Incorporating Signature Verification on Handheld Devices with User-dependent Hidden Markov Models

M. Martinez-Diaz, J. Fierrez and J. Ortega-Garcia
Biometric Recognition Group - ATVS, EPS - Univ. Autonoma de Madrid
C/ Francisco Tomas y Valiente, 11 - Campus de Cantoblanco - 28049
Madrid, Spain
{marcos.martinez, julian.fierrez, javier.ortega}@uam.es

Abstract

A dynamic signature verification system based on Hidden Markov Models is presented. For each user model, the number of states and Gaussian mixtures of the Hidden Markov Model is automatically set in order to optimize the verification performance. By introducing this user-dependent structure in the statistical modeling of signatures, the system error rate is significantly decreased in the challenging scenario of dynamic signature verification on handheld devices. Experimental results are given on a subset of the recently acquired BIOSECURE multimodal database, using signatures captured with a PDA.[1]

Keywords: Biometrics, signature verification, Hidden Markov Models, handheld, PDA

1. Introduction

Security has raised a great amount of interest in our society during the last few years. Biometric systems have become popular in secure applications like access control and user verification as they free the user from tokens or passwords. These systems may be based on behavioral biometric traits (e.g. voice, signature) or biological traits (e.g. fingerprint, iris) [1]. One of the main advantages of biometric traits is the fact that they cannot be easily stolen, forgotten or forged.

Within biometrics, signature is one of the most socially accepted traits. It has been used in legal and financial transactions for centuries, and is still now the principal mean of user authentication in most retail commercial transactions and other applications. Despite the wide range of systems proposed by researchers and commercial developers [2, 3], automatic signature verification is nowadays still a challenging task.

[1]This work has been supported by the Spanish Ministry of Education under project TEC2006-13141-C03-03. J. Fierrez is supported by a Marie Curie Fellowship from the European Commission.

The main difficulties in automatic signature verification are derived from the variability among signatures. Signatures from the same user may vary depending on the signing conditions or evolve during medium to large periods of time, leading to a considerable *intra-class* variability. Moreover, the existence of skilled forgers, which can imitate signatures with high precision can cause a very low *inter-class* variability among signatures. Forgeries are additionally difficult to model during the design phase of a system (and to obtain during the acquisition of research-oriented databases), as highly skilled or motivated forgers are rarely available.

Automatic signature verification systems can be classified in two types: *off-line* verification systems employ captured static signature images, which may have been scanned or acquired with a camera, to perform verification; *dynamic* or *on-line* systems capture signature time-functions via digitizer tablets or touch-screens (e.g. Tablet-PCs, smartphones, etc.). Dynamic systems have traditionally achieved a better verification performance than off-line systems as more levels of information than the signature static image are available [2].

On-line signature verification has followed two main approaches. *Feature-based* or global systems extract a set of global features from each signature and create a holistic n-dimensional vector describing it. Signatures are then compared using distance measures like Euclidean or Mahalanobis distance or statistical classifiers such as Parzen-Windows or GMMs (Gaussian Mixture Models). *Function-based* systems operate directly with captured or derived time sequences (position, velocity, inclination, etc.). These systems perform signature matching via elastic or statistical techniques such as DTW (Dynamic Time Warping) [4], or HMMs (Hidden Markov Models) [5]. Some authors have proposed the fusion of the two previous approaches (feature- and function-based) reporting a better performance than the individual systems [6]. The typical architecture of an on-line signature verification system is represented in Fig. 1.

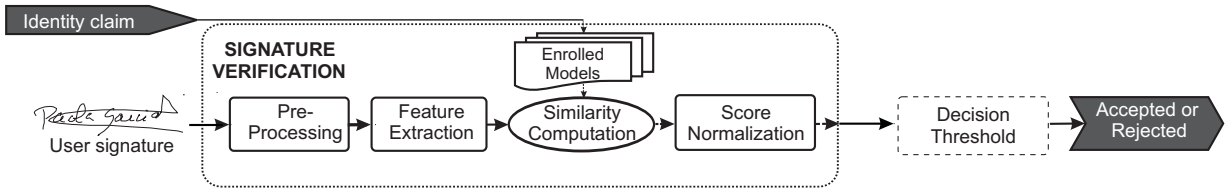


Figure 1. Signature Verification System Architecture.

Smartphones, PDAs and other portable devices that feature pen-based inputs provide a feasible platform to host a dynamic signature verification system. A great social and commercial interest on smart devices has raised in the last few years in the context of convergence and ubiquitous access to information and services [7]. In these devices, user validation and network access relies mostly on traditional PIN codes.

In this paper, an automatic signature verification system based on user-dependent Hidden Markov Models is presented and its performance is analyzed for the case of handheld devices. While some authors have reported HMMs with a variable number of states per user [8, 9, 10], no HMM approach has been reported in the literature with a varying number of Gaussian mixtures per state, to the extent of our knowledge. We present a verification system in which each user model has a variable number of states and Gaussian mixtures, allowing a significant improvement in the verification performance.

The system is a modified version of the one presented in our previous works [5] to adapt it to the challenging handheld scenario. The goal is to automatically find the HMM for each user that produces the best verification performance. The signatures used for experiments were captured on a PDA and belong to the recently acquired multimodal database under the BIOSECURE European Network of Excellence [11].

The work is structured as follows: the problem of signature verification on handheld devices is addressed in Sect. 2. Hidden Markov Models and related works are summarized in Sect. 3. The proposed user-depentent system is presented in Sect. 4. Experiments and results are reported in Sect. 5 and conclusions are finally drawn in Sect. 6.

2. Automatic Signature Verification on Handheld Devices

Touch-screen enabled handheld devices provide an appropriate hardware platform for a signature verification system. Most commercial handheld devices in the market are already able to perform handwritten character recognition as a text input alternative [7]. Signature verification systems on handheld devices allow multiple applications, including remote payments and legal transactions, network login or client validation. Documents can be electronically signed with a verifiable signature (including a confidence measure), leading to applications such as ubiquitous access to services or the paperless office.

2.1. Challenges of Signature Verification on Handheld Devices

Despite representing a very promising and convenient application, many challenges must be faced during the design of a signature verification system on handheld devices. Signature verification on such devices is affected by factors not present in other scenarios. Smartphones and PDAs usually have a reduced pen-input size, as they must combine both usability and portability. Processing power or memory is no longer a constraint while designing these devices, which are mostly limited by their screen and keypad size. As a consequence of small input areas, poor ergonomics or the fact that the user may be in movement, the signing process is degraded. The users must also face a new signing scenario, which presents differences in the signing surface (touch screen instead of paper) and in the signing instrument (PDA stylus instead of traditional pen).

Handheld device screens may also provide a poor sampling quality, with a variable sampling rate and sampling errors. Moreover, only position signals are made available by touch-screens, while pen azimuth or pressure among other signals that may enhance the verification performance [3], cannot be captured. The recent BIOSECURE Multimodal Evaluation Campaign [12], in which several independent research institutions have participated, has shown that there exists still room for improvement in the signature verification task on handheld devices, as verification results have been significantly lower than those with other databases captured using a pen tablet [13].

Another key concern while developing a signature verification application on handheld devices is security. The user template must be appropriately secured and encrypted as well as communication channels over which signature information may be transmitted.

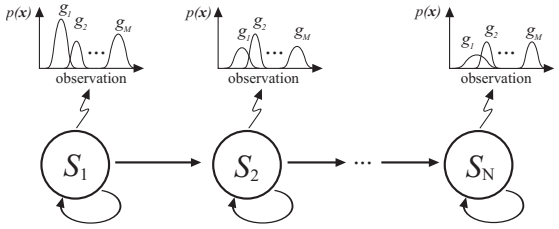


Figure 2. Graphical representation of a left-to-right N-state HMM, with M-component GMMs representing observations and no skips between states.

3. Related Work

3.1. Hidden Markov Models for Dynamic Signature Verification

Hidden Markov Models have been widely used by the speech recognition community as well as in many handwriting recognition applications. Several approaches using HMMs for dynamic signature verification have been proposed in the last years [5, 8, 9, 14]. An HMM represents a double stochastic process, governed by an underlying Markov chain with a finite number of states and a set of random functions (GMMs in most speech and handwriting recognition applications) that generate symbols each of which is associated with one state [14].

Finding a reliable and robust model structure for dynamic signature verification is not a trivial task. While too simple HMMs may not be able to model properly the user signatures, too complex models may not be able to model future realizations due to overfitting. On the other hand, as simple models have less parameters to be estimated, their estimation may be more robust than for complex models. Two main parameters are commonly considered while selecting an optimal model structure: the number of states and the number of Gaussian mixtures per state [5]. Most of the proposed systems consider a left-ro-right configuration without skips between states, also known as Bakis topology (Fig. 2).

3.2. User-dependent Hidden Markov Models

A recent study [10] has proven the benefits of using user-dependent models by specifically setting the number of states and Gaussian mixtures for each user in an HMM-based dynamic signature verification system. Nevertheless, results are obtained using exhaustive search for the best EER (Equal Error Rate) over all possible configurations, reflecting consequently a theoretical upper bound for the system performance and being thus non-implementable in practice.

Few works have been carried out to study user-dependent HMMs in the field of automatic signature verification. Examples of systems using a specifically selected number of states per user can be found in the literature. For example, in [8, 9], the number of states is proportional to the length of the training signatures, while in [15] the number of states is determined by the number of changes on the quantized pen trajectory. In all these referenced works, using a subject-dependent number of HMM states provided enhanced verification performance as compared to a fixed number of states. None of these works, however, studied the use of varying number of Gaussian mixtures for different subjects (in this way modulating the model complexity for different signers), as it is done in the present work.

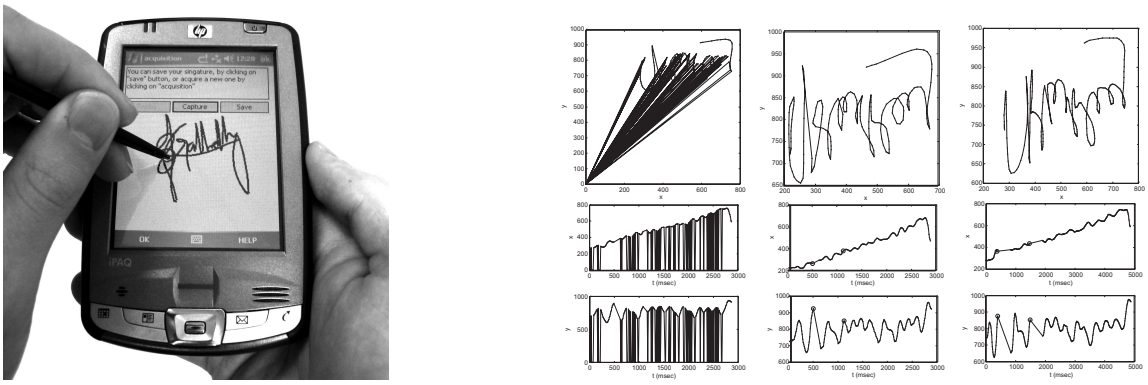
Another closely related work is described in [16], where a modified version of the Minimum Description Length (MDL) criterion is applied to on-line signature verification based on GMM with user-dependent complexity (number of Gaussian mixtures either 16 or 32). The MDL criterion increases with the likelihood of the model with respect to the training data and decreases with the increase of the model complexity (i.e. the number of parameters to be estimated). They demonstrate that minimizing the MDL criterion to automatically set the number of Gaussian Mixtures, the verification performance is enhanced with respect to fixed structure models.

4. Proposed System

The system implemented in this work is based on the one described in [5]. Each signature is modeled with 6 functions and their first differences leading to a $12-$dimensional time function describing it. The extracted time functions are the pen trajectory (coordinates x and y), path tangent-angle, path velocity magnitude, log curvature radius and total acceleration magnitude. A detailed description of the individual functions can be found in [5], where pressure information was additionally used as it was available in the considered pen tablet scenario.

An initial step is added to the original HMM training scheme, leading to the following stages: *i)* the global mean and covariance of the training signatures is assigned to all the mixtures, *ii)* k-means segmentation and Maximum Likelihood training is performed, *iii)* Baum-Welch re-estimation is finally carried out. The first step allows to have a trainable model for step *iii* (despite being inaccurate) in the case where step *ii* fails due to the large number of parameters to be estimated, or other computational problems.

Similarity scores are computed as the log-likelihood of the signature (using the Viterbi algorithm) divided by the total number of samples of the signature. No score alignment between users is applied [17], so this approach is equivalent to the baseline system proposed in [5] without using pen pressure information.



(a) (b)

Figure 3. (a) Signature capture process. (b) Signatures and associated signals from a user of the database. From left to right: genuine signature without preprocessing, genuine signature and skilled forgery. Pen-ups are marked with circles in the *x*- and *y*-coordinate time series. The signatures on the picture and the graphs are from different users.

4.1. Selection of the Number of States

A simple approach for the adaptation of the number of states is taken in this work. The number of states N_{opt} for a given user is obtained by dividing the mean length of the training signatures by a constant as follows:

$$N_{opt} = \frac{\sum_{i=1}^{K} T_i}{K \cdot D} \tag{1}$$

where T_i is the number of sample points of the i-th training signature of a given user, K is the number of training signatures per user and D is a division factor that is set by the system designer. The result is rounded to obtain integer values, being $N_{opt} = 1$ the smallest allowed value.

4.2. Selection of the Number of Gaussian Mixtures

The selection of the number of Gaussian mixtures for each specific user is performed for the whole HMM instead of for each state. Thus, all the states of the HMM model for each user have the same number M of mixtures. Preliminary experiments performing adaptation of the number of mixtures for each state (e.g. using Gaussian splitting approaches [18]) proved poor verification performances and were outperformed by models with the same number of mixtures for all states.

In our system, the optimum number of mixtures is selected as the one that maximizes the average likelihood of the training signatures. More precisely, the optimum number of mixtures M_{opt} is computed as follows:

$$M_{opt} = \underset{M<M_{max}}{\arg\max} \frac{\sum_{i=1}^{K} l(S_i, \lambda)/T_i}{K} \tag{2}$$

where K is the number of training signatures and $l(S_i, \lambda)$ is the likelihood of the training signature S_i given the user model λ (normalized by the number of samples T_i of the signature S_i). A maximum number of Guassian mixtures M_{max} is manually set to avoid overfitting.

This approach can be considered a simplified version of the MDL criterion proposed in [16], as no information about the model complexity is used to compute M_{opt}. This simplification is based on the following reasons. First, in [16], the proposed MDL criterion did provide positive results for full covariance matrices but not for diagonal, which are the ones used in our system (full covariance matrices are impractical in HMMs with scarce training data). Additionally, M_{opt} is computed by averaging the likelihood of all the training signatures from the enrolled user, providing enough variability to allow the model to generalize over unseen signatures. Moreover, as our approach could lead to overfitting, M_{max} is used to limit the number of mixtures.

5. Experiments

5.1. Database and Experimental Protocol

A subset of the signature corpus of the BIOSECURE multimodal biometric database is used for experiments. This subset was released prior to the Biosecure Multimodal Evaluation Campaign [11] to all participants for development purposes. It consists of 50 users, with 20 genuine signatures and 20 skilled forgeries per user, leading to $50 \times (20+20) = 2000$ signatures. The genuine signatures were acquired in two different sessions separated by an average period of two months, being 5 signatures from the first session and the remaining 15 from the second session. In each session, signatures were produced by the user in blocks of 5, leaving a gap of some minutes be-

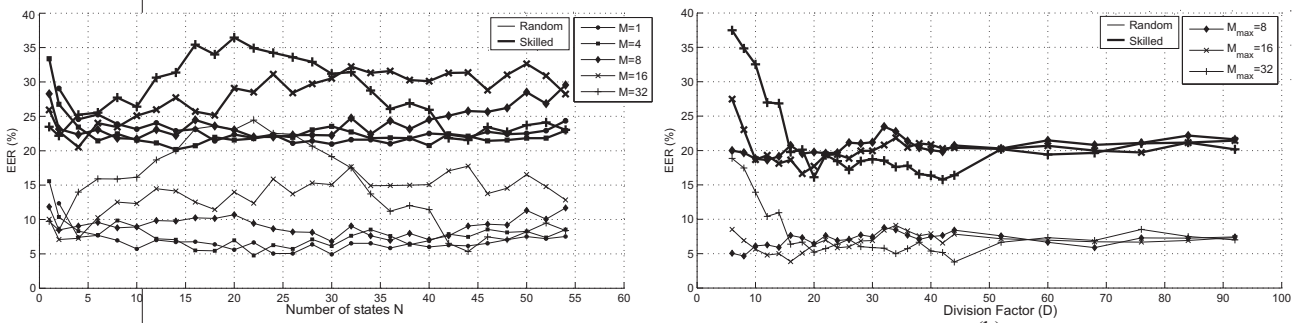
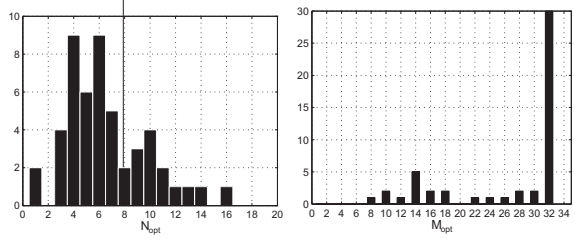


Figure 4. Verification performance for the experimental setup (a) Baseline system (b) User-specific for N and M.



(a) (b)

Figure 5. Histograms of N_{opt} and M_{opt} for $D = 42$ and $M_{max} = 32$. M_{opt} was computed in steps of 2.

tween each block. Signatures were captured with an HP iPAQ 2790 PDA while the user was standing and holding the PDA with one hand (Fig. 3.a). This emulates real operating conditions.

Only the *x* and *y* position signals and the sample timestamps were captured by the PDA. Skilled forgeries for each user were performed by 4 different users (5 forgeries each) in a "worst case" scenario: each forger had visual access to the dynamics of the genuine signature and a tracker tool allowing to see the original strokes.

An example of the capture process is shown in Fig. 3.a and examples of a genuine captured signature and a skilled forgery are shown in Fig. 3.b. Due to the degraded capture conditions, a pre-processing step is first performed, where incorrectly detected samples (see Fig. 3.b, left column) are linearly interpolated. As no pen pressure information is provided, pen-ups are assigned and linearly interpolated wherever a gap of 50 or more milliseconds between two consecutive samples exists.

In our experiments, training of the user models is performed with the 5 genuine signatures from the first acquisition session. The remaining genuine signatures from the second session are left for testing. Random forgery scores (the case where a forger uses his own signature claiming to be another user of the system) are obtained by comparing the user model to one signature sample of all the remaining users (thus, 50×49 random forgery scores are computed). Skilled forgery scores are computed by comparing all of the 20 available skilled forgeries per user with its own model (leading to 50×20 skilled forgery scores).

Experimental results are presented as follows. Baseline results with the original system, for different values of the number of HMM states and mixtures are first presented. Next, results for the proposed scheme, automatically setting the number of states N and mixtures M for each user using Eqs. (1) and (2) are analyzed.

5.2. Experimental Results

Baseline results are represented in Fig. 4.a. As can be seen, the verification performance for skilled forgeries and random forgeries is relatively stable when the number of states N varies for the case of a low number of mixtures M. More complex models ($M = \{16, 32\}$) lead to a poorer verification performance, due to the low amount of training data compared to the high number of parameters to be estimated.

Intermediate experiments, automatically setting either the number of states or the number of mixtures led to no significant improvements in the system performance.

Final experiments, where both the number of mixtures and states are specifically set for each user, show a considerable improvement in the system performance (Fig. 4.b), reaching an EER of 15.8% (with $D = 42$ and $M_{max} = 32$) for skilled forgeries compared to results no lower than 20% for the baseline system. The improvement in the system performance is mainly observed for the skilled forgery scenario. In Fig. 5, histograms of N_{opt} and M_{opt} for $D = 42$ and $M_{max} = 32$ are presented. As can be seen, 30 models (60% of the 50 users) are set to the maximum allowed number of mixtures. This shows the tendency to overfitting for some of them, which is controlled by M_{max}. Results with higher values of M_{max} proved this, with poorer verification performances.

Table 1. EER values for random (rd) and skilled (sk) forgeries.

Scenario	EER_{rd}	EER_{sk}
Baseline$_{(N\,=\,4,\,M\,=\,16)}$	7.3%	20.5%
User-dependent$_{(D\,=\,42,\,M_{max}\,=\,32)}$	5.2%	15.8%

In Table 1 we finally summarize the verification performance for a selected operating point with the baseline and the user-dependent system. The performance improvement with the proposed user-dependent scheme can be clearly observed.

6. Conclusions and Future Work

Experimental results have shown a significant improvement of 20% in the verification performance when both the number of states and mixtures are specifically set for each user. The best results are obtained for a high division factor D (thus, HMMs with a low number of states as seen in Fig. 5) and a high value for the maximum number of Gaussian mixtures M_{max}. This is aligned with the results presented in [5], where the best verification performance was found for a 2-state and 32 Gaussian mixtures HMM fixed structure. On the contrary, intermediate approaches where only the number of states or mixtures are specifically set have led to similar or even worse results than the baseline system.

As a result, we have observed the need of highly user-specific models to achieve a reasonable improvement in the verification performance for the challenging handheld scenario. The lack of pressure signals and the higher quality of skilled forgeries in the database (due to the tracking tool) may be also affecting the verification performance. Moreover, only raw scores have been used to study the system performance. The use of normalization techniques to align scores can reveal other behaviors of the system performance against variations of its structure or different user-dependent model setups [17].

The acquisition scenario offers challenges not studied in depth in related works using pen tablets, such as the interpolation of missing samples and the absence of pressure signals. The higher intra-user variability due to the adverse capture conditions can affect the system performance, contrary to more favorable capture conditions present in most research works like the user sitting while signing on a pen tablet. The adverse effect of the higher variability can be increased by the reduced amount of training data, leading to poor model parameter estimates. A more comprehensive comparison of traditional schemes for signature recognition between the established pen tablet scenario and the increasingly important handheld scenario, is therefore needed and will be the source of future research.

References

[1] A. K. Jain, A. Ross and S. Pankanti, "Biometrics: a tool for information security", *IEEE Trans. on Information Forensics and Security*, 1(2):125–143, 2006.

[2] R. Plamondon and G. Lorette, "Automatic signature verification and writer identification: the state of the art", *Pattern Recognition*, 22(2):107–131, 1989.

[3] J. Fierrez and J. Ortega-Garcia, *Handbook of Biometrics*, chapter On-line signature verification, Eds. A. K. Jain and A. Ross and P. Flynn, Springer, 2007.

[4] A. Kholmatov and B. Yanikoglu, "Identity authentication using improved online signature verification method", *Pattern Recognition Letters*, 26(15):2400–2408, 2005.

[5] J. Fierrez, D. Ramos-Castro et al., "HMM-based on-line signature verification: feature extraction and signature modeling", *Pattern Recognition Letters*, 28(16):2325–2334, 2007.

[6] J. Fierrez-Aguilar, L. Nanni et al., "An on-line signature verification system based on fusion of local and global information", *Proc. of IAPR Intl. Conf. on Audio- and Video-Based Biometric Person Authentication, AVBPA*, 2005, vol 3546 of *LNCS*, pp 523–532. Springer.

[7] R. Ballagas, J. Borchers, M. Rohs and J. Sheridan, "The smart phone: a ubiquitous input device", *IEEE Pervasive Computing*, 5(1):70–77, 2006.

[8] J. G. A. Dolfing, E. H. L. Aarts and J. J. G. M. van Oosterhout, "Online signature verification with Hidden Markov Models", *Proc. of the Intl. Conf. on Pattern Recognition, ICPR*, 1998, pp 1309–1312.

[9] B. L. Van, S. Garcia-Salicetti and B. Dorizzi, "On using the Viterbi path along with HMM likelihood information for online signature verification", *IEEE Trans. on Systems, Man, and Cybernetics, Part B*, 37(5):1237 – 1247, 2007.

[10] J. Pascual-Gaspar and V. Cardenoso-Payo, "Automatic online signature verification using HMMs with user-dependent structure", *Proc. IAPR ICB*, 2007, vol 4642 of *LNCS*, pp 1134–1143. Springer.

[11] F. Alonso-Fernandez, J. Fierrez et al., "Dealing with sensor interoperability in multi-biometrics: the UPM experience at the BioSecure Multimodal Evaluation 2007", *Defense and Security Symposium, BTHI, Proc. SPIE*, 2008.

[12] GET-INT, "BioSecure Multimodal Evaluation Campaign 2007 Mobile Scenario - experimental results", Technical report, 2007(http://www.int-edu.eu/biometrics/BMEC2007/files/Resultsmobile.pdf).

[13] D. Y. Yeung, H. Chang et al., "SVC2004: First International Signature Verification Competition", *Proc. of Intl. Conf. on Biometric Authentication, ICBA*, 2004, vol 3072 of *LNCS*, pp 16–22. Springer.

[14] L. Yang, B. K. Widjaja and R. Prasad, "Application of Hidden Markov Models for signature verification", *Pattern Recognition*, 28(2):161–170, 1995.

[15] D. Muramatsu and T. Matsumoto, "An HMM signature verifier incorporating signature trajectories", *Proc. of the Intl. Conf. on Document Analysis and Recognition, ICDAR*, 2003, vol 1, pp 438–442. IEEE Press.

[16] J. Richiardi and A. Drygajlo, "Gaussian Mixture Models for online signature verification", *Proc. of ACM SIGMM Workshop on Biometric Methods and Applications, WBMA*, 2003, pp 115–122.

[17] J. Fierrez-Aguilar, J. Ortega-Garcia et al., "Target dependent score normalization techniques and their application to signature verification", *IEEE Trans. on Systems, Man and Cybernetics, part C*, 35(3):418–425, 2005.

[18] A. Sankar, "Experiments with a Gaussian merging-splitting algorithm for HMM training for speech recognition", *Proc. of DARPA Speech Recognition Workshop*, 1998.