Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

Preprocessing of online handwritten Telugu character recognition

Srilakshmi Inuganti [1,*], Rajeshwara Rao Ramisetty [2]

[1]Computer Science and Engineering, GMR Institute of Technology, Rajam, India
[2]Computer Science and Engineering, UCEV, JNTUK, Vizainagaram, India

ARTICLE INFO

Article history:
Received 28 February 2017
Received in revised form
22 June 2017
Accepted 29 June 2017

Keywords:
Online handwriting recognition
Preprocessing
Telugu strokes
Character recognition

ABSTRACT

Online Handwritten Character Recognition (OHCR) is the method of recognizing characters by a machine while the user writes, in which the handheld devices record (x, y) coordinates of the track of the character. With the advent of handheld devices, there is a great attention towards OHCR of regional languages. Preprocessing is the main phase, in OHCR, as it increases the performance of succeeding phases, by removing the inconsistency or the redundancy present in the data collected in real-world environment. In this paper, we depict the model of Preprocessing of Online Handwritten Telugu Strokes. The preprocessing steps we address in our article are Normalization, Smoothing, Duplicate Point Removal, Interpolation, Dehooking and Resampling. Preprocessing data performance is evaluated through parameters namely recognition accuracy, recognition speed, false acceptance rate and false rejection rate over HP labs dataset hpl-Telugu-ISO-char-online-1.0. The dataset contains samples of the 166 character classes collected of different writers on ACECAD Digimemo (A4 sized) using an AcecadDigi memo DCT application. It consists of 270 samples on average for each of 166 Telugu "characters" written by native Telugu writers.

1. Introduction

With handheld devices reaching new heights of popularity every day and becoming almost indispensable in our busy lives, digital pens become a great alternative to keyboards, especially in case of PDAs, Hand Held PCs and high end mobile devices. A digital pen captures the handwriting of a user, converts handwritten information into digital data, enabling the data to be utilized in various applications. In this context Handwritten Character Recognition (HCR) is an immediate challenge in the area of pattern recognition.

HCR can be classified into Online HCR and Offline HCR. OHCR is the task of identifying character written by a machine while the user writes, in which transducer required for capturing dynamic handwriting information. The dynamic information contains numbers, order, length, writing direction and speed of stroke and some devices record pressure information also (i.e. at pen tip). A stroke is the writing form pen-down to pen-up.

Offline HCR is a sub category of Optical Character Recognition. In offline HCR character is recognized after completion of writing. Offline HCR takes a raster image from digital input source and converts into binary image, so that image pixel values are either 0 or 1.

The progressive study of handwritten character Recognition shows online HCR has many advantages over offline HCR. Offline data is not associated with temporal information. It only represents the final result as an image. So knowledge about the character is less. Online data are associated with temporal information, so that accuracy is high in adverse to offline. Online data are highly interactive.

Hence, errors can be debugged immediately with repeated tests. Online data offers reduction in memory and therefore space complexity. Even though many years of research in handwriting recognition (Plamondon and Srihari, 2000; Bharath and Madhvanath, 2009), very less has been made towards Indian languages. OHCR is more realistic for Indian languages which have huge character set.

1.1. Framework of OHCR

The block diagram OHCR illustrated in Fig. 1. The details of each step are described in the following paragraphs.

* Corresponding Author.
Email Address: srilakshmi.i@gmrit.org (S. Inuganti)
https://doi.org/10.21833/ijaas.2017.07.025
2313-626X/

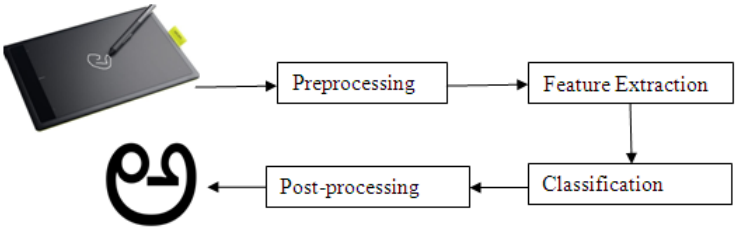


Fig. 1: Steps in OHCR

1.2. Data collection

Online handwriting recognition software incorporates the automatic conversion of the text simultaneously with the user's writing. This type of online processing makes use of special digitizers. These digitizers like PDAs use sensors to track movement of the input device like a stylus pen. When the pen tip makes contact with the screen, the sensors are activated. When the contact is broken, the sensors are automatically turned off. Some users refer to this processing method as digital inking, which can also be considered as a type of dynamic representation in the handwriting process. The acquisition interface outputs a sequence of (x, y) - coordinates representing the location of pen tip and binary value indicates pen up/pen down, the coordinates are recorded only period when the pen is in contact with the interface. This period is known as stroke.

1.2.1. Commercial products available

Numerous handwriting recognition software vendors take into account the type of digital device that can host their programs. This is because the use of handwriting recognition technology is fast becoming a standard not only for mobile devices, but for other equipment as well. PDAs, laptops, tablets, and mobile phones are but some of the digital devices which make use of handwriting recognition software in their operation. Nowadays Digimemos are also used which operate with only normal pen. In following paragraphs some commercial products available are described.

The Wacom Intuos Pro is a tablet with stylus. It is available in multiple sizes and its pen technology is pressure sensitive and cordless. This product acts as an input device by connecting it to the computer via USB. The pressure level is of 1024 (pen tip only). It has a resolution of 100 lines/mm (2540 lpi). The reading speed of pen is 133 pps. This sleek tablet is compatible with a high range of operating systems such as the Windows 7, Mac OS, Windows XP, Vista SP2, and so on.

A tablet PC is a special notebook computer with a digitizer tablet and stylus. It allows user to write on screen, the operating system recognizes the character. iBall Pen Tablet offers users the flexibility to write, interact with the graphical user interface and performs other input related tasks in a very user friendly manner. It offers 1024 pressure levels.

The ACECAD Dig memo is a standalone device that captures and stores everything you write without the use of computer and special paper. When connected to a PC, it offers on-line handwriting functions which can synchronize your writing on paper with its digital page in its software window. It is very portable, so that people can easily operate while standing or sitting. People feel as comfortable as with a regular pen on paper. So that data can be easily collected from computer novice also.

HP labs Data Collection Tool (DCT) provides user interface by which handwriting samples from different writers can be collected. Given the script and the data elements to be collected, the writer can give their handwriting samples.

The current version of DCT is supported on a Tablet PC with Microsoft Windows XP Tablet PC edition and Desktop PC with Microsoft Windows XP operating system. Fig. 2 shows some of the commercial products available.

Fig. 2: Some commercial products

1.2.2. Standards for online data representation

The following are the main standards for representing online-data:

- *UNIPEN format:* The UNIPEN format is a common data format, easy to exchange (Guyon et al., 1994). Users can easily convert their data to UNIPEN format, collect new data directly. The data were

developed and tested in collaboration with many industrial experts. The strokes are recorded from PEN_DOWN to PEN_UP.

- *Digital Ink markup language:* InkML is an XML-based markup language to describe "ink" data input with an electronic pen or stylus (Chee et al., 2006). The recommended specification was published by the World Wide Web Consortium (W3C) in September 2011. All content of an InkML document is contained within a single <ink>element. The fundamental data element in an InkML file is the <trace>. A trace represents a sequence of contiguous ink points, where each point captures the values of particular quantities such as the X and Y coordinates of the pen's position.
- *UPX:* UPX, an XML-based successor of UNIPEN. It addresses the limitations of UNIPEN and InkML (Agrawal et al., 2005). It allows the user to easily add specific information to ink files to suit the needs of the application.

1.3. Preprocessing

Before applying input to the system to get correct recognition result, data need to be pre-processed. As the data collected in real environment, it can be noisy and inconsistent. The important target of preprocessing is to eliminate the effect of noise, variation in writing size and style and repetition of points. It is carried out in five steps- Normalization, De-Hooking, Smoothing, Duplicate Point Removal, Interpolation and Resampling. Even though preprocessing enhances recognition accuracy, excessive preprocessing is undesirable because it may result in loss of valuable information.

1.4. Feature extraction

Feature extraction starts with measured data and builds features, which are informative and non-redundant. These features extracted should maximize inter-class similarity and minimize intra-class similarity.

1.4.1. Fixed length vs. variable length feature vector representation

Features of online handwriting units can be fixed length or variable length (Swethalakshmi et al., 2006). In fixed length representation, predefined length of feature vector is extracted from stroke. Fixed length representation uses reserved space large enough to accommodate larger data. These features include direction information, direction feature, structural features, etc.; Variable length feature vector is suitable for complex stroke and structure variations. This representation is useful for template matching like Dynamic Time Wrapping (DTW) and Hidden Markov Model (HMM) based approaches. Fixed length representations are useful in Support Vector Machine (SVM) and Principal Component Analysis (PCA) based approaches (Sundaram and Ramakrishnan, 2008).

1.4.2. Local vs. global features

Basing on the granularity at which features are extracted, they are categorized as local and global. Local features are extracted at a point of the stroke. The most common local features are x-y coordinates, local direction features, i.e. the relative vector of two adjacent points (Shashikiran et al., 2010). Global feature is defined as a relative vector between any arbitrary points (Rampalli and Ramakrishnan, 2011). These features are extracted at a stroke level or sub stroke level. Examples of global features are moments, Fourier descriptors, projections, etc. Global features are good at capturing overall information. They don't work well with similar classes that have minor variations. Local features are extracted at each point, good for inter-class separation. The features proposed in the literature so far are either local or global which fail to capture essential information about the character. A combination of local and global features has been proposed to capture local and global variations (Hollerbach, 1981).

1.5. Classification

A number of different models have been applied to Indian OHCR. Different models of online handwriting recognition are illustrated by Bellegarda et al. (1993) and Plamondon and Maarse (1989). These recognition models are Motor Models, Structured Based Models, Statistical Models, and Neural Network Models. Work in each of the above mentioned methods is illustrated in the following sections. The merits and demerits of each of these models given in Table 1.

1.5.1. Motor methods

Motor models (Uchida and Sakoe, 2005) are a technique commonly used in what is known as Analysis by Synthesis in which models of stroke segments are created along with rules for connecting them to form characters. Motor models represent these stroke segments as parameterized models of the motion of the pen tip, simulating the physical properties of human hand motion.

1.5.2. Structured based models

In Structure Based Methods different examples of strokes are considered as primitives (Chan and Yeung, 1998). The distance of test pattern with reference pattern is calculated. The distance measures can vary from the Euclidian distance to Mahalanobis distance. Structure based methods are weak at collecting data, but good at recognizing variations. The elastic matching approach followed in Lei et al. (2012), which works on sequence of

sample points directly by comparing the alignment of input pattern with reference pattern. A direction string approach is described in Zeng et al. (2006), in which each stroke is represented in the form of direction followed. DTW method is followed in Cho and Kim (2004). DTW compares online trajectories of the coordinates; trajectories include temporal and spatial information. If the character is represented as graph, graph matching algorithms can be applied to classify the character. Delaunay triangulation features are used in Do and Artières (2006).

1.5.3. Statistical models

These methods are probabilistic and need powerful calculators and considerable calculation time. This character is classified by selecting the class which is most probable or has a minimum amount of classification error.

Well known probabilistic model Bayesian decision rule is applied for OHCR in Babu et al. (2007). A probabilistic discriminate model Conditional Random Fields is applied in Kerrick and Bovik (1988). Another popular statistical method is HMM, the success of which motivated towards the application of HMM to OHCR (Poisson et al., 2002). HMM is trained from each stroke class using the observation sequence obtained when stoke is written. HMM allows variable length feature vector. Supervised learning method SVM is also another widely used OHCR method (Swethalakshmi et al., 2006). Decision trees can also be applied for classification, where prior probabilities can be used (Marukatat et al., 2001).

1.5.4. Neural network models

Neural Network methods for OHCR are gaining popularity, because of their performance in other areas. Multi Convolution Neural Networks are applied in Graves et al. (2008). Combination of HMM and Time delay Neural Networks have shown good performance for cursive script recognition in Madhvanath et al. (2007).

A Recurrent Neural Network's approach is applied in Reddy et al. (2012). It has the ability to make use of previous context.

Table 1: Review of different techniques of online handwriting

Techniques	Merits	Demerits
Motor Models	Uses advantages of Pen Dynamics	May lack robustness when writing style variations are large
Structured Based Models	Works well for writer-dependent data	Does not work very well for write-independent data, Recognition time rises linearly with the number of training examples
Statistical Models	Temporal relations are very well modelled	Requires a large amount of training data
Neural Network Models	Less Classification time	Temporal relations are not very well modelled

1.6. Post processing

After analysis of the confusion matrix, confusing pairs are identified. Script specific features can be used to resolve ambiguities in confusing characters.

2. Database

The Dravidian language Telugu is the official language in the states of Andhra Pradesh and Telangana. The government of India designated Telugu as the one of six classical languages of India. Telugu is the native language of 75 million people; according to 2011 census.

The syllables in Telugu script are vowels, consonants, and their combinations. The typical forms of syllables are V, CV, CCV and CCCV, and so they have a generalized form of C*V. Thus the basic units of character of script $O(10^2)$, these units forming $O(10^4)$ number of composite characters. Table 2 shows the possible combinations of Telugu characters.

In Telugu the data set available is Hp-Labs data in UNIPEN format. This dataset contains nearly 270 samples for each of 166 Telugu "characters" written by native Telugu writers (Sharma, 2009). The data are collected using Acecad Digimemo electronic clipboard devices using the Digimemo-DCT application. Fig. 3 shows 166 distinct symbols of Telugu.

These 166 symbols are collected from 146 users in two trials. Among these collected 45,219 samples, 33,897 samples are used for training and remaining is used for testing.

Table 2: Combination of telugu characters

Character	Type
V	13
C	35
CV	455
CCV	15925
CCCV	557375
NUMERALS	10
Total	573813

The symbol set containing vowels and consonants of Telugu script is shown in Fig. 3.

3. Preprocessing techniques of Telugu strokes

Preprocessing is the main phase of online handwritten character recognition, as Strokes obtained by a digitizer may contain a small amount of inconsistency or redundancy, most likely caused by digitizing device or inexperienced users having an erratic handwriting. Generally, these types of collected data may influence next phases, such as feature extraction and classification.

A preprocessing technique over benchmark UNIPEN database is applied by Huang et al. (2009), which reports an increase in accuracy rate by 10%. Swethalakshmi et al. (2006) proposed Gaussian filter for smoothing and normalize the data with respect to size and position. Sharma et al. (2009) has applied beizer interpolation in this article we propose preprocessing techniques for online handwritten Telugu strokes (Inuganti and Rao, 2015).

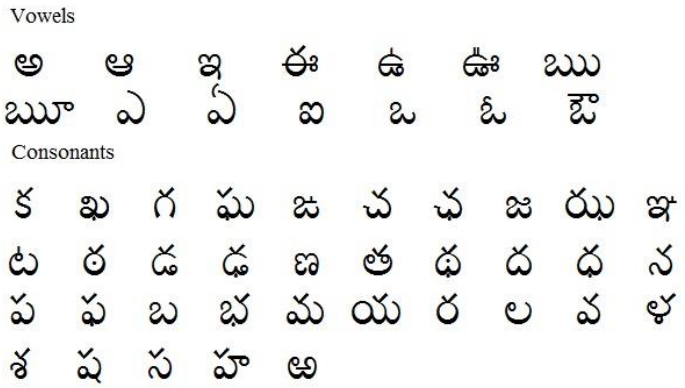


Fig. 3: Distinct symbols in Telugu

Initially, we normalize the character with respect to size and starting position. After Normalization duplicate points are removed and linear interpolation is applied for finding missing points of the stroke. Finally sharp points are detected to remove hooks and equi-distant resampling is applied.

The preprocessed character is recognized using K-Nearest Neighbor based on dynamic time warping (DTW) distance measure and by using pen-tip position as feature. As mentioned previously these techniques are applied over the HP dataset. The data set is in the form of Coordinates. These co-ordinates are input through writing pads by using stylus/pen. The strokes are recorded from PEN_DOWN to PEN_UP. The representation of one stroke is shown in the Fig. 4.

```
.VERSION 1.0
.HIERARCHY CHARACTER
.COORD X Y T
.SEGMENT CHARACTER 0 OK "0"
.H_LINE 749 1499
.V_LINE 249 999
.X_DIM 749
.Y_DIM 749
.X_POINTS_PER_INCH 1000
.Y_POINTS_PER_INCH 1000
.POINTS_PER_SECOND 125
.COMMENT CALIB X: -11 Y: -104
.PEN_DOWN
621 981 0
613 982 0
605 983 0
597 985 0
590 986 0
583 988 0
576 989 0
569 990 0
563 991 0
557 994 0
551 997 0
.PEN_UP
.PEN_DOWN
544 1001 0
539 1008 0
532 1014 0
526 1020 0
521 1027 0
516 1034 0
.PEN_UP
```

Fig. 4: Representation of stroke

The variation in writing a character by same user is shown in Fig. 5 and the variations in writing the same character by different users is shown in Fig. 6.

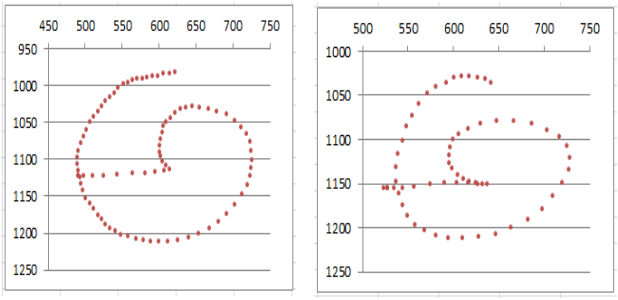


Fig. 5: Variants "Id_0" written by User 1

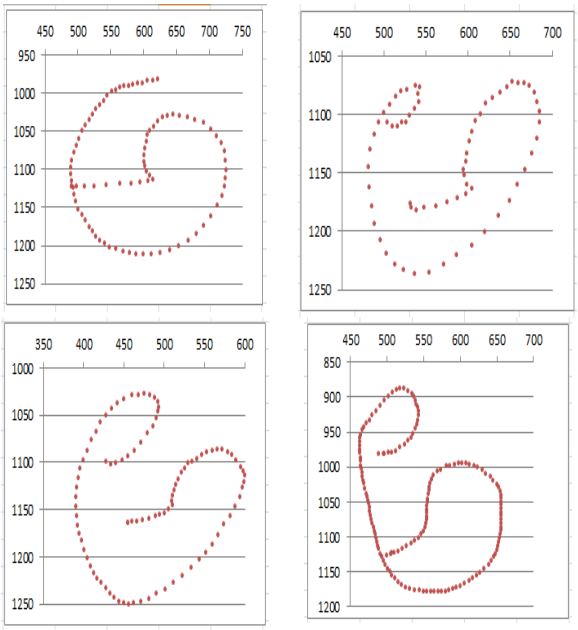


Fig. 6: "Id_0" written by 4 different users

3.1. Normalization

Usually the recognition rate is high, if we normalize the character with respect to the width and height, along with a starting point. In this paper we normalize the size and starting position of the stroke. In our window of size 260X250 pixels are considered for writing area. The size of the character

varies from one user to another and time to time also.

In size normalization the x and y coordinates are scaled both horizontally and vertically. The scale factors are calculated based on the ratio of height and width of the character with respect to height and width of the display window. In positional normalization every character is normalized with respect to starting position of the first stroke by using the translation of coordinates (Guerfali and Plamondon, 1993). The algorithm for size and position normalization of stroke is given below:

Algorithm:

In this algorithm the starting point is considered as (x_c, y_c) and set of pixels in which a Telugu stroke is represented as

$\{(x_i, y_i): xW_{min} \leq x_i \leq xW_{max}, yW_{min} \leq y_i \leq yW_{max}, i = 0,1, \ldots n,$

{n No. of pixels in the Telugu Character}

where:

$xW_{min} = min\{x_i\}, xW_{max} = max\{x_i\}, yW_{min} = min\{y_i\}, yW_{max} = max\{y_i\}$

Algorithm:

1. Set $xV_{min} = 0, xV_{max} = 260, yV_{min} = 0, yV_{max}$=250

2. $S_x = (xV_{max} - xV_{min})/(xW_{max} - xW_{min})$

$S_y = (yV_{max} - yV_{min})/(yW_{max} - yW_{min})$

3. $P_{xc} = x_c - x_0$

$P_{yc} = y_c - y_0$

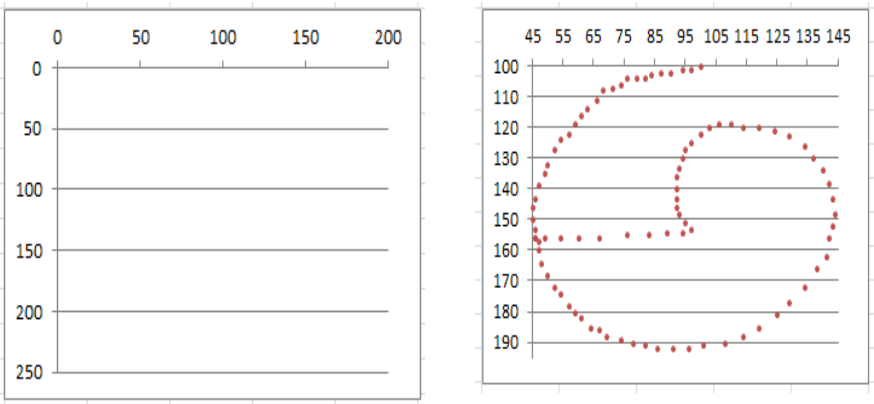
4. $x_i = (x_i - xW_{min}) * S_x$ $\forall$ *points i=1,2,... n*

$y_i = (y_i - yW_{min}) * S_y$

5. $x_i = (x_i + P_{xc}) \forall$ *points i=1,2,... n*

$y_i = (y_i + P_{yc})$

The above algorithm normalizes the stroke in size and starting position (100,100). The result is depicted in Fig. 7.



a) Input Character larger than display area b) Size and Position Normalized character

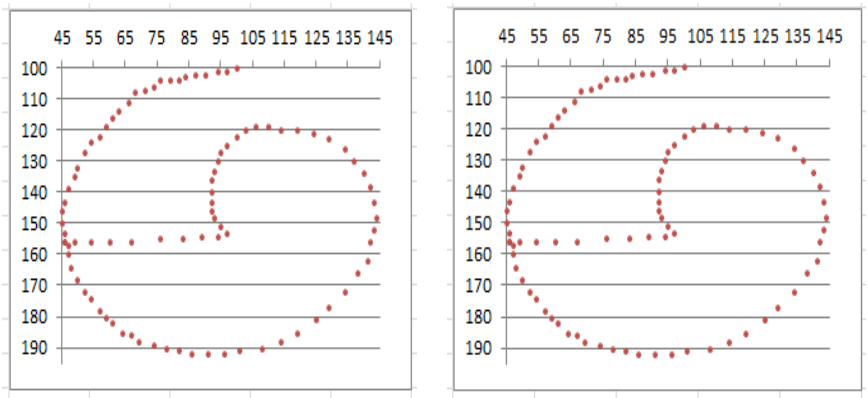
Fig. 7: Normalized character

3.2. Smoothing

Smoothing is performed to reduce the jitters in input obtained from the hardware or hand motion. In this paper a linear smoothing approach is adopted. The end points are preserved by taking special care. Each pattern is smooth both in horizontal and vertical directions separately (Li and Yeung, 1997). In linear smoothing a new coordinates (x_i, y_i) is calculated as Eqs. 1 and 2. The result is given in the Fig. 8.

$$x_i = (x_{i-1} + 2x_i + x_{i+1})/4 \quad (1)$$

$$y_i = (y_{i-1} + 2y_i + y_{i+1})/4 \quad (2)$$



(a)Normalized Character (b)Smoothed Character

Fig. 8: Smoothed Character

3.3. Removal of repetition points

Sometimes input data contains duplicate points and does not contain any useful information for classification. If P_i and P_j are two consecutive points, then these points will be preserved if the following equation is satisfied (Eq. 3):

$$x^2 + y^2 > d^2, \quad (3)$$

where $x = x_i - y_i$ and $y = y_i - y_j$.

We set d equal to zero; The Eq. 3 removes all consecutive repeated points.

3.4. Interpolation

Interpolation is the prerequisite for applying Re-sampling. Interpolation generates missing points; usually with the constraint that distance cannot be more than a certain threshold (Sharma et al., 2009). In this paper the missing points between P_i and P_{i+1} is calculated using the algorithm below. The result is illustrated in Fig. 9.

Algorithm:

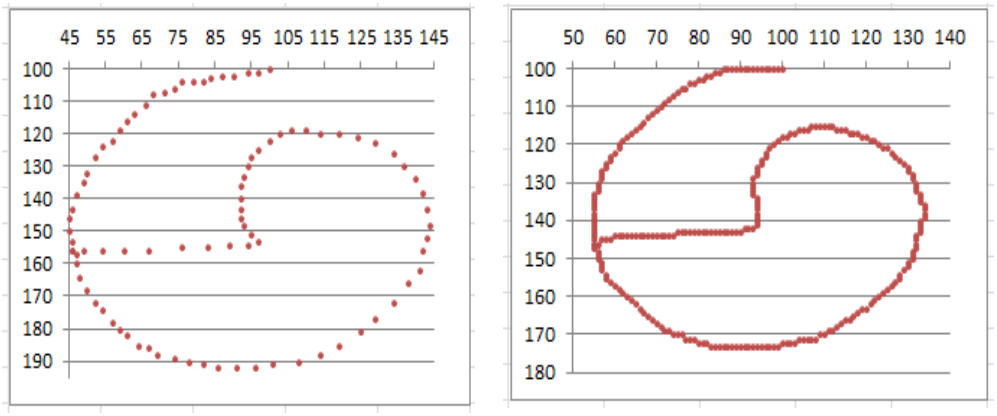
1. Initial the coordinates of two points $A(x_1, y_1)$ and $B(x_2, y_2)$ between which to calculate missing points.
2. [Calculate d_x and d_y]

$d_x = (x_2 - x_1)$ *and* $d_y = (y_2 - y_1)$

3. [Calculate the length L]
If $abs(x_2 - x_1) \geq abs(y_2 - y_1)$ *then* $L = abs(x_2 - x_1)$
Else $L = abs(y_2 - y_1)$
4. [Calculate the increment factor]

$\Delta x = (x_2 - x_1)/L$ *and* $\Delta y = (y_2 - y_1)/L$

This step makes either Δx or Δy equal to 1, because L is either $|x_2 - x_1|$ or $|y_2 - y_1|$. Therefore a step increment in x or y direction is equal to 1.
5. [Obtain the new pixel between the points]
Intialize I to 1
while ($i \leq 1$)
{
$x_{new} = x_{new} + \Delta x$
$y_{new} = y_{new} + \Delta y$
New pixel is (Integer (x_{new})*, Integer* (y_{new})*)*
i=i+1
}



(a) Smoothed Character (b) Interpolated Character

Fig. 9: Interpolated character

3.5. Sharp point detection and dehooking

The peak point of the wavelet where the writing direction has changed is known as sharp point. The algorithm (Bawa and Rani, 2011) is used for identifying sharp points based on angle change of pen direction. Initially, for a given stroke S, the slope angle of any two successive points is calculated.

Then the changed angles calculated based on these slopes. In these changed angles a subset of consecutive increasing and the subset of consecutive decreasing changed angles can be identified, which represents a sharp point. Along with theses, begin and end points of a stroke are sharp points.

The following algorithm detects sharp points of a given stroke S. All sharp points are stored in SSP. In this algorithm $\theta_{i,i+1}$ is the angle between two lines: one is (x_i, y_i) and (x_{i+1}, y_{i+1}) and the other is (x_{i+1}, y_{i+1}) and (x_{i+2}, y_{i+2}), and $\Delta\theta$ is a variable which determines whether an angle is turn angle or not. If Δθ is positive, then the pen direction has not changed.

Algorithm:

1. *add the first point to SSP.*
2. *Calculate the slope angles of the lines, each of which is defined by two consecutive points in strokes ,as* $\theta = \{\theta_{1,2}, \ldots\ldots\ldots\ldots \theta_{N-1,N}\}$*, where N is the number of points in stroke.*
3. *i=1, where I is the index ogf points.*
4. *while* $i < N$ *do*
5. $d\theta_{i,i+1} = \theta_{i,i+1} - \theta_{i+1,i+2}$
6. *if* $d\theta_{i,i+1} = 0$ *then*
7. *i++*
8. *Goto step 4*
9. *end if*
10. $\Delta\theta = d\theta_{i,i+1} * d\theta_{i-1,i}$
11. *If* ($\Delta\theta \leq 0$) *and* $d\theta_{i-1,i} \neq 0$) *then*
12. *The pen direction changed at point* p_{i+1} *, add this point to SSP.*
13. *end if*
14. *i++*

15. end while
16. add the last point to SSP.

After the detection of sharp points, the hooks of a stroke can be found, if there exists more than two sharp points.

Assume segment s_b is between the first two sharp points, and segment s_e is between the end two sharp points, and there slope angles be α_b and α_e respectively.

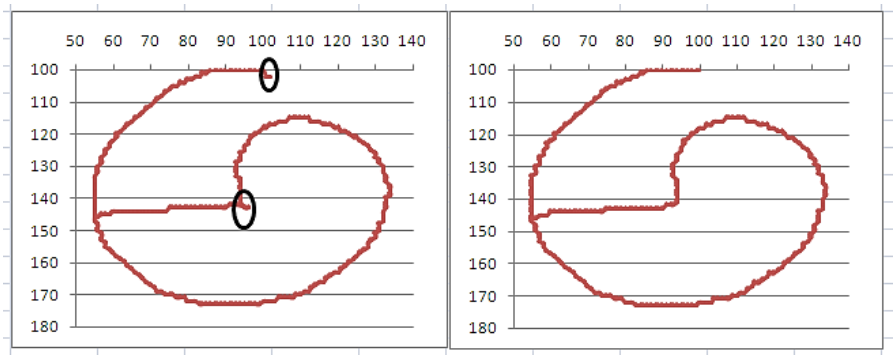
The length of the segment L_{seg} is either s_b or s_e. In addition, define two segments s_{b+1} is between the first second and third sharp points, s_{e-1} is between the last second and third sharp points. Let the angle between segments s_e and s_{e-1} be λ. Thus, if the length L_{seg} and angle λ match the following conditions, the segment is a hook:

$\lambda \leq 90$ &&$L_{seg} \leq 3\%$ of diagonal line

The diagonal lineL_{dia} of a stroke S is computed based on following formula (Eq. 4):

$$L_{dia} = \sqrt{(x_{max} - x_{min})^2 + (y_{max} - y_{min})^2} \quad (4)$$

The black circles in Fig. 10a Indicates hooks, hooks removed using above algorithm is shown in Fig. 10b.



(a) Character with Hooks (b) Character after removing Hooks

Fig. 10: Dehooked character

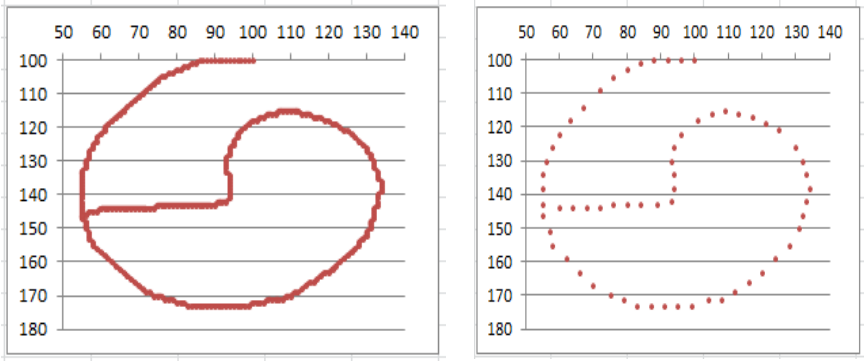
3.6. Resampling

Re-Sampling is performed to normalize input character to a constant number of points which are at equal distances. In this paper each character is resampled to 64 intervals. The total length of the character is computed by adding the Euclidean distances between successive points. This is divided by the number of intervals required after re-sampling. The original points are replaced with a new set at this constant spacing using piece-wise linear interpolation. When a character has multiple strokes, each stroke is resampled separately such that the total number of points using the technique below (Sharma et al., 2009). All training characters having the same number of strokes are considered as a set. The number of points in each stroke is made proportional to the average length of strokes obtained from the corresponding set. The k^{th} point from the series of N points is selected according to the following Eq. 5:

$$K = i * (\frac{N}{64}) \quad (5)$$

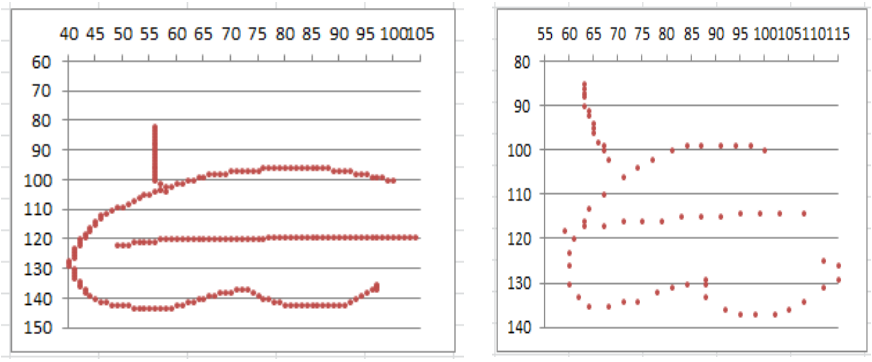
The value of i is successively taken as i =0, 1, 2, …, 64, and right-hand side of Eq. 5 is rounded to the nearest integer value to get all the 64 successive selected points. The resampled character is shown in the Fig. 11.

The three strokes of character are resampled in the ratio of 37:13:12 intervals. The resampled character "Id_4" with three strokes is shown in the Fig. 12.



(a) Interpolated Character (b) Resampled Character

Fig. 11: Resampled character with single stroke



(a) Interpolated Character (b)Resampled Character

Fig. 12: Resampled character with multiple strokes

4. Feature extraction and character recognition

We experiment on two different features:

- Pen-tip position: The preprocessed *(x, y)* coordinates are taken as pen-tip positions. P={p_i} where i=0,1,…..,64 and p_i =(x_i, y_i)
- Tangent Angle: This feature gives direction at pen-tip position. The tangent angle is calculate between two consecutive points of P as

$$\theta_i = \tan^{-1}(y_{i+1} - y_i / x_{i+1} - x_i)$$

where i=0, 1,..., 64.
The tangent angle at $\theta_{64} = \theta_{63}$

K-Nearest Neighbor classifier using DTW as a distance measure is used for character recognition. This is especially useful for comparing two patterns of equal or unequal length. To compare two sequences P and Q of length m and n respectively, using DTW, we first construct an n-by-m matrix where the (i^{th}, j^{th}) element of matrix represents the Euclidian distance between the points pi and qi. The best match between two strokes is obtained by considering all possible alignments between events and finding the alignment for which the distance is minimum using dynamic programming. The distance between two strokes A and B is calculated as follows:

$$D(i, j)=d(i, j)+\min(D(i-1, j-1)+D(i-1, j)+D(i, j-1)), \quad (6)$$

where d(i, j) is the local distance at (i, j) and D(i, j) is the global distance up to (i, j).

$$DTW(A, B) = D(N_A, N_B),$$

where N_A is the length of stroke A and N_B is the length of stroke B. In our process the class of test character is determined to be the class of its nearest neighbor in the train database.

5. Experimental results and discussion

The evaluation of preprocessing methods is carried out by a classifier. The parameters evaluated in our approach are recognition time, recognition accuracy, False Acceptance Rate (FAR) and False Rejection Rate (FRR). The average recognition time is calculated by dividing the total number of recognized characters by the total time taken to recognize all test samples. Our system's sensitivity parameters FAR is calculated as the ratio of the number of false acceptances divided by the number of all imposter attempts and FRR is calculated as the ratio of the number of false rejections divided by the number of genuine attempts. All the experiments are conducted on an Intel i5 processor 3.10 GHz and 4GB RAM. Here the experiments are carried out as follows: first the characters are recognized without preprocessing. Then, the same character is recognized with preprocessing. Therefore, the recognition process of the characters without preprocessing is to first extract the features of the characters, and then to classify the characters. In contrast, for the recognition of characters with preprocessing, the preprocessing for online handwritten data was carried out before the feature extraction and classification. In both approaches the feature pen-tip position and preprocessed pen-tip position are used, respectively. Fig. 13 depicts the recognition accuracy of unprocessed and preprocessed data with varying training set size. It shows recognition accuracy increases by 10% with preprocessing. Fig. 14 illustrates reduction in recognition time of preprocessing data of varying training set size. The plots of the FAR and FRR are shown in Fig. 15. We can see that the preprocessed data performance is better than that of unprocessed data. Fig. 13 also shows results of experiments conducted over the same dataset by including additional feature tangent angle. This additional feature increases the recognition accuracy of preprocessed data by 3%.

6. Conclusion

We have demonstrated preprocessing techniques over online data of HP data set Telugu strokes. The preprocessing techniques used are Normalization, Smoothing, Duplicate Point Removal, Interpolation, Dehooking and Resampling.

This is also first step towards Online Handwritten Telugu Character Recognition using HP data set. There is lot of scope for future enhancement towards the implementation of more preprocessing techniques. The experiment shows that this

proposed preprocessing approach is efficient for online handwritten data. In future the comparison between the proposed approach and other existing approaches will be focused.

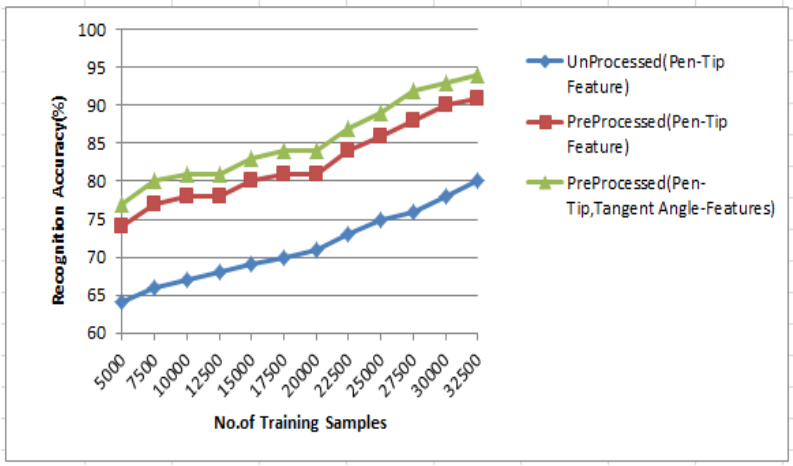


Fig. 13: Recognition Accuracy vs. number of traing samples for the unprocessed and processed data

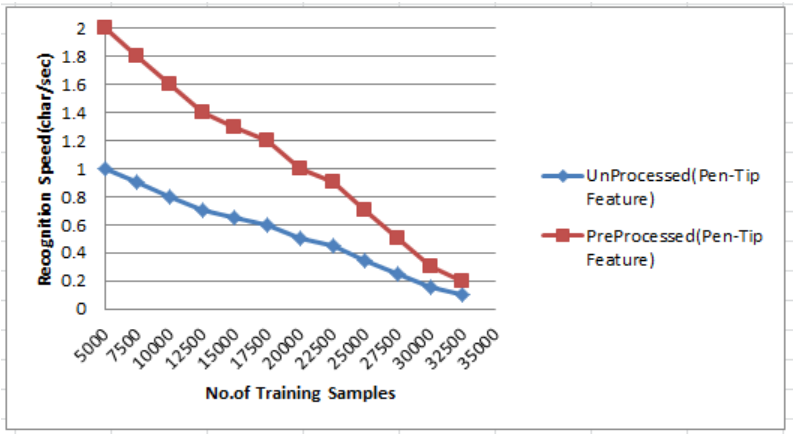


Fig. 14: Recognition Speed vs. number of traing samples for the unprocessed and processed data

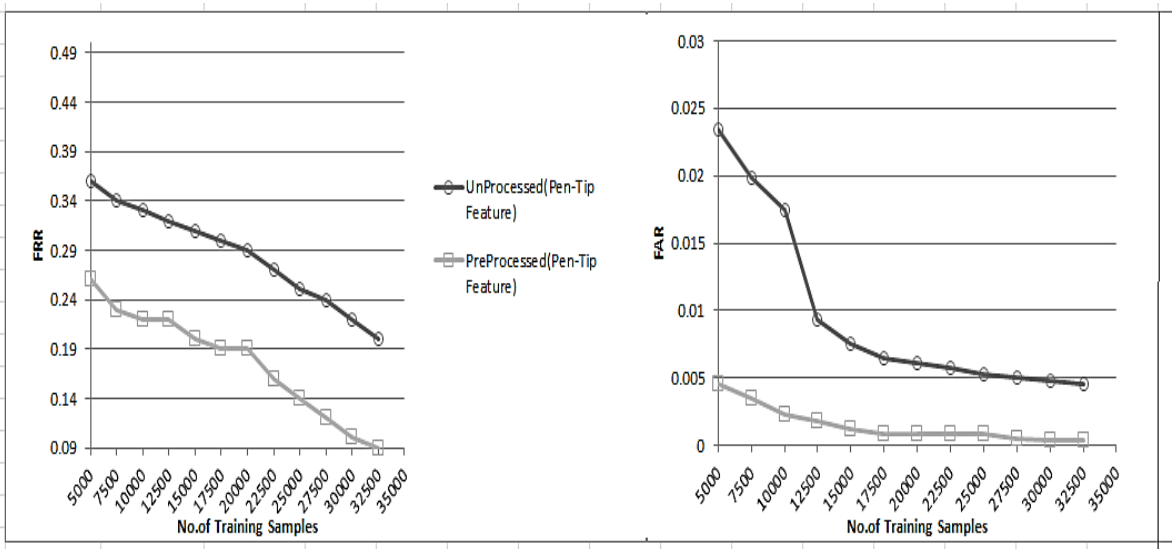


Fig. 15: FRR, FAR vs. number of traing samples for the unprocessed and processed data

References

Agrawal M, Bali K, Madhvanath S, and Vuurpijl L (2005). UPX: A new XML representation for annotated datasets of online handwriting data. In the 8th International Conference on Document Analysis and Recognition, IEEE, Seoul, South Korea: 1161-1165. https://doi.org/10.1109/ICDAR.2005.248

Babu VJ, Prasanth L, Sharma RR, and Bharath A (2007). HMM-based online handwriting recognition system for Telugu symbols. In the 9th International Conference on Document Analysis and Recognition, IEEE, Parana, Brazil, 1: 63-67. https://doi.org/10.1109/ICDAR.2007.4378676

Bawa RK and Rani R (2011). A preprocessing technique for recognition of online handwritten Gurmukhi numerals. In: Nandi AMS and Kumar GKS (Eds.), High Performance Architecture and Grid Computing: 275-281. Springer Berlin Heidelberg, Berlin, Germany.

Bellegarda EJ, Bellegarda JR, Nahamoo D, and Nathan KS (1993). A probabilistic framework for on-line handwriting recognition. In the 3rd International Workshop on Frontiers in Handwriting Recognition, Buffalo, New York: 225-234.

Bharath A and Madhvanath S (2009). Online handwriting recognition for Indic scripts. In: Setlur S (Ed.), Guide to OCR for Indic Scripts: 209-234. Springer, London, UK.

Chan KF and Yeung DY (1998). Elastic structural matching for online handwritten alphanumeric character recognition. In the 14th International Conference on Pattern Recognition, IEEE, Brisbane, Queensland, Australia, 2: 1508-1511. https://doi.org/10.1109/ICPR.1998.711993

Chee YM, Froumentin M, and Watt SM (2006). Ink markup language (InkML). W3C Working Draft, World Wide Web Consortium. Available online at: http://www.w3.org/TR/2006/WD-InkML-20061023.

Cho SJ and Kim JH (2004). Bayesian network modeling of strokes and their relationships for on-line handwriting recognition. Pattern Recognition, 37(2): 253-264.

Do TMT and Artières T (2006). Conditional random fields for online handwriting recognition. In the 10th International Workshop on Frontiers in Handwriting Recognition. Suvisoft, La Baule, France.

Graves A, Liwicki M, Bunke H, Schmidhuber J, and Fernández S (2008). Unconstrained on-line handwriting recognition with recurrent neural networks. In: Platt JC, Koller D, Singer Y, and Roweis ST (Eds.), Advances in Neural Information Processing Systems: 577-584. Curran Associates, New jersey, UK.

Guerfali W and Plamondon R (1993). Normalizing and restoring on-line handwriting. Pattern Recognition, 26(3): 419-431.

Guyon I, Schomaker L, Plamondon R, Liberman M, and Janet S (1994). UNIPEN project of on-line data exchange and recognizer benchmarks. In the 12th IAPR International. Conference on Pattern Recognition, Computer Vision and Image Processing, IEEE, Jerusalem, Israel, 2: 29-33. https://doi.org/10.1109/ICPR.1994.576870

Hollerbach JM (1981). An oscillation theory of handwriting. Biological Cybernetics, 39(2): 139-156.

Huang B, Zhang Y, and Kechadi M (2009). Preprocessing techniques for online handwriting recognition. In: Nedjah N, Mourelle LDM, Kacprzyk J, França FM, and De Souza AF (Eds.), Intelligent Text Categorization and Clustering: 25-45. Springer, Berlin, Germany.

Inuganti S and Rao RR (2015). Preprocessing of HP data set Telugu strokes in online handwritten Telugu character recognition. International Journal of Computer Technology and Applications (IJCTA), 8(5): 1939-1945

Kerrick DD and Bovik AC (1988). Microprocessor-based recognition of handprinted characters from a tablet input. Pattern Recognition, 21(5): 525-537.

Lei LI, Zhang LL, and Su JF (2012). Handwritten character recognition via direction string and nearest neighbor matching. The Journal of China Universities of Posts and Telecommunications, 19: 160-165.

Li X and Yeung DY (1997). On-line handwritten alphanumeric character recognition using dominant points in strokes. Pattern Recognition, 30(1): 31-44.

Madhvanath S, Vijayasenan D, and Kadiresan TM (2007). LIPITK: A generic toolkit for online handwriting recognition. In the ACM SIGGRAPH 2007 Courses, ACM, San Diego, USA. https://doi.org/10.1145/1281500.1281524

Marukatat S, Artieres T, Gallinari R, and Dorizzi B (2001). Sentence recognition through hybrid Neuro-Markovian modeling. In the 6th International Conference on Document Analysis and Recognition, IEEE, Seattle, WA, USA: 731-735. https://doi.org/10.1109/ICDAR.2001.953886

Plamondon R and Maarse FJ (1989). An evaluation of motor models of handwriting. IEEE Transactions on Systems, Man, and Cybernetics, 19(5): 1060-1072.

Plamondon R and Srihari SN (2000). Online and off-line handwriting recognition: A comprehensive survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(1): 63-84.

Poisson E, Gaudin CV, and Lallican PM (2002). Multi-modular architecture based on convolutional neural networks for online handwritten character recognition. In the 9th International Conference on Neural Information Processing, IEEE, Singapore, Singapore, 5: 2444-2448. https://doi.org/10.1109/ICONIP.2002.1201933

Rampalli R and Ramakrishnan AG (2011). Fusion of complementary online and offline strategies for recognition of handwritten Kannada characters. Journal of Universal Computer Science, 17(1): 81-93.

Reddy GS, Sharma P, Prasanna SRM, Mahanta C, and Sharma LN (2012). Combined online and offline assamese handwritten numeral recognizer. In the National Conference on Communications, IEEE, Kharagpur, India: 1-5. https://doi.org/10.1109/NCC.2012.6176859

Sharma A (2009). Online handwritten Gurmukhi character recognition. Ph.D. Dissertation, Thapar University, Patiala, India.

Sharma A, Sharma RK, and Kumar R (2009). Online preprocessing of handwritten Gurmukhi strokes. Machine Graphics and Vision International Journal, 18(1): 105-120.

Shashikiran K, Prasad KS, Kunwar R, and Ramakrishnan AG (2010). Comparison of HMM and SDTW for Tamil handwritten character recognition. In the International Conference on Signal Processing and Communications, IEEE, Bangalore, India: 1-4. https://doi.org/10.1109/SPCOM.2010.5560498

Sundaram S and Ramakrishnan AG (2008). Two dimensional principal component analysis for online character recognition. In the 11th International Conference on Frontiers in Handwriting Recognition, Concordia University, Montreal, Canada: 88-94. Available online at: mile.ee.iisc.ac.in/mile/publications/softCopy/DocumentAnalysis/Suresh_ICFHR2008.pdf

Swethalakshmi H, Jayaraman A, Chakravarthy VS, and Sekhar CC (2006). Online handwritten character recognition of Devanagari and Telugu Characters using support vector machines. In the 10th International workshop on Frontiers in Handwriting Recognition. Centre de Congres Atlantia, La Baule, France.

Uchida S and Sakoe H (2005). A survey of elastic matching techniques for handwritten character recognition. IEICE Transactions on Information and Systems, 88(8): 1781-1790.

Zeng W, Meng X, Yang C, and Huang L (2006). Feature extraction for online handwritten characters using Delaunay triangulation. Computers and Graphics, 30(5): 779-786.