Handwritten Signature Verification for Mobile Phones

Nilakantha Paudel[1], Marco Querini[2] and Giuseppe F. Italiano[2]
[1]*Roma Tre University, Department of Mathematics, Largo San Leonardo Murialdo, 1, 00146, Roma, Italy*
[2]*University of Rome "Tor Vergata" via del Politecnico 1, 00133, Roma, Italy*

Keywords: Handwritten Signature, Signature Verification, Mobile Signature, Identity Verification, Biometric Security.

Abstract: Handwritten Signature Verification (HSV) systems have been introduced to automatically verify the authenticity of a user signature. In offline systems, the handwritten signature (represented as an image) is taken from a scanned document, while in online systems, pen tablets are used to register signature dynamics (e.g., its position, pressure and velocity). The main contribution of this work is a new HSV algorithm specifically designed for running on low-end mobile devices. Towards this end, we report the results of an experimental evaluation of our system on different handwritten signature datasets.

1 INTRODUCTION

Biometrics examine the physical or behavioral traits that can be used to determine a person's identity. Biometric recognition is the automatic recognition of a person based on one or more of these traits. This method of authentication ensures that the person is physically present at the point-of-identification and makes unnecessary to remember a password or to carry a token. The most popular biometric traits used for authentication are face, voice, fingerprint, iris and handwritten signature.

In this paper, we focus on handwritten signature verification (HSV), which is a natural and trusted method for user identity verification. HSV can be classified into two main classes, based on the device used and on the method used to acquire data related to the signature: online and offline signature verification. Offline methods process handwritten signatures taken from scanned documents, which are, therefore, represented as images. This means that offline HSV systems only process the 2D spatial representation of the handwritten signature (i.e., its shape). Conversely, online systems use specific hardware, such as pen tablets, to register pen movements during the act of signing. For this reason, online HSV systems are able to process dynamic features of signatures, such as the time series of the pen's position and pressure.

Online HSV has been shown to achieve higher accuracy than offline HSV (Qiao et al., 2007; Kalera et al., 2004; Jain et al., 2002), but unfortunately it suffers from several limitations.

In fact, handwritten signatures are usually acquired by means of digitizing tablets connected to a computer, because common low-end mobile devices (such as mobile phones) may not be able to support the verification algorithms (due to their hardware configuration capacity to compute the algorithm) or may be too slow to run the verification algorithm (due to limited computational power). As a result, the range of possible usages of the verification process is strongly limited by the hardware needed. To overcome this limitation, one needs techniques capable of verifying handwritten signatures acquired by smartphones and tablets in mobile scenarios with very high accuracy.

Online HSV systems (such as (Xyzmo, 2013; SutiDSignature, 2013; Andxor Corporation, 2013; Trevathan and McCabe, 2005; Mailah and Lim, 2012)) are able to address only partially these issues: they are supported by mobile devices, but they are not inherently designed for common low-end mobile devices such as mobile phones; several approaches make use of pen pads (special purpose hardware for handwriting), signature tablet (special purpose desktop and mobile hardware for signing), interactive pen displays (complete instruments for working in digital applications), Kiosk systems and PC Tablets.

As for the online HSV systems described in (Krish et al., 2013; Mendaza-Ormaza et al., 2011; Blanco-Gonzalo et al., 2013), even if experiments related to online HSV were carried out on low-end devices in

Paudel, N., Querini, M. and Italiano, G.
Handwritten Signature Verification for Mobile Phones.
DOI: 10.5220/0005675200460052
In *Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP 2016)*, pages 46-52
ISBN: 978-989-758-167-0

order to evaluate the verification accuracy, no analysis addressing the computational time is used in the algorithm design (which is particularly important, due to the limited computational power of mobile devices).

The goal of this paper is to address the above challenges by designing a new method which can be run on low-end devices too. The novelties of our approach lie mainly in the following aspects.

First, we propose a method for the verification of signature dynamics which is compatible to a wide range of low-end mobile devices (in terms of computational overhead and verification accuracy) so that no special hardware is needed.

Secondly, our new method makes use of several technical features that, to the best of our knowledge, have not been previously used for handwritten signature recognition.

Finally, in order to assess the verification accuracy of our HSV system, along with the average computational time, we conduct an experimental study whose results are reported for different data sets of signatures.

2 FEATURES OF THE ONLINE SIGNATURES

2.1 Dynamics

An online handwritten signature on a digital device is a series of points, and each point is represented by a vector in four dimensions, X, Y, Pressure and Time. We define these series of points as dynamics of the signature. When the user writes the signature, s/he might do pen-up and pen-down moves rather than moving the pen tip continuously. We define a stroke (ST) as the trajectory of a pen tip between a pen-down and a pen-up. A signature can be can be partitioned into multiple strokes as shown in Figure 1 and in Figure 2.

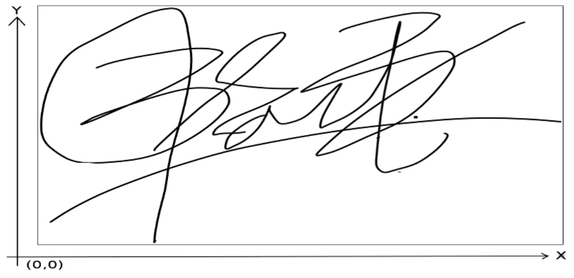

Figure 1: Handwritten online signature.

- *X,Y*: The x and y coordinates of each sampled point which is captured from the device screen.

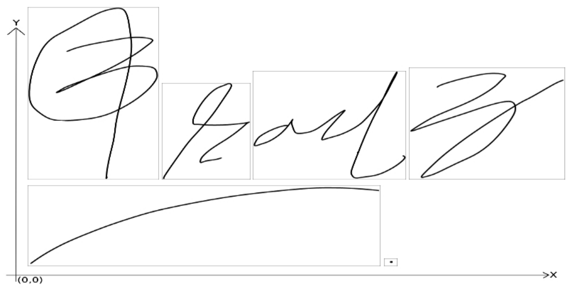

Figure 2: Strokes of the signature shown in Figure- 1, blocks are in left to right and top to bottom order.

Since the user may put his/her signature on any region of the screen, the X-Y coordinates are always translated so that the point given by the mean coordinates becomes the new origin of the coordinate system.

- *Pressure (P)*: The pressure with which the screen is pressed. When the pen is down, or when the user draws the line continuously, then the pressure value becomes 1 (maximum value) for that points. Similarly, when the pen is released from the screen, then pressure value becomes 0 (minimum value) for that specific point.
- *Time Series (TS)*: The sequence of equispaced sampling time instants. The sampling period, i.e., the time difference between two consecutive samples, is constant and exactly equal to the inverse of the device sampling frequency.

2.2 Features

We use the features to study the structure of the signature, and its strokes from the different perspective. Each features are equally important for the registration and verification steps. The only things is they will verify one after another. Why they are important and how they do the work for the signature registration and verification steps, we will explain in section 3.1 and 3.2. The statistical and mathematical tools over the dynamics are used to calculate the features as follows.

2.2.1 Features of the Signature

(i) *Pen-Up number*: Total number of pen-ups done by the user while s/he write his/her full signature.

(ii) *Path length(PL)*: The path length is the total path length travel by the user pen tip during the signature creation. The device sampling frequency gives the value of each dynamics in equal interval of time and the euclidean distance formula

calculate the distance between them in each interval as follows.
$PL = \sum_{i=2}^{n} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$ where $x_i \in X$ and $y_i \in Y$

(iii) *Diagonal length(DL)*: We extract the maximum (x_{max}, y_{max}) and minimum (x_{min}, y_{min}) points in *X,Y*. Then we use the euclidean distance formula for *2D, DL=* $(\sqrt{(x_{max} - x_{min})^2 + (y_{max} - y_{min})^2})$.

(iv) *Time Length(TL)*: The total time in milliseconds, which has taken by the user to write his complete signature (the time duration between the first pen down and last pen up).

(v) *Mean Speed(MS)*: The average of user writing speed for the signature. We have four different dynamics sets *(X, Y, TS, P)* of equal size. All points of these sets are sequential and tracked on the equal time interval from the device's screen with frequency. We calculate the speed/velocity between two subsequent points and sum them. At the end, we divide the total sum by the total number of points.
$MS = \frac{1}{n} \sum_{i=2}^{n} \frac{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}{(t_i - t_{i-1})}$, where $x_i \in X$, $y_i \in Y$ and $t_i \in TS$.

(vi) *Covariance-XY(CXY)*: In order to measure the scattered of the points on the signature path we calculate the *CXY* as follows: $CXY = \frac{1}{n} \sum_{i=1}^{n} \sqrt{(x_i)^2 + (y_i)^2}$, where, $x_i \in X$, $y_i \in Y$. We already translated X,Y with mean origin. So, we don't have to calculate the mean again.

(vii) *Vector length ratio (VLR)*: Each point of the signature captured by the acquiring device has a 4 dimensional representation *(X, Y, TS, P)*. As for *VLR*, we only focus on x-axis and y-axis. We calculate the sum of the length of all the vectors drawn from the origin to each point *(X,Y)* coordinates. Finally, the sum is divided by *PL*.
$VLR = \frac{1}{PL} \sum_{i=1}^{n} \sqrt{(x_i - x_{origin})^2 + (y_i - y_{origin})^2}$, where, $x_i \in X$, $y_i \in Y$.

2.2.2 Features of the Strokes

Stroke is the subsequence of a signature sequence. It has exactly the same features and dynamics that signature has.So, we calculate the following proportion of dynamics for each stroke over the full signature.

(i) *Path Length Ratio (PLR):* $\frac{\textit{PL of the stroke}}{\textit{PL of the signature}}$

(ii) *Time Length Ratio (TLR):* $\frac{\textit{TL of the stroke}}{\textit{TL of the signature}}$

(iii) *Diagonal Length Ratio (DLR):* $\frac{\textit{DL of the stroke}}{\textit{DL of the signature}}$

(iv) *Mean Speed Ratio (MSR):* $\frac{\textit{MS of the stroke}}{\textit{MS of the signature}}$

(v) *Covariance XY Ratio (CXYR):* $\frac{\textit{CXY of the stroke}}{\textit{CXY of the signature}}$

(vi) *Stroke Vector Length Ratio (STVLR):* $\frac{\textit{VLR of the stroke}}{\textit{VLR of the signature}}$

3 THE SIGNATURE VERIFICATION ALGORITHM

We describe next the registration and verification process from a technical perspective.

3.1 Signature Registration Phase

In this phase, the system takes the user's genuine signatures as input and generates the biometric template of the features with the following steps.

3.1.1 Acquisition and Pre-processing

In the acquisition phase, the user has to write the signatures with the same number of pen ups for three rounds as input. In each round, whenever the signature is captured from the screen, the pre-processing starts immediately. Then, the system eliminates the noise, normalizes the path and all kind of features are calculated and then checked with the features of existing signatures. In the checking process, the signature should have exactly the same number of pen ups.

The area covered by the signature and its length depend on the screen sizes. Since various devices may have different screen sizes and the feature values *(PL, DL, TL, MS, CXY, VLR, PLR, DLR, TLR, MSR, CXYR, STVLR)* depend on the screen pixel density, we use a certain tolerance factor to register the signature. For this reason, each feature should be similar with the features of existing signatures and their corresponding strokes by a certain tolerance factor. Otherwise, the user has to write the signature again for that round.

3.1.2 Template Generation and Store

Once pre-processing is completed, then the system has the features and dynamics of all the three signatures. So in this step, we calculate the average of each features as follows: $\frac{1}{3} \sum_{i=1}^{3} Feature_{(i)}$ and create an interval for each feature with its average value by certain threshold factor.

We also used the dynamic time warping (DTW) for template generation and signature verification process. In time series analysis, dynamic time warping (Müller, 2007) is an algorithm for measuring similarity between two temporal sequences which may vary in time or speed. In addition it has also been used for partial shape matching application. Moreover, it has been used in literature for both on-line and off-line HSV successfully (Faundez-Zanuy, 2007; Miguel-Hurtado et al., 2007; Piyush Shanker and Rajagopalan, 2007). There are different kind of algorithms to check the similarity between the sequences like Frèchet distance but we use DTW. This is because of its high accuracy and efficiency (in terms of computational time) which is well suited for our algorithm that is specially designed for mobile devices.

3.2 Signature Verification Phase

In the verification phase, the system makes the decision on whether the claimed signature is genuine or forged. We already calculated the accepted interval for each features in the template generation phase. The steps for the verification process are as follows:

3.2.1 Check with Global Features of the Signature

(i) *Check with Pen up Number*: If the claimed signature has a different number of *pen-ups*, then it will be rejected.

(ii) *Check with all features of the signature (PL, DL, TL, MS, CXY, VLR) respectively*:

If each feature of the claimed signature does not fall in its corresponding interval generated by template then it will be rejected.

3.2.2 Check with Features of the Strokes

The claimed signature may have more than a single stroke. For every stroke, the system checks all the features *(TLR, DLR, MSR, CXYR)*. Each feature should lie in the corresponding interval that was generated at the template generation phase. The system counts how many strokes pass the test. If this percentage is lower than a certain threshold then the signature is rejected.

3.2.3 Check with DTW

If the claimed signature passes all the above verification steps, then *DTW* is applied on it as follows.

Let m be the total number of strokes in a single signature. Then by using the feature of each signature, the following m-dimensional vector is computed. Let the i^{th} stroke (related to feature f of signature) of the j^{th} signature in a 1D time series be denoted as S_j^i. $DTW(S_j^i, S_k^i)$ denotes the 1D DTW method applied to the i^{th} segments of the j^{th} and k^{th} signatures.

$$\left\| \begin{matrix} f^1 \\ f^2 \\ \ldots \\ f^m \end{matrix} \right\| = \left\| \begin{matrix} \frac{DTW(S_1^1,S_2^1)+DTW(S_1^1,S_3^1)+DTW(S_2^1,S_3^1)}{3} \\ \frac{DTW(S_1^2,S_2^2)+DTW(S_1^2,S_3^2)+DTW(S_2^2,S_3^2)}{3} \\ \ldots \\ \frac{DTW(S_1^m,S_2^m)+DTW(S_1^m,S_3^m)+DTW(S_2^m,S_3^m)}{3} \end{matrix} \right\|$$

When $\|f\|$ vector is computed for each feature f, we get a $\|X\|$ vector (x coordinates), a $\|Y\|$ vector (y coordinates), a $\|P\|$ vector (P coordinates), and a $\|T\|$ vector (TS coordinates).

Finally, we combine the metrics with the following sums,

$$\left\| \begin{matrix} d^1 \\ d^2 \\ \ldots \\ d^m \end{matrix} \right\| = \left\| \begin{matrix} X^1 + Y^1 + P^1 + \ldots + T^1 \\ X^2 + Y^2 + P^2 + \ldots + T^2 \\ \ldots \\ X^m + Y^m + P^m + \ldots + T^m \end{matrix} \right\|$$

The output distance vector $\|d\|$ represents the "distance" among the three signatures. The whole process is repeated twice; the first time between the genuine registered signatures ($\|d_g\|$ as output, which is already calculated during the template generation phase) and the second time between the claimed signature and registered signatures ($\|d_v\|$ as output). In the template generation phase, we also calculated the interval by using the threshold factor in $\|d_g\|$. So if $\|d_v\|$ does not lie in that interval, then the claimed signature is rejected, otherwise accepted.

Now, we turn to describe our algorithm. First, the total pen up number is considered. If the signature to be verified has a different number of pen-ups, then the signature is assumed to be a forgery. If the forger writes the signature very fast then he/she produces the better line quality with less accuracy. Similarly, if he/she writes very slowly then the signature may be more accurate but the line quality is poor, and the time length is unnaturally high. So in either case, our algorithm works because of *TL*.

During the template generation phase, the user is totally free to write the genuine signature on the device screen. So, we calculate the *features* and *DL* for his/her signature from device perspective. Now, if the forger writes the signature on all the available area then it has a very high *features* and *DL*. Similarly, if he writes in a small area then it will have very low *features* and *DL*. In either case the algorithm works to reject it.

Even if the forger writes the signature within a given area with expected length and time, it is really difficult to write the signature with tolerable *MS* for the forger, even for the real user, s/he cannot write the signature with same *MS* as before, but s/he can write his/her signature within the tolerable interval of *MS*. Whereas forger can't do it and our algorithm can easily recognize his speed and reject it.

CXY measures the scatter value of all points in a signature that are distributed on the device screen. So, even if the forger writes a signature matching *PL, TL, MS*, it is unlikely to match the distribution of the points with the genuine signatures. So, whenever his/her signature does not match with *CXY* then our algorithm detects that it is a forgery.

A signature may have multiple strokes and each stroke has its own features (*PL, TL, DL, MS, CXY and STVLR*) because it is just a subsequence of the signature sequence. The features of each stroke are different from each other. So, our algorithm calculates all the features of each stroke and then finds out its ratio to the whole signature. So, even if the forger is able to write a signature which successfully passes all the global features test, still, if it does not passes the stroke ratio verification process then the signature is rejected.

Finally, if the forgery passes all the global and stroke feature tests, then, the signature undergoes *DTW* testing. *DTW* compares the similarity between two sequences. We find out two distance vectors: $\|d_g\|$ represents the "distance" among the three genuine signatures, while $\|d_v\|$ represents the "distance" among three genuine with claimed signature. If $\|d_v\|$ does not lie in the interval which is calculated on the basis of $\|d_g\|$ by certain threshold at the template generation phase, then it is rejected as a forgery.

4 EXPERIMENTATION

In this section we present experimental results concerning identity verification with our system. The accuracy of a recognition algorithm is generally measured in terms of two potential types of errors: false negatives (*fn*) and false positives (*fp*). *fp* are cases where a claimed identity is accepted, but it should not be, while *fn* are cases where a claimed identity is not accepted, while it should be. The frequency at which false acceptance errors occur is denoted as False Acceptance Rate *(FAR)*, while the frequency at which false rejection errors occur is denoted as False Rejection Rate *(FRR)*. Two metrics building on true/false positives/negatives (*tp,fp,tn,fn*) are widely adopted: precision and recall. Recall $(tp/(tp+fn))$ is the probability that a valid identity is accepted by the system (i.e., true positive rate) while precision $(tp/(tp+fp))$ is the probability that a claimed identity which is accepted by the system is valid. F-measure (which is the harmonic mean of precision and recall) combines both metrics into a global measure $(f\text{-}measure = (2 \times prec \times recall)/(prec + recall))$.

A threshold on the similarity score must be identified for determining whether two signatures are similar (accept the identity) or significantly different (reject the identity). The higher the threshold, the higher the precision (i.e., the lower the risk of accepting invalid identities). However, a high threshold also decreases the recall of the system (i.e., the higher the risk to reject valid identities).

The performance of the proposed scheme has been assessed in terms of false positives, false negatives, precision, recall and f-measure on three different datasets: on the SigComp2011 Dutch and Chinese datasets (Liwicki et al., 2011); on the SigComp2013 Japanese dataset (Malik et al., 2013).

We start by describing the experimental set-up. Several mobile devices have been involved in our experiments (i.e., Google Nexus 5, GalaxyS2, XperiaZ2 and ZTE Blade A430), along with several standard datasets. The specification of the datasets involved are as follows:

- The SigComp2011 (Liwicki et al., 2011) competition involved (online) dutch and chinese data. The purpose of using these two data sets was to evaluate the validity of the participating systems on both Western and Chinese signatures. Signature data were acquired using a WACOM Intuos3 A3 Wide USB Pen Tablet and collection software, i.e., MovAlyzer.
  - **Dutch Dataset**. The dataset is divided in two non-overlapping parts, a training set (comprised of 10 authors with 330 genuine signatures and 119 forgeries) and a test set (comprised of 10 authors with 648 genuine signatures and 611 corresponding forgeries).
  - **Chinese Dataset**. The dataset is divided in two non-overlapping parts, a training set (comprised of 10 authors with 230 genuine signatures and 430 forgeries) and a test set (comprised of 10 authors with 120 genuine signatures and 461 corresponding forgeries).
- The SigComp2013 (Malik et al., 2013) competition involved (online) data collected by PResearchers at the Human Interface Laboratory, Mie University Japan.

– **Japanese Dataset**. The signature data were acquired using a HP EliteBook 2730p tablet PC and self-made collection software built with Microsoft INK SDK. The whole dataset consists of 1260 genuine signatures (42 specimens/individual) and 1080 skilled forgeries (36 specimens/forgery). The dataset is divided in two non-overlapping parts, a training set (comprised of 11 authors with 42 genuine signatures of each author and 36 forgeries per author) and a test set (comprised of 20 authors with 42 genuine signatures each and 36 corresponding forgeries per author).

The experimental results in terms of precision, recall and f-measure (that vary according to the chosen thresholds) have been used for tuning the thresholds in order to get better performance.

The remainder of this section illustrates our results.

Table 1: Precision*(PCR)*, recall*(RCL)*, f-measure*(FMR)*, *FAR* and *FRR* as functions of a tolerance factor*(TF)*.

TF	*PCR*	*RCL*	*FMR*	*FRR*	*FAR*
34%	0.983	0.919	0.943	0.008	0.008
35%	0.969	0.934	0.945	0.014	0.065
36%	0.953	0.936	0.936	0.021	0.063

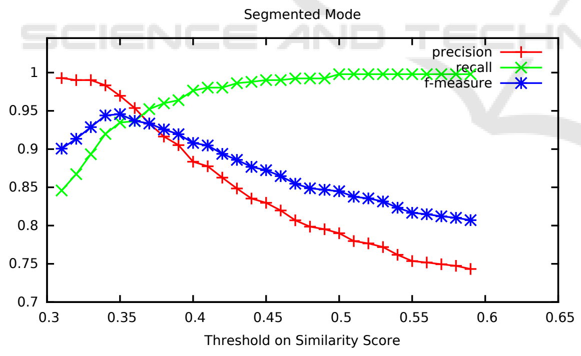

Figure 3: Precision, recall and f-measure as functions of threshold on tolerance factor. (Viewed better in color). (Viewed better in color).

The graph of Figure 3 plots precision, recall and f-measure as functions of the chosen tolerance factor, i.e., the threshold. Claimed identities are accepted whenever the score is above the threshold, rejected otherwise. The higher the threshold, the higher the precision, but the lower the recall. Table 1 shows the results related to precision, recall, f-measure, FAR, and FRR for values which maximize the f-measure. The best results were achieved using a 35% tolerance factor.

Finally, we address the computational overhead introduced by the color classifiers considered. We stress that the overall running time is important, since in many applications handwritten signatures could be decoded on low-end devices, such as mobile phones or tablets. Figure 4 illustrates the Box-and-Whisker plot for computational time distributions related to the different mobile devices involved in our experiments.

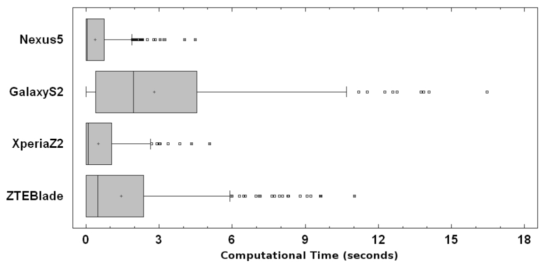

Figure 4: Computational time for different mobile devices: Box-and-Whisker plot indicating the smallest observation, lower quartile (Q1), median (Q2), upper quartile (Q3), and largest observation. (Viewed better in color).

The plots of Figure 4 show that even low-end devices (such as Samsung Galaxy S2) are able to verify the signature quickly (i.e., in a few seconds), while devices with high performance (such as Google Nexus 5) are really fast in verifying signatures (i.e., in a few hundreds of milliseconds).

5 CONCLUSIONS

Our work presented a new HSV system for document signing and authentication, whose novelties lie mainly in the following aspects. First, we proposed a method for the verification of signature features which is compatible to a wide range of low-end mobile devices (in terms of computational overhead and verification accuracy) so that no special hardware is needed. Secondly, our new method makes use of several technical features that, to the best of our knowledge, have not been previously used for handwritten signature recognition. In our experiments, precision and recall cross at 94%. As for the overall computational time, the average verification time is under 1 second for devices such as Nexus 5 or Xperia Z2. This is an interesting result, especially when the limited computational power of mobile devices is considered.

REFERENCES

Andxor Corporation (2013). View2sign. http://www.view2 sign.com/supported-signatures.html [Online; accessed 01-April-2014].

Blanco-Gonzalo, R., Sanchez-Reillo, R., Miguel-Hurtado, O., and Liu-Jimenez, J. (2013). Performance evaluation of handwritten signature recognition in mobile environments. *IET Biometrics*, 3(3):139–146.

Faundez-Zanuy, M. (2007). On-line signature recognition based on VQ-DTW. *Pattern Recognition*, 40(3):981–992.

Jain, A. K., Griess, F. D., and Connell, S. D. (2002). On-line signature verification. *Pattern recognition*, 35(12):2963–2972.

Kalera, M. K., Srihari, S., and Xu, A. (2004). Offline signature verification and identification using distance statistics. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(07):1339–1360.

Krish, R. P., Fierrez, J., Galbally, J., and Martinez-Diaz, M. (2013). Dynamic signature verification on smart phones. In *Highlights on Practical Applications of Agents and Multi-Agent Systems*, pages 213–222. Springer.

Liwicki, M., Malik, M. I., van den Heuvel, C. E., Chen, X., Berger, C., Stoel, R., Blumenstein, M., and Found, B. (2011). Signature verification competition for online and offline skilled forgeries (sigcomp2011). In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 1480–1484. IEEE.

Mailah, M. and Lim, B. H. (2012). Biometric signature verification using pen position, time, velocity and pressure parameters. *Jurnal Teknologi*, 48(1):35–54.

Malik, M. I., Liwicki, M., Alewijnse, L., Ohyama, W., Blumenstein, M., and Found, B. (2013). Icdar 2013 competitions on signature verification and writer identification for on-and offline skilled forgeries (sigwicomp 2013). In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 1477–1483. IEEE.

Mendaza-Ormaza, A., Miguel-Hurtado, O., Blanco-Gonzalo, R., and Diez-Jimeno, F.-J. (2011). Analysis of handwritten signature performances using mobile devices. In *Security Technology (ICCST), 2011 IEEE International Carnahan Conference on*, pages 1–6. IEEE.

Miguel-Hurtado, O., Mengibar-Pozo, L., Lorenz, M. G., and Liu-Jimenez, J. (2007). On-line signature verification by dynamic time warping and Gaussian mixture models. In *International Carnahan Conference on Security Technology*, pages 23–29. IEEE.

Müller, M. (2007). *Information retrieval for music and motion*, volume 2. Springer.

Piyush Shanker, A. and Rajagopalan, A. (2007). Off-line signature verification using DTW. *Pattern Recognition Letters*, 28(12):1407–1414.

Qiao, Y., Liu, J., and Tang, X. (2007). Offline signature verification using online handwriting registration. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07*, pages 1–8. IEEE.

SutiDSignature (2013). SutiDSignature. http://www.sutisoft.com/sutidsignature. [Online; accessed 01-April-2014].

Trevathan, J. and McCabe, A. (2005). Remote handwritten signature authentication. In *ICETE*, pages 335–339. Citeseer.

Xyzmo (2013). Xyzmo signature solution. http://www.xyzmo.com/en/products/Pages/Signature-Verification.aspx. [Online; accessed 01-April-2014].