to give a feel for the character.



(a) (b)

Fig. 5. (a)-(b) DWT smoothened Devanagari characters *"aa" and "k"*



(a) (b)

Fig. 6. (a)-(b) Curvature points identified for the characters *"aa" and "k"*

sition using Daubechies wavelet [2]. The DWT decomposition helps in removing noise due to small undulation caused due to the sensitiveness of the sensors on the electronic pen and inherent shake while writing. The results of DWT based smoothing is shown in Figure 5

A. Identification of curvature points

The curvature points (also called critical points) are extracted from the smoothed handwritten data. The sequence $(x_i, y_i)_{i=0}^{n}$ represents a hand written stroke. We treat the sequence x_i and y_i separately and calculate the critical points for each of these sequences. For the x sequence, we calculate the first difference (x'_i) as

$$x'_i = sgn(x_i - x_{i+1})$$

where $sgn(k) = +1$ if $x_i - x_{i+1} > 0$, $sgn(k) = -1$ if $x_i - x_{i+1} < 0$ and $sgn(k) = 0$ if $x_i - x_{i+1} = 0$. We use x' to compute the critical point in x sequence. The point i is a critical point *iff*

$$x'_i - x'_{i+1} \neq 0$$

Similarly we calculate the critical points for the y sequence independently. The final list of critical points is the union of all the points marked as critical points in both the x and the y sequence (see Figure 6). It must be noted that the position and number of curvature points computed for different samples of the same strokes vary.

B. Fuzzy Directional Feature (FDF) extraction

Several temporal features have been used for script recognition in general and for on-line Devanagari script recognition in particular [8]–[11]. We propose a simple and effective feature set based on fuzzy directional features [3]. Let k be the number

[2]We do not dwell on this since this is well covered in Digital Signal Processing literature

[3]Note that [12] talks of fuzzy feature set for Devanagari script albeit for offline handwritten character recognition

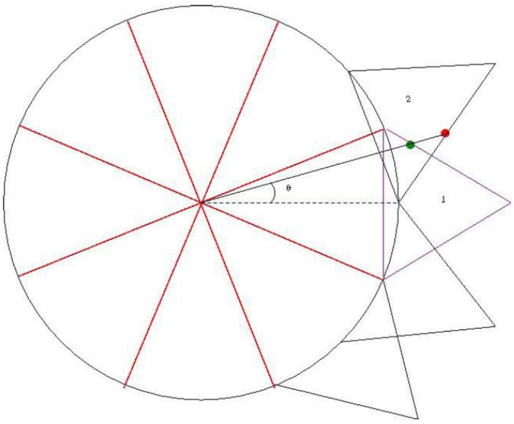


Fig. 7. θ contributing to two directions $(1, 2)$ with fuzzy membership values (green and red dot)

of curvature points (denoted by $c_1, c_2, \cdots c_k$) extracted from a stroke of length n; usually $k << n$. The k critical points form the basis for extraction of the directional features and the FDF. We first compute the angle between two adjacent critical points, say c_l and c_{l+1}, as

$$\theta_l = \tan^{-1}\left(\frac{y_l - y_{l+1}}{x_l - x_{l+1}}\right) \quad (1)$$

where (x_l, y_l) and (x_{l+1}, y_{l+1}) are the coordinates corresponding to the curvature point c_l and c_{l+1} respectively.

Note that we divide 2π into eight overlapping directions $1, 2, \cdots, 7$. Every θ_l (for example the angle θ maked in Figure 7) has two directions (say $d_l^1 = 1, d_l^2 = 2$). Note that the green and red dots in Figure 7 positioned in both the triangles represented by direction 1 and direction 2) associated with it having m_l^1, m_l^2 membership values respectively. Also note (a) $m_l^1 + m_l^2 = 1$ and (b) d_l^1, d_l^2 are adjacent directions, for example if $d_l^1 = 5$ then d_l^2 could be either 4 or 6.

Algorithm 1 Triangular Fuzzy Membership Function

fuzzy-membership(θ_c, θ);
$m = 1.0 - \frac{(|(\theta_c - \theta)|)}{\pi/4}$;
return(m);

We use θ_l in Algorithm 2 assisted by triangular membership function described in Algorithm 1 for computing the FDF set (refer Figure 8). Here $\theta_1, \cdots \theta_{k-1}$ are the angles between two consecutive curvature points (where k is the total number of curvature points) in a handwritten primitive and $d_1, \cdots d_8$ are the respective directions. The fuzzy membership values assigned to each direction are represented as $m_1^1, m_1^2 \cdots, m_{k-1}^8$ and the corresponding to $f_1, \ldots, f_8$ feature vector values and k curvature points. It should be noted that the sum of the membership functions of a particular row in Figure 8) is always 1. We calculate the FDF by averaging across the columns, so as to form a vector of dimension eight. The mean is calculated as follows; for each direction (1 to 8),

Algorithm 2 Computation of Fuzzy Directional Features

deg2dir(double θ)
int i=1; d[i]=-1; m[i] = -1;
if ($\theta > -\pi/8$ & $\theta < \pi/8$) **then**
  d[i] = 1; m[i] = fuzzy-memebership(0,θ)
end if
if ($\theta >= 0$ & $\theta < 2\pi/8$) **then**
  d[i] = 2; m[i] = fuzzy-memebership($2\pi/8$,θ)
  i ++
end if
if ($\theta >= \pi/8$ & $\theta < 3\pi/8$) **then**
  d[i] = 3; m[i] = fuzzy-memebership($3\pi/8$,θ)
  i ++
end if
if ($\theta >= 2\pi/8$ & $\theta < 4\pi/8$) **then**
  d[i] = 4; m[i] = fuzzy-memebership($4\pi/8$,θ)
  i ++
end if
if ($\theta >= 3\pi/8$ & $\theta < 5\pi/8$) **then**
  d[i] = 5; m[i] = fuzzy-memebership($5\pi/8$,θ)
  i ++
end if
if ($\theta >= -5\pi/8$ & $\theta < -3\pi/8$) **then**
  d[i] = 5; m[i] = fuzzy-memebership($-3\pi/8$,θ)
  i ++
end if
if ($\theta >= -4\pi/8$ & $\theta < -2\pi/8$) **then**
  d[i] = 6; m[i] = fuzzy-memebership($-2\pi/8$,θ)
  i ++
end if
if ($\theta >= -3\pi/8$ & $\theta < -\pi/8$) **then**
  d[i] = 7; m[i] = fuzzy-memebership($-\pi/8$,θ)
  i ++
end if
if ($\theta >= -2\pi/8$ & $\theta < 0$) **then**
  d[i] = 8; m[i] = fuzzy-memebership(0,θ)
  i ++
end if
return(d[i],m[i]);

collect all the membership values and divide by the number of occurrences of the membership values in that direction. For example, in Figure 8, the mean for direction 1 is calculated as $f_1 = \frac{(m_2^1 + m_3^1)}{2}$. In all our experiments we have used these mean values to construct the 8 directional FDF to represent a stroke.

$$F = [f_1, f_2, \ldots, f_8] \quad (2)$$

Clearly, the fuzzy aspect comes into picture due to the membership function which associates the angle between two curvature points into two directions with different membership values. In the commonly used Directional Features only one direction is associated with each θ (the angle between two consecutive curvature points). The experimental results shows

TABLE I
NUMBER OF STROKES UNDER EACH PRIMITIVE IN THE TEST AND TRAIN DATA SET.

Sl.No	Primitive	Train	Test
1	R	82	73
2	k	69	57
3	nn	63	67
4	v	50	36
5	p	48	36
6	dd	54	50
7	m	39	51
8	tt	28	24
9	y	40	36
10	aa	33	28
11	h	34	16
12	T	26	25
13	g	24	25
14	D	22	21
	Total	612	545

that the use of Fuzzy Directional Features improves the performance of recognition compared to Directional Features.

III. Experimental Results

We collected on-line handwritten data from 10 different writers, each of whom wrote a Devanagari text paragraph (Figure 3) using Mobile e-Notes Taker[4]. The Mobile e-Note Taker is a portable pen based handwriting capture device which allows user to write on an ordinary paper using an electronic pen to capture handwritten text. We initially hand tagged each stroke in the collected paragraph data based on the 46 primitives (see Figure 1). The strokes that did not fall into this primitive set were marked as being out of vocabulary. We used 5 user paragraph data for training and the other 5 user paragraph for the purpose of testing. All the strokes corresponding to a given primitive in the training and test data set were collected separately. We retained those primitives that occurred at least 15 times in both the train and the test data set. From the collected paragraph data set, we obtained 14 such primitives. Table I shows the number of strokes retained in the test and train data set corresponding to the 14 selected primitives.

For training, we computed the Fuzzy Directional Features for all stroke corresponding to the same primitive as described in section II. Then we computed the mean (μ) and co-variance (Σ) for each class to model the primitive. If there are β strokes corresponding to a primitive, then we have β Fs, say, $F_1, F_2, \ldots, F_\beta$ then each primitive is modeled as

$$\mu = \frac{1}{\beta}\sum_{i=1}^{\beta} F_i \tag{3}$$

$$\Sigma = \frac{1}{(\beta-1)}\sum_{i=1}^{\beta}(F_i-\mu)(F_i-\mu)^T \tag{4}$$

[4]http://www.hitech-in.com/mobile-e-note-taker.htm

Here a primitive is represented by a mean vector (μ) of size 1×8 and co-variance matrix of size 8×8. The class reference models are represented as $(\mu_i, \sum_i)_{i=1}^{14}$. For testing purpose, we took a stroke (t) from the test data set and extracted FDF as described in section II and compared it with 14 reference models. The likelihood of the test primitive t with reference primitive k (where $k = 1, 2, \cdots, 14$) is computed as

$$P(t/k) = \left[\frac{1}{(2\pi)^{\frac{8}{2}}\sqrt{|\sum_k|}}\right]\exp^{-\frac{1}{2}(t-\mu_k)\Sigma_k^{-1}(t-\mu_k)^T} \tag{5}$$

The k for which equation (5) is maximum is identified as the recognized primitive. Table II shows the experimental results obtained for both the train and the test data for FDF and commonly used Directional Features (DF). Note that the values in Table II are computed as follows. For $N = \alpha$, the test stroke t is recognized as the primitive l if l occurs at least at the α^{th} position from the best match (this is generally called the N-best in literature). It should be noted that the recognition accuracies are for writer independent unconstrained strokes. As expected the recognition accuracies are not very high (very similar to the phoneme recognition in speech literature) for $\alpha = 1$ but improves with increasing α. It is also noted that similar experiments with Directional Features (DF) shows $\pm 10\%$ lower recognition efficiency.

TABLE II
RECOGNITION RESULTS FOR TRAIN AND TEST DATA SET

Feature Set	Data	$\alpha = 1$	$\alpha = 2$	$\alpha = 5$
FDF	Train	57.84%	79.90%	96.89%
	Test	48.26%	60.18%	82.75%
DF	Train	56.86%	73.69%	85.46%
	Test	45.14%	53.76%	74.86%

IV. Conclusion

In this paper we have introduces a new on-line script feature set, called the Fuzzy Directional Feature. We have evaluated the performance of the novel feature set by presenting the recognition accuracies obtained for writer independent unconstrained stroke level data set. The recognition results shows that the Fuzzy Directional Features (FDF) performs much better than commonly used Directional Features (DF). We plan to use (a) Viterbi trace back and (b) the spatio-temporal information to enhance character (constitute of multiple strokes) recognition. This we believe will lead to better accuracies of writer independent script recognition.

References

[1] N. Joshi, G. Sita, A. Ramakrishnan, and V. Deepu, "Machine recognition of online handwritten Devanagari characters," in *ICDAR05*, 2005, pp. 1156–1160.

[2] A. Namboodiri and A. Jain, "Machine recognition of online handwritten Devanagari characters," in *PAMI*, vol. 26(1), 2004, pp. 124–130.

$\theta \downarrow)\ d \rightarrow)$	1	2	3	4	5	6	7	8
θ_1			m_1^1	m_1^2				
θ_2	m_2^1	m_2^2						
θ_3	m_3^2							m_3^1
$\vdots$								
θ_l							m_l^2	m_l^1
$\vdots$								
θ_{k-1}					m_{k-1}^2	m_{k-1}^1		
F	$f_1 = \frac{(m_2^1+m_3^1)}{2}$	$f_2 = m_2^2$	$f_3 = m_1^1$	$f_4 = m_1^2$	$f_5 = m_{k-1}^2$	$f_6 = m_{k-1}^1$	$f_7 = m_l^2$	$f_8 = \frac{(m_3^1+m_l^1)}{2}$

Fig. 8. Computation of Fuzzy Directional Features

[3] S. Parui, K. Guin, and U. Bhattacharya, "Online handwritten Bangla character recognition using hmm," in *ICPR08*, 2008, pp. 1–4.

[4] U. Bhattacharya, B. Gupta, and S. Parui, "Direction code based features for recognition of online handwritten characters of Bangla," in *ICDAR07*, 2007, pp. 58–62.

[5] V. Babu, L. Prasanth, R. Sharma, G. Rao, and A. Bharath, "HMM-based online handwriting recognition system for Telugu symbols," in *ICDAR07*, 2007, pp. 63–67.

[6] S. Sundaram and A. Ramakrishnan, "A novel hierarchical classification scheme for online Tamil character recognition," in *ICDAR07*, 2007, pp. 1218–1222.

[7] A. Bharath and S. Madhvanath, "Hidden Markov Models for online handwritten Tamil word recognition," in *ICDAR07*, 2007, pp. 506–510.

[8] G. Menier, G. Lorette, and P. Gentric, "A genetic algorithm for on-line cursive handwriting recognition," in *ICPR*, 1994, pp. 522–525.

[9] S. Garcia Salicetti, B. Dorizzi, P. Gallinari, and Z. Wimmer, "Maximum mutual information training for an online neural predictive handwritten word recognition system," in *IJDAR*, vol. 4(1), 2001, pp. 56–68.

[10] J. Schenk and G. Rigoll, "Neural net vector quantizers for discrete HMM-based on-line handwritten whiteboard-note recognition," in *ICPR*, 2008, pp. 1–4.

[11] S. Connell and A. Jain, "Template-based online character recognition," in *Pattern Recognition*, vol. 34(1), 2001, pp. 1–14.

[12] P. Mukherji and P. Rege, "Fuzzy stroke analysis of Devnagari handwritten characters," in *W. Trans. on Comp.*, vol. 7(5), 2008, pp. 351–362.