Paper

Hidden Signature for DTW Signature Verification in Authorizing Payment Transactions

Joanna Putz-Leszczyńska and Michał Kudelski

Research and Academic Computer Network NASK, Warsaw, Poland
Institute of Control and Computation Engineering, Warsaw University of Technology, Warsaw, Poland

Abstract—Traditional use of dynamic time warping for signature verification consists of forming some dissimilarity measure between the signature in question and a set of "template signatures". In this paper, we propose to replace this set with the hidden signature and use it to calculate the normalized errors of signature under verification. The approach was tested on the MCYT database, using both genuine signatures and skilled forgeries. Moreover, we present the real-world application of the proposed algorithm, namely the complete biometric system for authorizing payment transactions. The authorization is performed directly at a point of sale by the automatic signature verification system based on the hidden signature.

Keywords—dynamic time warping, hidden signature, payment transactions, signature verification.

1. Introduction

Despite an abundance of computers systems, handwritten signature is still widely used in everyday life. It is still essential for most of the financial transactions, notarial deeds, etc. However, there exists a considerable risk that someone can forge a signature. Forensic handwriting experts, whose discipline is forensic document examination, can tell a forgery from a genuine with almost 100% accuracy, but it is hard to imagine to seat a graphologist next to every bank worker. What is more, a forensic document examiners usually needs more than few seconds to perform his procedures. In spite of problems related to handwritten signature, it is still irreplaceable in many places. That is why there exists a great demand for automatic signature verification systems.

The first article describing automatic signature verification systems was published in 1979 by Herbst [1], that is more than 30 years ago. Even if the idea of automatic signature verification systems is a very good solution for banks and other places, it is still not commonly used in everyday life. This has a number of causes. First, in the very beginning, this idea was more like science fiction, and paying with plastic cards instead of paper money was not very common outside the US. The high cost and low quality of digitalizing tablets also curbed the on-line verification development. But it changed. Now, there exist tablets designed for signature verification. They are easy to get and their prices are not too steep.

The digitalizing tablets are important, because they alow for on-line signature capture. It means that the signature is represented as a sequence in time. The characteristic feature of signatures is that the instances of one signature can differ strongly between each other. This can be caused both by natural fluctuations and physical or emotional states. Two given instances can differ in amplitudes or values at certain points, but also in the signature dynamics. In other words, the time scales of two signatures can be different, which makes direct (point by point) comparisons impossible. For that issue, a nonlinear sequence alignment method is needed.

One approach that can be employed is dynamic time warping (DTW). The basic idea behind DTW is to "warp" the time dimension of two varying time sequences in a nonlinear manner. This allows for measuring dissimilarity between them.

We successfully used this approach in our original signature verification method. That is why in this paper we concentrate on methods based on DTW. These methods may differ in many important aspects. For example, DTW can be used only to compute the final score or to construct a classifier. However, what all such systems have in common is the template creation stage, in which either several (e.g., three, [2], [3]) or a single signature ([4], [5], [6]) is selected to represent the training set. While this selection has a big impact on the system quality, it is often arbitrary, and typically consists of choosing the signature(s) being closest (in some metric) to the rest of the training signatures. Some authors selected the template signature at random, or without elaboration on the statistic used for the selection [6].

The diversity of solutions shows two important problems associated with it. Both of these problems lie in template creation, which is seemingly very easy, and generally is limited to selecting a subset of a persons signatures. However – and this is the first problem – we do not know how many signatures should be selected for the template. Second, even if we resolve the first problem, we still do not know, which of the signatures should be selected.

The approach presented in this paper makes an effort to meet the mentioned needs. It is based on the well known property of least square solutions, namely, that the average minimize the squared norm distance to all averaged elements. While a direct averaging of signatures is not possible (even signature instances of the same person may have different length and may differ not only in magnitudes of the measured quantities but also in their "local speed"),

we may transform all training signatures to a common space by warping and then find their average in this space. This approach carries on various properties of least squares, while bringing in a possibility of local variation by proper warping. The averaged warped signature is called the hidden signature. It is an artificial signature which has a feature of minimizing the mean dissimilarity between itself and the signatures from the training set. By the limit laws of probability theory, the hidden signature is "close" to the mean value of the real signature after warping, thus assuring the proper level of invariance with the training set. The hidden signature can be calculated numerically.

Authorizing payment transactions is a straightforward area of application for signature verification systems. People are accustomed to using their signature to confirm the identity when paying with debit cards or credit cards. However, the verification method that is commonly used, namely the visual comparison performed by a cashier, leaves much to be desired. The sales staff is usually barely educated in the field of signatures verification and the risk of both false acceptance and false rejection may be significant. The above facts argue for the need of developing the automatic signature verification system for the purpose of authorizing payment transactions.

The paper is organized as follows. In the next Section 2, we show the proposed hidden signature estimation methodology. In Section 3, we introduce the verification algorithm that employs hidden signature for verification. Section 4 presents the application of the presented algorithm in authorizing payment transactions. Section 5 concludes the paper.

2. Hidden Signature Estimation

The hidden signature approach extends the least squares approach. In the least squares approach, the least squares model approaches the expected value as the number of observations increases to infinity, and for independent data it is also the most effective (in the statistical sense, e.g., it leads to a given error variance) for the least sample size. Since our approach is a conjunction of the least square modeling and optimal warping, the resulting model has some optimality properties, and it approaches the expected value of the signature, calculated with the use of warping. In other way, it approaches a form which is independent on particular signature instances. It is also independent on sampling frequency (which is not true for other warping models). This is why we may call it a }perfect instance of a signature.

In [7], we proposed two main directions for hidden signature estimation. First, *iterative point-by-point averaging*, and second, *evolutionary algorithms*. Each of those algorithms is constructed as an iterative procedure that alternates the warping steps and the averaging steps. Let $(G_1,\ldots,G_N)$ be a set of training signatures (the results presented here were calculated for $N = 10$). At each iteration stage, the algorithm corrects its approximation of the hidden signature $G^* = g^*(t),\ \ t = 1,2,\ldots,M_{G^*},\ \ M_{G^*} \in \mathbb{N}$, by minimizing the quality index:

$$\mathscr{V}_{\mathscr{H}} = \min_{j=1\ldots H} \sum_{i=1..N} \hat{\mathscr{D}}(G^*{}_j, G_i)\,, \tag{1}$$

where $\hat{\mathscr{D}}(G_1,G_2)$ denotes the dissimilarity measure between two signatures G_1 and G_2 (wide Eqs.(6), (7)).

Regardless of the estimation method, the resulting hidden signature depends on the coordinates selected to compute the dissimilarity measure and the warping path. This method was first proposed in [7] together with several others. All those methods yielded comparable results; the method presented was chosen as preferred because of its computation times, which were considerably shorter than the times of the other methods.

2.1. DTW Background

Dynamic time warping is used to compute the optimal alignment, written in a form of *warping path w*, between variable length discreet sequences:

$$R = r(t),\ \ t = 1,2,\ldots,M_R,\ \ M_R \in \mathbb{N} \tag{2}$$

and

$$G = g(\tau),\ \ \tau = 1,2,\ldots,M_G,\ \ M_G \in \mathbb{N}\,. \tag{3}$$

A warping path w is a parametric discrete curve parameterised by ℓ, that aligns R and G via a point-to-point mapping (Fig. 1). The warping path can be defined as:

$$w(\ell) = [w_t(\ell)\ \ w_\tau(\ell)]^T;\ \ell \in 1,\ldots,L_w\,, \tag{4}$$

where

$$\begin{aligned} w_t(\ell) &\in \{t\},\ \ t = 1,\ldots,M_R\,,\\ w_\tau(\ell) &\in \{\tau\},\ \ \tau = 1,\ldots,M_G\,, \end{aligned} \tag{5}$$

where $w_t(\ell)$ and $w_\tau(\ell)$ are two parameterised by ℓ functions of aligned sequence indexes. Thus, $w(\ell)$ maps successive steps ℓ to a points (t,τ) where $1 \le t \le M_R$ and $1 \le \tau \le M_G$. The warping path length L_w is a consequence of the minimisation process that minimise the overall distortion Eqs. (6 and 7).

The overall distortion $\mathscr{D}(R,G,w)$:

$$\mathscr{D}(R,G,w) = \sum_{\ell=1}^{L_w} d\left(r(w_t(\ell)), g(w_\tau(\ell))\right) \tag{6}$$

is based on the sum of local distances $d\left(r(w_t(\ell)), g(w_\tau(\ell))\right)$ between sequences elements at points, belonging to warping path. In the set of possible warping paths $w \in \mathrm{W}$, we can find the *optimal warping path* $\hat{w}$, such that its associated distortion (*dissimilarity measure*) is at a minimum:

$$\begin{aligned} \hat{\mathscr{D}}(R,G) &= \min_{w\in\mathrm{W}} \mathscr{D}(R,G,w)\,,\\ \hat{w} &= \arg\min_{w\in\mathrm{W}} \mathscr{D}(R,G,w)\,. \end{aligned} \tag{7}$$

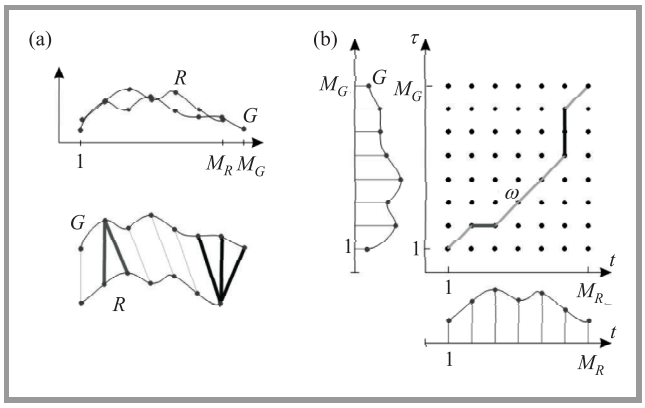

Fig. 1. The compared curves, and its alignment visualization (a). Warping path on a two-dimensional grid: discrete warped time $\tau \in [0, M_G]$ versus discrete reference time $t \in [0, M_R]$ (b). The compared curves are also plotted (the reference curve is plotted vertically).

Allowing for differences caused by time warping, we still encounter differences in sequence values at the aligned time instances $d(r(w_t(\ell)), g(w_\tau(\ell)))$, where $t = 1, \ldots, M_R$ and $\tau = 1, \ldots, M_G$. These differences can be measured at the aligned times using typical distance metrics, for instance with the L_2 distance but it is not limited to:

$$d(a,b) = \|a-b\|_2 . \tag{8}$$

2.2. Transformation from One Time Space to Another

The iterative point-by-point averaging (IPPA) is a method built on the concept of an average signature. If the signatures from a training set were the same length, the hidden signature could be simply computed as an average in each point.

However, like it was presented before, in each of the persons signature realizations the time flows differently. To overcome this problem we proposed the algorithm that transforms the training signatures into one common time space, thus obtaining signatures of the same length. The common time space, means that both signatures have the same length and the optimal warping path has only diagonal steps, which means that alignment between signatures is liner (Fig. 2b). The result of proposed transformation is presented in Fig. 2.

We assumed that we need a simple and fast transformation from one signature time space to a time space of second signature. Therefore we proposed the following procedure, transforming signature G into a time space of signature R, thus obtaining $G' = g'(t), \;\; t = 1, \ldots, M_R$:

$$\hat{w} = \arg\min_{w \in \mathrm{W}} \mathscr{D}(R, G, w), \tag{9}$$

$$g'(t;\hat{w}) = \frac{\sum_{\ell: w_t(\ell)=t} g(\hat{w}_\tau(\ell))}{|l : w_t(l) = t|}, , \quad t = 1, .., M_R ,$$

where t denotes the time of $R = r(t), \;\; t = 1, \ldots, , M_R$ and operator $|.|$ set cardinality. The illustration of this transformation is presented in Fig. 3.

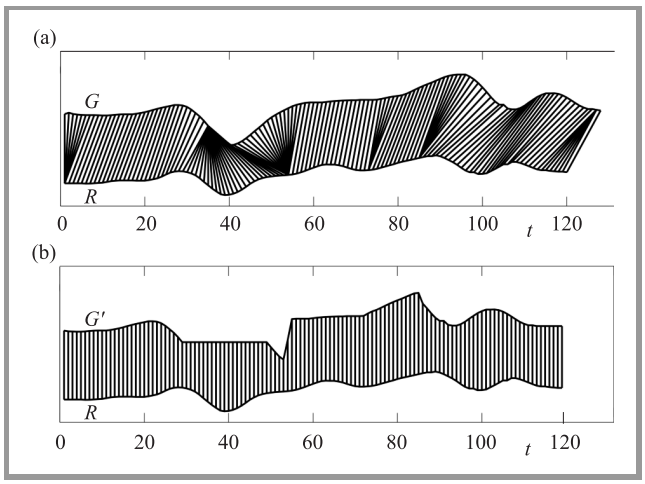

Fig. 2. A transformation into a different signature time space: (a) the DTW nonlinear alignment of two time sequences R and G; (b) the linear alignment between R and G' (G in R time space).

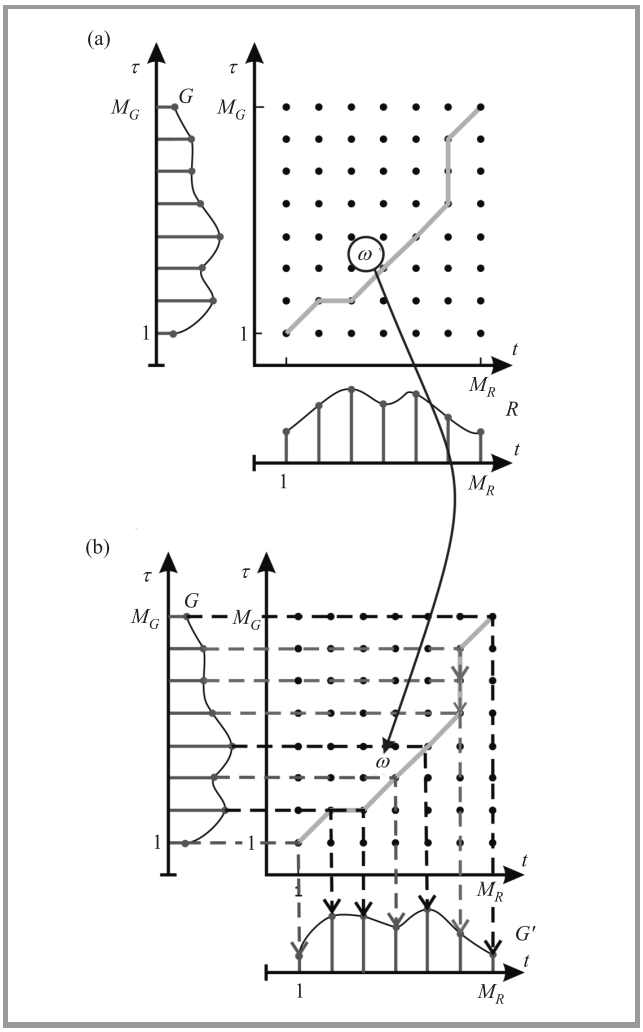

Fig. 3. A transformation into a different signature time space: (a) calculation of the warping path w between R and G; (b) G is transformed into the time space of R using the warping path w.

2.3. Iterative Point-by-Point Averaging

The Eq. (9) denotes a method of a signature transformation into a time space of another signature. However, it is assumed that the target time space is known. We assume that during each successive iteration only one estimation of hidden signature is calculated. Additionally, the hidden signature size is given by M, that can be set equally for the all users templates, or individually for each. In the beginning, we proposed to calculate the hidden signature length as the simple average (rounded down) of signatures from the training set, independently for each user, namely:

$$M = \frac{1}{N}\sum_{n=1}^{N} M_n\,. \tag{10}$$

Finally, we assume that in the initial iteration, all training signatures $\{G_1,\ldots,G_N\}$ are linearly graduated into a time length M, the assumed size of a hidden signature (Fig. 4).

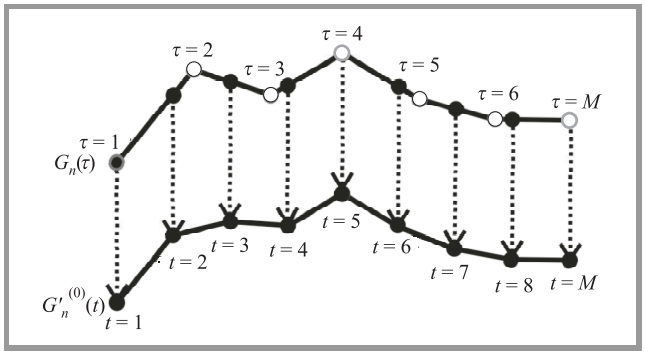

Fig. 4. Linearly graduation into a time length M.

As the result we obtain N training signatures of length M, namely $\left\{G_1'^{(0)},\ldots,G_N'^{(0)}\right\}$.
Detailed mathematical description has been omitted for the sake of simplicity. Using this starting set, it is possible to calculate the initial estimation of hidden signature $G^{*(0)}$:

$$g^{*(0)}(t) = \frac{1}{N}\sum_{n=1}^{N} g_n'^{(0)}(t), \quad t = 1,\ldots,M\,. \tag{11}$$

Then, in each successive iteration $k = 1,2,\ldots$ newly computed hidden signature approximation is used to calculate N warping paths between itself and the signatures from the training set, thus minimizing the quality index:

$$\hat{w}_n^k = \hat{w}_n^k(G^{*(k-1)},G_n) = \arg\min_{w} \mathscr{D}(G^{*(k-1)},G_n,w), \quad n = 1,\ldots,N \tag{12}$$

then, the warping paths are used to transform the training signatures into the new hidden signature approximation $G^{*(k)}$ space using DTW. As the result of this operation, we obtain N signatures with the lengths equal to the hidden signature length M. We can then calculate a new hidden signature approximation as a weighted mean for each point:

$$g^{*(k)}(t) = \frac{1}{N}\sum_{n=1\ldots N} g_n'(t;\hat{w}_n^{(k)}),, \quad t = 1,\ldots,M \tag{13}$$

where g_n' is calculated from Eq. (9). This process repeats in every iteration, because changing warping paths imply changes in the hidden signature approximation. Visualization of this method is presented in Fig. 5.

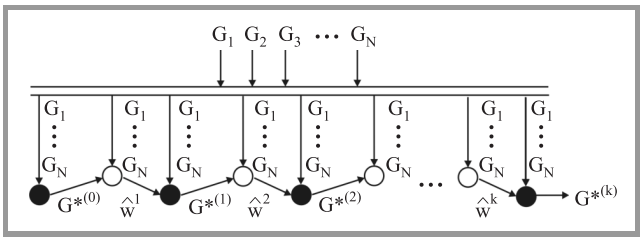

Fig. 5. The iteration process of iterative point by point averaging with one minimization at each stage.

To apply the hidden signature concept, we used it in our two stage approach, which employs the DTW for computing the final score. In [7], we showed that replacing the representative of a training set with the hidden signature allows to achieve better verification results in the existing global on-line system. Here, we extend our approach to exploit the properties of the hidden signature by using its certain statistics.

3. Verification Algorithm – Error Signal Approach

Error signals approach is a simple method dedicated for on-line verification with hidden signature.

3.1. Template

For the new approach to DTW, we proposed a parametric template:

$$\lambda_u = \{G^{*(u)},\sigma^{(u)}\}, \tag{14}$$

where u denotes the user identifier. This template consist of the hidden signature G^* and its corresponding standard deviation σ sequence. Hidden signature is created form users u training signatures $\{G_1,\ldots,G_N\}$. Standard deviation is calculated as the roots of mean-square errors between the hidden signature and the training signatures at each hidden signature point, independently for each feature $f \in \{\Delta x,\Delta y,p\}$ sequence:

$$\sigma_f(i) = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(g_n'(i,f) - g^*(i,f))}\,, \quad i = 1,\ldots,M, \tag{15}$$

where feature sequences are used according to definition of signature for each feature f sequence as:

$$G_f = g(\tau,f), \quad \tau = 1,2,\ldots,M_G, \quad M_G \in \mathbb{N}. \tag{16}$$

Table 1
On-line systems comparison

Authors	Year	FAR [%]	FRR [%]	EER [%]	No. of users	Database
Quan *et al.* [9]	2006			7	100	MCYT_Online
Miguel-Hurtado, O. *et al.* [5]	2007			8	100	MCYT_Online
Garcia-Salicetti, Dorizzi [10]	2007			3.37	100	MCYT_Online
Guru, Prakash [11]	2007	9.16	5.42	5.3	100	MCYT_Online
Galbally, Ortega-Garcia [12]	2007			3.5	330	MCYT_330
Faundez-Zanuy [13]	2007			5.4	330	MCYT_330
Nanni and Lumini [14]	2008			5.2	100	MCYT_Online
Yanikoglu and Kholmatov [15]	2009			7.22	100	MCYT_Online
This work	2010	**1.74**	**1.82**	**1.72**	100	MCYT_Online

The features f were selected, due to our previous experiments, presented in [8]. Additionally, we want the system to work with LCD tablets, and they allows only for x, y and pressure acquisition.

3.2. Verification Stage

In the verification stage, a signature in question G_q, after the transformation into the hidden signature space G'_q Eq. (9), can be standardized for every feature, by point-by-point subtraction of the hidden signature G^* as the sequence of mean values, and division by the standard deviations σ, given by Eq. (15):

$$g''_q(i,f) = \frac{g'_q(i,f) - g^*(i,f)}{\sigma_f(i)}\ ,\ i = 1,\ldots,M\,, \qquad (17)$$

where $f \in \{\Delta x, \Delta y, p\}$. The new resulting standardized sequences $G''_{q,\Delta x}$,$G''_{q,\Delta y}$,$G''_{q,p}$, are called the error signals, because they represent a normalize errors between signature in question and hidden signature at each point of a sequence.
We propose the final $SCORE^H$ used in at the verification stage. For the signature in question G_q its selected values are used for the scores calculations:

$$SCORE^H(G''_q; G^*, \sigma) = \left\| \left[s^2(G''_{q,\Delta x})\ \ s^2(G''_{q,\Delta y})\ \ s^2(G''_{q,p}) \right]^T \right\|_2 , \qquad (18)$$

where: $\overline{Q}$ denotes the arithmetic mean value of the Q sequence, $s^2(Q)$ denotes the standard variance of non-zero values of Q sequence, and $\|q\|_2$ denotes the Euclidean norm of q.
The logical sense of these scores is that the error signal G''_f is a resulting signal of normalization with a use of the hidden signature. This means, the closer is the verified signature G^Q to the hidden signature, the lower are the values at each point of the error signal.
The final stage of verification is very simple. The signature in question G_q is accepted if following condition is satisfied:

$$SCORE^H \leq \theta\,.$$

The threshold θ is calculated during the estimation phase.

3.3. Testing Methodology and Results

The tests were conducted on the MCYT on-line database [16] (MCYT_Online for 100 users). Our methodology was to evaluate in simulated real-world conditions. For this goal, the database was divided into two parts. The *estimation data* used for the estimation phase included 40 persons, and the remaining 60 person data was used as the *testing data*. The first part of the data was used for the estimation of the equal error rate (EER) and the corresponding threshold θ. In the testing phase, we checked the repetitiveness of our approach. We used the remaining 60 persons of the 100-people database for testing data, and fixed the same threshold θ as that adjusted for the first database part. In practice, it was the level of threshold θ for the EER obtained in the estimating phase. We then obtained false acceptance rate (FAR) and false rejection rate (FRR). If results of FAR and FRR are close to the EER obtained for the first part of the database, then they can be trusted. The presented results were obtained after 1000 different distributions of database for phases:

- estimating phase: $EER = 1.72 \pm 0.25\%$;
- testing phase:
  - $FRR = 1.82 \pm 0.86\%$,
  - $FAR = 1.74 \pm 0.3\%$ (skilled forgeries),
  - $FAR = 0.06 \pm 0.05\%$ (random forgeries).

4. Implementation

We propose to employ the automatic signature verification system based on the hidden signature for the purpose of authorizing payment transactions. In NASK Biometric Laboratories, we designed an adaptation of the verification algorithm presented in Section 3 to the platform of payment terminals. Together with the MCX Systems Company that specializes in developing software for payment terminals, we prepared the implementation that can be run directly on the terminal used at a *point of sale* (POS). We also built a prototype of a complete biometric system for authorizing

payment transactions. The system consists of three components: an enrollment stand, a POS terminal with biometric signature verification and a technique for storing biometric templates.
The automatic signature verification system can replace the cashier in performing the verification or can serve as a decision support system for the cashier.

4.1. Hardware Platform

Payment terminals make a demanding programming environment due to a limited computational power, reduced memory and (usually) only a software support for floating point operations.

Fig. 6. Terminal with tablet.

For the purpose of our prototype system, we used the Ingenico i5100 terminal. It is equipped with ARM7TDMI processing unit with computational power of about 30 MIPS. There is 2 MB of RAM memory available, yet only 256 kB can be used by the algorithm due to the requirements of other applications and the operating system. The terminal is connected with the Wacom STU-500 LCD signature tablet that is specially designed to capture digital signatures.
The specification of the terminal, together with time constraints related to payment transactions, required developing a specialized implementation of the verification system.

4.2. DTW Implementation

An effective implementation of the DTW algorithm is crucial from the performance point of view. The comparison of two signatures in a given space is relatively cheap, yet finding the optimal warping path requires solving a dynamic programming task with the quadratic time and space complexity.
In order to overcome the problem, we performed numerous steps that allowed us to reduce time and memory requirements of the DTW algorithm. First, we limited the maximum length of signatures to 500 points (longer signatures were subjected to downsampling). Second, we employed the Sakoe-Chuba Band constraint [17] to limit the number of cells that are evaluated in the cost matrix. We chose the window size according to a trade-off between memory requirements and the verification quality (only a minor increase of the ERR was observed after applying the constraint on the MCYT database). Finally, we reduced the memory requirements by using two moving buffers combined with a bitmap of the cost matrix (Fig. 7).

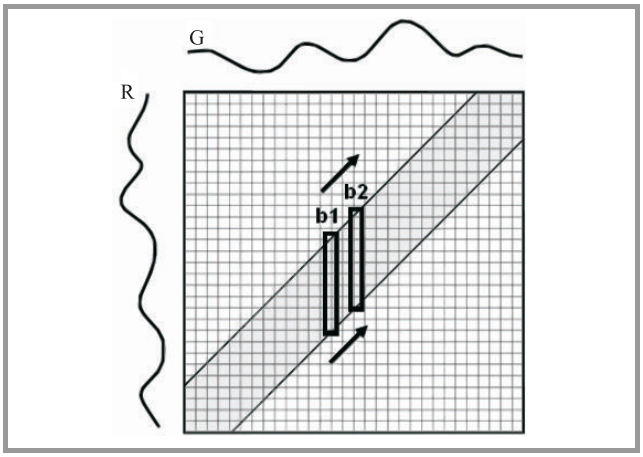

Fig. 7. Visualization of a discrete DTW implementation using a cost matrix. The number of cells that are evaluated is limited and two moving buffers are used to reduce the memory requirements.

The idea of using moving buffers is the following. Dynamic programming algorithms can be visualized as "filling an array". In DTW, each cell of the array represents the total cost of the best warping path reaching this cell (i.e., the partial warping path ending in this cell). The array is filled starting from the lower-left corner. After the whole array is filled, the value in the upper-right corner represents the minimal cost of the whole warping path. It is possible to reconstruct the best warping path by moving back from the upper-right corner, choosing always the neighboring cell with the lowest (partial) cost value (according to the Bellman's Dynamic Programming rule).
However, one may notice that it is not necessary to remember the whole array. Suppose that from a given cell we can move only in three directions (right, up and diagonal upper-right). In order to fill the nth column of the array (starting form the lowest cell), we need only to know the $(n-1)$th column. Hence, instead of keeping the whole cost matrix in memory, we may use two moving buffers, each one to store one column. After performing the whole run, we obtain the same optimal cost in the upper-right corner of the matrix. In addition, we need to store the information that would allow us to reconstruct the best warping path. We achieve this by remembering in each cell the information about the previous cell of the warping path. We code such information using two bits (as there are only three directions possible: left, down and diagonal lower-left) and

we store it an additional bitmap. As a result, instead of keeping all values of the cost matrix in memory (usually floating point values with double precision), we need only the memory for two moving buffers and the bitmap.

4.3. Performance and Memory Requirements

A combination of the techniques described in the previous section allowed us to obtain the reasonable memory requirements presented in Table 2.

Table 2
Memory requirements

Source of the requirement	Memory [kB]
Buffer for the raw signature	18
Buffer for the template	6
Temporary buffer for an exemplary signature	3
Temporary buffer for a candidate signature	3
Buffer for the warping path	5
Bitmap	20
Two moving buffers	1
Total:	**56**

The effective implementation of the DTW algorithm determined the following running times on the terminal:

- verification of a single signature below 5 s,
- template creation (from 5 signatures): 2–5 min, depending on the signatures length.

From the practical applications perspective, only the verification time is important as the enrollment procedure is usually performed outside the terminal. The verification time of 5 s is satisfactory as far as payment transactions are concerned. Moreover, this time may be further reduced using more powerful modern terminals.
The correctness of the results has also been verified: we performed tests on the MCYT database using the terminal for verifications. We observed some minor differences in the obtained scores (resulting from different floating point implementations), yet the verification results were identical on both PC and the payment terminal platforms.

4.4. Prototype of the System

Basing on the running times obtained in Subsection 4.3, we proposed the following operational scenario: signature templates should be created using a specialized enrollment stand (e.g., equipped with a PC computer) and payment terminals should only perform signatures verification during payment transactions. According to this scenario, we build a prototype biometric system for authorizing payment transactions (Fig. 8).
The first element of the system is an enrollment stand. The stand is composed of a PC computer, a smart card reader

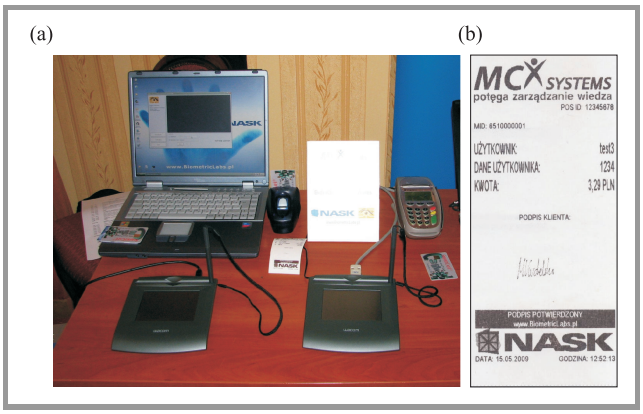

Fig. 8. Prototype of the system: a demonstration stand (a) and a sample printout (b).

and the STU-500 tablet for gathering signatures. The process of enrollment is controlled by a specialized application. The functionality of the application includes:

- creating templates basing on exemplary signatures and performing signatures verification,
- analyzing the template's consistency,
- saving the template in the internal database or on a smart card based on the JavaCard technology,
- making templates accessible via the HTTP protocol,
- visualizing signatures in real-time during writing.

The second element of the system is a payment terminal with a similar STU-500 tablet. The application installed on the terminal allows to simulate a real payment transaction at a point of sale: the cashier scans the customer's credit card, enters the amount of the transaction and waits for the authorization. At the end, the confirmation of the transaction is printed (Fig. 8b).
The transaction is authorized using automatic signature verification. The application running on the terminal downloads the template, captures the signature in question from the tablet and performs the verification. Both magnetic cards and smart cards are supported by the terminal.
The last element of the system is a mechanism of storing and transferring biometric templates. There are two alternative ways of storing templates, namely an *on-line scheme* and an *off-line scheme*. In the on-line scheme, templates are stored in a database and can be accessed via the HTTP protocol. The terminal scans the unique number of the customer's magnetic card, connects to the database using Ethernet connection, downloads the customer's template and performs verification. In the off-line scheme, the customer's template is stored in the customer's smart card. The terminal reads the template directly from the card and performs verification.

5. Conclusions

The presented novel approach to signature verification draws from the hidden signature concept and differs from the traditional DTW-based methods. This approach is based on replacing the template signatures with an artificial signature – the hidden signature.

The proposed method can be regarded as model-based, since the hidden signature can be thought as an approximation of a perfect signature of a given person.

The verification algorithm is based on computing error signals between the signature in question and the model. In comparison to results of other systems, obtained on the MCYT database (see Table 1), the complex system for hidden signature results is placed very high.

While the results obtained in tests on the MCYT database are promising, the method is mostly based on an engineering approach to error signals processing, and could benefit from more sophisticated comparison algorithms.

The overall simplicity allows this method to be used in mobile or embedded systems. One of them is the system for signature verification in authorizing payment transactions. The effective implementation of the proposed approach allowed to meet the memory and computational power constraints of payment terminals, as well as the time constraints of payment transactions. The automatic signature verification based on the hidden signature can be performed directly on the payment terminal at a point of sale.

Acknowledgements

This paper has been financed by the Ministry of Science and Higher Education grant OR00 0026 07 "*A platform of secure biometrics implementations in personal verification and identification*".

References

[1] N. Herbst and C. Liu, "Automatic signature verification based on accelerometry", in *IBM J. Res. Develop.*, pp. 245–253, 1977.

[2] D. Sakamoto, H. Morita, T. Ohishi, Y. Komiya, and T. Matsumoto, "On-line signature verifier incorporating pen position, pen pressure and pen inicliation trajectories", in *Proc. 3rd Int. Conf. AVBPA 2001*, Halmstad, Sweden, 2001, pp. 318–323.

[3] A. Pacut and J. Putz-Leszczyńska, "Dynamic time warping in subspaces for on-line signature verification", in *Proc. 12th Conf. Int. Graphonom. Soc. IGS 2005*, Salerno, Italy, 2005, pp. 108–112.

[4] A. Kholmatov and B. Yanikoglu, "Identity authentication using improved online signature verification method", *Pattern Recogn. Lett.*, vol. 26, no. 15, pp. 2400–2408, 2005.

[5] O. Miguel-Hurtado, L. Mengibar-Pozo, M. Lorenz, and J. Liu-Jimenez, "On-line signature verification by dynamic time warping and gaussian mixture models", in *41st Ann. IEEE Int. Carnahan Conf. Secur. Technol.*, Ottawa, Canada, 2007, pp. 23–29.

[6] B. Fang, C. Leung, Y. Tang, K. Tseb, P. Kwokd, and Y. Wonge, "Off-line signature verifcation by the tracking of feature and stroke positions", *Pattern Recogn.*, vol. 36, pp. 91–101, 2003.

[7] A. Putz-Leszczyńska, J.and Pacut, "'Hidden signature' – a new solution for on-line verification ", in *42nd Ann. IEEE Int. Carnahan Conf. Secur. Technol.*, Prague, Czech Rebulic, pp. 68–73, 2008.

[8] J. Putz-Leszczyńska, "On-line signature verification using dynamic time warping with positional coordinates", in *Proc. SPIE – Vol. 6347, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments*, R. S. Romaniuk, Ed., 2006 (doi: 10.1117/12.714578).

[9] X.-L. X. M. R. L. Z.-H. Quan, D.-S. Huang and T.-M. Lok, "Spectrum analysis based on windows with variable widths for online signature verification", in *Proc. 18th Int. Confe. Pattern Recogn. ICPR 2006)*, Hong Kong, China, 2006, vol. 2, pp. 1122–1125.

[10] 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. Sys. Man. Cybernetics, Part B: Cybernetics*, vol. 37, no. 5, 2007.

[11] D. Guru and H. Prakash, "Symbolic representation of on-line signatures", in *Proc. Int. Conf. Comput. Intellig. Multimedia Appl. 2007*, Sivakasi, India, 2007, pp. 313–317.

[12] J. Galbally, J. Fierrez, M. Freire, and J. Ortega-Garcia, "Feature Selection Based on Genetic Algorithms for On-Line Signature Verification", in *Proc. IEEE Worksh. Autom. Identif. Adv. Technol. 2007*, Alghero, Italy, 2007, pp. 198–203.

[13] M. Faundez-Zanuy, "On-line signature recognition based on VQ-DTW", *Pattern Recogn.*, vol. 40, no. 3, pp. 981–992, 2007.

[14] L. Nanni and A. Lumini, "A novel local on-line signature verification system,", *Pattern Recogn. Lett.*, vol. 29, no. 5, pp. 559–568, 2008.

[15] B. Yanikoglu and A. Kholmatov, "Online signature verification using fourier descriptors", *EURASIP J. Adv. Sig. Proces.*, 2009.

[16] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon, J. Gonzalez, M. Faundez-Zanuy, V. Espinosa, A. Satue, I. Hernaez, J.-J. Igarza, C. Vivaracho, D. Escudero, and Q.-I. Moro, "MCYT baseline corpus: a bimodal biometric database", *IEE Proc.-Vis. Image Sig. Process.*, vol. 150, no. 6, pp. 3412–3426, 2003.

[17] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition", *IEEE Trans. Acoust., Speech Sig. Proces.*, vol. 26, no. 1, pp. 43–49, 1978.

Joanna Putz-Leszczyńska received her M.Sc. in 2004 from the Faculty of Electronics and Information Technology of the Warsaw University of Technology, Poland. She is presently a Ph.D. student at the Institute of Control and Computation Engineering at the Faculty of Electronics and Information Technology of the Warsaw University of Technology. Since 2003 she works as a Research Assistant at Biometric Laboratory of Research and Academic Computer Network NASK. She is interested in biometrics, identification, security systems and global optimization heuristics.

e-mail: J.Putz@elka.pw.edu.pl
Research and Academic Computer Network (NASK)
Wąwozowa st 18
02-796 Warsaw, Poland

Institute of Control and Computation Engineering
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland

Michał Kudelski received his M.Sc. in 2005 from the Faculty of Electronics and Information Technology of the Warsaw University of Technology, Poland. He is presently a Ph.D. student at the Institute of Control and Computation Engineering at the Faculty of Electronics and Information Technology of the Warsaw University of Technology. Since 2008, he is also a Research Assistant at the Biometric Laboratory of Research and Academic Computer Network NASK. He is interested in artificial intelligence, adaptive systems, biometrics and related areas.

e-mail: M.Kudelski@elka.pw.edu.pl
Research and Academic Computer Network (NASK)
Wąwozowa st 18
02-796 Warsaw, Poland

Institute of Control and Computation Engineering
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland