Knotless spline based smoothing for on-line hand written Character recognition

Lajish VL, Vinod Kumar Pandey, and Sunil Kumar Kopparapu
TCS Innovation Labs - Mumbai, Tata Consultancy Services,
Yantra Park, Thane (West), Maharashtra, India - 400601
email: {Lajish.VL,Vinod.Pande,SunilKumar.Kopparapu}@TCS.Com

Abstract**—**There is inherent noise in on-line hand written characters due to shaking of the hand while writing and due to the process of digitization which causes degradation of performance of any character recognition algorithm. In this paper we describe a novel noise removal technique based on knotless splines which can significantly enhance the performance of the character recognition algorithm. We first compare the noise removal technique with a standard wavelet based noise removal technique. We conduct experiments on real on-line devanagari data and show that noise removal based in knotless splines enhances the performance of the character recognition algorithm. The recognition algorithm is based on the extraction of a novel feature set called the Fuzzy Directional Features (FDF).

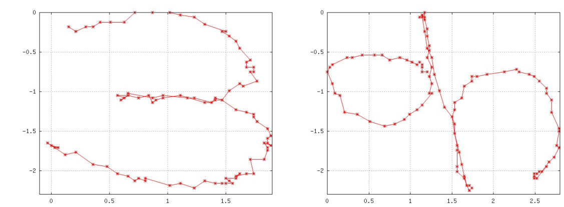
Keywords**—**On-line handwriting recognition, pre-processing, smoothing, free knote spline, filtering, fuzzy directional features.

I. Introduction

With advancement in wireless technology, PDAs, palm and handheld PCs are more frequently being used for composing short messages and e-mails. Electronic pen (e-pen) and /or stylus touching a pressure sensitive digital tablet are some of the popular non-keyboard data entry devices that is gaining popularity. The devices themselves use on-board hand writing recognition algorithms to transforms graphical form of characters into an electronically transferable character string. Generally strokes, defined as a trace of a pen between and pen-down and an pen-up, have non-uniformly sampled data points; the number of data points is very sparse especially when the pen movement is fast and when the pen movement is slow they are prone to be contaminated by high frequency noise, caused by digitization error of the input device and natural fluctuations by the writer during writing.

Noise in inherent in a hand written on-line character data. There are essentially two types of noise that contributes to the noisy data, the first is the inherent shake by the hand of a person while writin the character espeecially at the begining and end of the stroke while the second is contributed by the noise creeping in due to the digitization process.

Figure 1 (a) and (b) shows sample on-line Devanagri characters *”aa” and ”k”* collected from a writer. We observe that the characters are contaminated by high frequency noise. The noise severely effects the performance of a on-line character recognition algorithm. There are essentially two ways of taking care of the noise, in the first case an approriate noise removal algorithm is used on the raw and noisy data and in the second case one comes out with a feature extration algorithm that can compensate for the noise. In this paper, we propose a new noise reduction technique based on knotless spline fit for on-line handwritten data. We compare the noise removal performance of the proposed technique with the wavelet based noise removal technique and show that the proposed technique has better noise removal property. We also show the enhanceent of a character recognition algorithm due to noise removal on a real data set using a Fuzzy Directional Features (FDF) set. The rest of the paper is organized as follows. We review noise reduction techniques applied on on-line handwriting data in section II and introduce a novel noise reduction technique in Section III. In Section IV we describe the procedure for identifying the curvature points followed by Fuzzy Directional Features extraction technique and conclude in Section **??**.



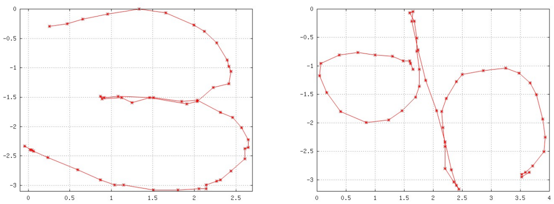
(a) Devanagri ”aa” (b) Devanagri ”ka”

Fig. 1. Sample on-line Devanagari characters *”aa” and ”k”*: Original

II. Noise reduction techniques

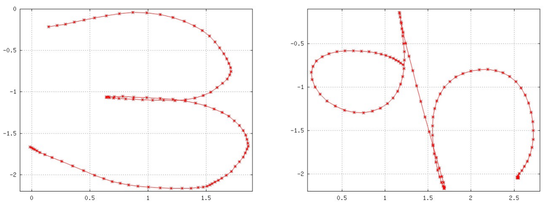
Noise in on-line handwritten script data typically appeares due to the hand shaking/fluctuation during writing and also due to digitizing process of the on-line data capturing devices. Invariably the raw on-line data are often found noisy and inconsistent, this necessitates use of noise removal as a necessary pre-processing step prior to feature extraction and recognition algorithms. Noise removal helps in enhancing the performance of charaacter recognition algorithm. Figure 1 (a) and (b) shows sample on-line Devanagri characters *”aa”* and *”k”*. It can be crearly seen that there is a large amount of noise.

Smoothing and filtering are the two important techniques used for noise reduction. Smoothing usually averages a data point with respect to its neighbouring data points such that there is no *large* variation between adjacent data points (for



(a) (b)

Fig. 2. (a)-(b) On-line Devanagari characters *"aa"* and *"k"*: Filtered uisng DWT



(a) (b)

Fig. 3. (a)-(b) On-line Devanagari characters *"aa" and "k"* : Smoothed using proposed procedure

example [1], [2], [3], [4], [5], [6]). Filtering on the other hand eleminates duplicate[1] data points and reduces the number of points. Some filtering techniques force a minimum distance between adjacent data points (for example [2], [3], [4], [7], [8]) which tend to produce data points that tends to be equally spaced. According to the writing speed, the distance between the points may vary significanly and iterpolation can be used to obtain uniformly sapced points. In some filtering, minimum change in the direction of the tangent to the drawing for consecutive points are maintained [9]. This produces more points in the greater curvature regions. Filtering can also be done by the use of convolution with one dimensional Gaussian kernels [10], which reduces the noise due to pen vibrations and errors in the sensing mechanism. Joshi et al.[11] reduced the effect of noise with the help of 5-tap low pass Gaussian filter, where each stroke is filtered separately. Malik et al.[12] proposed time domain filter by using the convolution of input sequence with a finite impulse response for smoothing a jitter appeared in the sequence. Although many techniques exist in literature to suppress noise, it is very difficult to select a technique such that it can work equally for all types of strokes. In some other studies smoothing and filtering are performed as part of single operation. An example of this is piecewise-linear curve fillting [13], [14].The following section describes the proposed technique for noise removal. Figure 2 (a) and (b) show the noise removal due to application of a single level one dimensional Discrete Wavelet Transformation (DWT) using Daubechies wavelet on the Devanagri characters *"aa" and "k"*.

III. Knotless Spline for noise removal

Noise in the data can severely affect the recognition performance of the most effective algorithms. We use a technique based on splines for on-line character noise removal. Different spline based smoothening have been successfully applied for denoising noise contaminated signal for example, [15], [16]. However, the degree of smoothness depends on the number and position of the control points and chosen knots. If the knots are *close* to each other, the smooth curve between the two knots would be linear. If the knots are *far* apart, a higher order polynomial would be needed for fitting a smooth curve between the two knots. We propose the use of a cubic spline based polynomial approximation with knots being selected automatically[2]. Hence, the smoothening technique becomes a cubic polynomial curve fitting with a variable span.

Let the sequence $(x_i^r, y_i^r)_{i=0}^n$ represents a hand written stroke made up on n points. For the pupose of noise removal we treat the sequence x_i^r and y_i^r separately and remove noise from each of these sequences independently. The noise removal process is described below for the sequence x_i^r.

1) Set the span to be $n/2$ data points (consider only $n/2$ of the orginal n points, namely, $\{x_i^r\}_{i=0}^{n/2}$)
2) Fit a cubic spline in this span[3] compute and a mean squared error (MSE) is calculated between the fitted spline and the actual data points, namely, find $\{a_i\}_{i=1}^3$ such that $f(x_i^r) = a_0 + a_1 x_i^r + a_2 {x^r}_i^2 + a_3 {x^r}_i^3$ such that $MSE = \sum_{i=1}^{n/2} (f(x_i^r) - x_i^r)^2$ is minimum
3) Reduce the span by 25% (namely, consider $\{x_i^r\}_{i=0}^{n/2-n/8}$) and repeat Step 1 and 2. Until the span is 20% of the initial span.
4) The span with smallest MSE is selected as the optimum span with the starting and end points of the span are the chosen knots and a cubic spline is fitted in this span.
5) Repeat on the remaining data points

It is to be noted that this process automatically selects the number and the location of the knots unlike other spline denosing techniques which requires the user to specify the number of knots.

Figure 3 (a) and (b) shows on-line devanagri characters *"aa" and "k"* smoothed using the proposed procedure.

We compare the effect of noise removal using (a) wavelet denoising technique and (b) the method proposed above; the results are shown visually in Figure **??**. It can be clearly seen that the noise removal is better using the proposed knotless spline denoising process compared to the wavelet based denoising.

IV. Experimental Results

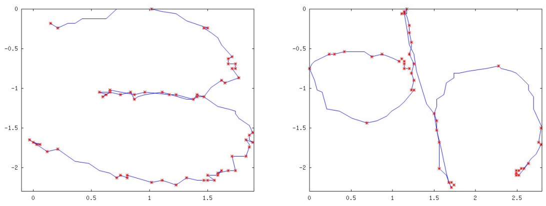
We collected on-line handwriting data from 10 different writers using Mobile e-Note Taker[4]. It is a portable pen based handwriting capture device which allows user to write on an ordinary paper using an electronic pen to capture handwritten

[1]Lajish: Need to define

[2]both the number of knots and the location of the knots

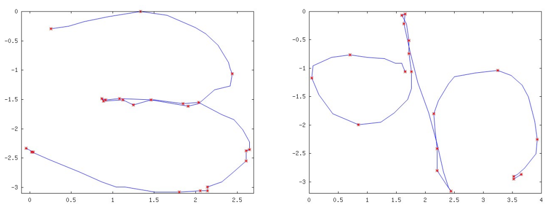
[3]span is defined as the distance between the two consecutive knots

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



(a) (b)

Fig. 4. (a)-(b) Curvature points identified on On-line Devanagari characters *"aa" and "k"* : Filtered uisng DWT



(a) (b)

Fig. 5. (a)-(b) Curvature points identified on On-line Devanagari characters *"aa" and "k"* : Filtered uisng DWT

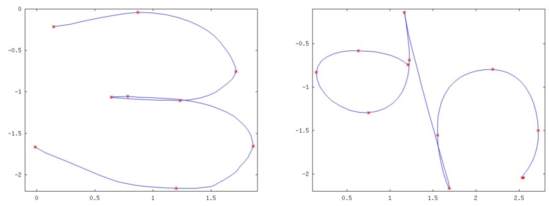
text the output from the device was a $\{x^r, y^r\}$ trace of the pen trace (raw data), sampled uniformly in time at 100 Hz. The raw data was preprocessed[5] to get the denoised or smoothned sequence (x, y).

A. Curvature point extarction

The curvature points (also called critical points) are extracted from the smoothed handwriting data. The denoised sequence $(x_i, y_i)_{i=0}^{n}$ represents a noiseless hand written stroke. We treat the sequence x_i and y_i separately and calculate the critical points for each of these sequence. 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. 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 Figures **??**). It must be noted that the position and number of curvature points computed for different samples of the same strokes may vary. Figure 4 shows the curvature points identified from the original samples of on-line devanagri characters *"aa" and "k"*.

We have applied single level one dimentional Discrete Wavelet Transformation (DWT) based decomposition for noise reduction using Daubechies wavelet. Figure 5 (a) and (b) shows the curvature points identified from the DWT filterd on-line devanagri characters *"aa" and "k"*. Figure 6 shows the curvature points identified from the on-line devanagri characters *"aa" and "k"* smoothed using the proposed procedure.

[5]in this paper we used both wavelet denoising technique and the proposed denoising technique



(a) (b)

Fig. 6. (a)-(b) Curvature points identified on On-line Devanagari characters *"aa" and "k"* : Filtered uisng DWT

By observing the curvature points identified from the above patterns, it is evident that the number and position of the curvature points are more consistant while we apply the proposed free knot spline based smoothing[6]. The follwoing sessions describe the feature extraction technique and the recognition experiments.

B. Feature extraction and recognition experiments

Several temporal features have been used for script recognition in general and for on-line Devanagari script recognition in particular [17], [18], [19], [20]. We propose a simple yet effective feature set based on fuzzy directional feature set[7]. Let k be the number 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 critical points, say c_l and c_{l+1}, as

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

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 in to eight directions with overlap. Every θ_l (for example the angle θ that the blue dotted line makes with the horizontal axis in Figure 7) has two directions (say $d_l^1 = 1, d_l^2 = 2$, note that the line in dotted line in Figure 7 lies in both the triangles represented by direction 1 and direction 2) associated with it having m_l^1, m_l^2 membership values respectively (represented by the green and the red dot respectively in Figure 7. 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 $\theta_{1,k-1}$ is the angle between

[6]Lajish: Check

[7]Note that [21] talks of fuzzy feature set for Devanagari script albeit for offline handwritten character recognition

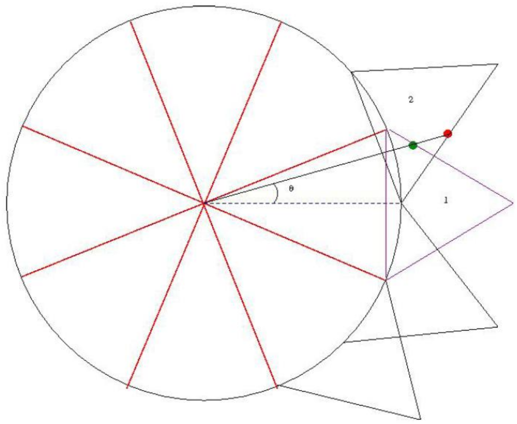


Fig. 7. θ contributing to two directions $(1, 2)$ with fuzzy membership values (green and red dot)

two consecutive curvature points (where k is the total number of curvature points) in a handwritten primitive and $d_{1,8}$ is the respective direction. The fuzzy membership values assigned to each direction are represented as $m^{1,2}_{1,k-1}$ and the corresponding feature vector values as $f_1, \ldots,$ and f_8. 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), 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 $f_1 = \frac{(m^1_2 + m^1_3)}{2}$. In all our experiments we have used these mean values to construct 8 directional FDF

$$F = [f_1, f_2, \ldots, f_8] \tag{2}$$

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

The recognition experiments are then conducted as follows. As a training step, we calculated FDF for all traing strokes corresponding to the same primitive and the average FDF is considered as the cluster centre for that primitive. For testing purpose, we took a stroke t from the test data and extracted FDF. Then the distance between the test DFD and all reference cluster centers are found out using the Euclidian distance measure. We then took the minimum distance of the stroke from all the references. The primitive with the least distance from the test stroke t is declared as being recognition of stroke t. The recognition experiments are then separatley conducted for raw data, DWT based smoothed data

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]);

and the data denoised with the proposed free knote spline based pattern smoothing technique. The average recognition accuracies obtained for 44 Devanagari character primitives were about 57.84% using raw data, 82.75% based on DWT based smoothed data and about 87.8% using the proposed method.

V. Conclusion

In this paper we have evaluated a new free knote spline based technique for denoising on-line handwriting data. The proposed method is tested on devanagari on-line handwriting primitives and evaluted. We also compared the recognition performance when fuzzy direction features are formulated using the row data, dwt based smoothed data and the proposed free knote spline based amoothed data. The average

$\theta \downarrow)\ d \rightarrow)$	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

recognition rates obtained for 44 Devanagari character primitives were about 57.84% using original on-line handwriting data, 82.75% based on DWT based smoothed data and about 87.8% using the proposed pattern smoothing technique.

References

[1] K. Odaka H. Arakawa and I. Masuda, "On-line recognition of handwritten characters-alphamlmerics," in *Proc. 4th Int. Joint Conf. Pattern Recognition*, 1978, pp. 810–812.

[2] M. I. Bernstein, "A method for recognizing handprinted characters in real-time," in *in L. N. Kanal, Ed., Pattern Recognition. Washington: Thompson*, 1968, pp. 109–114.

[3] B. Blesser, "Multistage digital filtering utilizing several criteria," in *U.S. Patent 4 375 081*, 1983.

[4] G. F. Groner, "Real-time recognition of handprinted text," in *Proc. FJCC*, 1966, pp. 591–601.

[5] M. Hosaka and F. Kimura, "An interactive geometrical design system with handwriting input," in *Proc. 6th IFIPS Congr.*, 1977, pp. 167–172.

[6] Ching Y. Suen Charles. C. Tappert and T. Wakahara, "The state of the art in on-line handwriting recognition," in *IEEE Trans. on PAMI*, 1990, pp. 787–808.

[7] S. Hanaki, T. Temma, and H. Yoshida, "An on-line character recognition aimed at a substitution for a billing machine keyboard," in *Pattern Recognition*, 1976, vol. 8, pp. 63–71.

[8] E. F. Yhap and E. C. Greanias, "An on-line chinese character recognition system," in *IEMJ. Rex Develop*, 1981, vol. 25, pp. 187–195.

[9] M. Berthod and J. P. Maroy, "Morphological features and sequential information in real-time hand-printing recognition," in *Proc. 2nd Int. Joint Conf. Pattern Recognition*, 1974, pp. 358–362.

[10] Anoop M. Namboodiri and Anil K. Jain, "On-line handwritten script recognition," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004, vol. 26, p. 124130.

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

[12] Sumaila Malik and Shoab A. Khan, "Urdu online handwriting recognition," in *IEEE International Conference on Emerging Technolgies*, 2005, p. 2731.

[13] K. Asai. S. Hanaki and K. Kiji, "Real time recognition of handprinted characters-japanese character katakana recognition," in *Proc.Znd Hawaii Int. ConJ Syst. Sci.*, 1969, pp. 345–348.

[14] M. R. Ito and T. L. Chui, "On-line computer recognition of proposed standard ansi (usa) handprinted characters," in *Pattern Recognition*, 1978, vol. 10, pp. 341–349.

[15] Solomon.D, ," in *Computer Graphics and Geometrical Modeling*. 1999, Springer.

[16] De Boor. C, ," in *A practical guide to splines*. 1978, Springer Verlag.

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

[18] 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*, 2001, vol. 4(1), pp. 56–68.

[19] 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.

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

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