The Needed Input Data for Accurate On-line Signature Verification

THE RELEVANCE OF PRESSURE AND PEN INCLINATION FOR ON-LINE SIGNATURE VERIFICATION

THOMAS SJÖHOLM
The Needed Input Data for Accurate On-line Signature Verification

The relevance of pressure and pen inclination for on-line signature verification

Indatan som behövs för bra signaturverifiering
Relevansen av Tryckkänslighet och penvinklar för signatur verifiering

Thomas Sjöholm
thsj@kth.se

Subject: Computer Science
Programme: Computer Science and Engineering

Syntronic Supervisor: Håkan Sjörling - hast@syntronic.com
CSC Supervisor: Giampiero Salvi - giampi@kth.se
CSC Examinator: Olov Engwall - engwall@kth.se

19 November 2015
Abstract

Signatures have been used to authenticate documents and transactions for over 1500 years and are still being used today. In this project a method for verifying signatures written on a tablet has been developed and tested in order to test whether pressure information is vital for a well performing on-line signature verification systems. First a background study was conducted to learn about the state-of-the-art methods and what features several research systems used, then the method was developed. The method is a Dynamic Time Warp with 8 local features, 2 of them were pressure values or derived from pressure, and 1 global feature. The developed method was tested on SUSig visual corpus containing signatures from 94 persons. The Equal Error Rate (EER) when not using pressure was 5.39% for random forgeries and 3.24% for skilled forgeries. EER when using pressure was 5.19% for random forgeries and 2.80% for skilled forgeries.

The background study concluded that pen inclination is not required for a well performing system. Considering the result of this project and the result of others, it seems that pressure information is not vital, but provide some valuable information that can be used to classify signatures more accurately.

Sammanfattning


Givet resultatet av det här projektet samt andra projekt utforskade i bakgrundsforskningsn så verkar tryckkänslighet inte vara kritiskt men ger en del värdefull information för klassificera signaturer mer träffsäkert. Bakgrundsforskningsn gav att vinkeln på pennan inte var kritisk för att välpresterande system.
Contents

1 Introduction 1
  1.1 Commercial Interest ........................................ 1
  1.2 The Problem .................................................. 1

2 Background 3
  2.1 Features and Feature Extraction ................................ 3
  2.2 Dynamic Time Warping ........................................ 4
    2.2.1 Multidimensional Dynamic Time Warp ..................... 6
    2.2.2 Dynamic Time Warping for On-line Signature Verification 7
  2.3 Hidden Markov Model .......................................... 7
  2.4 Accuracy Description ......................................... 8
  2.5 Signature Verification Contest 2004 .......................... 8
    2.5.1 Signature Data Collection ............................... 9
    2.5.2 Competition Results .................................... 9
    2.5.3 Research using the Database ............................ 10
  2.6 BioSecure Signature Evaluation Campaign 2009 ................. 11
    2.6.1 Signature Data Collection ............................... 11
    2.6.2 Competition Results .................................... 12
  2.7 Sabanci University Signature Database ......................... 14
  2.8 Summary ..................................................... 14

3 Method 16
  3.1 Preprocessing and Feature Extraction ........................ 16
  3.2 Distance Calculation .......................................... 16
  3.3 Reference Set ................................................ 17
  3.4 Verification .................................................. 17
  3.5 Evaluation .................................................... 18
  3.6 Calculation of Equal Error Rate .............................. 18

4 Result 19
  4.1 Without Pressure ............................................. 20
  4.2 With Pressure ................................................ 22

5 Discussion 25
  5.1 Sources of Error ............................................. 25
  5.2 Ethical Aspects ............................................... 25
  5.3 Future Work .................................................. 26
  5.4 Summary of The Results ..................................... 26
  5.5 Conclusion ................................................... 27

References 28

Appendices 30
  A Raw Data 30
  B Figures 31
1 Introduction

Signatures have been used for authentication for more than a millennium. The signing as authentication as we know it today began during the rule of Roman emperor Valentinian III in the Roman Empire in the year AD 439. At first signatures were used to authenticate wills but the practice spread rapidly to be used to authenticate other documents and agreements [1].

A signature today is used by a person to authenticate a document or verify the identity of the person using a service [2]. Identity theft could occur with the combination of signature and some other personal identification information [3]. Identity theft is a fairly common problem in the world; it is in fact the number one complaint to the Federal Trade Commission in the US in 2013 [4]. The number of identity theft has increased in Sweden in the last few years [5]. Identity theft is costly for society [4]. Some identity thefts could be averted by improving the verification of the identities where the identity is used. There are several ways to verify the identity by using biometric information [6].

The most common and popular biometric modality is fingerprint recognition [6]. Biometric systems can also use other aspects than fingerprints to recognise a person, e.g. face, iris and hand geometry. This project focuses on the behavioural biometric of handwritten signatures.

There are two ways to approach the problem of verifying a handwritten signature depending on how the signature was captured. The system can either use a signature that is captured after the signing process is finished called off-line (static) system or signature is captured during the signing process called on-line (dynamic) system [7]. On-line signatures can be captured in several ways including cameras, motion sensitive pens and tablets [7]. This project will exclusively focus on on-line signature verification using tablet-captured signatures as input.

1.1 Commercial Interest

Companies are interested in this technology because it can be used internally to store signatures digitally instead of the currently used paper lists. Examples where paper lists are common today include visiting logs, key retrieval acknowledgement and equipment retrieval acknowledgement. The benefits of storing the acknowledgements include that if the signer and signature receiver do not know each other it could be possible to verify the identity of the receiver either right then and there or at some later point in time.

Banks, insurance companies and other signature heavy companies could be interested in a verification system. Most of the time signatures are written to authenticate some document handed over to the company. The signatures written on the documents are stored in paper formats and the best case scenario is that the signature receiver verifies the signature by looking at it and a reference signature found on the id-card that the signer shows. Many verification of this kind could be replaced with a system. Newly enrolled people will probably need human verification on the first few signature while the system is learning.

1.2 The Problem

Is pressure sensitivity and pen inclination vital input for an accurate on-line signature verification system? Most of the publicly available databases has pressure data and some have pen inclination as well [25, 26, 27]. This project will explore if a tablet that is only capturing the coordinates is enough for an on-line signature verification system or if the tablet needs to be pressure sensitive.

The need for pressure data and pen inclination data can be explored in the literature by comparing how well different signature verification systems perform when only having access to...
certain information or what set of features are being used. After exploring the literature, the system used in this project was built. The system was tested with various combinations features on a well known signature database freely available for the research community according to domain standards.
2 Background

This section will bring up the theory of commonly used on-line signature verification techniques but the main focus will be on the importance of pressure and pen inclination information in on-line signature verification systems. The goal of the background is to see if it is possible for a well performing systems to not be dependent on pen inclination or pressure information.

Two methods will be covered in this report: Dynamic Time Warping and Hidden Markov Model.

Dynamic Time Warping is an algorithm used to compare two sequences (possibly of different lengths) that could represent the same sequence but with speed varying over the sequence.

Hidden Markov Model is a statistical algorithm that given a set of visible observations can estimate the likelihood of hidden states.

There are many other methods including Support Vector Machine, Neural Network, Vector Quantisation, Fuzzy Logic, Partitioning the Signature, Hilbert Scanning Patterns and String/Graph/Tree matching. These are not covered in this report and can be read about in other literature. [7, 8, 18]

The result of two competitions, SVC2004 and BSEC2009, will be presented along with how these competitions collected their data and evaluated the competing systems.

2.1 Features and Feature Extraction

Features in the context of on-line signature verification are the values that the verification algorithm can utilize during the preprocessing or verification process. The most commonly used input features are the horizontal and vertical position of the pen, denoted by $x$ and $y$, the pressure applied to the surface denoted by $p$ and azimuth and altitude denoted by $\phi$ and $\theta$. A clarification for azimuth and altitude can be found in figure 1. The timestamp for a sample is denoted with $t$. Each signature has some number of samples of each feature, the number of samples are $N$ for that given signature.

Common databases have $x, y, p, \phi, \theta, t$ as input values for a signature [25, 26, 27]. Some databases do not have the $\phi$ and $\theta$ [27].

It is common to calculate several additional features from the actual input. Some commonly calculated local and global features are described below.

The velocity is the change in position since the last sample. The horizontal velocity is calculated below, but all velocities are calculated the same way but with its input signal instead of $x$.

$$\frac{dx_i}{dt} = \frac{x(i) - x(i-1)}{(t(i) - t(i-1))} \quad \forall i = 2, 3, \ldots, N \quad (1)$$

Other features can be derived using a similar method. The direction of the stroke can be calculated with:

$$\sin(i) = \frac{dy_i}{\sqrt{dy_i^2 + dx_i^2}} \quad \forall i = 2, 3, \ldots, N \quad (2)$$

$$\cos(i) = \frac{dx_i}{\sqrt{dy_i^2 + dx_i^2}} \quad \forall i = 2, 3, \ldots, N \quad (3)$$

The speed of the pen is calculated as:

$$\text{speed}(i) = \sqrt{dy_i^2 + dx_i^2} \quad \forall i = 2, 3, \ldots, N$$
Position, velocities, acceleration, pressure and direction of movement are some of the most common features to use \cite{7, 8}. The verification systems may calculate many other features not mentioned here considering that there are systems using a significantly larger set of features given the same input data \cite{8, 12}.

### 2.2 Dynamic Time Warping

A dynamic programming way of calculating similarity given two time sequences is called Dynamic Time Warping (DTW). DTW is used to match data with non-linear variations as close as possible. \cite{9}

DTW can be specified as follows:

There are two sequences: $P = p_1, p_2, \ldots, p_N$ and $Q = q_1, q_2, \ldots, q_M$ ($N$ and $M$ are the number of samples in the $P$ and $Q$ sequences). The sum of the minimum absolute differences from start $\Delta(p_1, q_1)$ to element $\Delta(p_N, q_M)$ is stored in a $N$-by-$M$ matrix $\Delta$. The difference between the sequences element $(p_i, q_j)$ ($1 \leq i \leq n, 1 \leq j \leq M$) is $\text{diff}(i, j) = (p_i - q_j)^2$ with an appropriate minus operator depending on what context this algorithm is used.

Then the DTW algorithm is as follows:

\[
\Delta(i, j) = \min \left( \Delta(i - 1, j) + \epsilon, \Delta(i, j - 1) + \epsilon, \Delta(i - 1, j - 1) + \text{diff}(i, j) \right) \tag{4}
\]

For $1 \leq i \leq N, 1 \leq j \leq M$, $\Delta(0, 0) = 0$, $\Delta(i, 0) = \infty$, $\Delta(0, j) = \infty$.

The local difference between the two sequences is added to the sum of the shortest path up to that cell. It is common to restrict the warping amount to abort the comparison if the path wanders too far away from the diagonal center and add a minor cost ($\epsilon$) for warping. Comparison of two signals can be seen in Figure\textsuperscript{2}. 

Figure 1: Visual clarification of Azimuth $\phi$, the angle the projected vector and a reference vector, and the Altitude $\theta$, the angle between the pen and the plane.
Figure 2: The DTW takes the path that minimizes the difference between the two sequences. Two equal sequences would result in diagonal steps always. The blue signal needs two cells to represent the first dip in the green signal to result in the minimal difference.
Figure 3: A, B, C, D are time-series with one input signal and one reference signal with the lines between indicating the warping of the Dynamic Time Warp. A, B is dependent and C, D are the same signals as A, B but with independent warping. One line is highlighted to show that there is a significant difference between dependent and independent.

2.2.1 Multidimensional Dynamic Time Warp

When several features are being compared in a DTW it can be done several ways. Each feature is a dimension and the dimension can be considered dependent on each other or independent. With the dependent approach, the path will warp in the same way for each feature meaning that while a step can be suboptimal for one feature, averaging the cost for all features will be optimal for each step. With the independent approach, each feature will warp optimally for that feature producing an equal or shorter path than the dependent way, but not necessarily more accurate considering all the features. [13]

As seen in Figure 3, the warping for all dependent features are the same and the warping for the independent are optimal for each feature. The pink highlighted warping line is just to highlight that the warping line for A, B have the same horizontal distance while C, D warps differently compared to each other. When features can be warped independently, the ratio between horizontal and vertical position is not preserved, as the case is for dependent features.

With the Equation 4 formalizing the single-dimension DTW, the multidimensional case is covered below. The \( P \) and \( Q \) used here are multidimensional sequences, each dimension has
length $N$ and $M$ with $F$ number of dimensions (or features). $P_f = p_{1f}, p_{2f}, \ldots, p_{Nf}$ is describing dimension $f$ of sequence $P$. DTW$_I$ features are independent and DTW$_D$ features are dependent. DTW$_D$ is essentially the same as the single-dimension DTW but the diff-function needs to be redefined to Equation 5. DTW$_I$ is defined with several calls of the single-dimension DTW as in Equation 6.

$$\text{diff}(p_i, q_j) = \sum_{f=1}^{F} (p_{i,f} - q_{j,f})^2$$  \hfill (5)

$$\text{DTW}_I(P, Q) = \sum_{f=1}^{F} \text{DTW}(P_f, Q_f)$$  \hfill (6)

### 2.2.2 Dynamic Time Warping for On-line Signature Verification

The sequence in the context of on-line signature verification is the set of features including derived features. The feature comparison method varies greatly but it seems that the most common methods are Euclidean distance and Mahalanobis distance\[7\].

An input signature is compared to either a reference average signature that includes how the different features deviate or to all of the reference signatures. It is fairly common that the input value is compared to the minimum, the average and the maximum of all dissimilarity values and then these values are compared to some threshold when judging the authenticity; the threshold can either be an individual value depending on the reference signatures for the specific signer or a global value for all signatures.

### 2.3 Hidden Markov Model

Hidden Markov Models (HMM) is a robust statistical pattern recognition method that is used to solve a wide range of real-world problems\[14, 15\]. HMM is a set of hidden states and a set of observable symbols. Each state has some probability to transition in to some other state, but the actual transitions and the current state is unknown (hidden); a symbol observation is not sufficient to deduce the state. Each state is associated with a subset of the observable outputs and has some probability for its output to be the specific observable symbol.

Formally the HMM can be described as follows:

There are $n$ states in the model and $m$ different possible observations; the states are denoted $S = \{S_1, S_2, \ldots, S_n\}$ and the observations $O = \{O_1, O_2, \ldots, O_m\}$.

The state transition probability matrix $A = a_{ij}$ is a $n \times n$ matrix, describing $a_{ij} = P(\text{probability to transition to } S_j \text{ when current state is } S_i)$.

The observation probability matrix $B = b_j(k)$ is a $n \times m$ matrix, describing $b_j(k) = P(\text{probability for observation } k \text{ when current state is } S_j)$.

The initial state distribution $\pi = \{P(S_1), P(S_2), \ldots, P(S_n)\}$ represents the probability for the initial state to be $S_i$ for $1 \leq i \leq n$. \[15\]

The model $\lambda$ can then be described as:

$$\lambda = (A, B, \pi)$$  \hfill (7)

The signatures are usually represented as a matrix letting each column represent each sample of a feature. The columns are zero-mean normalized. A system can be constructed with observation probabilities $b_j(o)$ as mixtures of $X$ multivariate Gaussian densities (with $X = 2^x$ for some number $x$). After initialization and initial estimation\[14\], re-estimation steps are performed with
Baum-Welch equations. The verification of input signatures are done using the Viterbi algorithm.

2.4 Accuracy Description

How well an on-line signature verification system performs can be evaluated in several ways. The most common way of describing how well the system performs is giving the Equal Error Rate (EER) for the method. The EER is the value of when False Acceptance Rate (FAR) of forgeries and False Rejection Rate (FRR) of genuine signatures is equal. Precision and Recall are other terms commonly used for other applications, but not commonly used with on-line signature verification.

Precision is a measure of how many of the items classified as some class A was correctly classified (actually A), the inverse of precision is how many of the items classified as A was incorrectly classified (not A). Recall is a measure of how many of the tested As were classified as As, the inverse is how many of the tested As were classified as non-As.

False Rejection Rate (FRR) and False Acceptance Rate (FAR) are both the inverse of recall when applied to forgeries and genuine, respectively.

Most commonly the EER is derived from a Receiver Operating Characteristic (ROC), a plot of True Positive (other word for recall) against False Positive as the threshold for classifying A as A is varied. The EER is the point where True Positive is equal to False Positive, when used for plotting data from On-line Signature Verification, EER is the point closest to origin. Figure 4 shows a ROC curve on this project’s results when using all the features.

![ROC Curve](image)

Figure 4: The ROC curve for this project’s result using all the features (8,7,6,5,4,3,2,1). Skilled is against skilled forgeries, random is against random forgeries.

2.5 Signature Verification Contest 2004

This section contains results of the SVC2004 competition and research using the SVC2004 databases in the same way as the SVC2004 competition as input to get comparable results.

The accessibility of comparable result after the first Signature Verification Competition (SVC2004) improved by its open database to test verifications systems against. Several
other databases have appeared since [26, 27] and with these databases it is possible for the EER to hold some information. Without databases and standardized evaluation methods, it is hard to compare different systems. When looking at two different methods using different data sets it is more often than not a comparison of the data set and how well the method is adapted to the set. With larger data sets it is mostly DTW or HMM based methods that perform well, while with smaller data sets it is possible for a wider variety of methods to perform well [7].

SVC2004 contains two tasks. In both tasks each team’s system was first provided with 5 genuine reference signatures (from the first session), 10 genuine test signatures (from the second session), 20 skilled forgeries and 20 random forgeries; whenever randomness was used, it had the same random seed to produce the same randomization for each team. The SVC2004 judging system expected each team’s system to output a similarity score between 0 and 1. In the first task the input signatures features consisted only of $x, y, t$; in the second task the input signatures had all the common features, $x, y, p, \phi, \theta, t$ [17].

2.5.1 Signature Data Collection

The evaluation set used the same data for Task 1 and Task 2 but with pressure, azimuth and altitude omitted while evaluating Task 1. The competition participants had access to one Task 1 set of 40 signatures and one set of Task 2 set of 40 signatures that were different from each other and from the evaluation set.

The evaluation database consists of 60 sets of signatures. Each set has 20 genuine signatures and 20 skilled forgeries produced in two sessions with at least one week between the sessions. 10 genuine signatures were provided the first session and 10 genuine signature and four skilled forgeries for five other contributors the second session. The contributors could replay the signature writing process on a computer screen and practice the forgery a few times before the data collection began. The released developer sets were captured in a similar way.

The contributors were advised not to use their real signatures and to design a new signature for privacy reasons. The signatures were acquired using a tablet that gave the signers no visual feedback during the signing. The signatures were mostly in either English or Chinese. The data were captured at a stable frequency of 100 Hz and were not re-sampled in any way by the hosts of SVC2004. [17]

2.5.2 Competition Results

The competition ended up with 15 teams for task 1 and 12 teams for task 2. 7 teams managed to successfully complete both tasks but one team wanted their result omitted giving 6 public results on both tasks with a total of 8 programs. The result of interest comes from the 6 teams with public results that completed both tasks. The teams are found in table 1. Team 19 had three different programs and will have the identifier $a, b, c$ after their team id to specify the program in the SVC2009 result tables.

The result of interest is the Skilled Forgery (SF) Equal Error Rate (EER) for the test set of both task 1 and task 2. Task 1 uses the same data as Task 2, but have the pressure, azimuth and altitude values stripped [17].

As seen in Table 2 the additional data provided in Task 2 gave positive results for 5 out of 8 programs on the skilled forgeries; for the SVC2004 winning team, team 6, the additional data had a 0.05 % negative effect on the EER result.

As seen in Table 3 the additional data provided in Task 2 gave positive results for 6 out of 8 programs on the random forgeries; for the SVC2004 winning team, team 6, the additional data had a 0.28 % positive effect on the EER result.
Table 1: The teams with public data that successfully completed both task 1 and 2 of SVC2004

<table>
<thead>
<tr>
<th>Team ID</th>
<th>Institution</th>
<th>Country</th>
<th>Member(s)</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>anonymous</td>
<td></td>
<td>A. Kholmatov and B. Yanikoglu</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Sabanci University</td>
<td>Turkey</td>
<td>A. Kholmatov and B. Yanikoglu</td>
<td>DTW [10]</td>
</tr>
<tr>
<td>14</td>
<td>anonymous</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>anonymous</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>anonymous</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Laboratory, Universidad</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Politecnica de Madrid</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: The EER result of skilled forgeries on SVC2004. $\Delta$ is the difference between Task 1 and Task 2.

<table>
<thead>
<tr>
<th>Team ID</th>
<th>Task1(%)</th>
<th>Task2(%)</th>
<th>$\Delta$(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>16.22</td>
<td>16.34</td>
<td>-0.12</td>
</tr>
<tr>
<td>6</td>
<td>2.84</td>
<td>2.89</td>
<td>-0.05</td>
</tr>
<tr>
<td>14</td>
<td>8.77</td>
<td>8.02</td>
<td>0.75</td>
</tr>
<tr>
<td>17</td>
<td>11.85</td>
<td>12.51</td>
<td>-0.66</td>
</tr>
<tr>
<td>18</td>
<td>11.81</td>
<td>11.54</td>
<td>0.27</td>
</tr>
<tr>
<td>19a</td>
<td>6.88</td>
<td>5.91</td>
<td>0.97</td>
</tr>
<tr>
<td>19b</td>
<td>5.88</td>
<td>5.01</td>
<td>0.87</td>
</tr>
<tr>
<td>19c</td>
<td>6.05</td>
<td>5.13</td>
<td>0.92</td>
</tr>
</tbody>
</table>

Table 3: The EER result of random forgeries on SVC2004. $\Delta$ is the difference between Task 1 and Task 2.

<table>
<thead>
<tr>
<th>Team ID</th>
<th>Task1(%)</th>
<th>Task2(%)</th>
<th>$\Delta$(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>6.89</td>
<td>6.17</td>
<td>0.72</td>
</tr>
<tr>
<td>6</td>
<td>2.79</td>
<td>2.51</td>
<td>0.28</td>
</tr>
<tr>
<td>14</td>
<td>2.93</td>
<td>5.19</td>
<td>-2.26</td>
</tr>
<tr>
<td>17</td>
<td>3.83</td>
<td>3.47</td>
<td>0.36</td>
</tr>
<tr>
<td>18</td>
<td>4.39</td>
<td>4.89</td>
<td>-0.50</td>
</tr>
<tr>
<td>19a</td>
<td>2.18</td>
<td>1.70</td>
<td>0.48</td>
</tr>
<tr>
<td>19b</td>
<td>2.12</td>
<td>1.77</td>
<td>0.35</td>
</tr>
<tr>
<td>19c</td>
<td>2.13</td>
<td>1.79</td>
<td>0.34</td>
</tr>
</tbody>
</table>

2.5.3 Research using the Database

There are many researchers using the SVC2004 database to test their on-line signature verification system. Table 4 shows the result of some on-line signature verification systems using SVC2004 databases.
Table 4: The EER test result of research SVC2004 database testing skilled forgeries. The Task # is 1 if the Task 1 data set is used, 2 if Task 2 data set is used. (*) only $x,y,t$ were used, but the report say Task 2. (**) Marks that they used 10 training signatures instead of the SVC2004 standard of 5.

<table>
<thead>
<tr>
<th>Source</th>
<th>Method</th>
<th>EER(%)</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. Ahrary and S. Kamata (2009) [18]</td>
<td>Hilbert Scanning Pattern</td>
<td>4.0</td>
<td>1</td>
</tr>
<tr>
<td>A. Ahrary and S. Kamata (2009) [18]</td>
<td>Hilbert Scanning Pattern</td>
<td>2.2</td>
<td>2</td>
</tr>
<tr>
<td>L. Hu and Y. Wang (2007) [19]</td>
<td>Enhanced DTW</td>
<td>4.00</td>
<td>1*</td>
</tr>
<tr>
<td>J. Fierrez, J. Ortega-Garcia, D. Ramos and J. Gonzalez-Rodriguez (2007) [22]</td>
<td>HMM</td>
<td>0.78</td>
<td>2**</td>
</tr>
</tbody>
</table>

2.6 BioSecure Signature Evaluation Campaign 2009

BioSecure Signature Evaluation Campaign 2009 (BSEC2009) was a competition that used three different data sets, one with $x,y,t$, one with $x,y,p,t$ and one with $x,y,p,\phi,\theta,t$.

The testing protocol was similarly to SVC2004. For each tested individual the system receives 5 genuine reference signatures then the test is performed with 10 other genuine signatures, 20 skilled forgeries and 30 random forgeries. The reference signatures were only taken from the first session.

2.6.1 Signature Data Collection

There were two data sets used in BSEC2009. Data Set 3 (DS3) was acquired with a PDA of the model HP iPAQ hx2790. Data Set 2 (DS2) was acquired with a digitizing tablet of the model WACOM INTUOS 3 A6. Both data sets contain signatures from the same 382 people.

The participants had access to a subset of both data sets, each containing the data from 50 people. The complete sets are referred to as DS2-382 and DS3-382, the data set number and the number of signatures in the data set. The developers subset will be referred to as DS2-50 and DS3-50, the data set number and the number of signatures in the subset.

Both data sets were acquired on two sessions. Each session the donor was asked to alternately sign three times five genuine signatures and two times five skilled forgeries.

The DS3 was collected while the donor was standing up, keeping the PDA in the persons hand. The skilled forgeries of DS3 were collected by first animate a genuine signature allowing the forger to see the dynamics of the signature, then sign while an image of the signature was on the screen allowing the skilled forgery to have both a good shape and a good dynamics. The sessions of DS3 were around 5 weeks apart. DS3 was sampled with an event based method.
producing close to, but not actually, 100 Hz but the data used in both the development subset and test data were re-sampled to 100 Hz. Each sample in DS3 contained $x, y, t$.

The DS2 was collected while the donor was sitting down, signing the digitizer covered by a paper with an inking pen. The skilled forgeries were collected by showing a genuine signature allowing the forger to train reproducing the image of the genuine signature. The two sessions of DS2 were around 2 weeks apart. DS2 was captured in a stable 100 Hz frequency and not resampled in any other way. Each sample in DS2 contained $x, y, p, \phi, \theta, t$. [23]

2.6.2 Competition Results

Only the results using DS2 will be presented here because it was evaluated in three steps:

1. only coordinates
2. coordinates and pressure
3. coordinates, pressure and pen inclination

For the full results of BSEC2009, see its results document [24].

12 of the systems, including the reference system, had data for step 1 and step 2 above; 6 of the systems had data for all of the steps above. Only the data testing session 1 will be presented here.

A short overview of the systems participating in BSEC2009 can be found in Table 5 [23].

As seen in Table 6, the pressure data added in Step 2 gave a significant positive results for 11 out of 12 programs on the skilled forgeries. The additional information about pen inclination had a negative impact for 3 out of 6 systems and a positive impact for 1 system. The pressure information improved the best performing system for skilled forgeries, system 12.

As seen in Table 7, the pressure data added in Step 2 had a positive impact for 10 out of 12 programs on the random forgeries. The additional information about pen inclination had a negative impact for 4 out of 6 systems and a positive impact for 1 system. The pressure information improved the best performing system for random forgeries, system 8.
Table 5: Short description of each system with information available participating in BioSecure Signature Evaluation Campaign 2009

<table>
<thead>
<tr>
<th>Sys ID</th>
<th>Approach</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Biometric Dispersion Match</td>
<td>Discrete Cosine Transform</td>
</tr>
<tr>
<td>2</td>
<td>Euclidian distance</td>
<td>16 local features extracted from pen coordinates, individual scoring</td>
</tr>
<tr>
<td>3</td>
<td>DTW</td>
<td>- Information Missing -</td>
</tr>
<tr>
<td>4</td>
<td>DTW</td>
<td>Local features: Coordinates, pen direction, velocity. Score computation combining perceptions, using adaboost algorithm</td>
</tr>
<tr>
<td>6</td>
<td>DTW</td>
<td>Local features: time derivative of pen coordinates. Normalization with data from reference signatures and data derived from MCYT-100</td>
</tr>
<tr>
<td>8</td>
<td>DTW tuned for random forgeries</td>
<td>27 local features. Feature selection via Sequential Forward Floating Selection. Score computation by min and mean distance of the test to the reference signatures.</td>
</tr>
<tr>
<td>9</td>
<td>DTW tuned for skilled forgeries</td>
<td>Same as system 8, optimized for skilled forgeries.</td>
</tr>
<tr>
<td>10</td>
<td>HMM tuned for skilled forgeries</td>
<td>27 local features. Feature selection via Sequential Forward Floating Selection. Likelihood score computation.</td>
</tr>
<tr>
<td>11</td>
<td>Mahalanobis distance</td>
<td>100 global features. Score computation using Mahalanobis distance</td>
</tr>
<tr>
<td>12</td>
<td>Fusion</td>
<td>Weighted fusion of system 8, 9, 10, 11.</td>
</tr>
<tr>
<td>13</td>
<td>DTW</td>
<td>- Information Missing -</td>
</tr>
<tr>
<td>ref</td>
<td>HMM</td>
<td>25 local features. Score computation by likelihood by Viterbi algorithm.</td>
</tr>
</tbody>
</table>

Table 6: The EER result of skilled forgeries on BSEC2009. $\Delta_{1,2}$ is the difference between Step 1 and Step 2.

<table>
<thead>
<tr>
<th>Sys ID</th>
<th>Step1(%)</th>
<th>Step2(%)</th>
<th>$\Delta_{1,2}$(%)</th>
<th>Step3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8.71</td>
<td>4.03</td>
<td>4.68</td>
<td>N/A</td>
</tr>
<tr>
<td>2</td>
<td>7.38</td>
<td>4.50</td>
<td>3.88</td>
<td>4.52</td>
</tr>
<tr>
<td>3</td>
<td>18.32</td>
<td>13.69</td>
<td>4.63</td>
<td>13.41</td>
</tr>
<tr>
<td>4</td>
<td>6.37</td>
<td>2.76</td>
<td>3.61</td>
<td>3.02</td>
</tr>
<tr>
<td>6</td>
<td>5.69</td>
<td>2.19</td>
<td>3.50</td>
<td>2.19</td>
</tr>
<tr>
<td>8</td>
<td>10.40</td>
<td>3.26</td>
<td>7.14</td>
<td>N/A</td>
</tr>
<tr>
<td>9</td>
<td>5.24</td>
<td>2.38</td>
<td>2.86</td>
<td>N/A</td>
</tr>
<tr>
<td>10</td>
<td>24.79</td>
<td>27.76</td>
<td>-2.97</td>
<td>N/A</td>
</tr>
<tr>
<td>11</td>
<td>10.49</td>
<td>5.90</td>
<td>4.59</td>
<td>N/A</td>
</tr>
<tr>
<td>12</td>
<td>4.93</td>
<td>1.71</td>
<td>3.22</td>
<td>N/A</td>
</tr>
<tr>
<td>13</td>
<td>5.98</td>
<td>2.84</td>
<td>3.14</td>
<td>17.94</td>
</tr>
<tr>
<td>ref</td>
<td>11.27</td>
<td>4.07</td>
<td>7.20</td>
<td>4.07</td>
</tr>
</tbody>
</table>
Table 7: The EER result of random forgeries on BSEC2009. $\Delta_{1,2}$ is the difference between Step 1 and Step 2.

<table>
<thead>
<tr>
<th>Sys ID</th>
<th>Step1(%)</th>
<th>Step2(%)</th>
<th>$\Delta_{1,2}$ (%)</th>
<th>Step3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.22</td>
<td>1.70</td>
<td>0.52</td>
<td>N/A</td>
</tr>
<tr>
<td>2</td>
<td>1.85</td>
<td>1.96</td>
<td>−0.11</td>
<td>1.91</td>
</tr>
<tr>
<td>3</td>
<td>8.36</td>
<td>8.62</td>
<td>−0.26</td>
<td>8.63</td>
</tr>
<tr>
<td>4</td>
<td>2.00</td>
<td>1.33</td>
<td>0.67</td>
<td>1.49</td>
</tr>
<tr>
<td>6</td>
<td>1.50</td>
<td>0.97</td>
<td>0.53</td>
<td>0.97</td>
</tr>
<tr>
<td>8</td>
<td>0.70</td>
<td>0.42</td>
<td>0.28</td>
<td>N/A</td>
</tr>
<tr>
<td>9</td>
<td>2.09</td>
<td>1.17</td>
<td>0.92</td>
<td>N/A</td>
</tr>
<tr>
<td>10</td>
<td>27.29</td>
<td>20.51</td>
<td>6.78</td>
<td>N/A</td>
</tr>
<tr>
<td>11</td>
<td>2.93</td>
<td>2.02</td>
<td>0.91</td>
<td>N/A</td>
</tr>
<tr>
<td>12</td>
<td>1.41</td>
<td>0.65</td>
<td>0.76</td>
<td>N/A</td>
</tr>
<tr>
<td>13</td>
<td>1.44</td>
<td>1.38</td>
<td>0.06</td>
<td>24.06</td>
</tr>
<tr>
<td>ref</td>
<td>4.80</td>
<td>1.65</td>
<td>3.15</td>
<td>2.39</td>
</tr>
</tbody>
</table>

2.7 Sabanci University Signature Database

Sabanci University Signature database (SUSig) consist of two parts, a visual sub-corpus from 2009 and a blind sub-corpus from 2005. The visual sub-corpus displayed the signature whilst it was written on the LCD display; the blind sub-corpus had no visual feedback for the signer. The visual sub-corpus were collected with 100 Hz sample rate using a pressure sensitive touch-pad with registering 128 levels of pressure.

Each sub-corpus of SUSig were collected in a similar manner. 100 signers (29 women, 71 men) between 21 and 52 years old donated their signatures. Each signer supplied 10 signatures per session for two sessions with approximately one week between the sessions. During the second session, each signer was shown the signature of another person and were able to train several times to forge that signature and finally supply 5 skilled forgeries of that signature. Finally 5 highly skilled forgeries are collected on each person.

Each sample in the data base contains $x,y$ coordinates, time stamp and pressure level [27].

2.8 Summary

Considering the result on both SVC2004 and BSEC2009 it seems that the azimuth and altitude is not needed for a well performing system. The pressure information however could add some accuracy, specially against skilled forgeries, but is not vital if the system is used as a signature verification assistant with a human supervisor comparing as well.

The winning system of SVC2004 performed significantly better than the other teams for skilled forgeries but worse than some teams on random forgeries. The winning team of SVC2004 used a DTW [10, 17]. However the system that performed best on random forgeries used a HMM [13] [17].

The best performing system of BSEC2009 on skilled forgeries was a Fusion of several systems including the second best performing system, a DTW tuned for skilled forgeries. The best performing system of BSEC2009 on random forgeries was a DTW tuned for random forgeries followed by the same Fusion system as the skilled forgeries winner. The DTW tuned for random forgeries is also part of that Fusion system.

HMM systems can perform well, specially for random forgeries, but the winners of SVC2004 and BSEC2009 used DTW or had well performing DTWs in fusion with HMM and global features
compared with Mahalanobis distance.

Given these facts, the approach chosen for this project is DTW. The features will be considered independently, the distance function will be Euclidean distance and the features will be compared to an individual threshold for that feature. The individual thresholds are depending on how the reference signatures behave for the considered feature. The input signals that will be explored more in-depth are coordinates, pressure and time-stamp. The azimuth and altitude will not be explored because most modern tablet do not have the possibility to capture them and most systems do not improve its accuracy when the azimuth and altitude information are used. The first order derivatives are processed for each signal (velocities) and the angle of the stroke (the $\sin$ and $\cos$).

The system used in this project is designed so it is easy to enable and disable features. The reason for developing a system is to gain information about how pressure data can impact the result of a system when not changing anything else.
3 Method

The input is collected through a file reader or an Android app saving the MotionEvent data. The coordinates and the pressure data are normalized and additional feature are extracted. First the reference set is saved and processed then the processed set is compared to the input signature. The comparison produces a value on how close the input signature is to the reference set and based on that value a decision is made if it is a genuine signature. If database values are being used the signatures are loaded from files.

3.1 Preprocessing and Feature Extraction

The preprocessing calculates all the max, min, avg and sum values for the input signals \(x, y, p, t\). With this data, the \(x, y\) are centered around the average and normalized to values between \(-1\) and \(1\) with \(0\) at the average value. The pressure is normalized to values between \(0.0\) and \(1.0\). The timestamps are preprocessed to start at \(0\) and the unit is milliseconds.

The velocities are calculated as Equation 1 for all input signals (except for the time). \(\sin\) and \(\cos\) is calculated as described in Equation 2 and 3.

Worth noting is that no resampling is done. It was expected that the Android sample rate would be unstable and needed a resampling step in order to get accurate estimates for the derivative, but the sample rate proved to be stable so no resampling is done for this reason. Removing duplicate points is a classic resampling but it result in information loss that hurt accuracy so it is not done. After the feature extraction, the features available can be found in Table 8.

Table 8: The available features, the index it is being referred to and their values. The listed interval is the edge values. Some of the features are never close their edge values.

<table>
<thead>
<tr>
<th>Feature Index</th>
<th>Feature</th>
<th>Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Horizontal Position</td>
<td>-1 .. 1</td>
</tr>
<tr>
<td>2</td>
<td>Horizontal Velocity</td>
<td>-1 .. 1</td>
</tr>
<tr>
<td>3</td>
<td>Vertical Position</td>
<td>-1 .. 1</td>
</tr>
<tr>
<td>4</td>
<td>Vertical Velocity</td>
<td>-1 .. 1</td>
</tr>
<tr>
<td>5</td>
<td>Pressure Value</td>
<td>0 .. 1</td>
</tr>
<tr>
<td>6</td>
<td>Pressure Velocity</td>
<td>-1 .. 1</td>
</tr>
<tr>
<td>7</td>
<td>Sin</td>
<td>0 .. (2\pi)</td>
</tr>
<tr>
<td>8</td>
<td>Cos</td>
<td>0 .. (2\pi)</td>
</tr>
<tr>
<td>Global 1</td>
<td>Total Writing Time</td>
<td>0 .. (\infty)</td>
</tr>
</tbody>
</table>

3.2 Distance Calculation

The distance between two signatures is calculated using a multidimensional dynamic time warp described in section 2.2.1. The difference sum is calculated as described in Equation 6 with each DTW is calculated using Equation 4 with the diff-function described below in Equation 8.

\[
diff(a, b) = \min(0, (a - b)^2 - \gamma)
\]

The \(\gamma\) in Equation 8 is a very small value to reduce noise and results in setting the difference to zero if \(a\) and \(b\) are closer than \(\sqrt{\gamma}\).

The single dimension DTW used is defined in Equation 4 with the diff-function of Equation 8. The \(\epsilon\) (warping penalty) was selected using a small subset of data.
The warping amount is restricted to be at most 25% of the length of the shortest signal and is visualised in Figure 2 only the open (no diagonal line) cells in the figure are available. The unavailable cells have infinity in them and will therefore never be part of the shortest path. The warping amount is calculated with Equation 9 with $L_1, L_2$ being the number of samples in the signals currently being compared.

$$\text{window}(L_1, L_2) = 1 + \max(\abs{L_1 - L_2}, \min(L_1, L_2) \times 0.25)$$  (9)

The distance result is the sum of the difference of each feature for each sample in the signature.

### 3.3 Reference Set

The reference set is the set of signatures that the input signature will be compared to. First $n$ signatures are inserted to a set and each signature is compared to all other signatures to get min, max, avg distances and total writing time to the other signatures. The min, max values (from each signature to each other signature in the reference set) are averaged to get the average minimum and average maximum distances (avg min, avg max). The signature with the smallest avg distance is called template signature (or closest signature) for that set.

With the template signature found, the distance from each signature to the template signature is calculated and averaged to get avg tdist.

After this, the reference set has min, max, avg total writing time and avg min, avg max, avg tdist distances. Each signature is then verified (though not verified against itself) and min, max, avg verification values are saved in the reference set. The actual verification is explained in Section 3.4.

### 3.4 Verification

When verifying an input signature, the distance is calculated from each reference signature to the input signature ($s_{in}$) and min, max distance ($d_{min}, d_{max}$) is stored as well as the distance to the template signature ($d_{template}$). The time difference is calculated with Equation 10 where $r_{t_{min}}, r_{t_{max}}, r_{t_{avg}}$ is the reference set’s total writing time.

$$\text{timediff}(t) = \begin{cases} 
2 \times (t - r_{t_{max}})/r_{t_{avg}} & \text{if } t > r_{t_{max}} \\
(r_{t_{min}} - t)/r_{t_{avg}} & \text{if } t < r_{t_{min}} \\
0 & \text{otherwise}
\end{cases}$$  (10)

The verification value is calculated with Equation 11. The avg min, avg max, avg tdist are the reference set distance values as defined in Section 3.3.

$$\text{verify}(s_{in}, rs) = (d_{min} - \text{avg min})^2 + (d_{max} - \text{avg max})^2 + (d_{template} - \text{avg tdist})^2 + \text{timediff}(\text{time}(s_{in}))$$  (11)

When preparing the reference set for input signatures, a reference set value is calculated by comparing each reference signature to each other reference signature. This is done by letting one signature from the reference set take the place of $s_{in}$ and temporarily remove it from the reference set. The outputted reference set values ($rsv$) from Equation 11 are then used to determine the genuine breakpoint.

For verification of non-reference-set input signature, the verification value is compared against the reference set verification value to decide if the input signature should be classified as genuine
or forgery. If the verification value is smaller than the genuine-breakpoint, it is considered genuine. The genuine breakpoint is calculated using Equation 12 with \( i = 6 \). During evaluation, several is are tested in the equation to get enough data to calculate Equal Error Rate (EER).

\[
bp(i, rsv_{\text{min}}, rsv_{\text{max}}) = \frac{i \times rsv_{\text{max}}}{6} + \frac{rsv_{\text{min}}}{2}
\]  

(12)

### 3.5 Evaluation

The system is evaluated using the SUSig cross session evaluation method (similar to SVC2004 evaluation method) on the SUSig data set (Information about SUSig are found in subsection 2.7). 5 random genuine signatures from session 1 are used as reference, 10 (all) genuine from session 2 are used to test genuine verification, 10 (all) skilled forgeries to test the genuine vs forgery and 20 random other users genuine (session 2) signatures to test against some completely different signatures. Note that the random forgery test has twice the number of input signatures than the other tests.

Each test is done 10 times to get a representative result with all the randomness involved. Each evaluation uses the same random seed to ensure that each combination of features for each breakpoint \((i)\) uses the same random sets of signatures for the tests.

### 3.6 Calculation of Equal Error Rate

Comparing skilled and random forgeries against the reference set is done to calculate False Acceptance Rate (FAR) for skilled and random forgeries; comparing the genuine signatures against the reference set is done to calculate False Acceptance Rate (FAR). The FRR and FAR are calculated for several breakpoints (using Equation 12) and the EER is calculated by finding the point where FRR is equal to FAR. Finding the point of EER can be done by plotting FRR and FAR against the breakpoint changing variable and find the intersection or plotting FRR against FAR and finding the point where FRR is equal to FAR.

The False Acceptance Rate is reported separately for skilled forgeries and random forgeries because the forgeries are acquired differently and to see if there is a difference in how the signature is produced. The False Rejection Rate of genuine signatures is independent of the FAR, but needed to calculate EER.
4 Result

In this section, several graphs and tables with the data of this algorithm will be brought up. The database used is SUSig [27] visual sub-corpus because it used a tablet with visual feedback, much like one would expect an Android tablet to behave.

The features and their indexes are found in Table 8. The features will be referred to by their index.

Figure 5: Breakpoint \( i \) on x-axis, % classified on y-axis. For a better logarithmic function for FAR-random, the first 4 values are removed.

Figure 5 is to show how most data behaved when the data were growing. The features used are shown in the top of the figure, 8, 7, 4, 3, 2, 1, and reading from Table 8 it is possible to read that the resulting features are cos, sin, vertical velocity, vertical position, horizontal velocity and horizontal position. Figure 6 uses the same features but the function fitted for FAR-random is a power function on the form \( a \times x^b \) while Figure 5 the FAR-random is a logarithmic function on the form \( a \times \ln(x) + b \) for some constants \( a, b \). The FRR shrinks with a negative power, FAR-skilled linearly and FAR-random grows with a positive power up to a certain point, after that it stops growing like a logarithmic function. Reading the \( y \)-value from the intersection of FRR and FAR-random gives EER-R (Equal Error Rate - Random forgeries). For the example used here, the EER-R is 5.83% and EER-S is 3.24%. More figures can be found in Appendix B.
Figure 6: Breakpoint $i$ on x-axis, % classified on y-axis. For a better power function, most values outside the shown range is ignored when fitting the function to make the intersection between FAR-random and FRR fit the data as good as possible.

4.1 Without Pressure

In this subsection, the results of feature sets without pressure (or features derived from pressure) will be presented. Without any local features the algorithm either accepts all input as genuine signatures or rejects all input as forgeries, i.e. the EER is 50%.

Table 9: Combination of features without pressure. EER-R is the Equal Error Rate for Random forgeries, EER-S is the Equal Error Rate for Skilled forgeries. Only top 5 combination of features and the featureless results are presented here.

<table>
<thead>
<tr>
<th>Features</th>
<th>EER − R(%)</th>
<th>EER − S(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>50.00</td>
<td>50.00</td>
</tr>
<tr>
<td>8,3,2,1</td>
<td>5.39</td>
<td>3.66</td>
</tr>
<tr>
<td>8,4,3,2,1</td>
<td>5.47</td>
<td>3.64</td>
</tr>
<tr>
<td>7,4,3,2,1</td>
<td>5.58</td>
<td>3.46</td>
</tr>
<tr>
<td>8,7,4,3,2,1</td>
<td>5.83</td>
<td>3.24</td>
</tr>
<tr>
<td>4,3,2,1</td>
<td>7.20</td>
<td>4.65</td>
</tr>
</tbody>
</table>

As seen in Table 9, the lowest EER-R and EER-S result are two separate feature sets. Lowest EER-R of 5.39% is the combination of cos, vertical position, horizontal velocity and horizontal position and can be seen in Figure 7. Lowest EER-S of 3.24% is the combination of cos, sin, vertical velocity, vertical position, horizontal velocity and horizontal position and can be seen in Figure 8.
Figure 7: Breakpoint $i$ on x-axis, % classified on y-axis.

Figure 8: Breakpoint $i$ on x-axis, % classified on y-axis.
4.2 With Pressure

In this subsection, the results of feature sets containing pressure (or pressure velocity) in combination with other features will be presented.

Table 10: Combination of features containing pressure. EER-R is the Equal Error Rate for Random forgeries, EER-S is the Equal Error Rate for Skilled forgeries. Only top 10 combinations is featured here.

<table>
<thead>
<tr>
<th>Features</th>
<th>EER – R(%)</th>
<th>EER – S(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7,6,5,3,2,1</td>
<td>5.19</td>
<td>3.01</td>
</tr>
<tr>
<td>8,7,6,5,4,3,2,1</td>
<td>5.26</td>
<td>2.88</td>
</tr>
<tr>
<td>8,7,5,3,2,1</td>
<td>5.36</td>
<td>2.84</td>
</tr>
<tr>
<td>8,6,5,4,3,2,1</td>
<td>5.37</td>
<td>3.20</td>
</tr>
<tr>
<td>7,6,5,4,3,2,1</td>
<td>5.37</td>
<td>3.07</td>
</tr>
<tr>
<td>6,5,3,1</td>
<td>5.40</td>
<td>3.48</td>
</tr>
<tr>
<td>8,7,5,4,3,2,1</td>
<td>5.44</td>
<td>2.81</td>
</tr>
<tr>
<td>8,7,6,5,4,3,1</td>
<td>5.44</td>
<td>2.80</td>
</tr>
<tr>
<td>8,7,6,5,3,2,1</td>
<td>5.44</td>
<td>2.80</td>
</tr>
<tr>
<td>8,6,4,3,2,1</td>
<td>5.50</td>
<td>3.57</td>
</tr>
</tbody>
</table>

As seen in Table 10, the set of features giving the lowest EER-R of 5.19% is the set of sin, pressure velocity, pressure value, vertical velocity, vertical position and horizontal position seen in Figure 9. The lowest EER-S of 2.80% are shared among 8, 7, 6, 5, 3, 2, 1 and 8, 7, 6, 5, 4, 3, 1 with the difference between them are: one has the horizontal velocity and the other vertical velocity; the shared features are cos, sin, pressure velocity, pressure value, vertical position and horizontal position. The lowest ERR-S figures can be found in Figure 10 and Figure 11.
Figure 9: Breakpoint $i$ on x-axis, % classified on y-axis.

Figure 10: Breakpoint $i$ on x-axis, % classified on y-axis.
Figure 11: Breakpoint $i$ on x-axis, % classified on y-axis.
5 Discussion

This section will discuss various aspects of this report.

5.1 Sources of Error

Various factors that can have an impact on the result and thereby the conclusion are discussed in this sub-section.

Only one method was tested. In order to give a conclusive answer to the question, several methods have to be evaluated. Due to the scope of this project, only one method was developed and tested. If there were several state-of-the-art methods openly available to the research community, it would have been possible to evaluate many, but that is not the case.

Accepted Difference $\gamma$ from Equation 12 is a constant chosen when testing only on a small subset. It was not possible to test several values of the $\gamma$ value on the whole data set because it would take about two to three weeks of processing to get the optimal values. A low $\gamma$ value would result in a higher difference score in general (more likely to classify as forgery). A high $\gamma$ is lowering the difference score, allowing a higher difference at each sample point (more likely to classify as genuine). This has primarily an effect on noise or smaller (possibly continuous) differences; smaller continuous differences are commonly found in skilled forgeries.

Warp Penalty $\epsilon$ from Equation 4 is a constant chosen when testing only on a small subset. It was not possible to test several values of the $\epsilon$ value on the whole data set because it would take about two to three weeks of processing to get the optimal values. A low (possibly 0) warp penalty $\epsilon$ is allowing signatures with a few meeting points (points where the difference between the two signatures are very close) but otherwise very different to warp from meeting point to meeting point and end up on a small difference making the difference score lower (more likely to classify as genuine). A high warp penalty is punishing warping, possibly to a point so the method instead is more or less an Euclidean point-to-point matching (disallowing warping because it is always worse than the distance) making the difference score higher (less likely to classify as genuine). This has primarily an effect on signatures with meeting points. Note that a signature is required to have several meeting points for this to happen because of the warping window. The warping window only allow warping within 25% of the shortest signature’s samples; warp outside of the warping window cannot happen. How the warping window behaves is defined in Equation 9.

Actual Name doesn’t matter. In other word, the coordinates, velocity, pressure and direction is the only thing that matters, not what is written. The name of the person is never identified. This method is able to discriminate only based on reference set. The reference set does not evolve over time and if the signature changes (possibly over time), the signature could be rejected.

5.2 Ethical Aspects

The verification used in this project uses the preprocessed features to verify input. In other words, the actual (normalized) coordinates are needed to verify the input. This could be compared to storing the passwords of users in plain text. It is possible to recreate all aspects of the signature given the data (except for size because it is normalized, but it does not matter). No security method should store sensitive data in such a format. Even with great investments in computer security, leaks happen from websites and password (usually hashes) are leaked. Passwords can be changed and are hopefully only connected to one service, signatures on the other hand are very hard to change and are used to authenticate a person towards banks and governments.

Not only could the signatures leak, but there is also the question whether companies should have the power to decide whose signature is genuine and being able to recreate a signature to
be used as a forgery should they want to. While not likely to be abused by companies, should a system like this be implemented, the problem is still there.

The consequences of leaking signatures are severe and therefore this method should not be used in large scale. There are however other methods that use statistical data or other derived data that makes it impossible to recreate the original signature given the data. These methods are much more safe to use in large scale and reduce the above mentioned problems.

5.3 Future Work

In order to draw stronger conclusions, several different approaches (HMM, Mahalanobis distance, Support Vector Machines, Vector Quantization, Hilbert Scanning Pattern, Parzen window, different fusion methods) to the problem have to be explored with more features (some methods have over 100 features!), trying all combinations of features for each method. The methods have to be evaluated using the same evaluation methods and the same input signatures.

One problem that exist for to happen is that no research paper on the topic of On-line Signature Verification had released their source code or raw data, making it impossible to reproduce other research project results. Within the field of computer science, it is often easy to share reproducible experiments, but it is not common practice to do so. Without open source, it would be a huge project to test all the different approaches to this problem in order to evaluate them.

A smaller scale project would be to do either of:

- add more features to the method presented in this project.
- test several $\epsilon$ from Equation 4
- test several $\gamma$ from Equation 12
- test the used method on other signature databases.

5.4 Summary of The Results

The feature sets with the lowest EER with and without pressure can be seen in Table 11. The difference between with pressure and without pressure for EER-R is 0.20% and for EER-S it is 0.44%. The difference between the best EER-R and EER-S is 2.39%.

Table 11: The lowest EER results from without pressure and with pressure for both Equal Error Rate - Random forgeries (EER-R) and Equal Error Rate - Skilled forgeries (EER-S).

<table>
<thead>
<tr>
<th>Features</th>
<th>EER − R(%)</th>
<th>EER − S(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8,3,2,1</td>
<td>5.39</td>
<td>3.66</td>
</tr>
<tr>
<td>8,7,4,3,2,1</td>
<td>5.83</td>
<td>3.24</td>
</tr>
<tr>
<td>7,6,5,3,2,1</td>
<td>5.19</td>
<td>3.01</td>
</tr>
<tr>
<td>8,7,6,5,4,3,1</td>
<td>5.44</td>
<td>2.80</td>
</tr>
</tbody>
</table>

The method used is better to separate skilled forgeries from genuine signatures than random signatures from genuine signatures. It could be considered strange that a skilled forgery (made with the forger able to follow an animation of the signature and the opportunity to practice the signature several times) is classified more accurately than when a random genuine signature from another person is inserted, but it is likely that the warp penalty $\epsilon$ is too low. Most western signatures have several common traits such as going from left to right, low total height of the signature and fairly short total signing time (about 1 to 3 seconds). With both horizontal and
vertical traits in common, it is fairly likely for a *meeting point* to occur and only 3-4 needs to occur to get a low enough difference score given a low $\epsilon$ value, depending on how alike the signatures are in general. When exploring why skilled forgeries had a lower EER than random forgeries, the explanation above was the most likely one.

5.5 Conclusion

In the method presented, when used without any pressure related features, the best EER-R was 5.39% and EER-S was 3.24%. With pressure the best result for EER-R was 5.19% and EER-S 2.80%. The benchmark system for SUSig gives an EER of 2.10%. For the method presented, the EER for both Skilled and Random forgeries improved (lowered) slightly when adding pressure related features. However, only one method has been tested and with only that data it is impossible to conclusively state the importance of pressure other than it seems likely to have some impact. When looking at other sources, it seems that the research community is undecided with some projects stating that pressure is unimportant \[10\], while the result of this project and several others finds pressure to improve the result \[18, 19, 22\].

Considering the result of this project and the result of others, it seems that pressure information is not vital, but provide some valuable information that can be used to classify signatures more accurately. The background study concluded that pen inclination is not required for a well preforming system.
References

   An Introduction to Roman Law, Clarenden Law Series, Oxford, 1962, page 256


   Identity Theft: Who’s Using Your Name?, Information and Privacy Commissioner, 1997

   United States Federal Trade Commission - Commission and Staff Report, 2014

   Antalet registrerade UC bedrägerispärvar ökade med 48 procent 2014!, mynewsdesk, 2015


    Identity authentication using improved online signature verification method, Pattern Recognition Letters 26, ScienceDirect, 2005

    Enhanced DTW Based On-line Signature Verification, Image Processing (ICIP), IEEE, 2009

    Feature extraction based DCT on dynamic signature verification, Scientia Iranica, Vol.19, Iss.6, December 2012


    A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceedings of the IEEE, Vol.77, Iss.2, 1989

SVC2004: First International Signature Verification Competition, International Conference on Biometric Authentication, ICBA, Hong Kong, China, p. 16-22, July 15-17, 2004

[18] A. Ahrary and S. Kamata

[19] L. Hu and Y. Wang


Fusion of Local and Regional Approaches for On-Line Signature Verification, International Workshops on Biometric Recognition Systems, IWBR ’05, 2005


http://www.cse.ust.hk/svc2004/

[26] MCYT-Signature-100 Database
http://atvs.ii.uam.es/mcyt100s.html


29
Appendices

A Raw Data

The following output data was produced by the program.

**FRR** (False Rejection Rate) is the percentage of the genuine test that were rejected. Total FN divided by total tests against genuine \(\frac{\text{Total FN}}{9400}\).

**FAR-S** (False Acceptance Rate for Skilled forgeries) is the percentage of the skilled forgeries that were accepted as genuine. Total FPS divided by total tests against skilled forgeries \(\frac{\text{Total FPS}}{9400}\).

**FAR-R** (False Acceptance Rate for Random forgeries) is the percentage of random forgeries that were accepted as genuine. Total FPR divided by total tests against random forgeries \(\frac{\text{Total FPR}}{18800}\).

**Total FN** is the number of false negatives (false rejections) of genuine signatures (out of 9400 tested).

**Total FPS** is the number of False Positives (false accepted) Skilled forgeries (out of 9400 tested).

**Total FPR** is the number of False Positives (false accepted) Random forgeries (out of 18800).

**Featureset** is the set of features being tested. The different features are shown in Table 8 and the features in a features set is the ones with their index bit set to 1 in the binary number shown in the Featureset column (i.e. the decimal number 21 is found in the Featureset column, 21 is the binary number 00010101 meaning that feature 1, 3, 5 are the features being tested. Looking at Table 8 and you can see that the feature combination of Horizontal Position, Vertical Position and Pressure Value is being tested. The decimal number 3 is the binary number 00000011 and shows that feature 1 and 2 (Horizontal Position and Horizontal Velocity) are being tested).

**Breakpoint** is the \(i\) value sent to Equation 12, meaning that a lower value is rejecting more inputs as forgeries, while a higher value is allowing more inputs to be classified as genuine.

The result for feature set 11 is used to demonstrate the raw data.

<table>
<thead>
<tr>
<th>FRR</th>
<th>FAR-S</th>
<th>FAR-R</th>
<th>Total FN</th>
<th>Total FPS</th>
<th>Total FPR</th>
<th>Featureset</th>
<th>Breakpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7503194169</td>
<td>0.0015957447</td>
<td>0.0021808511</td>
<td>7053</td>
<td>15</td>
<td>41</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>0.6129787234</td>
<td>0.0029787234</td>
<td>0.0052659574</td>
<td>5762</td>
<td>28</td>
<td>99</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>0.5430851064</td>
<td>0.0057446809</td>
<td>0.0084574468</td>
<td>5105</td>
<td>54</td>
<td>159</td>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>0.500746809</td>
<td>0.0069148936</td>
<td>0.0112234043</td>
<td>4707</td>
<td>65</td>
<td>211</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>0.4715957447</td>
<td>0.0092553191</td>
<td>0.0149468085</td>
<td>4433</td>
<td>87</td>
<td>281</td>
<td>11</td>
<td>5</td>
</tr>
<tr>
<td>0.4456382979</td>
<td>0.0116382979</td>
<td>0.0183510638</td>
<td>4189</td>
<td>104</td>
<td>345</td>
<td>11</td>
<td>6</td>
</tr>
<tr>
<td>0.409787234</td>
<td>0.0132978723</td>
<td>0.0248404255</td>
<td>3852</td>
<td>125</td>
<td>467</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>0.3841489062</td>
<td>0.014680851</td>
<td>0.0311170213</td>
<td>3611</td>
<td>136</td>
<td>585</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>0.3591489062</td>
<td>0.0158510638</td>
<td>0.0375</td>
<td>3376</td>
<td>149</td>
<td>705</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>0.2869148936</td>
<td>0.0221276596</td>
<td>0.0606382979</td>
<td>2697</td>
<td>208</td>
<td>1148</td>
<td>11</td>
<td>20</td>
</tr>
<tr>
<td>0.2564893617</td>
<td>0.024680851</td>
<td>0.0711170213</td>
<td>2411</td>
<td>232</td>
<td>1337</td>
<td>11</td>
<td>24</td>
</tr>
<tr>
<td>0.2338297872</td>
<td>0.0276595745</td>
<td>0.0812234043</td>
<td>2198</td>
<td>260</td>
<td>1527</td>
<td>11</td>
<td>28</td>
</tr>
<tr>
<td>0.214893617</td>
<td>0.0306382979</td>
<td>0.0916489582</td>
<td>2020</td>
<td>288</td>
<td>1723</td>
<td>11</td>
<td>32</td>
</tr>
<tr>
<td>0.1587234043</td>
<td>0.037468085</td>
<td>0.1247340426</td>
<td>1492</td>
<td>352</td>
<td>2345</td>
<td>11</td>
<td>48</td>
</tr>
<tr>
<td>0.1223404255</td>
<td>0.0428723404</td>
<td>0.154893617</td>
<td>1150</td>
<td>403</td>
<td>2912</td>
<td>11</td>
<td>64</td>
</tr>
<tr>
<td>0.0613829784</td>
<td>0.0589361702</td>
<td>0.2620744681</td>
<td>577</td>
<td>554</td>
<td>4927</td>
<td>11</td>
<td>128</td>
</tr>
<tr>
<td>0.030106383</td>
<td>0.090746809</td>
<td>0.4129787234</td>
<td>283</td>
<td>853</td>
<td>7764</td>
<td>11</td>
<td>256</td>
</tr>
</tbody>
</table>
B Figures

Figure 12: Breakpoint $i$ on x-axis, % classified on y-axis.
Figure 13: Breakpoint $i$ on x-axis, % classified on y-axis. First 4 values removed.

Figure 14: Breakpoint $i$ on x-axis, % classified on y-axis. No local features.
Figure 15: Breakpoint $i$ on x-axis, % classified on y-axis.

Figure 16: Breakpoint $i$ on x-axis, % classified on y-axis.
Feature: 4,3,2,1

Figure 17: Breakpoint \( i \) on x-axis, \% classified on y-axis.

Feature: 6,5,3,1

Figure 18: Breakpoint \( i \) on x-axis, \% classified on y-axis.
Figure 19: Breakpoint $i$ on x-axis, % classified on y-axis.

Figure 20: Breakpoint $i$ on x-axis, % classified on y-axis.
Figure 21: Breakpoint \( i \) on x-axis, \% classified on y-axis.

Figure 22: Breakpoint \( i \) on x-axis, \% classified on y-axis.
Figure 23: Breakpoint $i$ on x-axis, % classified on y-axis.

Figure 24: Breakpoint $i$ on x-axis, % classified on y-axis.
Figure 25: Breakpoint $i$ on x-axis, % classified on y-axis.

Figure 26: Breakpoint $i$ on x-axis, % classified on y-axis.
Figure 27: Breakpoint $i$ on x-axis, % classified on y-axis.

Figure 28: Breakpoint $i$ on x-axis, % classified on y-axis.
Figure 29: Breakpoint $i$ on x-axis, % classified on y-axis.

Figure 30: Breakpoint $i$ on x-axis, % classified on y-axis.
Figure 31: Breakpoint $i$ on x-axis, % classified on y-axis.

Figure 32: Breakpoint $i$ on x-axis, % classified on y-axis.
Figure 33: Breakpoint $i$ on x-axis, % classified on y-axis.