PERGAMON

Pattern Recognition 35 (2002) 2963–2972

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

On-line signature verification

Anil K. Jain *, Friederike D. Griess, Scott D. Connell

Department of Computer Science and Engineering, Michigan State University, 3115 Engineering Building, East Lansing, MI 48824, USA

Abstract

We describe a method for on-line handwritten signature verification. The signatures are acquired using a digitizing tablet which captures both dynamic and spatial information of the writing. After preprocessing the signature, several features are extracted. The authenticity of a writer is determined by comparing an input signature to a stored reference set (template) consisting of three signatures. The similarity between an input signature and the reference set is computed using string matching and the similarity value is compared to a threshold. Several approaches for obtaining the optimal threshold value from the reference set are investigated. The best result yields a false reject rate of 2.8% and a false accept rate of 1.6%. Experiments on a database containing a total of 1232 signatures of 102 individuals show that writer-dependent thresholds yield better results than using a common threshold.

Keywords: On-line signatures; Biometric authentication; Verification; Template; String matching; Feature detection; Writer-dependent threshold

1. Introduction

Handwritten signatures are commonly used to approbate the contents of a document or to authenticate a financial transaction. Signature verification is usually done by visual inspection. A person compares the appearance of two signatures and accepts the given signature if it is sufficiently similar to the stored signature, for example, on a credit card. In the majority of situations where a signature is required, no verification takes place at all due to the amount of time and effort that would be required to manually verify signatures. Automating the signature verification process will improve the current situation and eliminate fraud.

The handwritten signature is a biometric attribute. Biometric identification and verification systems are being increasingly adopted in our environment. Well-known biometric methods include iris-, retina-, face- and fingerprint-based identification and verification [1]. While attributes like iris, retina and fingerprints do not change over time and thus have low intra-class variation, they require special and relatively expensive hardware to capture the image. An important advantage of the signature over other biometric attributes is its long standing tradition in many commonly encountered verification tasks. It has been used for decades in civilian applications while other methods (e.g., fingerprints) still have the stigma of being associated with criminal investigation. In other words, signature verification is already accepted by the general public. While we are unaware of any studies that show that an individual's signature is unique, it is generally accepted that this is the case. Nevertheless, signature verification is a difficult pattern recognition problem because the intra-class variations (i.e., the signature of one individual) can be large (see Fig. 1); even forensic experts cannot always tell whether a signature is authentic or not. In addition, signatures are easier to forge than other biometric attributes.

Automatic signature verification can be divided into two main areas depending on the data acquisition method: off-line and on-line signature verification. In off-line signature verification, the signature is available on a document which is scanned to obtain its digital image representation. On-line signature verification uses special hardware, such as a digitizing tablet or a pressure sensitive pen, to record the pen movements during writing. In addition to shape,

* Corresponding author. Tel.: +1-517-353-6484; fax: +1-517-432-1061.

E-mail address: jain@cse.msu.edu (A.K. Jain).

0031-3203/02/$22.00
PII: S0031-3203(01)00240-0

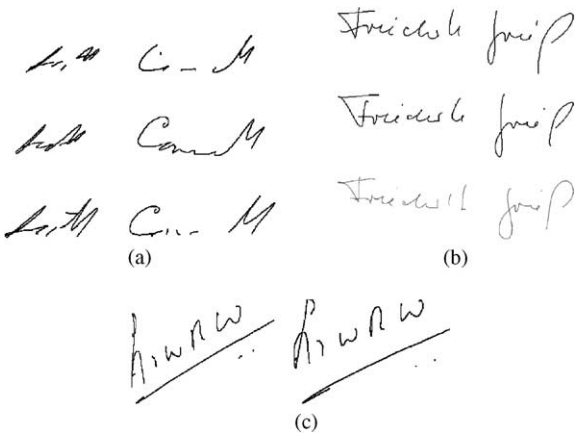

Fig. 1. Sample signatures of three writers.

the dynamics of writing are also captured in on-line signatures, which is not present in the 2-D representation of the signature and hence it is difficult to forge. Fig. 2 shows the same signature twice, as it would appear scanned (off-line) and with the time represented along the z-axis (on-line).

Automatic signature verification can be used in all applications where handwritten signatures are currently collected such as cashing a check, signing a credit card transaction or authenticating a legal document. The ability to capture the signature and have it immediately available in a digital form for verification also opens up a range of new application areas. Basically, any system that uses a password or PIN can instead use an on-line signature for access. This includes file and device access or secure physical entry systems. The advantages are evident; a signature is more difficult to steal or guess than a password and is also easier to remember for the user.

Fig. 3 shows the diagram of a typical signature verification system. To enroll into the system, the user has to provide a set of training signatures. Typically, a feature vector is extracted from the data which describes certain characteristics of the signature and stored as a template. For verification, the same features are extracted from the test signature and compared to the template.

The system implemented here uses a digitizing tablet (IBM CrossPad) from the A.T. Cross company [2] as the data capturing device. The IBM CrossPad has a sampling rate of 100–150 samples per second and records the x- and y-coordinates of the points in the signature. The pen has a touch sensitive switch in its tip such that only pen-down samples (i.e., when the pen touches the paper) are recorded. Evaluating a verification system requires the analysis of two types of errors. The percentage of genuine signatures that are incorrectly rejected by the system is called the false reject rate or type I error. The percentage of incorrectly accepted forgeries is called the false accept rate or type II error. The two types of errors usually have different costs associated with them depending on the security requirements of the application. The performance of a system is often measured by its equal error rate, which is the point where the false accept rate and the false reject rate are the same. A more meaningful performance measure is the error tradeoff curve (receiver operating characteristic curve), which shows how one error changes with respect to the other.

2. Related work

A wide range of methods for on-line handwritten signature verification have been reported, but more work has been done on off-line verification. Depending on the signature capture device used, features such as velocity, pen pressure and pen tilt are used in on-line verification in addition to spatial (derived from (x, y) coordinates) features. Different approaches can be categorized based on the model used for verification. Most of the approaches do a fair amount of preprocessing before extracting features from the signature. Features can be separated into two categories: global and local features. Global features describe properties of the whole signature. Examples of global features include total writing time, bounding box or the number of strokes. (A stroke is the sequence of points through which the pen moves while touching the paper. A signature is usually made up of several strokes.) Local features are properties that refer to a position within the signature, whose examples include local curvature and speed. The use of global features [3] alone

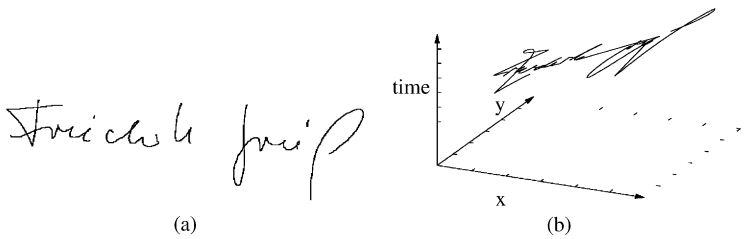

Fig. 2. Off-line versus on-line signature: (a) signature that is captured off-line. Only spatial information is available; (b) shows the same signature with the temporal information displayed along the z-axis.

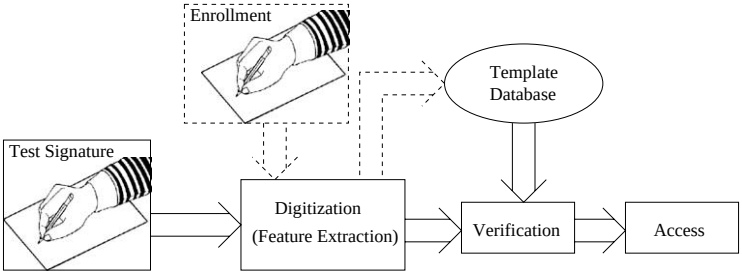

Fig. 3. A typical signature verification system.

has the advantage that the verification time is very short, but the error rates of algorithms that also incorporate local features are generally lower. The most common method to find the similarity between the input feature vector and the stored template is to use some variant of the Euclidean distance. Since the number of points differs between any two signatures, some form of string matching [4,5] is used. Hidden Markov models, well known for their success in speech recognition, have also been successfully applied to handwriting recognition. For signature verification, a variety of models [6,7] and features [8,9] have been evaluated. The number of signatures captured for a user during the enrollment phase varies between 6 and 20. The equal error rate generally lies between 1% and 6%. Since there does not exist a signature database in the public domain, every research group has collected its own data set, having between 9 and 105 individuals enrolled. This makes a comparison of different signature verification systems a difficult task.

3. System design

Fig. 4 shows the modules of our signature verification system. During enrollment of a new user, input to the system is a set of training signatures produced by that user. The training data is preprocessed and the features are extracted. This data is then saved in a database together with a unique identifier (ID) that is used to retrieve the signatures during matching. In addition, a threshold on the matching score is derived from the training data. For verification, a test signature along with the claimed writer identity is input to the system. The same preprocessing and feature extraction methods are applied. The signature is then compared to each of the reference (training) signatures which are retrieved from the database based on the writer identifier. The resulting difference values are combined and, based on the individual threshold for the writer, the signature is accepted as genuine or rejected as a forgery.

3.1. Preprocessing

The input signal from a digitizing tablet or digitizing pen can be very jagged. The physical space provided for writing

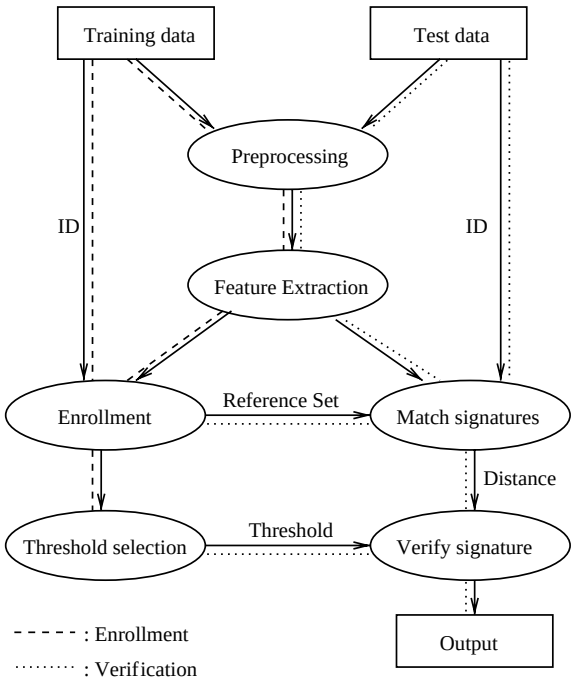

Fig. 4. Modules of a signature verification system.

the signature may vary between different applications and the pen used can affect the smoothness and the size of the signature. A commonly used method to smooth the signature is based on a Gaussian filter. In order to compare the spatial features of the signature, time dependencies have to be eliminated from the representation. This is achieved by resampling the signature uniformly with equidistant spacing. Certain points in the signature, such as start and endpoints of a stroke and points of trajectory change, carry important information. These points, referred to as critical points, are extracted before preprocessing and their positions are retained throughout the resampling and smoothing process. Temporal features must be extracted before resampling, and then propagated to the resampled points by interpolation. Fig. 5 shows one original signature and the output after all the preprocessing steps.

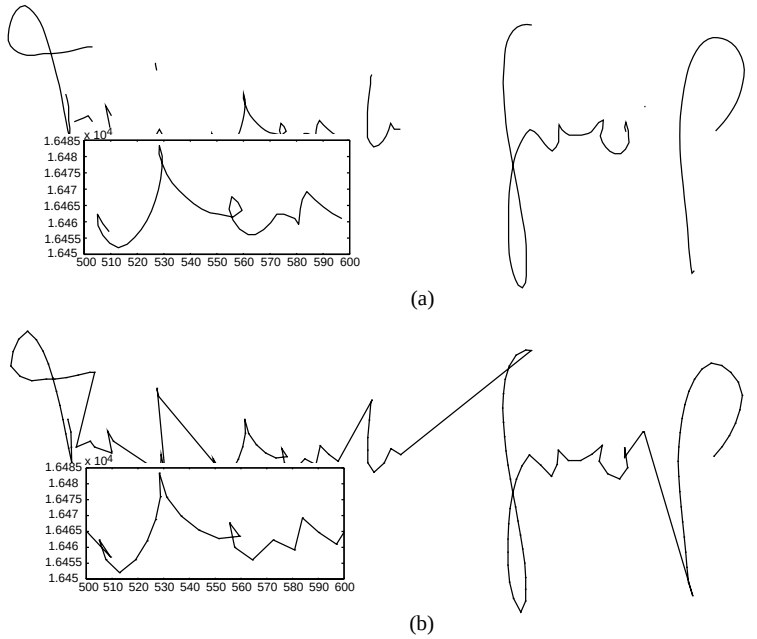

Fig. 5. Preprocessing of on-line signatures: (a) shows a signature before preprocessing. The sampling points are equally spaced in time; (b) shows the same signature after preprocessing; it has been smoothed and resampled. The individual strokes are concatenated for the subsequent matching process.

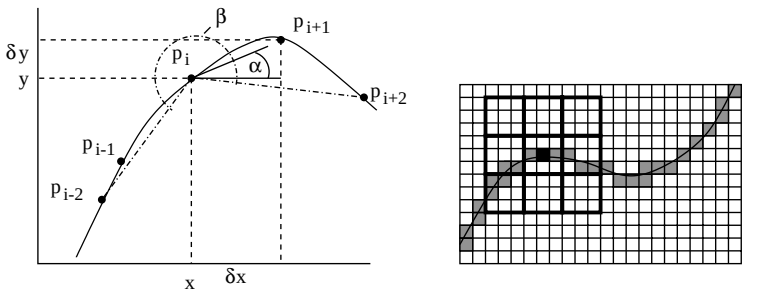

Fig. 6. Feature computation. The features are computed at point p_i; the two preceding points are p_{i-1} and p_{i-2} and the two succeeding points are p_{i+1} and p_{i+2}, respectively. The changes in the x- and y-coordinates for point p_i are the changes with respect to the subsequent point p_{i+1}. The absolute y-coordinate is the y-coordinate of each resampled point after preprocessing. The angle α is the angle between the x-axis and the line through points p_i and p_{i+1}. The curvature feature is the angle between the lines $\overline{p_i p_{i-2}}$ and $\overline{p_i p_{i+2}}$. The image feature calculates nine grey values in the neighborhood of the sampling point. A 9×9 pixel neighborhood is divided into nine 3×3 squares and a grey value is computed as the sum of the pixel values falling in that window.

3.2. Feature extraction

All strokes are combined into one long stroke during preprocessing. The original number of strokes is recorded and used as a global feature. From the x- and y-coordinates of the preprocessed image, a number of local features are extracted which are divided into two categories, spatial and temporal features. Spatial features are static features that are extracted from the shape of the signature.

Local spatial features that are extracted and studied for their saliency for signature verification are: (i) the x and y coordinate differences between two consecutive points, δx and δy, (ii) the absolute y-coordinate with reference to the center of the signature, y, (iii) the sine and cosine of the angle with the x-axis, $\sin\alpha$ and $\cos\alpha$, (iv) the curvature, β and (v) the grey values in a 9×9 pixel neighborhood. Fig. 6 shows all the spatial features.

In addition to the temporal order of the points, the speed of the writing at local points is a valuable feature. Two different variants to extract the speed from the signature are explored: (i) the absolute speed and the relative speed (absolute speed normalized by the average signing speed)

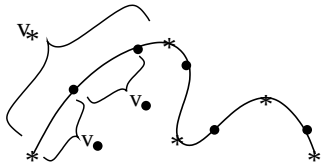

Fig. 7. Computation of speed features. The speed between two consecutive critical points, V_*, and the speed between any two points, $V_\blacklozenge$, is calculated as the distance between those points, since the points are equidistant in time.

at each resampled point and (ii) the absolute and relative average speeds between two critical points. In Fig. 7 the calculation of the speed before resampling is shown.

3.3. String matching

Once the local features are extracted from each point in the signature, a method must be chosen to compare two signatures. Each signature is described by a set of features extracted at each sampling position. Thus the signature can be represented as a string, i.e., a sequence of feature vectors whose size is the number of local features extracted. Global features are not included in this representation. String matching, also known as dynamic time warping [10,11], is a well-known method to compare strings of different lengths. It finds an alignment between the points in the two strings such that the sum of the differences between each pair of aligned points is minimal. To find the minimal difference, all possible alignments must be investigated. An efficient solution for this is based on dynamic programming. Our method of string matching extends the basic approach by adding a method of allowing strings with broken strokes to be reconnected, while including a penalty to discourage the matching of two strings with large differences in the number of strokes detected. After an alignment between the two signatures is found, the difference between the number of strokes in the two signatures is incorporated into the overall dissimilarity measure. The formula for the overall dissimilarity between an input signature string (I) and a template string (T) is given by

$$Dissimilarity(T,I) = \frac{Dist(T,I)^2}{Norm\_Factor(N_T,N_I)} + (SP)|S_T - S_I|,$$

where $Dist(T,I)$ is the distance measure obtained after aligning the two strings T and I, SP is the penalty for matching signatures with different stroke counts, $|S_T - S_I|$ is the difference between the number of strokes in the template and the input strings and $Norm\_Factor(N_T,N_I)$ is the maximum possible distance between two strings of lengths N_T and N_I scaled by a constant factor. This method was implemented by Connell and Jain for handwritten character recognition [12,13].

3.4. Verification

In the verification process a test signature must be compared to all the signatures in the reference set (template database). Three basic methods to combine the individual dissimilarity values (between the input and one of the templates) into one value are investigated: (i) the minimum of all the dissimilarity values, (ii) the average of all the dissimilarity values and (iii) the maximum of all the dissimilarity values.

After the dissimilarity value is computed, a decision regarding whether the signature is authentic or a forgery must be made. For this, the result of the matching will be compared to a threshold. If the dissimilarity value is above that threshold, the signature is rejected, otherwise it is accepted. The threshold can be chosen to be identical for all the writers or set individually for each writer.

3.4.1. Common threshold

A common threshold has the advantage that all the enrollment data from all the writers can be used to find an optimal threshold. The dissimilarities between all the signatures of all the writers who are enrolled into the system are computed and a threshold value is selected based on the minimum error criterion.

3.4.2. Writer-dependent threshold

To adapt the verification process to the properties of a single writer, writer-dependent thresholds should be used. In principle, a writer-dependent threshold can be derived only from that writer's enrollment data. However, to reliably estimate the writer-dependent threshold, more enrollment data than usually available are necessary. To circumvent this, one starts with a common threshold and then modifies it for each writer by adding a writer specific component. Three choices to calculate the writer specific component from the reference set are investigated: (i) the minimum distance between all the references, (ii) the average distance between all the references and (iii) the maximum distance between all the references.

4. Experimental results

The proposed method has been implemented and evaluated with 1232 signatures from 102 different writers. Two datasets, called DB1 and DB2, are used for evaluation. The first dataset, DB1, contains 520 signatures from 52 writers, approximately one-fifth female. Each writer was asked to contribute ten signatures. These signatures were collected in one session. Additionally, for 20 writers three forgeries each were collected from individuals who were shown the

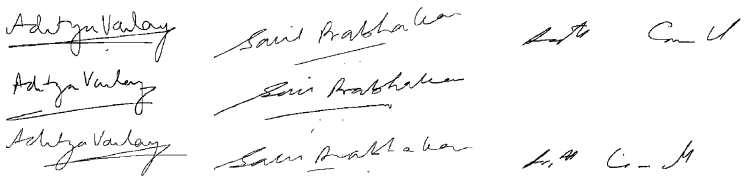

Fig. 8. Example signature from writers who contributed data over a period of 1 yr. The top signature in each column shows a sample from the first acquisition, the second row contains samples of signatures taken approximately eight months later and the last row contains signatures collected approximately 1 yr after the first acquisition. Writer C did not contribute any signature after eight months.

Table 1
Datasets for signature verification (DB1 ⊂ DB2)

Dataset	Number of writers	Number of signatures per writer	Total number of signatures	Total Number of forgeries
DB1	52	10	520	60
DB2	102	10–42	1232	60

original signature before being asked to produce the forgeries. No signatures from professional imitators are available, therefore these 60 forgeries will be called skilled forgeries. The second database, DB2, contains a total of 1232 signatures collected from 102 writers and is a superset of dataset DB1. Seventeen of these writers contributed more than ten signatures, which were collected in multiple sessions over a period of up to 1 yr. Fig. 8 shows some examples from these writers. Table 1 summarizes the data used. Fig. 9 shows sample signatures of four writers.

Forgeries are classified into random or zero-effort forgeries and skilled forgeries. For a random forgery, the forger has either no knowledge about the original signature or does not try to imitate the shape of the signature. Since only a limited number of forgeries exist in our database, authentic signatures from other writers serve as random or zero-effort forgeries.

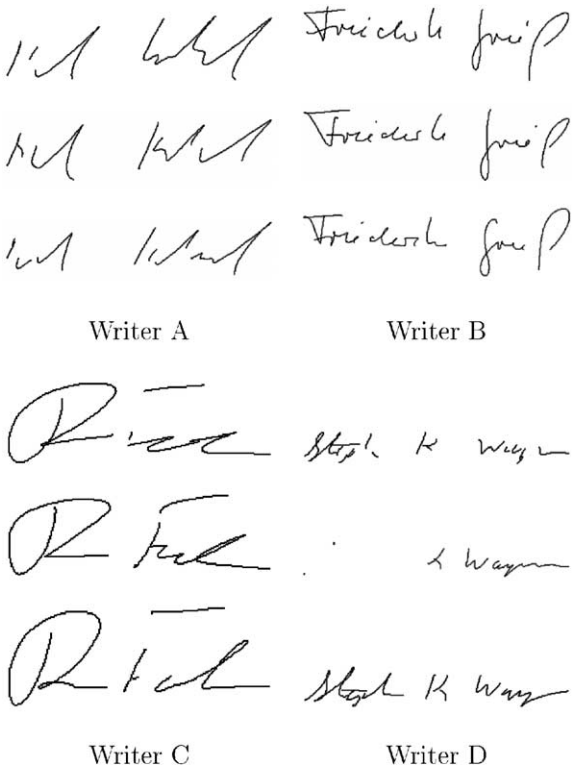

Fig. 9. Sample signatures of four writers. Writer D's signature has large intra-class variability. Note that in the second signature of writer D, the first name is missing almost completely because the writing pressure was not sufficient to record the signal.

4.1. Feature selection

For on-line handwriting recognition, several different feature sets have been evaluated in Ref. [14]. To see if these features are also applicable to on-line signature verification, each feature combination is first evaluated on our smaller database DB1. To evaluate the discriminative potential of the feature sets, every signature of every writer is compared to all the other signatures. The resulting dissimilarity value should be low for two signatures from the same writer and high for two signatures of different writers. Table 2 shows the results for the best feature subset consisting of the features δx, δy, $\sin\alpha$ and $\cos\alpha$. The fourth column of this table gives the percentage of time the signatures from the same writer differ by more than the threshold shown in column three. Column five reports the percentage of time the difference between signatures of two different writers is below the threshold. The threshold is selected such that these two percentages are approximately equal. The lower half of Table 2 shows the results when the thresholds are individually selected for each writer. The results for the forgeries are not considered to be of much statistical relevance, since only a total of 60 forgeries are available.

The best feature subset consisting of the spatial features, δx, δy, $\sin\alpha$ and $\cos\alpha$, is combined with different tempo-

Table 2
Performance of two different feature subsets on DB1

Feature subset	Type of forgery	Threshold (TH)	Percentage of genuine values above TH (%)	Percentage of forgery values below TH (%)
Common threshold				
δx, δy, $\sin\alpha$, $\cos\alpha$	Genuine	3.9	9.4	9.5
	Skilled	3.3	11.4	11.6
Writer-dependent thresholds				
δx, δy, $\sin\alpha$, $\cos\alpha$	Genuine	0.5–45.3	3.6	3.5
	Skilled	0.5–8.5	4	4.2

Table 3
Performance improvement using temporal features with writer-dependent thresholds on DB1

Speed feature	Type of forgery	Threshold (TH)	Percentage of genuine values above TH (%)	Percentage of forgery values below TH (%)
Absolute speed	Random	2–20	3.4	3.5
	Skilled	2–19	3.7	3.8
Normalized speed	Random	1.5–20	2.6	2.7
	Skilled	1.8–15	2.5	2.4
Absolute speed between critical points	Random	4–19	2.5	2.6
	Skilled	4–20	2.3	2.3
Normalized speed between critical points	Random	1.5–13	2.7	2.6
	Skilled	2.5–17.5	2.5	2.6

Table 4
Equal Error rates for different preprocessing methods and different number of reference signatures, and different methods for computing the dissimilarity value between the test signature and the reference set on DB1

Preprocessing	Features	No. of reference signatures	Type of forgeries	Equal error rate		
				Min. (%)	Avg. (%)	Max. (%)
Smoothing	δx, δy	3	Random	3	1.9	5.5
Resampling (8)	$\sin\alpha$, $\cos\alpha$		Skilled	11	11.2	16
Stroke concatenation		5	Random	2.7	1.1	5.8
			Skilled	12	6.6	16
Smoothing	δx, δy	3	Random	5.2	2.9	6.2
Resampling (8)	$\sin\alpha$, $\cos\alpha$		Skilled	11	12	17
Stroke concatenation		5	Random	6	3.9	9.5
Size normalization			Skilled	13	12	20

ral or speed features. Each of these speed features has been added to the "optimal" spatial feature set resulting in a feature vector of dimensionality five. The results of the various combinations using writer-dependent thresholds can be found in Table 3. While using the speed between all sampling points, the relative speed gives better results than the absolute speed. If we consider only the speed between critical points, the opposite is true; the absolute speed gives (slightly) better results (Table 4).

4.2. Threshold selection

In the previous section the discriminatory potential of different feature sets was investigated. To derive the error rates for the system, a set of sample signatures, called the reference set, must be chosen and compared to the database signatures. The reference signatures are drawn randomly from the available data. Reference sets of size three and five are evaluated. This process is

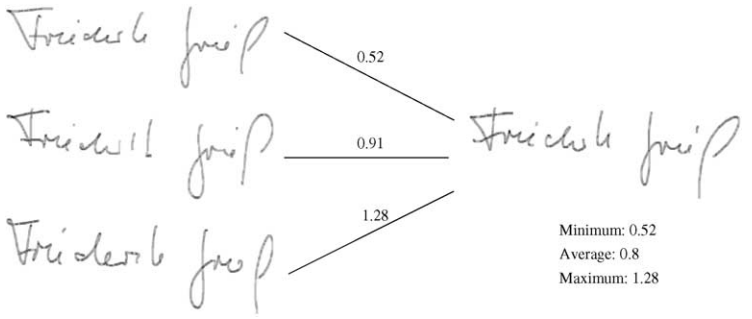

Fig. 10. The test signature on the right-hand side is compared to each of the three reference signatures shown on the left-hand side. Each comparison results in a dissimilarity value. The final dissimilarity value for each method using minimum, average and maximum operator, is shown.

Table 5
Error rates for a common threshold and the individual (writer-dependent) thresholds. The results are obtained with dataset DB2. Thresholds are chosen such that FAR and FRR are as close to each other as possible

Features	Type of forgeries	Common TH		Individual TH	
		FRR (%)	FAR (%)	FRR (%)	FAR (%)
δx, δy, $\sin\alpha$, $\cos\alpha$	Random	5.6	4.3	1.7	1.6
	Skilled	11.3	8.9	0.5	0.4
δx, δy, $\sin\alpha$, $\cos\alpha$, Absolute speed	Random	5.4	3.8	1.5	1.6
	Skilled	11	10.6	0.5	0.4
δx, δy, $\sin\alpha$, $\cos\alpha$, Normalized speed	Random	3.3	2.7	1.3	1.2
	Skilled	7.9	10.3	1.2	0.6
δx, δy, $\sin\alpha$, $\cos\alpha$, Absolute speed at critical points	Random	3.2	3.5	0.6	0.6
	Skilled	3.3	4.7	n/a	n/a
δx, δy, $\sin\alpha$, $\cos\alpha$, Normalized speed at critical points	Random	3.5	3.1	1.5	1.4
	Skilled	9.2	5.3	1.2	0

repeated 20 times on different randomly selected reference sets.

4.2.1. Common threshold

A signature is accepted or rejected based on its dissimilarities to the signatures in the reference set. Three different methods to combine the dissimilarity values from the comparison of the test signature with the reference set are investigated: the minimum, the average and the maximum dissimilarity value. Fig. 10 demonstrates how the combined dissimilarity value is obtained using three reference signatures. It can be seen that the minimum value yields the best error rates. Note that in using the minimum value, only the most similar reference signature is involved in the decision making process. For the evaluation of the various speed features, only the minimum distance is considered. Table 5 shows the results. The error tradeoff curves using database DB2 are shown in Fig. 11. For both types of forgeries, the false accept rate increases rapidly when the false reject error is reduced. Similarly, a slight increase in false rejects reduces the false accept rate by a significant amount.

4.2.2. Writer-dependent threshold selection

The writer-dependent threshold values are found empirically. The same range of potential thresholds is evaluated and the best (i.e., the one yielding (approximately) equal

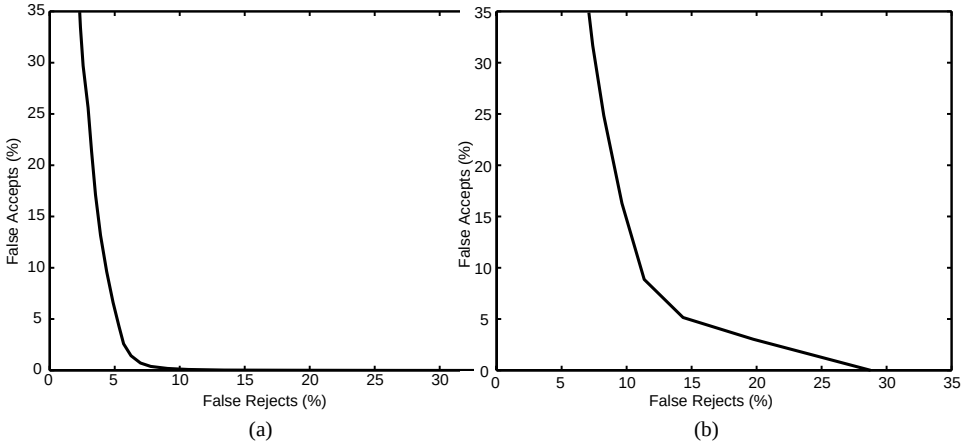

Fig. 11. Error tradeoff curves for a common threshold using only the spatial feature set on DB2: (a) Error tradeoff curve for random forgeries; (b) Error tradeoff curve for skilled forgeries.

Table 6
Error rates for automatic writer-dependent threshold selection incorporating speed features. The results are obtained with dataset DB2. Only the enrollment data was used to obtain the thresholds and the remaining data was used to estimate the error rate

Feature set: local spatial features and	FRR (%)	Random forgery FAR (%)	Skilled forgery FAR (%)
None	5.3	2.7	2.9
Absolute speed	4.0	1.3	10.1
Normalized speed	3.5	1.3	16.9
Absolute speed at critical points	2.8	1.6	n/a
Normalized speed at critical points	3.9	1.1	7.5

values for the false accept rate and false reject rate) threshold is chosen. This result is "ideal" in the sense that it reflects the best rates that can be achieved (see tables) by adjusting the threshold value using all the available signatures for a given user. The goal of every automatic threshold selection method must be to come as close as possible to these values. The three methods chosen to calculate the thresholds are the minimum, maximum and average dissimilarity values between the test signature and the reference set plus a user dependent offset. The offset depends on the feature set and must be determined empirically. From preliminary experiments reported in Table 5 it could be seen that the minimum value results in the lowest error rates. Table 6 shows the results for the feature sets incorporating various writing speed measures in addition to the local spatial features: δx, δy, $\sin\alpha$ and $\cos\alpha$. The results reported in Table 6 are obtained when only the enrollment data was used to obtain the thresholds and the rest of the data was used for testing. The threshold used is the combination of a global threshold and an offset calculated for each writer. This combined threshold can be altered to change the operating point of the system according to the needs of the application.

5. Conclusions and future work

A system for on-line signature verification has been implemented. The best results for a common threshold are obtained with the feature set consisting of the local features δx, δy, $\sin\alpha$, $\cos\alpha$, the relative speed between all sampling points and the number of strokes as a global feature. The best error rates for a common threshold are 3.3% false rejects and 2.7% false accepts. Writer-dependent thresholds are computed from the reference signatures. All the reference signatures are matched with each other. The best feature set for writer-dependent thresholds consists of the absolute speed between critical points as the speed feature. Using the minimum dissimilarity value plus a user dependent offset results in 2.8% false rejects and 1.6% false accepts.

It is still an open question as to how the reference set should be updated. The signature of an individual usually changes over time, so a deterioration of the verification rates can be expected if the reference set remains fixed. One possibility would be to ask the user to periodically provide new reference signatures. This would also ensure that no forged signatures are used for updates. An automatic update system would be most comfortable for the user. Here the choice must be made regarding which current reference signature should be updated with the new reference. A simple choice would be to always replace the oldest signature first. More sophisticated methods that incorporate the use of dissimilarity values to the other reference signatures for that signer should be taken into consideration.

Our signature database does not contain any data from skilled forgers. It is still unclear how such data should be col-

lected. The results would be more valuable if true forgeries that imitate the shape of the original signature were available. Evaluations of human performance on distinguishing such a set of forgeries from the true signatures could provide a baseline for system performance evaluation.

Finally, more signatures must be collected and over a longer period of time. The current test database, consisting of signatures from 102 writers, is at most representative of an application appropriate for a small organization. Larger organizations or applications that use signature verification for their clients will have a much larger signature database and the scalability of our system needs to be investigated. Studies on how signatures change over time, and how well our system will handle these changes need to be conducted.

References

[1] A.K. Jain, S. Pankanti, R. Bolle (Eds.), BIOMETRICS: Personal Identification in Networked Society, Kluwer, Dordrecht, 1999.
[2] A.T. Cross Company, http://www.cross.com.
[3] L.L. Lee, T. Berger, E. Aviczer, Reliable on-line human signature verification systems, IEEE Trans. Pattern Anal. Mach. Intelligence 18 (6) (1996) 643–647.
[4] V.S. Nalwa, Automatic on-line signature verification, Proc. IEEE 85 (2) (1997) 215–239.
[5] B. Wirtz, Stroke-based time warping for signature verification, in: Proceedings of the International Conference on Document Analysis and Recognition, Vol. 1, 1995, pp. 179–182.
[6] L. Yang, B.K. Widjaja, R. Prasad, Application of hidden Markov models for signature verification, Pattern Recognition 28 (2) (1995) 161–170.
[7] Q. Wu, I. Jou, S. Lee, On-line signature verification using LPC cepstrum and neural networks, IEEE Trans. Systems, Man Cybernet.—Part B: Cybernetics 27 (1) (1997) 148–153.
[8] J.G.A. Dolfing, E.H.L. Aarts, J.J.G.M. van Oosterhout, On-line signature verification with hidden Markov models, in: Proceedings of the International Conference on Pattern Recognition, Vol. 2, 1998, pp. 1309–1312.
[9] G. Rigoll, A. Kosmala, A systematic comparison between on-line and off-line methods for signature verification with hidden Markov models, in: Proceedings of the International Conference on Pattern Recognition, Vol. 2, 1998, pp. 1755–1757.
[10] R. Martens, L. Claesen, Dynamic programming optimization for on-line signature verification, in: Proceedings of the International Conference on Document Analysis and Recognition, 1997, pp. 653–656.
[11] Y.K.T. Ohishi, T. Matsumoto, On-line signature verification using pen-position, pen-pressure and pen-inclination trajectories, in: Proceedings of the International Conference on Pattern Recognition, 2000, pp. 547–550.
[12] S.D. Connell, A.K. Jain, Template-based online character recognition, Pattern Recognition 34 (1) (2001) 1–14.
[13] S.D. Connell, A.K. Jain, Learning prototypes for on-line handwritten digits, in: Proceedings of the International Conference on Pattern Recognition, 1998.
[14] S.D. Connell, Online handwriting recognition using multiple pattern class models, Ph.D. Thesis, MSU-CSE-00-27, Department of Computer Science, Michigan State University, May 2000.

About the Author—ANIL K. JAIN is a University Distinguished Professor in the Department of Computer Science and Engineering at Michigan State University. He served as the department Chair between 1995–1999. His research interests include statistical pattern recognition, computer vision, and biometric authentication. He received the best paper awards in 1987 and 1991 and certificates for outstanding contributions in 1976, 1979, 1992, and 1997 from the Pattern Recognition Society. He also received the 1996 IEEE Trans. Neural Networks Outstanding Paper Award. He was the Editor-in-Chief of the IEEE Trans. on Pattern Analysis and Machine Intelligence (1990–94). He is the co-author of Algorithms for Clustering Data, Prentice-Hall, 1988, has edited the book Real-Time Object Measurement and Classification, Springer-Verlag, 1988, and co-edited the books, Analysis and Interpretation of Range Images, Springer-Verlag, 1989, Markov Random Fields, Academic Press, 1992, Artificial Neural Networks and Pattern Recognition, Elsevier, 1993, 3D Object Recognition, Elsevier, 1993, and BIOMETRICS: Personal Identification in Networked Society, Kluwer in 1999. He is a Fellow of the IEEE and IAPR. He received a Fulbright research award in 1998 and was named a Fellow of the John Simon Guggenheim Memorial Foundation in 2001.

About the Author—FRIEDERIKE GRIESS received her university diploma from the Rheinisch-Westfaelische Technische Hochschule Aachen, Germany in 1998 and her MS degree from Michigan State University in 2000. She is currently working for AccuImage Diagnostics Corporation in San Francisco.

About the Author—SCOTT CONNELL received a BSE degree in Industrial and Operations Engineering from the University of Michigan in 1992, and MS and Ph.D. degrees in Computer Science from Michigan State University in 1996 and 2001, respectively. His research interests include pattern recognition, computer vision, hidden Markov models, offline and online handwriting recognition, and bioinformatics. He is currently a member of the technical staff at Agilent Technologies.