International Journal of Computer Science and Applications

Vol. 12 No. 2, pp. 37 - 56, 2015

A NEW SYSTEM FOR SECURE HANDWRITTEN SIGNING OF DOCUMENTS*

MARCO QUERINI, MARCO GATTELLI, VALERIO M. GENTILE and GIUSEPPE F. ITALIANO

Department of Civil Engineering and Computer Science Engineering,
University of Rome "Tor Vergata",
viale del Politecnico 1, 00133, Rome, Italy
marco.querini@uniroma2.it, querini.marco@gmail.com
marco.gattelli@gmail.com
valeriomaria.gentile@gmail.com
italiano@disp.uniroma2.it

Handwritten Signature Verification (HSV) is a natural and trusted method for user identity verification. HSV can be classified into two main categories: offline and online HSV. Offline systems take handwritten signatures from scanned documents, while online systems use specific hardware (e.g., pen tablets) to register pen movements during the act of signing. Online HSV systems may embed signatures (including the signature dynamics) into digital documents. Unfortunately, during their lifetime documents may be repeatedly printed and scanned, and digital to paper conversions may result in loosing the signature dynamics. The main contribution of this work is a new HSV system for secure handwritten signing of documents. First, we illustrate how to verify handwritten signatures so that signature dynamics can be processed during verification of every type of document (both paper and digital documents). Secondly, we show how to embed features extracted from handwritten signatures within the documents themselves, so that no remote signature database is needed. To accomplish the embedding task, we make use of 2D barcodes. The main challenge here is to be able to store the signature dynamics within the limited capacity of barcodes. Thirdly, we propose a method for the verification of signature dynamics which is compatible to a wide range of mobile devices so that no special hardware is needed. The main challenge here is to achieve a high verification performance, despite constrains due to the limited computational resources and pressure accuracy of mobile phones. We address the trade-off between discrimination capabilities of the system and the storage size of the signature model. Towards this end, we report the results of an experimental evaluation of our system on different signature datasets.

Keywords: handwritten; signature; documents.

1. Introduction

Biometric recognition refers to the automatic identification of a person based on his/her anatomical (e.g., fingerprint, iris) or behavioral (e.g., signature) character-

*A preliminary version of this paper was presented at the Federated Conference on Computer Science and Information Systems [Querini *et al.* (2014)].

istics or traits. This method of authentication offers several advantages over traditional methods involving authentication tokens (including ID cards) or passwords: it ensures that the person is physically present at the point-of-identification; it makes unnecessary to remember a password or to carry a token. The most popular biometric traits used for authentication are face, voice, fingerprint, iris and handwritten signature.

In this paper, we focus on handwritten signature verification (HSV). Since people are used to signing documents in their everyday life, HSV is a natural and trusted method for user identity verification. HSV can be classified into two main categories, depending on the hardware used and on the method used to acquire data related to the signature: online and offline signature verification. Offline systems take handwritten signatures (represented as an image) from scanned documents. This means that offline HSV systems only process the 2D spatial representation (i.e., the shape) of the signature. On the contrary, online systems use specific hardware (e.g., pen tablets) to register pen movements during the act of signing. As a result, online HSV systems are able to process dynamic features, such as the time series of the pen's position, pressure, velocity, acceleration, azimuth and elevation. Online signature verification has been shown to achieve higher verification rate than offline signature verification [Qiao *et al.* (2007); Kalera *et al.* (2004); Jain *et al.* (2002)], but unfortunately it suffers from several limitations.

First, the online approach works only for digital documents and it is currently unavailable for paper documents. In particular, during a document's life cycle, when a document is being printed, scanned or faxed, the signature dynamics are unavoidably lost. To overcome this limitation, there is an emerging need of designing new methods capable of embedding the signature dynamics within paper documents, along with the signature shape (the 2D spatial representation) which is the only feature usually preserved after printing. This will enable one to verify the authenticity of a document, regardless of its current (paper or digital) format, which is particularly important when the same document is repeatedly printed and scanned (or faxed) in a typical workflow.

Secondly, current online approaches raise privacy and security concerns since they store the genuine signatures of each user on a remote database server. Indeed, both commercial [Xyzmo (2015); SutiDSignature (2015); Andxor Corporation (2015)] and HSV systems proposed in the scientific literature [Qiao *et al.* (2007); Trevathan and McCabe (2005); Mailah and Lim (2012)] store genuine signatures of the users in a central database: during verification, specific signature data is retrieved from the database and compared to the actual signature. From the security viewpoint, an intruder who gains unauthorized access to the database containing dynamics of users' signatures can use this information to produce accurate forgeries. From the viewpoint of privacy, the recent news about the NSA surveillance program (see e.g., [The Guardian (2015)]) have definitely reduced our trust in providing sensitive data (such as signature features) to third parties. In order to address these

privacy and security concerns, we need to design novel HSV systems capable of supporting the online approach without using signature databases.

Thirdly, the online approach is often feasible only if special purpose hardware is available. Indeed, handwritten signatures are usually acquired by means of digitizing tablets connected to a computer, because smartphones and mobile tablets (that have worse sensitivity) may be not able to support the verification algorithms. As a result, the range of possible usages of the verification process is strongly limited by the hardware needed. To overcome this limitation, one needs techniques capable of verifying signatures acquired by smartphones and tablets in mobile scenarios.

To the best of our knowledge, there is no existing solution which is able to address all of those critical points simultaneously. The approach described in [Qiao *et al.* (2007)] addresses only the first point: by performing offline verification using online handwriting registration, the online approach is (partially) applicable to verifying signatures taken from paper documents, but this framework is not supported by mobile devices and requires an online signature database. Offline HSV solutions (such as [Kalera *et al.* (2004); Kumar *et al.* (2013); Pansare and Bhatia (2012)]) address only the second point: they do not use remote signature databases, but unfortunately they are not able to take into account signature dynamics. Online HSV systems (such as [Xyzmo (2015); SutiDSignature (2015); Andxor Corporation (2015); Trevathan and McCabe (2005); Mailah and Lim (2012)]) address only the third point: they are supported by mobile devices, but cannot verify signatures taken from paper documents and are inherently based on remote database servers.

The goal of this paper is to address all of the above challenges by considering new mobile scenarios in which HSV can play a significant role. The novelties of our approach lie mainly in the following three aspects.

First, we present a new system to sign and verify documents so that the online approach is applicable for all kind of documents (including paper documents). It performs verification in a way that the signature dynamics can be used also when the signed documents are printed and scanned, thus allowing the online approach to operate in those cases where only the offline approach was available.

Secondly, we show how to embed features extracted from handwritten signatures within the documents themselves, so that no remote signature database is needed. To accomplish the embedding task, we make use of 2D barcodes. In particular, we use HCC2D codes [Querini *et al.* (2011); Querini and Italiano (2014)], which are well-suited for this framework because of their high data capacity. The main challenge here is to be able to store the signature dynamics (into documents), within the limited capacity of barcodes. For this reason, we need to use a signature model whose size is small.

Thirdly, we propose a method for the extraction and verification of signature dynamics which is compatible to a wide range of mobile devices (in terms of computational overhead and verification accuracy) so that no special hardware is needed. The main challenge here is to achieve a high verification performance, despite con-

strains due to the limited computational resources and pressure accuracy of mobile phones. For this reason, we designed a verification algorithm that can be run on mobile phones in fractions of a second and that weights the signature features based on the accuracy of the given device.

In order to assess the precision and recall of our HSV system, we conduct an experimental study whose results are reported for different data sets of signatures.

2. A Logical View of the HSV System

Our HSV system consists of three main modules, corresponding to three main phases. We next describe (from a logical point of view) the registration phase, the document signing phase and the document verification phase with our system.

2.1. *The Registration Phase*

The objective of the registration phase is to compute a compact representation of the signature dynamics of a given person. The process starts with the user writing his/her signature on the device's screen and ends with the generation of a biometric template representative of the user signatures. Figure 1 shows a high level, logical view of the registration procedure.

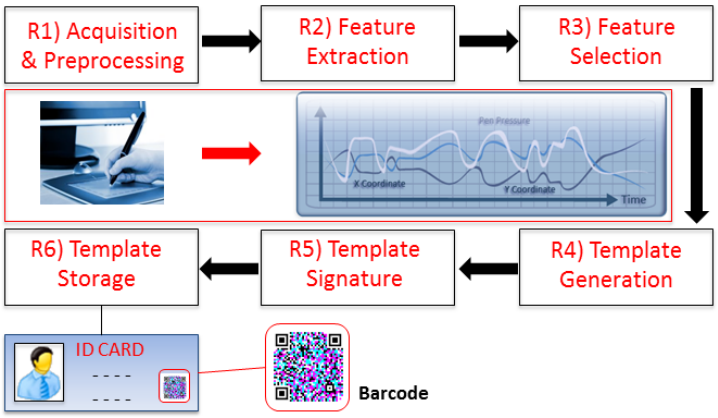


Fig. 1: Registration procedure for the proposed HSV system. (Viewed better in color).

The registration phase consists of the following tasks. First, in order to take into account the variability among signatures produced by the same user, signature dynamics for at least three signatures are captured. Then, the features extracted from the various signatures are combined to form a template representative of the given user. Finally, the template is digitally signed and securely stored in a barcode. The barcode is embedded within a card which is released only to the user to whom

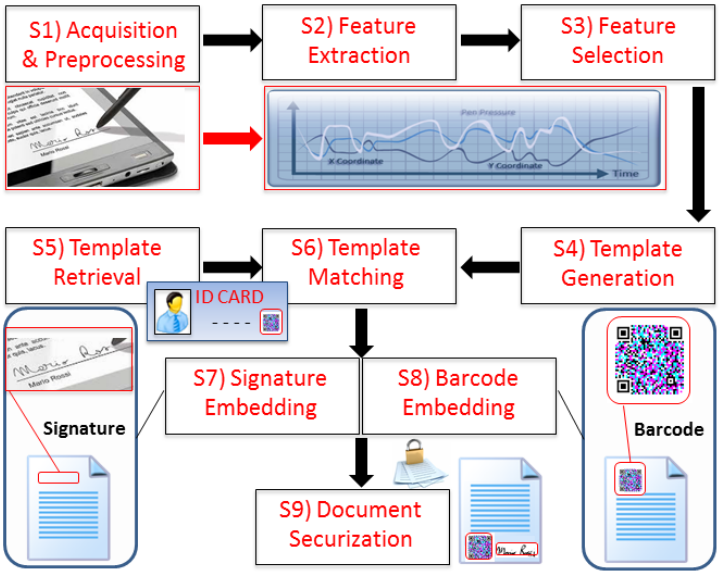


Fig. 2: Document signing procedure for our HSV system. (Viewed better in color).

the template belongs, in order to address any privacy concern. We remark that in order to make this scenario possible, it is necessary that the barcode is capable of storing the whole template and the digital signature associated with it.

We will describe how to perform identity registration in Section 3, including details of the acquisition, pre-processing, feature extraction and selection steps, along with the definition of the user template.

2.2. *The Document Signing Phase*

The objective of this phase is to sign documents on mobile devices so that: the signature dynamics are embedded within the documents themselves (along with the spatial representation of the signature); the signature dynamics survive the document's life cycle, when a document is being printed, scanned or faxed.

Figure 2 shows a logical view of the document signing phase, which consists of the following steps.

2.2.1. *Template Generation of the User to Be Verified (S1-S4)*

We need to generate a compact feature synthesis of the person to verify. In order to accomplish this task, we proceed through steps S1 to S4, which are equivalent to steps R1-R4 of the Registration phase, except for the fact that there is no need to capture three or more writings of the handwritten signature to be verified.

2.2.2. *Retrieval of the Secure Template (S5)*

We retrieve the secure template (generated at the end of the registration phase) of the user corresponding to the claimed identity of the user to be verified.

2.2.3. *Template Matching (S6)*

We compare the signature dynamics related to the identity to verify with the secure template retrieved at the previous step. This is a crucial step, because we do not allow unrecognized signatures to be embedded within documents. Note that in order to enhance security, we use strong authentication based on something the user has (i.e., a card storing the registered template) with something the user is (features of his/her handwritten signature captured at the moment).

2.2.4. *Signature Embedding (S7)*

If the matching is successfully, we embed the spatial representation of the signature within the document.

2.2.5. *Barcode Embedding (S8)*

If the matching is successfully, we embed the signature dynamics of the signature within the document by means of a high capacity barcode such as the HCC2D code.

The elements that need to be embedded by means of barcodes include the following:

- The template representing the signature dynamics.
- The timestamp of the signature.
- Information about the document the user is going to sign.
- The digital signature of all the above data.

The last three elements ensure that the barcode storing the signature dynamics cannot be copied and pasted on a new document for producing a forgery.

2.2.6. *Document Securization (S9)*

The secure document is generated (with the signature image and the barcode). Because of the binding [signature features, document], no other document can be signed with features that are used for the given document.

2.3. *The Document Verification Phase*

The aim of the verification phase is to verify the authenticity of a handwritten signature. This phase ends with the authenticity of the document signature being accepted or rejected. Figure 3 shows a high level view of the document verification procedure, consisting of the following tasks.

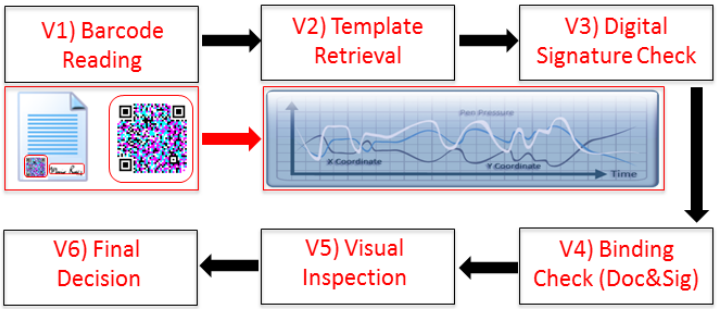


Fig. 3: Document verification procedure for our HSV system. (Viewed better in color).

2.3.1. *Barcode Reading (V1)*

The objective of this step is to decode the barcode which has been encoded in the document signing phase. This allows us to retrieve the secure template which has been stored within the barcode, along with the document metadata, the timestamp and the digital signature.

2.3.2. *Template Retrieval (V2)*

The signature features used to sign the given document are retrieved from the barcode.

2.3.3. *Digital Signature Check (V3)*

The digital signature is checked before the retrieved data are processed, in order to ensure that they are not tampered with.

2.3.4. *Binding Check (V4)*

The binding [signature features, document] is retrieved and showed to the user, in order to verify that those features were previously associated with the given document.

2.3.5. *Visual Inspection and Final Decision (V5-V6)*

The signature matching is graphically showed to the user so that he/she is able to sense the similarity and the likelihood of a forgery. This is a crucial step because it ensures that the final decision is taken by the user.

In this stage, the shape of the signature is reconstructed from the signature dynamics stored in the barcode. We cannot completely trust the signature shape

that is pasted on the document because it may be a forgery, but we can trust the reconstructed shape because data stored in the barcode are digitally signed.

3. The Signature Verification Algorithm

In this Section, we describe (from a technical point of view) our signature registration and verification methods.

3.1. *Signature Registration Phase*

The registration phase with our system starts with the user writing his/her signature on the device screen and ends with the generation of his/her biometric template (representative of his/her signature dynamics), which is embedded within a document (issued to the user) by means of a high capacity barcode.

3.1.1. *Acquisition and Pre-processing*

In the acquisition phase, the user is requested to make genuine signature 3 times. Once signatures are acquired, a pre-processing step is carried out. Pre-processing can successfully eliminate noise, normalize handwriting and fix the variations of handwriting. For instance, we use smoothing, which is one of the simplest approaches for data filtering. As most pre-processing methods, it consists of substituting the coordinates of the original point. We perform smoothing by computing a weighted sum of $(2n + 1)$ points in an interval centered on p_i:

$$p_i^* = \sum_{k=-n}^{n} \alpha_k \cdot p_{i+k}$$

where

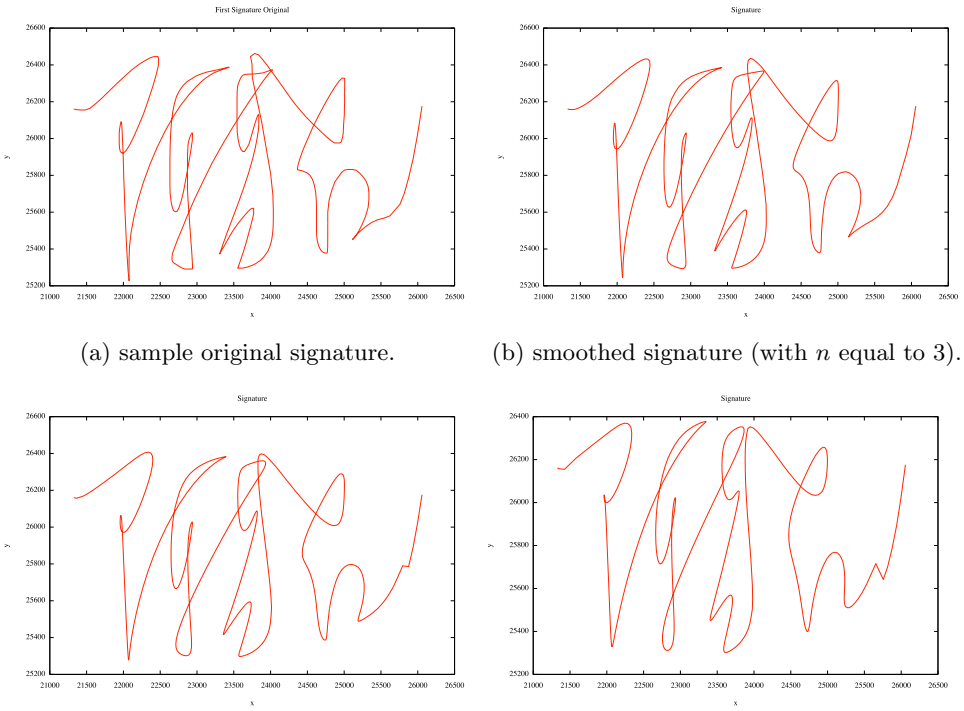
$$\sum_{k=-n}^{n} \alpha_k = 1$$

$$\alpha_k = 1/(2 \cdot n + 1)$$

Figure 4 shows a sample handwritten signature (Figure 4a) and the output of the smoothing procedure, with $n = 3$ (Figure 4b), $n = 5$ (Figure 4c) and $n = 7$ (Figure 4d), respectively. The larger n, the larger is the number of neighboring points considered in the weighted sum. If n is too large, significant features may be lost, and thus, in our experiments, we never consider values of n greater than 5.

3.1.2. *Feature Extraction and Selection*

Once signatures are pre-processed, the system performs an analysis of the intra-person variability of the signature features. This allows the system to determine



(a) sample original signature. (b) smoothed signature (with n equal to 3).

(c) smoothed signature (with n equal to 5). (d) smoothed signature (with n equal to 7).

Fig. 4: Sample output of the smoothing procedure. The origin of the x and y axes is located at the bottom-left corner of the device screen.

whether to accept or to reject the 3 signatures upon which the user model has to be built: if variability is too high, the user is requested to repeat the acquisition step.

If the intra-person variability is acceptable, then for each signature and for each sampled point, the following features are captured:

- *Event Type*. The event which led to the generation of a sample can be of three different types: Pen-down, i.e., a pressed gesture has started (the motion contains the initial starting location); Pen-up, i.e., a pressed gesture has finished (the motion contains the final release location as well as any intermediate points since the last down or move event); Pen-Move, i.e., a change has happened during a press gesture (between a Pen-down and a Pen-up).
- *Time*. The instant in which the pressed gesture has occurred (expressed as milliseconds since the first gesture event of the signature).
- *X-Y coordinates*. The X and Y coordinates of the sampled point (expressed in pixels).

- *Pressure*. The pressure with which the screen is pressed (expressed with a value ranging from 0, i.e., no pressure, to 1, i.e., maximum pressure).

3.1.3. *Template Generation, Signature and Storage*

Since we aim at minimizing the storage size of the signature model, derivatives such as velocity or higher-order derivatives such as acceleration are not stored in this stage, as they can be computed at run-time during the verification phase. Time series are also not stored, since they can be derived from the sampling frequency which is constant for a given device. For these reasons, the signature model consists of the series of the x-y coordinates and pressure only (for each of the three signatures). The model is digitally signed and stored in a high capacity barcode which is embedded within the document.

3.2. *Signature Verification Phase*

The verification phase determines the acceptance or the refusal of the claimed identity based on the similarity between the registered signature and the signature to verify. Our system can compute the similarity score in different ways depending on whether signatures are segmented or not (see below) before applying the verification algorithm, where signature segmentation refers to the process of partitioning the signatures into multiple strokes (i.e., segments). In our case, strokes are separated by discontinuities represented by pen-up events.

In the reminder of this section, we first describe the modes in which the system operates; then, we illustrate the verification algorithm. The same algorithm is used by the two modes: the main change is the way in which the algorithm is applied. We distinguish a segmented, an unsegmented and a mixed mode.

- *Segmented Mode*. The matching is done by comparing each segment (or subsequence) of the signature to verify with the corresponding segment of the registered signature. This means that the verification algorithm is applied several times (for each pair of segments) and the final decision (acceptance or refusal) is taken according to the result of each segment pairing.
- *Unsegmented Mode*. The matching is done by comparing the whole sequence representing the signature to be verified with the corresponding registered signature. The verification algorithm is applied just once and the output is the final decision.
- *Mixed Mode*. The matching is done using either the segmented or the unsegmented mode, depending on characteristics of the input signatures. For instance, the system may choose to apply the unsegmented mode in those cases in which the segmentation step generates, for a first signature, a number of segments significantly different from the number of segments in which a second signature is divided.

Now we turn to describe our verification algorithm, which is a scheme based on the Dynamic Time Warping (DTW) algorithm. DTW is a popular and robust technique for comparing time series, capable of handling time shifting and scaling, which has been successfully used in literature for HSV (prevalently for online approaches like the methods proposed in [Faundez-Zanuy (2007); Miguel-Hurtado *et al.* (2007)], but also for offline approaches such as the method described in [Piyush Shanker and Rajagopalan (2007)]). We use both global and local features, where global features are features related to the signature as a whole (e.g., the total signature time, that is, the total time taken in writing the signature), while local features correspond to a specific sample point along the trajectory of the signature (e.g., pressure in a given point). We describe our verification stage for the segmented mode only, being the unsegmented mode a sub-case of the segmented mode in which the number of segments is exactly one.

The algorithm is as follows. First, the total signature time is considered, since a forger is unlikely to be able to successfully reproduce the shape and the fluency of the signature writing fast. Basically, when a forger simulates a writing he must choose between writing fast (which produces generally better line quality but a less accurate reproduction) and slower writing (which results in a more accurate letter shapes but a loss of line quality). If the forger chooses to write slowly, the total signature time is unnaturally high, and thus, our algorithm immediately classifies such a signature as a forgery. In this way, we force the forger to write fast, in order to compromise the accuracy of the forgery.

If the total writing time of the signature to be verified is acceptable, the algorithm proceeds as follows. For each local feature f (such as X-Y coordinates, Pressure, X-Y Velocity, X-Y Acceleration) the following n-dimensional vector is computed, where n is the number of segments in which each of the three signatures is divided. S_j^i represents the i^{th} segment (related to feature f) of the j^{th} signature as a 1D time series, while $DTW(S_j^i, S_k^i)$ denotes the 1D Dynamic Time Warping method applied to the i^{th} segments of the j^{th} and k^{th} signatures.

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

Then, once the $\left\|f\right\|$ vector is computed for each feature of interest, we get a $\left\|X\right\|$ and a $\left\|Y\right\|$ vector (x and y coordinates), a $\left\|P\right\|$ vector (pressure), a $\left\|V_x\right\|$ and a $\left\|V_y\right\|$ vector (velocity on x and y directions), a $\left\|A_x\right\|$ and a $\left\|A_y\right\|$ vector (acceleration on x and y directions).

Finally, we combine the metrics with the following weighted sums, by giving a weight to each of the kinds of signature features.

$$\left\|\begin{matrix} d^1 \\ d^2 \\ ... \\ d^n \end{matrix}\right\| = \left\|\begin{matrix} w_x \cdot X^1 + w_y \cdot Y^1 + w_p \cdot P^1 + ... + w_{a_y} \cdot A_y^1 \\ w_x \cdot X^2 + w_y \cdot Y^2 + w_p \cdot P^2 + ... + w_{a_y} \cdot A_y^2 \\ ... \\ w_x \cdot X^n + w_y \cdot Y^n + w_p \cdot P^n + ... + w_{a_y} \cdot A_y^n \end{matrix}\right\|$$

The weights w_x, w_y, w_p, ... , w_{a_y} must be experimentally determined as they are dependent on the device. For instance, even if the pressure change is generally a very discriminating feature (often leading to a high w_p coefficient), the influence of the capability of sensing pressure change (which is specific for each device) is significant and the weight should be lowered accordingly on low end devices.

The output distance vector $\|d\|$ represents the "distance" among the three signatures. The whole process is repeated twice; the first time using genuine registered signatures ($\|d_g\|$ as output), the second time using signatures to be verified ($\|d_v\|$ as output).

Finally, we compare the two distance vectors with each other. The similarity function is defined as follows.

$$Similarity\left(\left\|\begin{matrix} d_v^1 \\ ... \\ d_v^n \end{matrix}\right\|, \left\|\begin{matrix} d_g^1 \\ ... \\ d_g^n \end{matrix}\right\|\right) = \left(\sum_{i=1}^{n} F(d_v^i, d_g^i)/n\right)$$

where n is the number of segments and the $F()$ function is defined as follows (c is a tolerance coefficient).

$$f(d_v^i, d_g^i) = \begin{cases} 1 & \text{if } d_v^i \leq c \cdot d_g^i \\ 0 & \text{otherwise} \end{cases}$$

The higher the value of the similarity function, the more likely the claimed identity is correct. The final decision (accept or reject the claimed identity) depends on whether the similarity score is above or below of a given threshold.

Note that all the computation happens at verification time (including the tasks which process genuine signatures). We cannot move any computation at registration time, because the priority is to minimize the storage size of the output of the registration phase (to be embedded by means of barcodes).

4. Experimentation

In this section we present experimental results concerning identity verification with our system. The accuracy of a recognition algorithm is generally measured in terms of two potential types of errors: false negatives (*fn*) and false positives (*fp*). False positives are cases where a claimed identity is accepted, but should not be, while false negatives are cases where a claimed identity is not accepted, while it should be. Two metrics building on true/false positives/negatives (*tp,fp,tn,fn*) are widely

adopted: precision and recall. Recall ($tp/(tp + fn)$) is the probability that a valid identity is accepted by the system (i.e., true positive rate) while precision ($tp/(tp + fp)$) is the probability that a claimed identity which is accepted by the system is valid. F-measure (which is the harmonic mean of precision and recall) combines both metrics into a global measure ($f\text{-}measure = (2 \times prec \times recall)/(prec + recall)$). A more general f-measure is generally defined as function of a β parameter, which is used to weight f-measure in favor of precision ($\beta < 1$) or recall ($\beta > 1$).

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

The performance of the proposed scheme has been assessed in terms of false positives, false negatives, precision, recall and f-measure on three different datasets: first, on the SVC database [SVC 2004 (2015c)]), involving WACOM digitizing tablets, 100 sets of signature data, with 20 genuine signatures and 20 skilled forgeries for each set; secondly, on the SigComp 2011 dataset [SigComp 2011 (2015b)], involving 1,790 dutch signatures and 960 chinese signatures collected using WACOM tablets. Thirdly, on a custom dataset, built for this purpose using different smartphones (mainly from the Google Nexus family), involving 250 signatures partitioned into 5 sets of signature data, with 10 genuine signatures and 40 skilled forgeries for each set.

We start by describing the experimental set-up:

- For each user, 3 genuine signatures out of 20 (first dataset) or out of 10 (second dataset) were selected in rotation to form the template of the user.
- Every time a user template (building on 3 signatures) is selected, the remainder 7 genuine signatures were matched to the template itself in order to compute the false rejection rate (i.e., the false negatives rate) by means of the matching error. This process is repeated for every user.
- Given a user, the skilled forgeries (provided by users other than the one named) were matched to his/her template in order to compute the false acceptance rate (i.e., the false positives rate). This process is repeated for every user involved in the experiment.
- As for the first dataset, the SVC 2004 competition [SVC 2004 (2015c)] consisted of two separate signature verification tasks, each of which was based on a different signature database. Each database of the SVC 2004 has 100 sets of signature data. Each set contains 20 genuine signatures from one signature contributor and 20 skilled forgeries from five other contributors. Contributors were asked to write on a digitizing tablet (specifically, a WACOM Intuos tablet).
- As for the second dataset, the SigComp 2011 competition made use of two sets of signatures (i.e., a set of dutch signatures and a set of chinese

signatures). Each set contains data related to 10 reference writers, along with skilled forgeries of their signatures. The total number of signatures in the dutch set is 1,790, while the total number of signatures in the chinese set is 960. Contributors were asked to write on a WACOM Intruos3 tablet.

- As for the third dataset, we used a custom dataset, built on data acquired by smartphones (i.e., no special purpose devices). We collected 250 handwritten signatures. Each user was requested to make genuine signature 10 times and to provide 10 (skilled) forgeries of any other user. The signatures were partitioned into 5 sets of signature data, where each set contains 10 genuine signatures of a specific user and 40 skilled forgeries of the signature of that user, produced by the other users involved in the experiment.

The experimental results in terms of precision, recall and f-measure (that vary according to the chosen thresholds) have been used for tuning the thresholds in order to get better performance (see Section 4.1). Then, once we fixed the threshold on the similarity score, we evaluated how subsampling the sequences (forming each signature) affected the overall precision and recall. This allowed us to identify the best trade-off between discrimination capabilities of the system and the storage size of the handwritten signature model, in order to ensure that the signature model fit into 2D barcodes (see Section 4.2).

The remainder of this section illustrates our results, which are in line with other work in the area, despite storage constrains due to barcode capacity and limitations in sensing and processing related to common devices such as smartphones and tablets.

4.1. *Tuning the Thresholds to Enhance Precision and Recall*

In this section we tune system thresholds by analyzing the curves of precision, recall and f-measure in order to get better performance (thresholds determine whether to accept or reject the claimed identity).

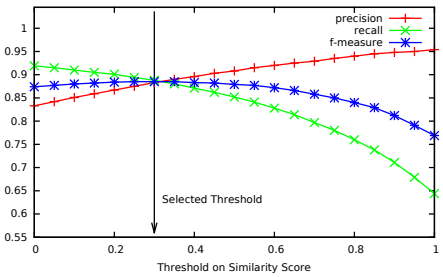
The graphs of Figure 5 show the curves of precision, recall and f-measure as functions of threshold on the similarity score for each dataset using the segmented and the unsegmented mode. The precision and recall curves related to the mixed mode are not reported here, since their values derive directly from the values related to the segmented and unsegmented mode (the mixed mode take the best from the two modes). Claimed identities are accepted whenever the score is above the threshold, rejected otherwise. The higher the threshold, the higher the precision, but the lower the recall. The threshold which maximizes the f-measure is identified for each working mode and highlighted with an arrow in each graph. The best results (plotted in Figures 5a, 5c, 5e and 5g) were achieved using the segmented mode. This is in agreement with the intuition that working at low granularity (i.e., comparing the subsequences segmented by pen-up events of a first signature with the corresponding subsequences of a second signature) leads to a more accurate comparison than just making a single comparison of the two signatures (considered

in their entirety).

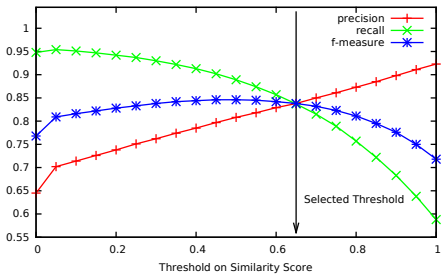
As for the SVC2004 dataset, precision and recall jointly reach a maximum of 89.50% in the graph of Figure 5a. For this reason, at this operating point, where the curves of precision and recall cross each other, the false positive rate (also called the false acceptance rate - FAR) and the false negative rates (also called the false rejection rate - FRR) are both 10.50% as complementary values. This value is denoted as the equal error rate (EER), that is, the point where FAR equals FFR. We got good results, considering the EERs reported at the SVC competition: for instance, the EERs related to the SVC training set (with skilled, not random forgeries) range from a low of 5.50% to a high of 31.32% in the first verification task [SVC 2004 (2015a)] and from 6.90% to 21.89% in the second verification task [SVC 2004 (2015b)].

As for the SigComp 2011 dataset, precision and recall cross at 95.40% (dutch dataset) and at 76.00% (chinese dataset). In the case of the dutch dataset, the results that we achieved are comparable with the best results obtained at the competition, while in the case of the chinese dataset, our results are only in line with the average results got at the competion. The difference between the results that we achieved using the dutch dataset and the results that we got using the chinese dataset can be explained by the fact that the two datasets have significant differences in the way they are built (in terms of the ratio between the number of authentic signatures and the number of skilled forgeries). Such a difference is not surprising, if noting that several algorithms that partecipated at the competition, such as xyzmo, got better results on the dutch dataset than on the chinese dataset, as reported in [SigComp 2011 (2015a)].

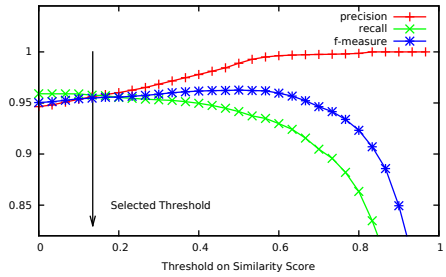
Finally, as for the custom dataset built using smartphones, precision and recall cross at 85.15%. This is an interesting result, if noting that the precision and recall related to the dataset built using smartphones (that is, no special-purpose hardware) are comparable with the precision and recall related to datasets built using special-purpose devices.



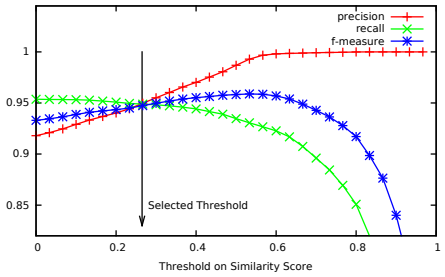
(a) SVC 2004 dataset; segmented mode.



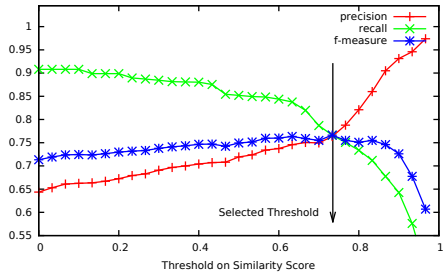
(b) SVC 2004 dataset; unsegmented mode.



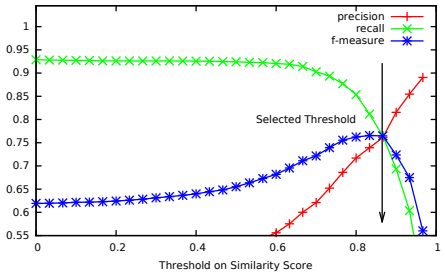
(c) SigComp 2011 (Dutch) dataset; segmented mode.



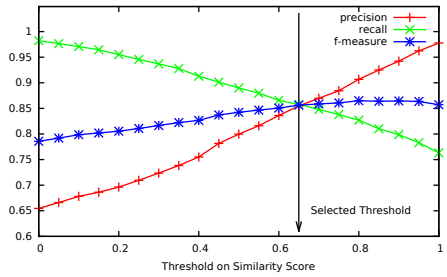
(d) SigComp 2011 (Dutch) dataset; unsegmented mode.



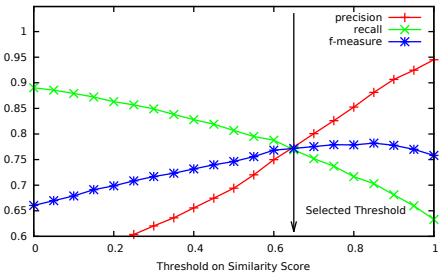
(e) SigComp 2011 (Chinese) dataset; segmented mode.



(f) SigComp 2011 (Chinese) dataset; unsegmented mode.



(g) custom dataset built using different smartphones; segmented mode.



(h) custom dataset built using different smartphones; unsegmented mode.

Fig. 5: Precision, recall and f-measure as functions of threshold on similarity score for each working mode and for each dataset (Viewed better in color).

4.2. *Trade-off between Precision/Recall and the Model Size*

In this section we address the trade-off between precision and recall of the system and the space used for storing models of signatures. This allow us to show how to embed features extracted from handwritten signatures within the documents themselves by means of barcodes.

Consider that the size of a model depends on the number of samples with which we represent each handwritten signature. We expect that the larger the space available for storing models of signatures, the more the precision and recall of the system. We also expect this to happen up to a certain threshold, after which precision and recall remain almost constant. This is consistent with the intuition that getting more samples than needed (the limit is due to the precision of the acquiring devices) does not improve the overall performance.

Signatures data were subsampled as follows: 1:1 (all the samples captured by the device are kept), subsampled 2:1 (1 sample out of 2 is filtered out), subsampled 3:1 (2 samples out of 3 are filtered out), ..., subsampled 20:1 (19 samples out of 20 are discarded). For instance, from a signature which is 600 samples length we produce subsampled signatures whose length is 300 (subsampled 2:1), 200 (subsampled 3:1), ..., 30 (subsampled 20:1) samples. However, the actual signature length (expressed as number of samples) depends not only on the sampling rate of the device but also on the path length of the handwritten signature (expressed as length unit such as the millimeter).

In order to isolate the impact of the path lengths so that results do not depend on the length of the words forming the signature, we introduce the concept of samples density as the number of samples per unit length. In order to compute the samples density, the number of samples of a given signature is divided by the total path length of the signature itself. This is computed in pixels and is then converted from pixels to millimeters by referring to the number of points per inch (ppi) characterizing the device screen and to the (inches to millimeters) conversion factor. As a result, samples density is expressed here as number of samples per millimeter, which is a measure free from the signature word lengths and from device-dependent features such as the screen size.

Figure 6 shows the precision, recall and f-measure of our system as functions of samples density. We used the f0.5-measure (with β equal to 0.5), which weights precision higher than recall. For our framework, increases in precision (that decrease the number of false positives) may be considered more important than increases in recall (that reduce the number of false negatives), since if a false instance is misclassified as true (i.e., a false positive), a forgery is accepted as genuine, while, if a true instance is misclassified as false (i.e., a false negative), the user has only to re-enter the signature. The X-axis is on a log scale for visual clarity. On the far right side of the curves data are not subsampled (all the samples are kept), while the further we go towards the left side of the curves the more the data are subsampled (up to a maximum of 20:1). In absolute terms, the average path length of the signatures of

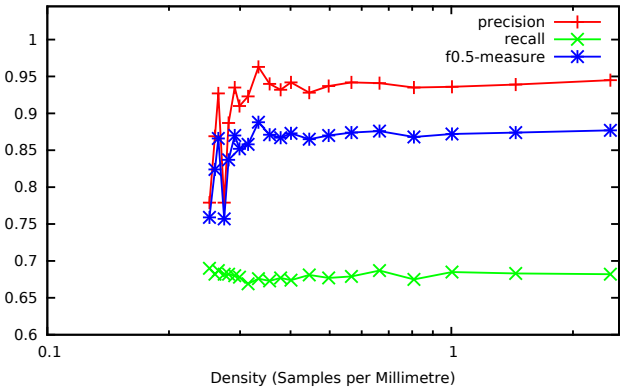


Fig. 6: Precision, recall and f-measure (with β equal to 0.5) as functions of density (expressed as samples per millimeter) shown in log scale (Viewed better in color).

our dataset was around 232 millimeters and the average number of samples (captured by the device for a single signature) was 581. Samples density ranges from around 2.5 to 0.125 samples per millimeter. This plot shows an interesting trend of decreasing precision and recall with decreasing samples density. The curves are almost constant initially (until a subsampling rate of 10:1 and a density of around 0.25), while at lower densities (or higher subsampling rates) the curves decrease sharply. This means that we are able to reduce the size of models of signatures by a factor of 10 (with respect to the sequence of samples acquired by the mobile device) without significant impacts on the overall precision and recall of the system. This, in turn, allows us to store models of signatures by means of barcodes, making our framework applicable to practical scenarios.

5. Conclusions

Our work presented a new HSV system for document signing and authentication, whose novelties lie mainly in the following aspects. First, we showed how to verify handwritten signatures so that signature dynamics can be processed during verification of every type of document (including paper documents). Secondly, we illustrated how to embed features extracted from handwritten signatures within the documents themselves, so that no remote signature database is needed. Thirdly, we proposed a method which is supported by a wide range of mobile devices so that no special hardware is needed. Finally, we showed how to reduce the size of models of signatures without significant impacts on the overall precision and recall of the system. In our experiments, precision and recall cross at 89.50% (SVC2004 dataset), at 95.40% (SigComp 2011 Dutch dataset), at 76.00% (SigComp 2011 Chinese dataset) and at 85.15% (custom dataset built using different smartphones). This is an interesting result, if noting that we used mobile devices (that is, no special-purpose

hardware) to capture the signature dynamics needed by our experiments.

Acknowledgments

This work has been partially supported by Filas S.p.A. (Rome, Italy; regional agency for the promotion of development and innovation; under project MYME - My Mobile Enterprise).

References

Andxor Corporation. View2sign. http://www.view2sign.com/supported-signatures.html, 2015. [Online; accessed 10-January-2015].

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

Firmani D., Italiano G.F., and Querini M. Engineering color barcode algorithms for mobile applications. In *13th International Symposium on Experimental Algorithms (SEA)*. 2014.

Grillo A., Lentini A., Querini M., and Italiano G.F. High capacity colored two dimensional codes. In *Proceedings of the 2010 International Multiconference on Computer Science and Information Technology (IMCSIT)*, pages 709–716. IEEE, 2010.

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

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

Kumar P., Singh S., Garg A., and Prabhat N. Hand written signature recognition & verification using neural network. *International Journal*, 3(3), 2013.

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

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

Pansare A. and Bhatia S. Handwritten signature verification using neural network. *International Journal of Applied Information Systems*, 1:44–49, 2012.

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

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

Querini M. and Italiano G.F. Color classifiers for 2D color barcodes. In *Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 611–618. IEEE, 2013. Full version submitted to the special issue of the conference in the Computer Science and Information Systems (ComSIS) journal.

Querini M. and Italiano G.F. Reliability and data density in high capacity color barcodes. *Computer Science and Information Systems (ComSIS)*, 2014.

Querini M., Grillo A., Lentini A., and Italiano G.F. 2D color barcodes for mobile phones. *International Journal of Computer Science and Applications (IJCSA)*, 8(1):136–155, 2011.

Querini M., Gattelli M., Gentile V.M., and Italiano G.F. Handwritten signature verification

with 2D color barcodes. In *Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 701–708. IEEE, 2014.

SigComp 2011. Final Report - Signature Verification Competition for Online and Offline Skilled Forgeries (SigComp2011). http://www.dfki.de/ liwicki/sigTutorial2011/2011-SigComp-report.pdf, 2015. [Online; accessed 10-January-2015].

SigComp 2011. Signature Verification Competition for Online and Offline Skilled Forgeries (SigComp2011). http://forensic.to/webhome/afha/SigComp.html, 2015. [Online; accessed 10-January-2015].

SutiDSignature. SutiDSignature. http://www.sutisoft.com/sutidsignature, 2015. [Online; accessed 10-January-2015].

SVC 2004. First Task Results. http://www.cse.ust.hk/svc2004/results-EER1.html, 2015. [Online; accessed 10-January-2015].

SVC 2004. Second Task Results. http://www.cse.ust.hk/svc2004/results-EER1.html, 2015. [Online; accessed 10-January-2015].

SVC 2004. Signature Verification Competition. http://www.cse.ust.hk/svc2004/, 2015. [Online; accessed 10-January-2015].

The Guardian. NSA Prism program taps in to user data of Apple, Google and others. http://www.theguardian.com/world/2013/jun/06/us-tech-giants-nsa-data, 2015. [Online; accessed 10-January-2015].

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

Xyzmo. Xyzmo signature solution. http://www.xyzmo.com, 2015. [Online; accessed 10-January-2015].