OFFLINE SIGNATURE VERIFICATION SCHEME USING FEATURE EXTRACTION METHOD

A THESIS SUBMITTED IN PARTIAL FULFILMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology

In

Computer Science and Engineering

By

MAHENDRA SINGH CHAUHAN

Roll No: 10306032

DEEPJYOTI NATH

Roll No: 10306007

Department of Computer Science and Engineering

National Institute of Technology, Rourkela

May, 2007

OFFLINE SIGNATURE VERIFICATION SCHEME USING FEATURE EXTRACTION METHOD

A THESIS SUBMITTED IN PARTIAL FULFILMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology

In

Computer Science and Engineering

By

MAHENDRA SINGH CHAUHAN

Roll No: 10306032

DEEPJYOTI NATH

Roll No: 10306007

Under the guidance of

Prof. B.MAJHI

Department of Computer Science and Engineering

National Institute of Technology

Rourkela

May, 2007

National Institute of Technology
Rourkela

CERTIFICATE

This is to certify that the thesis entitled, "**OFFLINE SIGNATURE VERIFICATION SCHEME USING FEATURE EXTRACTION METHOD**" submitted by **Mahendra Singh Chauhan** , **Roll No. 10306032** and **Deepjyoti Nath, Roll No. 10306007** in partial fulfillment of the requirements for the award of **Bachelor of Technology** Degree in **Computer Science and Engineering** at the National Institute of Technology, Rourkela (Deemed University) is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other university / institute for the award of any Degree or Diploma.

Date:

Prof. B. MAJHI
Dept. of Computer Science and Engineering
National Institute of Technology, Rourkela
Rourkela - 769008

ACKNOWLEDGEMENT

We are thankful to **Dr. B.Majhi,** Professor in the department of Computer Science and Engineering, NIT Rourkela for giving us the opportunity to work under him and extending every support at each stage of this project work.

We would also like to convey our sincerest gratitude and indebt ness to all other faculty members and staff of Department of Computer Science and Engineering, NIT Rourkela, who bestowed their great effort and guidance at appropriate times without which it would have been very difficult on our part to finish the project work.

Submitted By,

Deepjyoti Nath

Roll No. 10306007

Computer Sc. Engg. Deptt.

NIT Rourkela

Mahendra Singh Chauhan

Roll No. 10306032

Computer Sc. Engg. Deptt.

NIT Rourkela

Date: May 10, 2007

CONTENTS

ABSTRACT

In this project a new improved offline signature verification scheme has been proposed. The scheme is based on selecting 60 feature points from the geometric centre of the signature and compares them with the already trained feature points. The classification of the feature points utilizes statistical parameters like mean and variance. The suggested scheme discriminates between two types of originals and forged signatures. The method takes care of skill, simple and random forgeries. The objective of the work is to reduce the two vital parameters False Acceptance Rate (FAR) and False Rejection Rate (FRR) normally used in any signature verification scheme. Comparative analysis has been made with standard existing schemes.

The Algorithms are based on the Geometric Center of an image so images are splitted into different parts to get the geometric centers of each which are called as Feature points in our thesis. We have taken 60(30+30) Feature points for calculation purpose(in extended Algorithm). As Feature points increases results will be more accurate but complexity and time require for testing will be more. So we have taken 60 feature points which improves security and maintains same complexity level. All calculations are done on the basis of these feature points. Results are expressed in terms of FAR (False Acceptance Rate) and FRR (False Rejection Rate) and subsequently compare these results with other existing Techniques. Results obtained by this algorithm are quite impressive. Random and Simple forgeries are eliminated and skilled forgeries are also eliminated in greater extent. As signature image is tested rigorously so FRR is more in the Algorithm proposed by us.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

(i) Introduction
(ii) Types of Forgeries

1.1 INTRODUCTION

Signature verification is an important research area in the field of person authentication. The recognition of human handwriting is important concerning about the improvement of the interface between human-beings and computers. If the computer is intelligent enough to understand human handwriting it will provide a more attractive and economic man-computer interface. In this area signature is a special case that provides secure means for authentication, attestation authorization in many high security environment. The objective of the signature verification system is to discriminate between two classes: the original and the forgery, which are related to intra and interpersonal variability. The variation among signatures of same person is called Intra Personal Variation. The variation between originals and forgeries is called Inter Personal Variation.

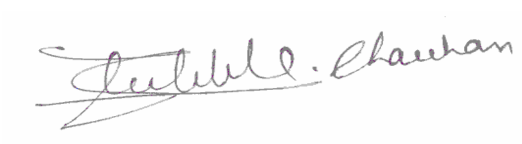
Signature verification is so different with the character recognition, because signature is often unreadable, and it seems it is just an image with some particular curves that represent the writing style of the person. Signature is just a special case of handwriting and often is just a symbol. So it is wisdom and necessary to just deal with a signature as a complete image with special distribution of pixels and representing a particular writing style and not as a collection of letters and words.

A signature verification system and the techniques using to solve this problem can be divided into two classes: online and off-line. In an online system, a signature data can be obtained from an electronic tablet and in this case, dynamic information about writing activity such as speed of writing, pressure applied, and number of strokes is available. In off-line systems, signatures written on paper as has been done traditionally are converted to electronic form with the help of a camera or a scanner and obviously, the dynamic information is not available. In general, the dynamic information represents the main writing style of a person. Since the volume of information available is less, the signature verification using off-line techniques is relatively more difficult. Our work is concerned with the techniques of off-line signature verification. The static information derived in an off-line signature verification system may be global, structural, geometric or statistical.

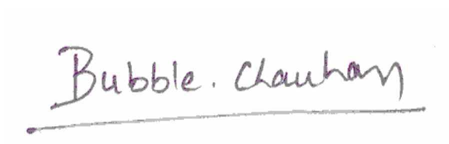
In this paper we concern with offline signature verification which is based on geometric centre and is useful in separating skilled forgeries from the originals. The algorithms used have given improved results as compared to the previously proposed algorithms based on the geometric centre. Our algorithms eliminate the random and simple forgeries as well as most of the skilled forgeries

1.2 Types of Forgeries:

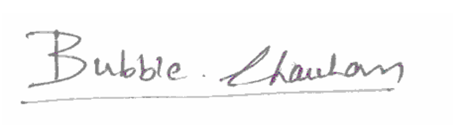
There are three different types of forgeries to take into account. First one is ***random forgery*** which is written by the person who doesn't know the shape of original signature. The second, called ***simple forgery***, which is represented by a signature sample, written by the person who knows the shape of original signature without much practice. The last type is ***skilled forgery***, represented by a suitable imitation of the genuine signature model. Each type of forgery requires different types of verification approach. Hybrid systems have also been developed. Fig. 1 shows the different types of forgeries and how much they vary from original signature.



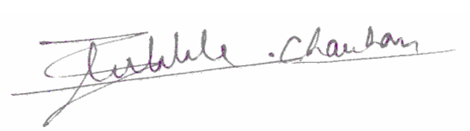
(A)



(B)



(C)



(D)

Fig1.1: Original signature(A), Random forgery(B), Simple forgery(C), Skilled forgery(D)

CHAPTER 2

EXISTING TECHNIQUES

i. Fixed-point Arithmetic method
ii. 12 feature point method

2.1 Fixed-point Arithmetic method:

This technique is based on geometrical features which are based on two vectors which represent the envelope description and the interior stroke distribution in polar and Cartesian coordinates**.**

2.1.1 Outline Detection and Representation:

The outline is calculated by means of morphological operations as is shown in Fig.2.1: First, we apply a dilatation in order to reduce the signature variability and, afterward, a filling operation is applied to simplify the outline extraction process. When several

objects are detected after filling, a horizontal dilatation is performed until all the objects are connected. The outline is represented as a sequence of its Cartesian coordinates $(Xt,Yt)_{t=1}^{T}$ being its length. This sequence follows the counterclockwise and starts in the point $(X1,Y1) = C_x$ max(Y_t !X_t=C_x), (C_x, C_y) being the geometric center of the outline (see Fig 2.2)

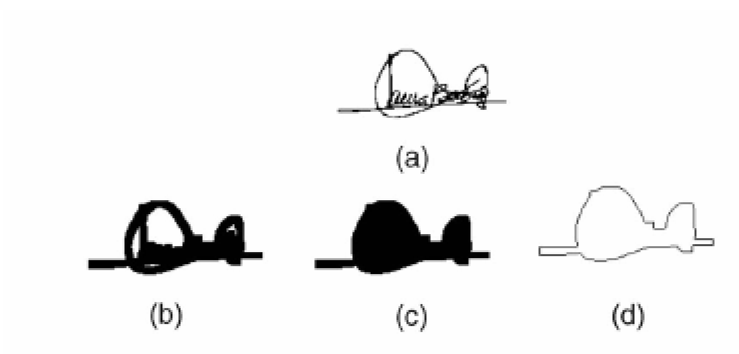


Fig2.1: (a) Original (b) Dilated, (c) Filled, (d) Outline of the signature

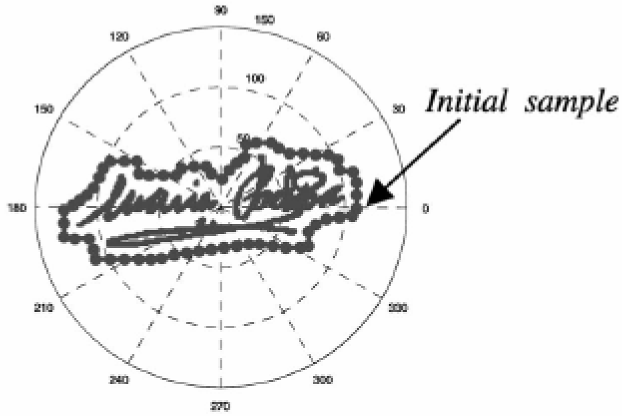


Fig2.2: Original signature, its outline, and its selected samples

2.1.2 Feature Vector Based on Polar Coordinates:

To represent the signature outline in the polar coordinates, it is decimate selecting Tr equidistant samples of the envelope $(X_{tp},Y_{tp})_{t=1}^{Tr}$ (being p=*fix*(T/T_r) and *fix* rounds to the nearest integers toward zero) and represent each sample as a three components feature vector, which are: the derivation of the radius, is angle and the number of black pixel that the radiuses cross when sweeping from one selected point to the next. The latter components have been obtained with an algorithm designed for fixed-point microprocessor.

The radius function r_t, $t=1,2,3,\ldots,T_r$ is calculated as the number of pixels from the geometric center(C_x,C_y) to each outline selected point(X_{tp},Y_{tp}) as:

$$d1 = X_{tp}-C_x, \quad d2 = Y_{tp}-C_y$$

$$r_t = \max(d1,d2) + \min(d1,d2)/4 \qquad (2.1)$$

The radius sequence is normalized to r1 = 127 and the first component of the feature vector, the derivative of the radius, is obtained as $\Delta r_t = r_{t+1}-r_t$, $t = 1,2,\ldots,T_r$. The radius is not used because the probability density functions of the radius of different signatures are very similar. The density of the radius derivate is more discriminative. Here it is verified that a $T_r = 64$ value is a good trade-off between the recognition ratio and computational requirements. The second component of the feature vector is the angle of each selected contour sample, which is calculated by means of the arctan function implemented through lookup table

$$\theta t = \arctan(X_{nT/Tr}/Y_{nT/Tr})\ ,\ t = 1,2,\ldots,Tr \qquad (2.2)$$

The third component contains the number of black pixels of the signature strokes that the radius crosses when sweeping from θt to θt_{+1}normalized to maximum value equal to 1 in order to increase the stroke thickness independence. This new component, denoted as A_t, t = 1; 2; . . . ; T_r, completes the information about the outline covered by the first and second components with a rough idea about the distribution of the signatures' strokes inside the outline. Therefore, the sequence of feature vectors based on polar coordinates that characterize the signature is $p_t = [\Delta r_t, \theta t, A_t]$, t = 1; 2; . . . ; T_r. Experimentally, $T_r = 64$. An example can be seen in Fig. 2.3.

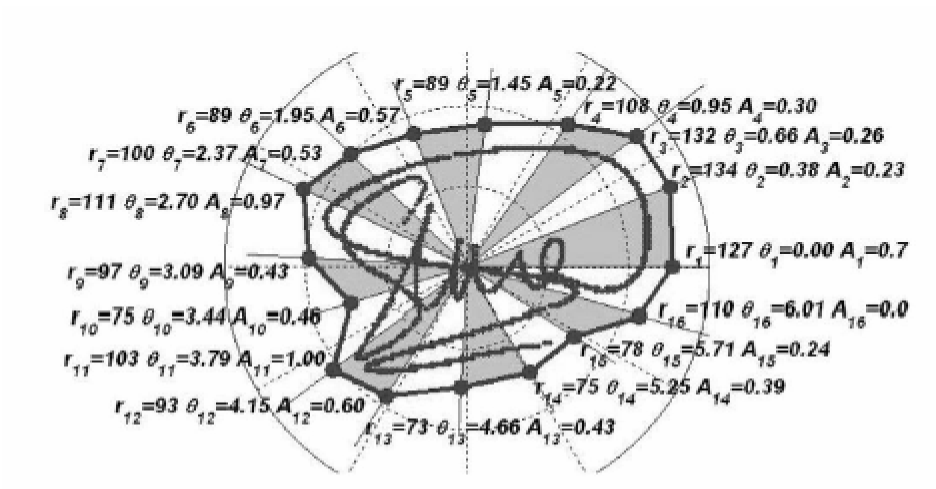


Fig 2.3: The signature and its envelop with the values r_t, θt, A_t of associated to selected sample on the envelop with $T_r = 16$

2.1.3 Feature Vector Based on Cartesian Coordinates:

The second feature vector is also based on the envelope and the signature strokes density parameterization, but, in this case, in Cartesian coordinates. The envelope is divided through the geometric center into top and bottom halves. The measurements are the height of the top half at T_h equidistant points, obtaining the sequence uh_t, t = 1; 2; . . . ; T_h. The bottom half height is also measured at the same T_h points, getting the sequence lh_t, t = 1; 2; . . . ; T_h.

Next, the envelope is divided in two halves once again, this time the left and right-hand sides through the geometric center. T_w equidistant measures of the width of the right and left-hand sides are taken, obtaining the sequences rw_t, $t = 1; 2; \ldots; T_w$, and lwt, $t = 1; 2; \ldots; T_w$.

The above sequences are combined, obtaining $T_h + T_w$ long sequences as follows:

$$ur_t = \begin{cases} uh_t & t = 1, \ldots, T_h \\ rw_{t-T_h} & t = T_h + 1, \ldots, T_h + T_w \end{cases}$$

and

$$ll_t = \begin{cases} lh_t & t = 1, \ldots, T_h \\ lw_{t-T_h} & t = T_h + 1, \ldots, T_h + T_w. \end{cases}$$

Both sequences characterize the signature envelope shape. Here $T_h = 42$ and $T_w = 22$ are used in the experiments, looking for a trade-off between recognition ratio and computational load.

Once both sequences have been obtained, the feature vector sequence is composed of four dimensional vectors. The first component of the feature vector is the sequence ur_t, $t = 1; 2; \ldots; T_h + T_l$, the second component is the values of ll_t, $t = 1; 2; \ldots; T_w + T_l$. The third feature vector component is the value of the index t in order to help the HMM synchronism. The fourth component is the sequence tr_t, $t = 1; 2; \ldots; T_h + T_l$, defined as

$$tr_t = \begin{cases} th_t & t - 1, \ldots, T_h \\ tw_{t-T_h} & t = T_h + 1, \ldots, T_h + T_w, \end{cases}$$

where th_t, $t = 1; 2; \ldots; T_h$, contains the number of transitions (black to white or white to black) originated by the signature strokes when we go from the bottom side to the top side of the signature envelope following the vertical lines used to work out uht or lh_t Similarly, tw_t, $t = 1; 2. \ldots; T_w$, contains the number of transitions when we go from the right-hand side to the left-hand side of the signature envelope following the horizontal lines used to work out rw_t or lw_t. These sequences are gathered in the sequence of vectors $c_t = [ur_t; ll_t; t; tr_t]$, $t = 1; 2; \ldots; T_h + T_l$. Here $T_h = 42$ and $T_w = 22$ sequence is used. These sequences are illustrated in Fig. 2.4.

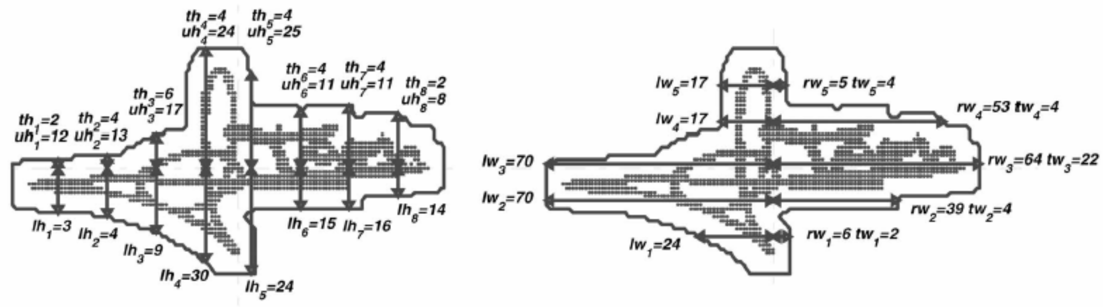


Fig 2.4: The signature, its envelope and sequences th_t, uh_t, lh_t, tw_t, rw_t, lw_t with T_h =8 and T_w = 5

2.2 12 feature point method:

In this method feature points are extracted from the signature image based on the geometric centre of the image.

Geometric Center: It is a point about which image is perfectly symmetric number of pixel wise. So we can say that it is the centre point of an image. It is a vital parameter used in the feature extraction. If a vertical line drawn from the feature point then the number of pixel at the right side must be equal to the left side and similarly a line drawn horizontally then number of pixel at top side must be equal to the bottom side.

2.2.1 Feature points based on vertical splitting

Six feature points are retrieved based on vertical splitting. Here feature points are nothing but geometric centers of different splitted images. The procedure for finding feature points by vertical splitting is mentioned in Algorithm.

Algorithm

This is the procedure for generating feature points based on vertical splitting.

Input: Static signature image after moving the signature to center of image

Output: *v*1;*v*2;*v*3;*v*4;*v*5;*v*6 (feature points)

Steps:

(a) Split image with vertical line at the center of image then we will get left and right parts of image.

(b) Find geometric centers $v1$ and $v2$ for left and right parts correspondingly.

(c) Split left part horizontal line at $v1$ and find out geometric centers $v3$ and $v4$ for top and bottom parts of left part correspondingly.

(d) Split right part horizontal line at $v2$ and find out geometric centers $v5$ and $v6$ for top and bottom parts of left part correspondingly.

Fig.2.5 shows the feature points retrieved from signature image and O is the center of image. These features we have to calculate for every signature image in both training and testing.

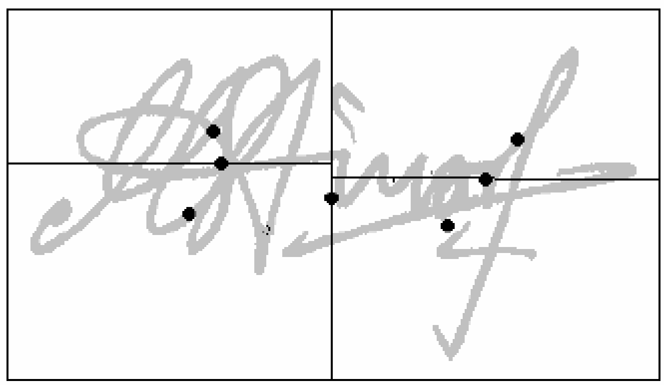


Fig 2.5: Feature points extraction based on vertical splitting

2.2.2 Feature points based on horizontal splitting

Six feature points are retrieving based on horizontal splitting. Here feature points are nothing but geometric centers. The procedure for finding feature points by horizontal splitting is mentioned in Algorithm.

Algorithm

This is the procedure for generating feature points based on horizontal splitting.

Input: Static signature image after moving the signature to center of image

Output: $h1;h2;h3;h4;h5;h6$ (feature points)

Steps:

(a) Split image with horizontal line at the center of image then we will get top and bottom parts of image.

(b) Find geometric centers $h1$ and $h2$ for top and bottom parts correspondingly.

(c) Split top part with vertical line at $h1$ and find out geometric centers $h3$ and $h4$ for left and right parts of top part correspondingly.

(d) Split bottom part with vertical line at $h2$ and find out geometric centers $v5$ and $h6$ for left and right parts of left part correspondingly.

Fig.2.6 shows the feature points retrieved from signature image and O is the center of image. These features we have to calculate for every signature image in both training and testing. Now total twelve feature points ($v1;...;v6$ and $h1;...;h6$) are calculated by vertical and horizontal splitting.

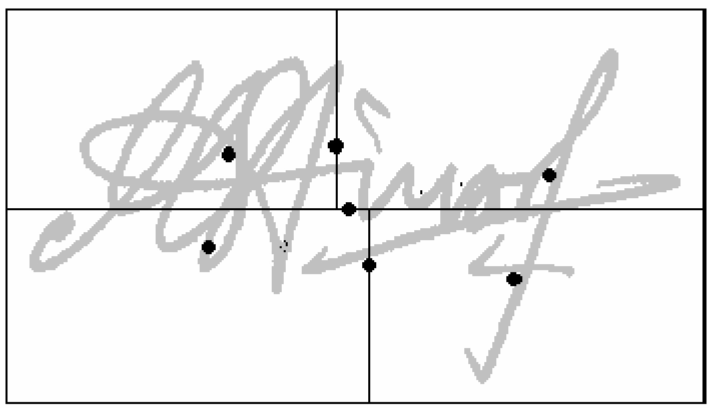


Fig 2.6: Feature points extraction based on horizontal splitting

CHAPTER 3

PROPOSED TECHNIQUE

i. Overview
ii. Processing of the Signature
iii. Classification
iv. Threshold
v. Experiment and Result

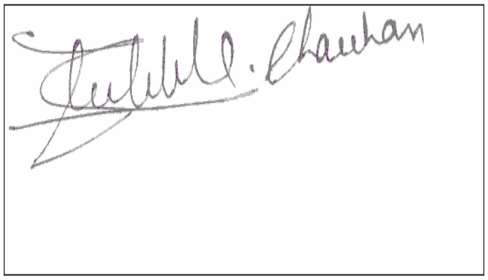
3.1 Overview

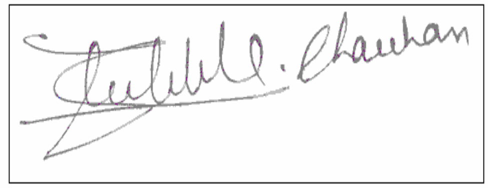
We have extended the previously explained technique which is based on feature extractions using geometric centers of the signature image. We obtained sixty feature points, thirty from vertical splitting and thirty by horizontal splitting to compute the threshold values. In this chapter we discuss about proposed method for offline signature verification. In section 3.2 we discuss about the preprocessing of the signature image to make geometric center to the center of image. In which feature extraction we use two type of splitting (Vertical and Horizontal) based on geometric center. Classifier used here is the Euclidean distance model which is suitable for the features proposed. Threshold is based on the statistical parameters like median, variance and standard deviation.

3.2 Processing of the Signature:

The geometric features proposed by this paper are based on two sets of points in two-dimensional plane. Each set having thirty feature points which represent the stroke distribution of signature pixels in image. These sixty feature points are calculated by Geometric Center. Vertical Splitting and Horizontal Splitting are two main steps to retrieve these feature points. Before finding feature points we have to do some adjustments to the signature image. The processing of the signature is discussed below:

3.2.1 Moving signature into the centre of image: The signature is moved to the centre by taking the signature image into a fixed calculated frame and the unnecessary white spaces are removed without affecting the signature image such that the image is in the middle of the frame. For this first we divide the whole frame of the signature into 10*10 square row-wise and column-wise and find the variance(signature is considered to be binary and consists of only black and white pixels).If a square block has a zero variance we remove that square, otherwise restore. Thus squares of unnecessary white spaces are removed and then the image is restored in the fixed frame.





(A) **(B)**

Fig 3.1: Captured signature before adjustment (A) and after adjustment (B)

3.2.2 Feature Extraction: The geometric features are based on two sets of points in 2-dimensional plane. The vertical splitting of the image results thirty feature points (v1,v2,v3,……..,v30) and the horizontal splitting results thirty feature points (h1,h2,h3,…….,h30).These feature points are obtained with relative to a central geometric point of the image.

Here the centered image is scanned from left to right and calculate the total number of black pixels. Then again from top to bottom and calculate the total number of black pixels. Then divide the image into two halves w.r.t. the number of black pixels by two lines vertically and horizontally which intersects at a point called the geometric centre. With reference to this point we extracted 60 feature points: 30 vertical and 30 horizontal feature points of each signature image.

3.2.3 Feature points based on Vertical Splitting:

Thirty feature points are obtained based on vertical splitting w.r.t. the central feature point. The procedure for finding vertical feature points is given below:

Algorithm 1:

Input: Static signature image after moving it to the centre of the fixed sized frame.

Output: Vertical feature points: v1,v2,v3,v4,……..v29,v30.

The *steps* are:

1) Split the image with a vertical line passing through the geometric centre (v0) which divides the image into two halves: Left part and Right part.
2) Find geometric centers v1 and v2 for left and right parts correspondingly.
3) Split the left and right part with horizontal lines through v1 and v2 to divide the two parts into four parts: Top-left, Bottom-left and Top-right, Bottom-right parts from which we obtain v3, v4 and v5, v6.
4) We again split each part of the image through their geometric centers to obtain feature points v7, v8, v9 ….v13, v14.
5) Then we split each part once again to obtain all the thirty vertical feature points.

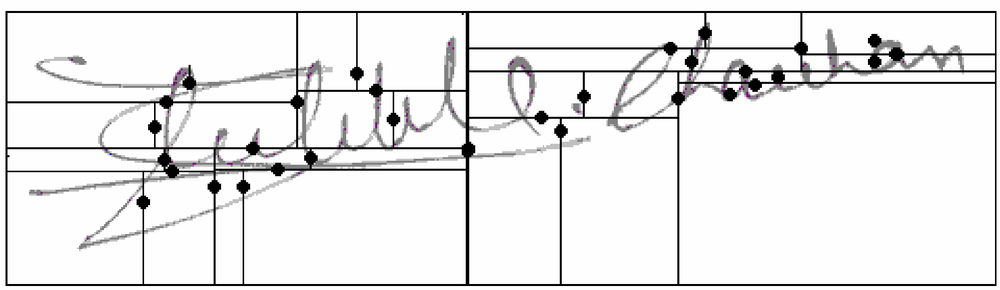


Fig 3.2: Vertical splitting of the signature image

3.2.4 Feature points based on Horizontal Splitting :

Thirty feature points are obtained based on horizontal splitting w.r.t. the central feature point. The procedure for finding horizontal feature points is given below:

Algorithm 2:

Input: Static signature image after moving it to the centre of the fixed sized frame.

Output: Horizontal feature points: h1,h2,h3,h4,……..h29,h30.

The *steps* are:

1) Split the image with a horizontal line passing through the geometric centre (h0) which divides the image into two halves: Top part and Bottom part.
2) Find geometric centers h1 and h2 for top and bottom parts correspondingly.
3) Split the top and bottom part with vertical lines through h1 and h2 to divide the two parts into four parts: Left-top, Right-top and Left-bottom, Right-bottom parts from which we obtain h3, h4 and h5, h6.
4) We again split each part of the image through their geometric centers to obtain feature points h7, h8, h9 ….h13, h14.
5) Then we split each part once again to obtain all the thirty vertical feature points.

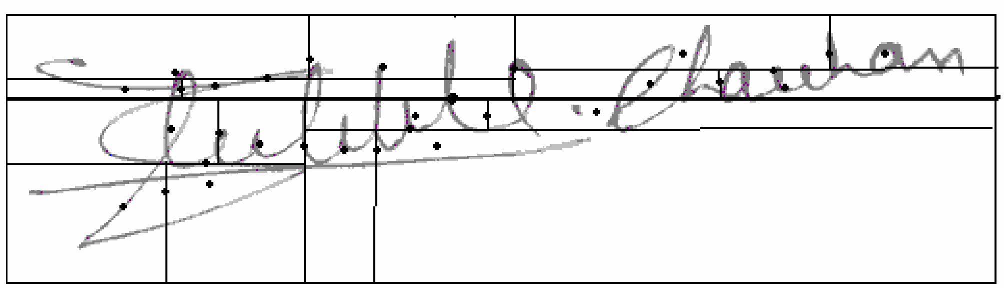


Fig 3.3: Horizontal splitting of the signature image

3.3 Classification:

In this paper features are based on geometric properties. So we use Euclidean distance model for classification. This is the simple distance between a pair of vectors of size *n*. Here vectors are nothing but feature points, so the size of vector is 2. How to calculate distance using Euclidean distance model is described in the following Section. In threshold calculation these distances are useful.

3.3.1 Euclidean distance model:

Let A ($a1$,$a2$…..an) and B($b1$,$b2$….bn) are two vectors of size n. We can calculate *distance* (d) by using equation 1.

$$distance(d) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \quad \text{………………….(3.1)}$$

In our application, vectors are feature points on plane. So d is the simple distance between two points.

3.4 Threshold

We have calculated individual thresholds for vertical splitting and horizontal splitting. Here we proposed one method for threshold selection. Fig. 5 shows the variations in single corresponding feature points of training signatures. Let n is the number of training signatures and $x1$,$x2$, ….,xn are corresponding single feature points of training signatures(taking one corresponding feature point from each signature). x_{median} is the median of n features from n signatures.

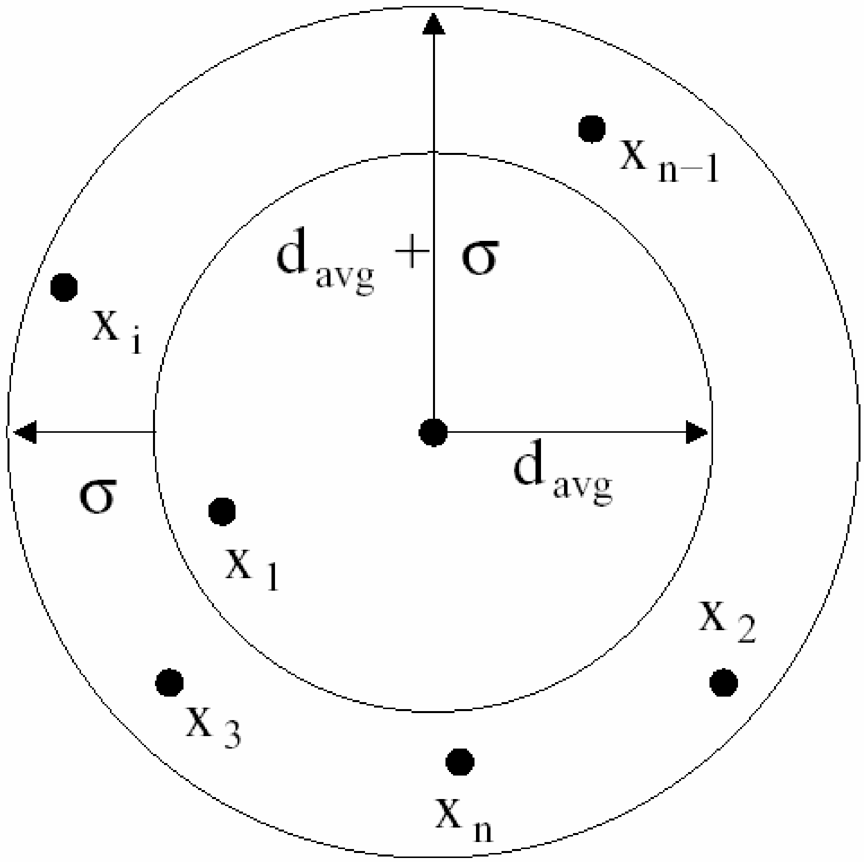


Fig3.4: *davg* (average distance) and s (standard deviation) derivation from distances

Let *d*1,d2,….,*dn* are distances defined here,

$$d1 = distance(x_{median};x1)$$
$$d2 = distance(x_{median};x2)$$
$$\ldots\ldots$$
$$dn = distance(x_{median};xn) \qquad \ldots\ldots\ldots\ldots\ldots \quad (3.2)$$

Two main parameters we used in threshold calculation are d_{avg} and s. Equations 3.3 and 3.4 shows the calculation of these two parameters.

$$d_{avg} = average(d1,d2,\ldots,dn) \quad \ldots\ldots\ldots\ldots\ldots(3.3)$$

$$\sigma = SD(d1,d2,\ldots\ldots,dn) \quad \ldots\ldots\ldots\ldots\ldots (3.4)$$

Like this total thirty different feature points are there for both vertical and horizontal splitting based on average distance (*davg*) and standard deviation (s). Equation 3.5 shows the main formula for threshold.

$$threshold(t) = \sqrt{\sum_{t=1}^{30}(d_{avg,t} + \sigma_t)^2} \quad \ldots\ldots\ldots\ldots\ldots(3.5)$$

3.5 Experiments & Results:

For experiment we took 8 original signatures from each person and selected 3 for training. These original signatures are taken in different days. Forgeries taken by three persons and 10 from each. Total 50 originals and 30 forgeries for each person signature are going to be tested. There are two thresholds (one based on vertical splitting and another based on horizontal splitting) for each person signature.

3.5.1 Training:

Let *n* signatures are taking for training from each person. There are 60 feature points from each original signature, 30 are taken by vertical splitting (Section2.2) and 30 are taken by horizontal splitting (Section2.3). Individual thresholds and patterns are calculated for vertical splitting and horizontal splitting. Pattern points based on vertical splitting are shown below.

$$Vpattern,1 = median(v1,1;v2,1;\ldots\ldots;vn,1)$$
$$Vpattern,2 = median(v1,2;v2,2;\ldots\ldots;vn,2)$$
$$Vpattern,3 = median(v1,3;v2,3;\ldots\ldots;vn,3)$$
$$Vpattern,4 = median(v1,4;v2,4;\ldots\ldots;vn,4) \qquad (3.6)$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$Vpattern,29 = median(v1,29;v2,29;\ldots\ldots;vn,29)$$
$$Vpattern,30 = median(v1,30;v2,30;\ldots\ldots;vn,30)$$

Where $vi,1;vi,2;\ldots\ldots;vi,30$ are vertical splitting features of *ith* training signature sample. Threshold based on vertical splitting is shown below:

Now we have to calculate the Vd which is the distance of the first feature point off all the training signatures from the geometric centers.

$$Vd_n = \text{Distance}(x_{median}, x_{n)}$$

So,
$$Vd1,1 = \text{Distance}(Vpattern,1 , V_{1,1})$$
$$Vd1,2 = \text{Distance}(Vpattern,1 , V_{1,2})$$
$$Vd1,3 = \text{Distance}(Vpattern,1 , V_{1,3})$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$Vd1,n = \text{Distance}(Vpattern,1 , V_{1,n})$$

Therefore,

davg1 = Average(Vd1,1; Vd1,2; Vd1,3; …….Vd1,n)

where davg1 is the average of Vd's for the first feature point of n signatures. Similarly we calculate davg2,davg3,…….,davg29,davg30 for the 2^{nd}, 3^{rd}, ………,29^{th}, and 30^{th} feature points.

Now we know that, Variance,

$$\sigma = SD\ (d1,d2,\ldots\ldots,dn)$$

Therefore,

$$\sigma 1 = SD\ (Vd1,1;\ Vd1,2,\ldots\ldots,Vd1,n)$$

$$\sigma 2 = SD\ (Vd2,1,Vd2,2\ldots\ldots,Vd2,n)$$

……………………………………

……………………………………

……………………………………

$$\sigma 30 = SD\ (Vd30,1;\ Vd30,2,\ldots\ldots,Vd30,n)$$

Hence,

$$v_{threshold} = \sqrt{\sum_{t=1}^{30} (vd_{avg,t} + \sigma_{v,t})^2} \quad \ldots\ldots\ldots (3.7)$$

Now we apply the same procedure to calculate the $h_{threshold}$ using the horizontal feature points.

$$hpattern,1 = median(h1,1;h2,1;\ldots\ldots;hn,1)$$

$$hpattern,2 = median(h1,2;h2,2;\ldots\ldots;hn,2)$$

$$hpattern,3 = median(h1,3;h2,3;\ldots\ldots;hn,3)$$

$$hpattern,4 = median(h1,4;h2,4;\ldots\ldots;hn,4) \quad \ldots\ldots\ldots\ldots\ldots (3.8)$$

…………………………………………

…………………………………………

…………………………………………

$$hpattern,29 = median(h1,29;h2,29;\ldots\ldots;hn,29)$$

$$hpattern,30 = median(h1,30;h2,30;\ldots\ldots;hn,30)$$

Where $hi,1;hi,2;\ldots\ldots;hi,30$ are horizontal splitting features of *ith* training signature sample. Threshold based on horizontal splitting is shown below:

$$h_{threshold} = \sqrt{\sum_{t=1}^{30} (hd_{avg,t} + \sigma_{h,t})^2} \tag{3.9}$$

We will store pattern points and thresholds of both horizontal splitting and vertical splitting. These values are useful in testing.

3.5.2 Testing:

When new signature comes for testing we have to calculate features of vertical splitting and horizontal splitting. Feature points based vertical splitting are *vnew;*1, *vnew;*2, *vnew;*3, *vnew;*4,…….. *vnew;29*, *vnew;30*. Distances between new signature features and pattern feature points based on vertical splitting are shown below:

$$\begin{aligned}
vdnew;1 &= distance(vpattern;1;vnew;1) \\
vdnew;2 &= distance(vpattern;2;vnew;2) \\
vdnew;3 &= distance(vpattern;3;vnew;3) \\
vdnew;4 &= distance(vpattern;4;vnew;4) \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
vdnew;29 &= \mathrm{d}istance(vpattern;29;vnew;29) \\
vdnew;30 &= \mathrm{d}istance(vpattern;30;vnew;30)
\end{aligned} \tag{3.10}$$

For classification of new signature we have to calculate *vdistance* and compare this with *vthreshold*. If *vdistance* is less than or equal to *vthreshold* then new signature is acceptable by vertical splitting.

$$v_{distance} = \sqrt{\sum_{i=1}^{30} vd_{new,i}^2} \qquad \text{………………… (3.11)}$$

Feature points based vertical splitting are *hnew;*1, *hnew;*2, *hnew;*3, *hnew;*4,…… *hnew;29; hnew;*30. Distances between new signature features and pattern feature points based on horizontal splitting are shown below:

$$
\begin{aligned}
hdnew;1 &= distance(hpattern;1;hnew;1)\\
hdnew;2 &= distance(hpattern;2;hnew;2)\\
hdnew;3 &= distance(hpattern;3;hnew;3)\\
hdnew;4 &= distance(hpattern;4;hnew;4) \qquad \text{………………(3.12)}\\
&\text{……………………………………….}\\
&\text{……………………………………….}\\
&\text{………………………………………}\\
hdnew;29 &= distance(hpattern;29;hnew;29)\\
hdnew;30 &= distance(hpattern;30;hnew;30)
\end{aligned}
$$

For classification of new signature we have to calculate *hdistance* and compare this with *hthreshold*. If *hdistance* is less than or equal to *hthreshold* then new signature is acceptable by horizontal splitting.

$$h_{distance} = \sqrt{\sum_{i=1}^{30} hd_{new,i}^2} \qquad \text{……………(3.13)}$$

New signature features have to satisfy both vertical splitting and horizontal splitting thresholds.

3.5.3 Results

False Acceptance Rate (FAR) and *False Rejection Rate (FRR)* are the two parameters using for measuring performance of any signature verification method. FAR is calculated by equation 3.14 and FRR is calculated by equation 3.15.

$$FAR = \frac{number\ of\ forgeries\ accepted}{number\ of\ forgeries\ tested} \times 100 \tag{3.14}$$

$$FRR = \frac{number\ of\ originals\ rejected}{number\ of\ originals\ tested} \times 100 \tag{3.15}$$

Table 3.1 shows the False Acceptance Rate of our method for different type of forgeries. Table 3.2 shows the False Rejection Rate for original signatures.

Table3.1: False Acceptance Rate(FAR)

Forgery Type	FAR(%)
RANDOM	**0.43**
SIMPLE	**0.98**
SKILL	**2.08**

Table3.2: False Rejection Rate(FRR)

FALSE REJECTION RATE(FRR)	
Original Signature	**20.83**

CHAPTER 4

PERFORMANCE AND ANALYSIS

i. Overview
ii. False Acceptance Rate (FAR)
iii. False Rejection Rate (FRR)

4.1 Overview:

This chapter analyses the results of our proposed technique and compares it performance with the existing techniques. Section 4.2 discuses the performance improvement in false acceptance rate. Section 4.3 discuses performance of False Rejection Rate.

4.2 False Acceptance Rate (FAR):

Table 4.1 compares the False Acceptance Rate of proposed technique with two existing technique that have discussed earlier. Proposed technique leads to better result then the existing two techniques.

Table4.1: Comparative analysis of FAR

Forgery Type	Existing Scheme	Existing Scheme with 12 feature point	Proposed Scheme with 60 feature point
RANDOM	5.61	2.08	0.43
SIMPLE	16.39	9.75	0.98
SKILL	19.3	16.36	2.08

4.3 False Rejection Rate(FRR):

Table 4.2 compares the False Rejection Rate of our proposed technique with the other two existing techniques.

Table2: Comparative analysis of FRR

FALSE REJECTION RATE(FRR)	
Existing Scheme	19.1
Existing Scheme(12 feature point)	14.58
Proposed Scheme(60 feature point)	20.83

CHAPTER 5

Conclusion

5.1 CONCLUSION

The Algorithm which is based on the 60 feature points is more efficient and gives more accurate results than the existing Techniques and survives against the skilled forgeries. The algorithm results the FAR which is very much less as compared to the FARs of the previously existing techniques. We compared our algorithms with other techniques based on feature extraction (12 feature points) and techniques based on Polar and Cartesian coordinates. But as our algorithm takes 60 feature points for threshold calculations, a small variation of a signature results in a large change in the values of threshold distance from the geometric center. Therefore in our algorithm the FRR value is increased. So it is important for a user to sign his signature with utmost care so that there is not a large variation of his signature to his training signatures. Otherwise there is a probability of rejection of an original signature. Moreover, since we have extracted 30 feature points by vertical splitting and 30 by horizontal splitting for the calculation of the threshold value, the time complexity is higher than the time complexity of the existing technique which uses 12 feature points for threshold calculations.

REFERENCES:

[1] Majhi Banshider, Reddy Y Santhosh, Babu D Prasanna "Novel Features for Off-line Signature Verification" International Journal of Computers, Communications & Control Vol. I (2006), No. 1, pp. 17-24

[2] J.J. Brault and R. Plamondon, "Segmanting Handwritten Signatures at Their Perceptually Important Points",
IEEE Trans. Pattern Analysis and Machine Intelligence, vol.15, no. 9, pp.953-957, Sept.1993.

[3] J Edson, R. Justino, F. Bortolozzi and R. Sabourin, "A comparison of SVM and HMM classifiers in the off-line
signature verification", *Pattern Recognition Letters* 26, 1377-1385, 2005.

[4] J Edson, R. Justino, F. Bortolozzi and R. Sabourin, "An off-line signature verification usingHMMfor Random,
Simple and Skilled Forgeries", *Sixth International Conference on Document Analysis and Recognition*, pp.
1031-1034, Sept.2001.

[5] J Edson, R. Justino, F. Bortolozzi and R. Sabourin, "The Interpersonal and Intrapersonal Variability Influences
on Off-line Signature Verification Using HMM", *Proc. XV brazilian Symp. Computer Graphics and Image
Processing*, 2002, pp. 197-202 Oct.2002.

24 Banshider Majhi, Y Santhosh Reddy, D Prasanna Babu

[6] J Edson, R. Justino, A. El Yacoubi, F. Bortolozzi and R. Sabourin, "An off-line Signature Verification System
Using HMM and Graphometric features", *DAS 2000*, pp. 211-222, Dec.2000.

[7] B. Fang, C.H. Leung, Y.Y. Tang, K.W. Tse, P.C.K. Kwok and Y.K. Wong, "Off-line signature verification by
the tracking of feature and stroke positions", *Patter Recognition* 36, 2003, pp. 91–101.

[8] Migual A. Ferrer, Jesus B. Alonso and Carlos M. Travieso, "Off-line Geometric Parameters for Automatic Signature
Verification Using Fixed-Point Arithmetic", *IEEE Tran. on Pattern Analysis and Machine Intelligence*,
vol.27, no.6, June 2005.

[9] R. Plamondon and S.N. Srihari, "Online and Offline Handwriting Recognition: A Comprehensive Survey",
IEEE Tran. on Pattern Analysis and Machine Intelligence, vol.22 no.1, pp.63-84, Jan.2000.

[10] A. Zimmer and L.L. Ling, "A Hybrid On/Off Line Handwritten Signature Verification System", *Seventh International
Conference on Document Analysis and Recognition*, vol.1, pp.424-428, Aug.2003.

Appendix A

STATISTICS:

A.1 Variance:

In general, the population variance of a *finite* population of size N is given by

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2$$

Or, if the population is an abstract population with probability distribution Pr:

$$\sigma^2 = \sum_{i=1}^{N}(x_i - \overline{x})^2 \ \Pr(x_i),$$

where $\overline{x}$ is the population mean. This is merely a special case of the general definition of variance introduced above, but restricted to finite populations

A.2 Standard Deviation:

Standard deviation of a random variable

The standard deviation of a random variable X is defined as:

$$\sigma = \sqrt{\mathrm{E}((X - \mathrm{E}(X))^2)} = \sqrt{\mathrm{E}(X^2) - (\mathrm{E}(X))^2}$$

where $\mathrm{E}(X)$ is the expected value of X.

Not *all* random variables have a standard deviation, since these expected value need not exist. For example, the standard deviation of a random variable which follows a Cauchy Distribution is undefined.

If the random variable X takes on the values $x_1,...,x_N$ (which are real numbers) with equal probability, then its standard deviation can be computed as follows. First, the mean of X, $\overline{x}$, is defined as a summation:

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i = \frac{x_1 + x_2 + \cdots + x_N}{N}$$

where N is the number of samples taken. Next, the standard deviation simplifies to:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2}$$

In other words, the standard deviation of a discrete uniform random variable X can be calculated as follows:

1. For each value x_i calculate the difference $x_i - \overline{x}$ between x_i and the average value $\overline{x}$.
2. Calculate the squares of these differences.
3. Find the average of the squared differences. This quantity is the variance σ^2.
4. Take the square root of the variance.