*Journal of Computer Sciences and Applications, 2013, Vol. 1, No. 2, 23-26*
Available online at http://pubs.sciepub.com/jcsa/1/2/2
© Science and Education Publishing
DOI:10.12691/jcsa-1-2-2

# An Off-Line Signature Verification

**Vahe Khachaturyan***

Digital Signal and Image Processing Laboratory, Institute for Informatics and Automation Problems, Yerevan, Armenia
*Corresponding author: vahekh@mail.ru

*Received January 01, 2012; Revised April 20, 2013; Accepted April 23, 2013*

**Abstract** An off-line signature verification system attempts to authenticate the identity of an individual by examining his/her handwritten signature, after it has been successfully extracted from, for example, a cheek, a debit or credit card transaction slip, or any other legal document. In this paper a novel off-line signature verification system selecting 120 feature points from the geometric center of the signature and compares them with the already trained feature points.

***Keywords:*** *offline signature, geometric center, feature point, FAR (false acceptance rate)*

## 1. Introduction

Signature verification is an important research area in the field of personal authentication. The recognition of human handwriting is important concerning about the improvement of the interface between human-beings and computers (e.g. [1-8]). 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 (e.g. [1]). 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 (e.g. [7]).

A signature verification system and the techniques using to solve this problem can be divided into two classes: online and off-line (e.g. [9]). 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, numbers of strokes are available (e.g. [4,5,6]). 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 (e.g. [2,3]).

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. We concern with offline signature verification which is based on geometric center 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 center.

## 2. Feature Extraction

The geometric features are based on two sets of points in 2-dimensional plane (e.g. [7]). The vertical splitting of the image results sixty feature points (v1,v2,v3,…,v60) and the horizontal splitting results sixty feature points (h1,h2,h3,…,h60). 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 center. With reference to this point we extracted 120 feature points: 60 vertical and 60 horizontal feature points of each signature image. We take 120 feature points based on many test. When we increase this number for current image size we will have very little sub images that we cannot split it. In this case 120 is optimal number for our algorithm.

### 2.1. Processing of the Signature

The geometric features proposed here are based on two sets of points in two-dimensional plane. Each set having sixty 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 (e.g. [1]). The processing of the signature is discussed below.

## 2.2. Moving Signature into the Center of Image

The signature is moved to the center 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 as shown in Figure 1.

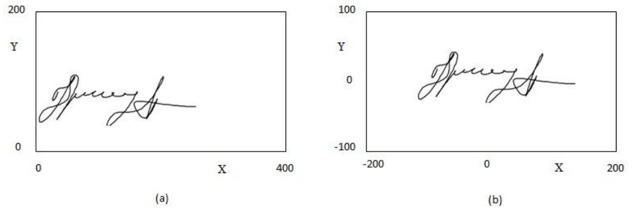

**Figure 1.** Captured signature (A) before adjustment and (B) after adjustment

## 2.3. Feature Points Based on Vertical Splitting

Sixty 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:**

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

*Output:* Vertical feature points: v1,v2,…,v59,v60.

The steps are:

1) Split the image with a vertical line passing through the geometric center (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 of the parts once again to obtain all the sixty vertical feature points (as shown in Figure 2).

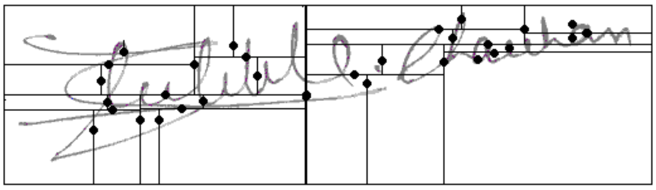

**Figure 2.** Vertical splitting of the signature image

## 2.4. Feature Points Based on Horizontal Splitting

Sixty 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:**

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

*Output:* Horizontal feature points: h1,h2,…,h59,h60.

The steps are:

1) Split the image with a horizontal line passing through the geometric center (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 of the parts once again to obtain all the sixty vertical feature points.

# 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.

*A. Euclidean distance model*

Let $A(a_1,a_2,\ldots,a_n)$ and $B(b_1,b_2,\ldots,b_n)$ are two vectors of size $n$. We can calculate ($d$) by using equation 1.

$$d = \sqrt{\sum_{l=1}^{n}\left(a_l - b_l\right)^2} \quad (1)$$

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

# 4. Threshold

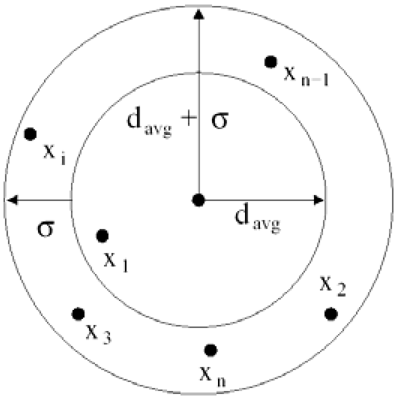

**Figure 3.** davg (average distance) and s (standard deviation) derivation from distances

We have calculated individual thresholds for vertical splitting and horizontal splitting. Here we proposed one method for threshold selection. Figure 3 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). *Xmedian* is the median of *n* features from *n* signatures.

Let *d1,d2,...,dn* are distances defined here,

$$\begin{aligned} d1 &= distance(xmedian,\ x1) \\ d2 &= distance(xmedian,\ x2) \\ &\ldots\ldots \\ dn &= distance(xmedian,\ xn) \end{aligned} \tag{2}$$

Two main parameters we used in threshold calculation are *davg* and s. Equations 3 and 4 shows the calculation of these two parameters.

$$davg = average(d1, d2, \ldots, dn) \tag{3}$$

$$\sigma = SD(d1, d2, \ldots\ldots, dn) \tag{4}$$

Like this total sixty different feature points are there for both vertical and horizontal splitting based on average distance (*davg*) and standard deviation (*s*). Equation 5 shows the main formula for threshold.

$$threshold\,(t) = \sqrt{\sum_{i=1}^{60}\left(d_{avg,i} + \sigma_i\right)^2} \tag{5}$$

## 5. Experiments and Results

For experiment we took 27 original signatures from each person and selected 9 for training. These original signatures are taken in different days. Forgeries taken by three persons and 9 from each. Total 18 originals and 27 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.

*A. Training*

Let *n* signatures are taking for training from each person. There are 120 feature points from each original signature, 60 are taken by vertical splitting (Section2 B) and 60 are taken by horizontal splitting (Section2 C). Individual thresholds and patterns are calculating for vertical splitting and horizontal splitting. Pattern points based on vertical splitting are shown below.

$$\begin{aligned} vpattern;1 &= median(v1;1, v2;1, \ldots, vn;1) \\ vpattern;2 &= median(v1;2, v2;2, \ldots, vn;2) \\ &\ldots\ldots\ldots \\ vpattern;59 &= median(v1;59, v2;59, \ldots, vn;59) \\ vpattern;60 &= median(v1;60, v2;60, \ldots, vn;60) \end{aligned} \tag{6}$$

Where vi;1,vi;2,…,vi;60 are vertical splitting features of i-th training signature sample. Threshold based on vertical splitting is shown below.

$$v_{threshold} = \sqrt{\sum_{i=1}^{60}\left(vd_{avg,i} + \sigma_{v,i}\right)^2} \tag{7}$$

In equation 9 $vd_{avg,i}$ is same as average distance and $\sigma_{v,i}$ is same as standard deviation shown in Section 4. Pattern points based on horizontal splitting are shown below.

$$\begin{aligned} hpattern;1 &= median(h1;1, h2;1, \ldots, hn;1) \\ hpattern;2 &= median(h1;2, h2;2, \ldots, hn;2) \\ &\ldots\ldots \\ hpattern;59 &= median(h1;59, h2;59, \ldots, hn;59) \\ hpattern;60 &= median(h1;60, h2;60, \ldots, hn;60) \end{aligned} \tag{8}$$

Where *hi;1,hi;2,...,hi;60* are horizontal splitting features of *i-th* training signature sample. Threshold based on horizontal splitting is shown below.

$$h_{threshold} = \sqrt{\sum_{i=1}^{60}\left(hd_{avg,i} + \sigma_{h,i}\right)^2} \tag{9}$$

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

*B. 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;60*. Distances between new signature features and pattern feature points based on vertical splitting are shown below.

$$\begin{aligned} vdnew;1 &= distance(vpattern;1, vnew;1) \\ vdnew;2 &= distance(vpattern;2, vnew;2) \\ &\ldots\ldots \\ vdnew;59 &= distance(vpattern;59, vnew;59) \\ vdnew;60 &= distance(vpattern;60, vnew;60) \end{aligned} \tag{10}$$

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.

$$v_{distance} = \sqrt{\sum_{i=1}^{60} vd_{new,i}^{2}} \tag{11}$$

Feature points based vertical splitting are *hnew;*1,*hnew;*2,*hnew;*3,…,*hnew;*60. Distances between new signature features and pattern feature points based on vertical splitting are shown below.

$$\begin{aligned} hdnew;1 &= distance(hpattern;1, hnew;1) \\ hdnew;2 &= distance(hpattern;2, hnew;2) \\ &\ldots\ldots \\ hdnew;59 &= distance(hpattern;59, hnew;59) \\ hdnew;60 &= distance(hpattern;60, hnew;60) \end{aligned} \tag{12}$$

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.

$$h_{distance} = \sqrt{\sum_{i=1}^{60} hd_{new,i}^{2}} \qquad (13)$$

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

*C. 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 14 and FRR is calculated by equation 15.

$$FAR = \frac{number\ of\ \ forgeries\ accepted}{number\ of\ \ forgeries\ tested} \times 100 \qquad (14)$$

$$FAR = \frac{number\ of\ originals\ rejected}{number\ of\ originals\ tested} \times 100 \qquad (15)$$

**Table 1. Comparative analysis of FAR**

| Forgery Type | Existing Scheme | Proposed Scheme with 60 feature point | Proposed Scheme with 120 feature point |
|---|---|---|---|
| Random | 20 | 18 | 16 |
| Simple | 18 | 16 | 14 |
| Skill | 24 | 20 | 18 |

**Table 2. Comparative analysis of FRR**

| FALSE REJECTION RATE(FRR) | |
|---|---|
| Existing Scheme | 20 |
| Proposed Scheme (60 feature point) | 18 |
| Proposed Scheme (120 feature point) | 16 |

## 6. Conclusion

The Algorithm which is based on the 120 feature points is more efficient and gives more accurate results than the existing Techniques and survives against the skilled forgeries. We compared our algorithms with other techniques based on feature extraction and techniques based on Polar and Cartesian coordinates. But as our algorithm takes 120 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 and FAR value are decreased.

## References

[1] Marc J.J. Brault and R. Plamondon, "Segmanting Handwritten Signatures at Their Perceptually Important Points", *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(9): 953-957, Sept.1993.

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

[3] 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.

[4] 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", *Pattern Recognition*, 36(1): 91-101. 2003.

[5] Migual A. Ferrer, Jesus B. Alonso and Carlos M. Travieso, "Off-line Geometric Parameters for Automatic SignatureVerification Using Fixed-Point Arithmetic", IEEE Tran. on Pattern Analysis and Machine Intelligence, 27(6), June 2005.

[6] R. Plamondon and S.N. Srihari, "Online and Offline Handwriting Recognition: A Comprehensive Survey", IEEE Tran. on Pattern Analysis and Machine Intelligence, 22(1): 63-84, Jan.2000.

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

[8] 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.

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