Feature Selection in a Low Cost Signature Recognition System Based on Normalized Signatures and Fractional Distances[*]

C. Vivaracho-Pascual[1], J. Pascual-Gaspar[2], and V. Cardeñoso-Payo[1]

[1] Dep. Informática, U. de Valladolid
{cevp,valen}@infor.uva.es
[2] Grupo ECA-SIMM, Universidad de Valladolid
pascualgaspar@gmail.com

Abstract. In a previous work a new proposal for an efficient on-line signature recognition system with very low computational load and storage requirements was presented. This proposal is based on the use of size normalized signatures, which allows for similarity estimation, usually based on DTW or HMMs, to be performed by an easy distance calcultaion between vectors, which is computed using fractional distance. Here, a method to select representative features from the normalized signatures is presented. Only the most stable features in the training set are used for distance estimation. This supposes a larger reduction in system requirements, while the system performance is increased. The verification task has been carried out. The results achieved are about 30% and 20% better with skilled and random forgeries, respectively, than those achieved with a DTW-based system, with storage requirements between 15 and 142 times lesser and a processing speed between 274 and 926 times greater. The security of the system is also enhanced as only the representative features need to be stored, it being impossible to recover the original signature from these.

1 Introduction

Although the evolution of the technology leads to a continuous increase in the data storage and processing capacity of computers while their price decreases, the reduction in costs will always be an important matter in commercial applications and in large-scale systems, but also in resource-limited systems like smart-cards or PDA terminals.

In on-line signature recognition, this reduction usually lies in the characteristic extraction stage. The different methods that can be found in the literature to reduce the signature representation can be grouped into: (1) The use of global features, (2) Size reduction of the time series representing the signature.

The first option achieves worse results than using local features. Then, it is usually used in a first, quick but weak decision [16], or fusing with local

---

information [4]. So most of the research effort focuses on the second option, where the methods used to reduce the time series size can be classified as follows:

- Selection of representative or critical points and information extraction [10][16].
- Signature resampling [7][9][11], generally performed before feature extraction. Theoretically, sampling rates higher than 40-60 Hz would produce redundant information [14]. However, the results are not always in accordance with that affirmation as, while some authors report better results [7][11] with the application of resampling, some others report worse performance [8].
- Signature resampling + representative points [1][7][12].

In [15] we have applied resampling techniques to normalize the signature size, being possible, then, that the similarity between signatures can be computed as a simple distance measurement between vectors. Normally, this distance calculation is based on the use of the Euclidean one. When data are high dimensional, however, $fractional$ p-norms (Minkowski-like norms with an exponent p less than one) are, in general, less sensitive to he concentration phenomenon [6], performing better, as can be seen in our work [15].

Here, we present a first approximation to the idea shown in the third item of the previous list. The goal is to select the signatory's most representative points/features from the normalized signatures. In this first approximation we have focused on the feature selection, considering the signature as a vector of p (points) $\times$ f (features per point) components.

This problem is usually approached by statistical methods, such as, for example, Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA). Here, we have defined the most representative features as the most stable in the training set, that is, those that show less intra-signatory variability. This selection method is similar to that shown in [2]. Each feature variability/dispersion is measured by its standard deviation (σ). Then, once a threshold, $\sigma_{threshold}$, is fixed, only features with $\sigma < \sigma_{threshold}$ will be used to compute the distance calculation (see section 3).

This approach has the additional advantage of increasing the system's security, since only the representative features need to be stored, it being impossible to recover the original signature from these.

The rest of the paper is organized as follows. In Sec. 2 the main characteristics of the baseline system, as shown in [15], will be described. The feature selection method can be seen in Sec. 3. After describing the experimental environment (Sec. 5), the results achieved will be shown, comparing them with those achieved both with the baseline system and the reference system [3]; the computational load and storage requirements will also be compared. Finally, the conclusions and future work can be seen in Section 6.

2   Baseline System

The approach shown in [15] is based on the use of normalized signatures, that is, the signatures are resampled (using a linear interpolation method) to get a fixed

size (number of points). This idea is based on the property of the "diagonal path" alignment of genuine signatures in DTW. That is, as is shown in the above reference work, the alignment path is more linear (diagonal) for genuine signatures than for those that are not.

Once the signatures have the same length, the problem of measuring similarity between signatures can be computed in a more straightforward way as a simple distance measurement between vectors. Due to the high dimensionality of these, to alleviate the concentration phenomena of the Euclidean distance, the use of *fractional* p-norms is proposed, achieving recognition rates comparable with the state-of-the-art classifiers.

This proposal can be seen as a simplification of the DTW-based systems.

In [15] the proposed system was tested with regard to several different configurations, proposing the final following two for the verification task:

- **Baseline_Ver_Sytem.** p value of the p-norm equals 0.3 and normalized signature size equals 50 points. The n signatures $\{tr_1, \ldots, tr_n\}$ of the training set are used as signatory model. Then, given a test sample (te), the distance with regard to each of the n training signatures $(d(tr_i, te))$ is calculated, these n distances achieve a final measured similarity by means of the min function $(\min_i(d(tr_i, te)))$.
- **Baseline_Ver_Sytem_Soft.** p value of p-norms equals 0.3 and normalized signature size equals 25 points. The n training signatures $\{tr_1, \ldots, tr_n\}$ are fused to create a prototype signature, that is called *averaged template (AT)*. The AT is the mean of the training signatures, that is, if a training signature is composed of m feature vectors $(tr^i = \{x_1^i, \ldots, x_m^i\})$, each of them with t components $(x_j^i = (x_{j1}^i, \ldots, x_{jt}^i))$, then the template signature will be composed of m vectors $(AT = \{\overline{x}_1, \ldots, \overline{x}_m\})$ calculated as shown in Eq. 1. The computational load and storage requirements of this option are lower than those in the *Baseline_Ver_Sytem*.

$$\overline{x}_j = \left(\frac{\sum_{i=1}^n x_{j1}^i}{n}, \ldots, \frac{\sum_{i=1}^n x_{jt}^i}{n} \right) \tag{1}$$

3   Feature Selection

The procedure proposed to select the features with less variability over the training set is shown here.

In the feature extraction stage, a set X^i of m (m being the size of the normalized signature) feature vectors (normalized signature points) of t components, $x_j^i = (x_{j1}^i, ..., x_{jt}^i)$, are extracted from the signature i. Then, this stage output can be seen as a matrix where each row is a feature vector:

$$X^i = \begin{pmatrix} x_{11}^i & \cdots & x_{1t}^i \\ \cdots & & \cdots \\ x_{m1}^i & \cdots & x_{mt}^i \end{pmatrix} \tag{2}$$

The selection method is similar to that shown in [2], being performed over each column as follows.

Given n training signatures, each feature x_{jl}^i distribution is modeled by means of a gaussian, with mean μ_{jl} and standard deviation σ_{jl} calculated as shown in Eq. 3.

$$\mu_{jl} = \frac{\sum_{i=1}^n x_{jl}^i}{n}$$

$$\sigma_{jl} = \sqrt{\frac{\sum_{i=1}^n (x_{jl}^i - \mu_{jl})^2}{n}}$$

(3)

The set of standard deviations achieved as seen in Eq. 3 can also be seen as a matrix:

$$\Sigma = \begin{pmatrix} \sigma_{11} & \cdots & \sigma_{1t} \\ \cdots & & \cdots \\ \sigma_{m1} & \cdots & \sigma_{mt} \end{pmatrix}$$

(4)

The standard deviation is a measure of dispersion, then, the smaller σ_{jl} is, the more stable the feature jl will be. Using a threshold, $\sigma_{threshold}$, only features with standard deviation less than this threshold will be used for distance calculation.

This threshold is fixed by column, such that the same number, N_f, of features per column will be selected, not necessarily the same for each vector (row/point). That is, N_f is fixed a priori and, then, the column l threshold, $\sigma_{threshold}^l$, is calculated such that the number of σ_{jl} that comply with Eq. 5 will be N_f.

$$\sigma_{jl} \leq \sigma_{threshold}^l \qquad \forall\, 1 \leq j \leq m$$

(5)

Once each $\sigma_{threshold}^l$ is calculated, the distance (using p-norm) between two normalized signatures X^a and X^b is calculated as shown in Eq. 6.

$$d(X^a, X^b) = (\sum_{j=1}^m \sum_{l=1}^t C_{jl} * |x_{jl}^a - x_{jl}^b|^p)^{\frac{1}{p}} \ \ with\ C_{jl} = 1\ if\ \sigma_{jl} \leq \sigma_{threshold}^l$$
$$C_{jl} = 0\ if\ \sigma_{jl} > \sigma_{threshold}^l$$

(6)

A feature selection per matrix was also tested, that is, using the N_f features with the lesser σ. The threshold, $\sigma_{threshold}$, is calculated such that the number of σ_{jl} that comply with Eq. 7 will be N_f.

$$\sigma_{jl} \leq \sigma_{threshold} \qquad \forall\, 1 \leq j \leq m \ \ and\ \ 1 \leq l \leq t$$

(7)

The distance was calculated in a similar way to that shown in Eq. 6, but replacing $\sigma_{threshold}^l$ with $\sigma_{threshold}$.

This selection technique showed a worse performance, so the results have not been included in this work. Therefore, in the following, "feature selection" will only refer to the first technique described, that is, feature selection per column.

4 Experimental Environment

In [3] the performance of the main classifiers used in signature recognition were compared. For this reason, this work was used as the reference in [15]. Then, the same experimental conditions as those in both works will be followed here.

4.1   Experiments

The features selection proposal has been tested for the system configurations shown in *Baseline_Ver_Sytem* and *Baseline_Ver_Sytem_Soft* (see Sec. 2).

Several values of N_f have been tested for each system, being calculated as a percentage of the normalized signature size. The following percentages have been tested: 20%, 40%, 60% and 80%. For these percentages, the corresponding N_f values for each system are:

- **Baseline_Ver_Sytem:** 10, 20, 30 and 40 features per column. For this system the normalized signature has 50 vectors ($m = 50$).
- **Baseline_Ver_Sytem_Soft:** 5, 10, 15 and 20 features per column. For this system the normalized signature has 25 vectors ($m = 50$).

In the following, to identify the systems where the feature selection has been applied, these will be called:

- **FS_Ver_System_N_f:** is the *Baseline_Ver_Sytem* with N_f features selected per column.
- **FS_Ver_System_Soft_N_f:** idem to the previous but for the *Baseline_Ver_Sytem_Soft*.

4.2   Database and Trials

We have used MCYT database [13]. The information acquired for each sampling instant (X, Y, pressure, azimuth and altitude) is used directly as features ($t = 5$).

Samples of 330 different people were acquired. Each target user produced 25 genuine signatures, and 25 skilled forgeries were also captured for each user. These skilled forgeries were produced by the 5 subsequent target users by observing the static images of the signature to imitate, trying to copy them (at least 10 times). Shape-based natural dynamics of highly skilled forgeries are obtained.

The center of mass of each signature was calculated and this point was displaced to the origin of coordinates.

We carried out verification experiments, using 5 signatures per person for training and 5 different genuine signatures per person for testing. In addition, we used the 25 forgeries. This implies a total number of 330×5 genuine tests plus 330×25 skilled impostor tests and $330 \times 329 \times 5$ random impostor tests. This procedure is similar to SVC 2004 [17], although our database is larger.

4.3   Performance Measure

To compare the performance with the reference work [3], the same measure has been used, the minimum value of the Detection Cost Function (DCF): $DCF = C_{miss} * P_{miss} * P_{true} + C_{fa} * P_{fa} * P_{false}$, where C_{miss} is the cost of a miss (rejection), C_{fa} is the cost of a false alarm (acceptance), P_{true} is the a priori probability of the target, and $P_{false} = 1 - P_{true}$. $C_{miss} = C_{fa} = 1$.

Due to the values of C_{miss} and C_{fa}, the minimum value of the DCF is similar, for a great number of tests (which is our case), to the Equal Error Rate (EER).

The minimum DCF or EER can be evaluated with a different threshold for each user or with the same for all, which is the approach used in this work. The latter is the less favorable case, due to the variability of the users scores.

4.4   Score Normalization in Verification Task

To improve the performance of the systems tested with common threshold, score normalization techniques can be applied. A detailed study of this problem is far beyond the scope of this work, so only the technique with the best performance in our baseline work [15] will be used:

– **Target-Centric (TC) Method.** Only target training signatures are used. Given a system score S, the normalized one is calculated as: $S_Z = S - \mu_T$, where μ_T is the mean of the target scores. To generate these scores with the training samples, we have used the rotation sampling method of error estimation [5], implemented by the leave-one-out technique (5 scores).

5   Results: Performance Comparison

Table 1 shows the results achieved with our feature selection proposal, compared with the baseline and reference systems. With regard to the last, only the best results achieved with each classifier tested are shown. In the column "Parameters" of the table, the main characteristics of each system are shown:

– For the VQ system, the size of the model (*cluster · vector dimension*).
– For the HMM-based system the number of states (Q) and mixtures (M).
– For the DTW-based system, the function used to integrate the n distances to the n training templates.
– For the baseline systems and with feature selection ones if the score normalization has been applied (*TC2 Rotat*) or not (*No ScorNor*).

From the results shown in Table 1, it can be seen that the best results are achieved with the score normalization application. However, it is interesting to notice that when no score normalization is used, the feature selection improves the system performance, achieving better results than the DTW-based system.

Focusing on the results achieved with score normalization, the use of feature selection allows us to achieve results close to, and even better than, those achieved with the *baseline system*, but with fewer requirements. For example, the **FS_Ver_Sys_30** system (which has the best performance) improves the skilled forgeries result, this being similar to random ones, while the number of vectors is 40% smaller. The "soft" approach is very interesting, since the system has a fairly good performance, but with very low requirements. The feature selection in this case shows that, even here, it is possible to decrease the number of vectors, without decreasing the performance. For example, the **FS_Ver_Sys_Soft_15** system outperforms the random forgeries recognition rates, decreasing by only 3% those with the skilled ones, while the number of vectors is also 40% smaller.

Table 1. Comparison of our feature selection proposal performance and that of the baseline and reference systems in verification. The best performance (best rate random/skilled)is bold face emphasized.

Method	Parameters	Minimum DCF (%)	
		Random	Skilled
VQ	128x5	4.3	11.9
HMM	Q=12,M=1	5.4	16.3
DTW	min($\cdot$)	2.4	8.9
Baseline_Ver_Sys	No ScorNor	2.4	8.6
FS_Ver_Sys_10	No ScorNor	2.8	9.4
FS_Ver_Sys_20	No ScorNor	2.1	7.3
FS_Ver_Sys_30	No ScorNor	2.1	7.2
FS_Ver_Sys_40	No ScorNor	2.2	7.9
Baseline_Ver_Sys	TC2 Rotat	1.8	6.6
FS_Ver_Sys_10	TC2 Rotat	2.5	7.8
FS_Ver_Sys_20	TC2 Rotat	2.0	6.2
FS_Ver_Sys_30	TC2 Rotat	**1.9**	**6.2**
FS_Ver_Sys_40	TC2 Rotat	1.8	6.4
Baseline_Ver_Sys_Soft	No ScorNor	2.9	8.4
FS_Ver_Sys_Soft_5	No ScorNor	3.5	9.4
FS_Ver_Sys_Soft_10	No ScorNor	2.9	7.9
FS_Ver_Sys_Soft_15	No ScorNor	2.6	8.1
FS_Ver_Sys_Soft_20	No ScorNor	2.5	8.3
Baseline_Ver_Sys_Soft	TC2 Rotat	2.3	6.4
FS_Ver_Sys_Soft_5	TC2 Rotat	3.2	8.1
FS_Ver_Sys_Soft_10	TC2 Rotat	2.6	6.8
FS_Ver_Sys_Soft_15	TC2 Rotat	2.1	6.6
FS_Ver_Sys_Soft_20	TC2 Rotat	2.3	6.5

In order to get a more complete comparison, it is interesting to compare computational load and storage requirements. With regard to the algorithms tested in the reference work [3], we shall focus this study on the best: DTW.

We will use the following nomenclature:

- J_i is the i signature reference template length.
- I is the candidate's signature length.
- K is the number of reference templates per user.
- m is the normalized signature length.

In our experiments, we have set $K = 5$, and in MCYT database the average length per signature is 454 vectors, so for the calculations we can fix the following $J_i = \overline{J} = 454$ and $I = \overline{I} = 454$.

The DTW algorithm applied in [3] requires the computation of about $\frac{KI\sum_{i=1}^{K} J_i}{3}$ distance measurements to be computed.

In the baseline systems, the number of distances computed is Km per candidate signature, with m the normalized signature length. The time necessary to normalize this signature must be added (an operation that must not be performed with DTW).

When feature selection is applied, we must calculate the standard deviations matrix. This must be performed only once, and can be done off-line, then this time will not be taken into account for the computational load estimation. The number of distances computed is KN_f per candidate signature.

As for the storage requirements, with DTW, it is necessary to save the original signature, that is, $K \sum_{i=1}^{K} J_i$ vectors. In the baseline proposal, the number of vectors to be stored is Km in *Baseline_Ver_Sys* and m in *Baseline_Ver_Sys_Soft*. When feature selection is applied, the number of "vectors" (the same number is selected for each feature, then, we can also use the "vector" unit to compute storage requirements) to be stored is kN_f in *FS_Ver_Sys_N_f* and N_f in *FS_Ver_Sys_Soft_N_f*. In this case we must also store the $m * t$ coefficients C_{jl} matrix. These coefficients have a value 0 or 1, then we need $m * t$ bits. Let us suppose that the data is codified in floating point IEEE 754, single-precision, that is, 32 bits per number (feature). Under this assumption, we need $m * t/32/t = m/32$ "vectors" to store the coefficients.

Table 2 shows a summary of the storage requirements (row *StReq*) and computational load comparison. For the first, the number of vectors needed to store the template/s for each system is shown. As for the computational load, a theoretical and practical estimation is shown. For the theoretical estimation (row *DisNr*) the previous calculations of computed distances have been used replacing I and J_i for their averaged values. The practical comparison has been performed with real processing times, using the same computer and experimental environment. Matlab has been used, performing 1000 verification operations. The following measurements were carried out:

1. Processing time comparison of the algorithms, that is, the input/output operations were excluded. The results of each comparison can be seen in the last four rows of Table 2. The comparison between the systems of the corresponding row and column appears in each cell. The value to the left of the ":" symbol is the processing time of the row system, and the value to the right is the processing time of the column system. Since these processing times are computer and software dependent, the relative values are shown, fixing as 1 the value of the quickest in each case.

2. Productiveness, i.e, the number of operations performed per second, including all the necessary operations: to open and read the test and template signature files, to normalize the test signature in our proposal and get the final score. The results can be seen in row *Op/s* of Table 2.

From the results shown in table 2, and focusing on the comparison with the DTW-based system, it can be seen that the storage requirements, that were 9 times lesser with the *Baseline_Ver_Sys* and 91 with *Baseline_Ver_Sys_Soft*, are 15 and 142, respectively, times lesser now, with the use of feature selection. With regard to the computer load, the processing speed (including I/O operations) that was 181 and 713 times greater with the *baseline systems*, it is 274 and 926, respectively, times greater now. The computer load ratios without I/O are clearly shown in the table.

A comparison with other approaches is far from the scope of this work. In [15] a comparison can be seen between the *baseline systems* and some of the most recent published work with the MCYT database, showing that our proposal is, in general, in the state-of-the-art scores.

Table 2. Computer load and storage requirements comparison between DTW and our systems proposals. In row **StReq (Storage Requirements)**, the number of vectors (5 features each) necessary to store the template/s is shown (the mean in the DTW column). The number of distances calculated per test sample is shown in row **DisNr (Distance Numbers)**; in the DTW column the mean also appears. The number of verification operations per second with each system can be seen in row **Op/s**, and the processing time comparison of the algorithms in the rest of the rows.

	Ba_Ver_Sys	Ba_Ver_Sys_Soft	FS_Ver_Sys_30	FS_Ver_Sys_Soft_15	DTW
StReq	250	25	152	16	2270
DisNr	250	25	150	15	343527
Op/s	98	385	148	500	0.54
Ba_Ver_Sys		5:1	1.6:1	8:1	1:328
Ba_Ver_Sys_Soft			1:3	1.6:1	1:1670
FS_Ver_Sys_30				4.8:1	1:557
FS_Ver_Sys_Soft_15					1:2662

6   Conclusions

In this work we have shown a new proposal of representative feature selection. This proposal has been successfully applied to an on-line signature recognition system based on size normalized signatures and fractional distances.

The proposed goal has been achieved, since it has been shown that the system storage requirements and computer load can be reduced with the use of representative points/feature selection, without a noticeable decrease in system performance, and the results are even better with skilled forgeries.

This reinforces the workline shown in our *baseline work* and other related ones, that shows that it is possible to reduce the signature points (undersampling, representative/critical points selection, etc.) without performance loss.

We have achieved a final system with a cost even lower than that of the *baseline system*, which is very interesting for large-scale systems and in resource-limited systems like smart-cards, but with a state-of-the-art performance of systems based on a single classifier, and even in some of those based on classifiers or features fusion. Besides, the security of the system is improved as only the representative features have to be stored, it being impossible to recover the original signature from these.

Although the score normalization is not a goal in this work, we think that it is interesting to remark the good performance of the Target-Centric Methods.

Finally, we should comment that what is shown in this work is a first approximation to the feature selection problem with normalized signatures. The good results achieved shows that it is an interesting workline, with interesting future work, as for example, testing with other normalized sizes and with other proposals of features/points selection.

References

1. Connell, S.D., Jain, A.K.: Template-based online character recognition. Pattern Recognition 34(1), 1–14 (2001)
2. Fabregas, J., Faundez-Zanuy, M.: Biometric dispersion matcher. Pattern Recognition (2009)

3. Faundez-Zanuy, M.: On-line signature recognition based on vq-dtw. Pattern Recognition 40(3), 981–992 (2007)
4. Fiérrez-Aguilar, J., Nanni, L., Lopez-Peñalba, J., Ortega-Garcia, J., Maltoni, D.: An on-line signature verification system based on fusion of local and global information. In: Kanade, T., Jain, A., Ratha, N.K. (eds.) AVBPA 2005. LNCS, vol. 3546, pp. 523–532. Springer, Heidelberg (2005)
5. Fierrez-Aguilar, J., Ortega-Garcia, J., Gonzalez-Rodriguez, J.: Target dependent score normalization techniques and their application to signature verification. IEEE Transactions on Systems, Man and Cybernetics, Part C 35(3), 418–425 (2005)
6. François, D., Wert, V.: The concentration of fractional distances. IEEE Transactions on Knowledge and Data Engineering 19(7), 873–886 (2007)
7. Jain, A.K., Griess, F.D., Connell, S.D.: On-line signature verification. Pattern Recognition 35(12), 2963–2972 (2002)
8. Kholmatov, A., Yanikoglu, B.: Identity authentication using improved online signature verification method. Pattern Recognition Letters 26(15), 2400–2408 (2005)
9. Li, B., Wang, K., Zhang, D.: Minimizing spatial deformation method for online signature matching. In: Lee, S.-W., Li, S.Z. (eds.) ICB 2007. LNCS, vol. 4642, pp. 646–652. Springer, Heidelberg (2007)
10. Li, B., Zhang, D., Wang, K.: Improved critical point correspondence for on-line signature verification. International Journal of Information Technology 12(7), 45–56 (2006)
11. Martinez-Diaz, M., Fierrez, J., Freire, M.R., Ortega-Garcia, J.: On the effects of sampling rate and interpolation in hmm-based dynamic signature verification. In: Proc. Intl. Conf. on Document Analysis and Recognition, ICDAR, vol. 2, pp. 1113–1117 (September 2007)
12. Ohishi, T., Komiya, Y., Matsumoto, T.: Online signature verification using pen-position, pen-pressure and pen-inclination trajectories. In: Proceedings of 15th International Conference on Pattern Recognition, 2000, September 3-7, vol. 4, pp. 547–550 (2000)
13. Ortega-Garcia, J., Fierrez, J., Simon, D., Gonzalez, J., Faundez-Zanuy, M., Hernaez, I., Espinosa, V., Satue, A., Igarza, J.J., Vivaracho, C., Escudero, D., Moro, Q.I.: Mcyt baseline corpus: a bimodal biometric database. IEE Proc. Visual Image Signal Processing 150(6), 395–401 (2003)
14. Plamondon, R., Lorette, G.: Automatic signature verification and writer identification: The state of the art. Pattern Recognition 22(2), 107–131 (1989)
15. Vivaracho-Pascual, C., Faundez-Zanuy, M., Pascual, J.M.: An efficient low cost approach for on-line signature recognition based on length normalization and fractional distances. Pattern Recognition 42(1), 183–193 (2009)
16. Wirotius, M., Ramel, J.-Y., Vincent, N.: Selection of points for on-line signature comparison. In: Ninth International Workshop on Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004, October 26-29, pp. 503–508 (2004)
17. Yeung, D.Y., Chang, H., Xiong, Y., George, S., Kashi, R., Matsumoto, T., Rigoll, G.: Svc2004: First international signature verification competition. In: Zhang, D., Jain, A.K. (eds.) ICBA 2004. LNCS, vol. 3072, pp. 16–22. Springer, Heidelberg (2004)