Combining Multiple Features for Text-Independent Writer Identification and Verification

Marius Bulacu, Lambert Schomaker

Combining Multiple Features for Text-Independent Writer Identification and Verification

Marius Bulacu *Lambert Schomaker*
Artificial Intelligence Institute, University of Groningen, The Netherlands
(bulacu, schomaker)@ai.rug.nl

Abstract

In recent years, we proposed a number of new and very effective features for automatic writer identification and verification. They are probability distribution functions (PDFs) extracted from the handwriting images and characterize writer individuality independently of the textual content of the written samples. In this paper, we perform an extensive analysis of feature combinations. In our fusion scheme, the final unique distance between two handwritten samples is computed as the average of the distances due to the individual features participating in the combination. Obtained on a large dataset containing 900 writers, our results show that fusing multiple features (directional, grapheme, run-length PDFs) yields increased writer identification and verification performance.

Keywords: writer identification / verification, directional / grapheme distributions, feature combination

1. Introduction

The identification of a person on the basis of scanned images of handwriting is a useful behavioral biometric modality with application in forensic and historic document analysis. A *writer identification* system retrieves, from a database containing handwritings of known authorship, those samples that are most similar to the query (see Fig. 1). The hit list is then analyzed in detail by a human expert. A *writer verification* system compares two handwriting samples and takes an automatic decision whether or not the input samples were written by the same person (see Fig. 2). Writer verification has potential applicability in a scenario in which a specific writer must be automatically detected in a stream of handwritten documents.

Stimulated also by the case of the anthrax letters, scientific research in this area has received renewed interest and many novel and effective approaches have been proposed recently. Writer identification and verification methods fall into two broad categories [8]: *text-dependent* vs *text-independent* methods. The *text-dependent* methods [14, 15, 16] are very similar to signature verification techniques and use the comparison between individual characters or words of known text (ASCII) content. These methods therefore require the prior segmentation by hand of the relevant information. The *text-independent* methods for writer identification [9, 2, 10, 11] use statistical features extracted from the entire image of a text block. A minimal amount of handwriting (e.g. a paragraph containing a few text lines) is needed in order to derive stable features insensitive to the text content of the samples. Our approach falls in this *text-independent* category. From the application point of view, two notable advantages are that human intervention is minimized and the compared samples are not required to have the same fixed textual content.

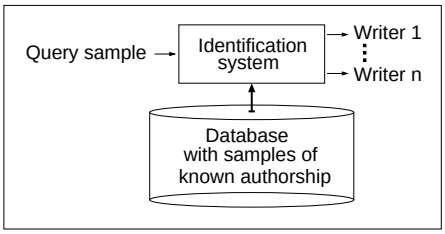

Figure 1. *Writer identifi cation* involves a 'one-to-many' search in a database with handwritings of known authorship and returns a likely list of candidates.

Proposed in the last several years, our writer identification and verification methods operate at two levels of analysis: the *texture level* and the *character-shape (allograph) level*. At the *texture level*, we use contour-based joint directional PDFs that encode orientation and curvature information to give an intimate characterization of individual handwriting style [4]. In our analysis at the *allograph level*, the writer is considered to be characterized by a stochastic pattern generator of ink-trace fragments, or graphemes [11, 12]. The PDF of these simple shapes in a given sample is characteristic for the writer and is computed using a common shape codebook obtained by grapheme clustering [3]. Two essential sources of behavioral information regarding handwriting individuality are thus exploited: the *texture-level features* are informative for the habitual pen-grip and preferred writing slant, while the *allograph-level features* reveal the character shapes engrained in the motor memory of the writer, as a result of educational, cultural and memetic factors [11].

In this paper, we specifically consider the problem of fusing these multiple features for improving performance on both tasks of identification and verification. This was not fully addressed in previous work. Here we provide an extensive analysis of feature combinations and show results obtained on a large dataset containing 900 writers.

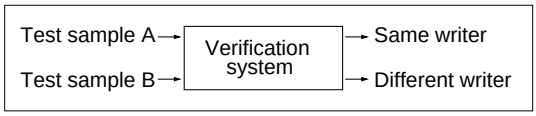

Figure 2. *Writer verifi cation* involves a 'one-to-one' comparison with a decision whether or not the two samples are written by the same person.

2. Experimental datasets

In our writer identification and verification experiments we used handwriting images originating from three datasets: Firemaker, IAM and ImUnipen.

The Firemaker set [13] contains handwriting collected from 250 Dutch subjects required to write 4 different pages. In the tests, we used pages 1 and 4 consisting of lowercase script: on page 1, the subjects copied a text of 5 paragraphs and, on page 4, they described the content of a given cartoon in their own words.

The IAM database [7] consists of forms with handwritten English text of varying content. This dataset includes a variable number of handwritten pages per writer, from 1 page (350 writers) to 59 pages (1 writer). In order to have comparable test conditions across all datasets, we modified the IAM set to contain always 2 samples per writer: we kept only the first 2 documents for those writers who contributed more than 2 pages to the original IAM dataset and we have split the document roughly in half for those writers with a unique page in the original set. Our modified IAM dataset therefore contains lowercase handwriting from 650 persons, 2 samples per writer.

We merged the Firemaker and IAM datasets to obtain a combined set which we named "Large". The Large dataset therefore contains 900 writers, 2 samples per writer, lowercase handwriting. All documents were originally scanned at 300 dpi, 8 bits / pixel, gray-scale. In this paper, we will report our experimental results obtained on the Large dataset. This merged set is comparable, in terms of number of writers, to the largest dataset used in writer identification and verification until the present [14]. It is significant to mention here that our approach to writer identification and verification is text-independent and does not require human effort for labeling. This gave us the advantage of being able to easily extend our methods to other datasets and to collect data from multiple sources and different languages in a common framework.

The ImUnipen set contains handwriting from 215 subjects, 2 samples per writer. The images were derived from Unipen on-line handwriting database. The time sequences of coordinates were transformed to simulated 300 dpi images using a Bresenham line generator and an appropriate thickening function. This set was not directly used in writer identification and verification tests due to the different origin of the images. A part of this set containing 65 writers (130 samples) was used in our allograph-level method for training the shape codebooks needed for computing the writer-specific grapheme emission probability. This also ensures a complete separation, at the level of writers, between training and testing data.

Table 1. Overview of the features used for writer identification and verification and their dimensionalities.

	Feature	Explanation	Dim
f1	$p(\phi)$	Contour-direction PDF	12
f2	$p(\phi_1, \phi_2)$	Contour-hinge PDF	300
		Direction co-occurrence PDFs	
f3h	$p(\phi_1, \phi_3)$ h	- horizontal run	144
f3v	$p(\phi_1, \phi_3)$ v	- vertical run	144
f4	$p(g)$	Grapheme emission PDF	400
		Run-length on white PDFs	
f5h	$p(rl)$ h	- horizontal run	60
f5v	$p(rl)$ v	- vertical run	60

3. Feature extraction methods

We use probability distribution functions (PDFs) extracted from the handwriting images to characterize writer individuality in a text-independent manner. The term "feature" will be used to denote such a complete PDF (an entire vector of probabilities). An overview of all features used here is given in table 1. We have designed features *f2*, *f3* and *f4*, while features *f1*, *f5* are classically known. The most discriminative features were selected here from a large number of features tested in [11].

The gray-scale images containing the scanned samples of handwriting are binarized using Otsu's method. Three primary representations of the document will then be used for feature computation: the binary image, the connected components and their contours (extracted using Moore's algorithm). Our methods work at two levels of analysis: the *texture level* and the *allograph level*.

3.1. Textural features

In these features, the handwriting is merely seen as a texture described by some probability distributions computed from the image and capturing the distinctive visual appearance of the written samples.

The most prominent visual attribute of handwriting that reveals individual writing style is slant. Further more, the whole distribution of directions in the script provides useful information for writer identification [6]. The directional PDF can be computed very fast using the contours by considering the orientation of local contour fragments determined by two contour pixels taken a certain distance apart (see Fig. 3). As the algorithm runs over the contours, the angle that the analyzing fragment makes with the horizontal is computed using equation 1 and an angle histogram is built thereby. This histogram is then normalized to a probability distribution $p(\phi)$ that constitutes the feature used in writer identification and verification.

$$\phi = arctan(\frac{y_{k+\epsilon} - y_k}{x_{k+\epsilon} - x_k}) \quad (1)$$

In our implementation $\epsilon = 5$ and this value was selected such that the length of the contour fragment is comparable to the thickness of the ink trace (6 pixels). The number of histogram bins spanning the interval 0° - 180° was set to $n = 12$ through experimentation. These settings will be used for all the directional features.

The directional PDF $p(\phi)$ was our starting point in designing more complex features that give a more intimate

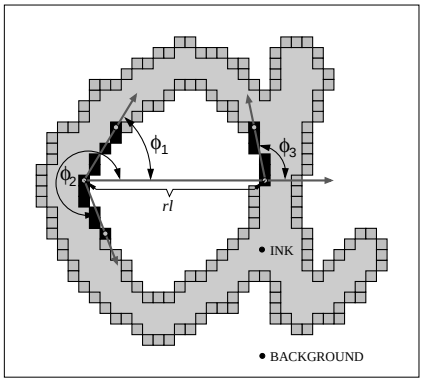

Figure 3. Schematic description for the feature extraction methods of directional and run-length PDFs.

description of handwriting individuality and ultimately yield significant improvements in writer identification and verification performance. In order to capture, besides orientation, also the curvature of the ink trace, which is very discriminatory between different writers, we designed the "hinge" feature. The central idea is to consider, not one, but two contour fragments attached at a common end pixel and, subsequently, compute the joint PDF of the orientations of the two legs of the "contour-hinge" (see Fig. 3). The feature $p(\phi_1, \phi_2)$ is therefore a bivariate PDF capturing both the orientation and the curvature of contours.

Building upon the same idea of combining oriented contour fragments, we designed another feature: the directional co-occurrence PDF. For this feature, we consider the combination of contour-angles occurring at the ends of run-lengths on the background (see Fig. 3). The joint PDF $p(\phi_1, \phi_3)$ of the two contour-angles occurring at the ends of a run-length on white captures longer range correlations between contour directions and gives a measure of the roundness of the written characters. Horizontal runs along the rows of the image generate *f3h* and vertical runs along the columns of the image generate *f3v*. Examples of $p(\phi_1, \phi_3)h$ for two writers are given in Fig. 4.

Run lengths were first proposed for writer identification in [1] and were also used on historical documents in [5]. They are determined on the binary image taking into consideration either the black pixels (the ink trace) or the white pixels (the background). We consider the white runs that capture the regions enclosed inside the letters and also the empty spaces between letters and words. There are

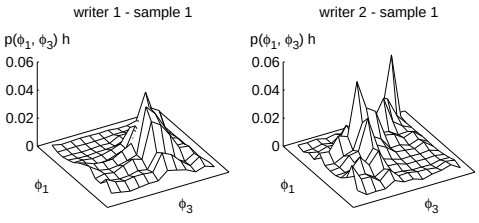

Figure 4. Surface plots of the contour-direction co-occurrence PDF $p(\phi_1, \phi_3)h$ for two writers. Every writer has a different "probability landscape".

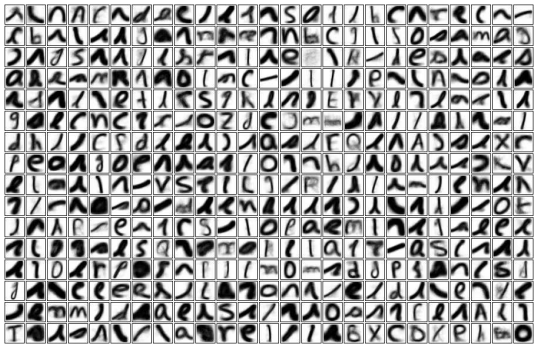

Figure 5. Shape codebook generated by k-means clustering and containing 400 graphemes.

two basic scanning methods: horizontal along the rows of the image (*f5h*) and vertical along the columns of the image (*f5v*). Similarly to the contour-based directional features presented above, the histogram of run lengths is normalized and interpreted as a PDF.

3.2. Allographic features

Our allograph-level method, similar to the approach described in [2], assumes that every writer is a stochastic generator of ink-blob shapes, or graphemes [11]. The PDF of grapheme usage in a given sample is characteristic of each writer and is computed using a common shape codebook obtained by clustering [3]. To make this approach applicable to free-style handwriting (both cursive and isolated), a segmentation method [12] is used yielding graphemes (sub- or supra-allographic fragments) that often will not overlap a complete character. This writer identification method involves three processing stages:

1) Handwriting segmentation: the ink is cut at the minima in the lower contour for which the distance to the upper contour is comparable to the ink-trace width. The graphemes are then extracted as connected components, followed by size normalization to 30x30 pixel bitmaps.

2) Shape codebook generation: k-means clustering ($k = 400$, see Fig. 5) was applied to a training set containing 41k graphemes extracted from 130 samples (65 writers) from the ImUnipen set. The codebook graphemes act as prototype shapes representative for the types of shapes to be expected as a result of handwriting segmentation. In [3], we show that the identification technique described here is robust to design choices regarding the size of the codebook and the clustering algorithm used to generate it.

3) Grapheme-usage PDF computation: one bin is allocated to every grapheme in the codebook and a shape occurrence histogram is computed for every handwritten sample. For every ink fraglet extracted from a sample after segmentation, the nearest codebook grapheme g is found using Euclidean distance and this occurrence is counted into the corresponding histogram bin. The histogram is normalized to a PDF $p(g)$ that acts as the writer descriptor used for identification and verification.

The perfect segmentation of individual characters in free-style script is still unachievable and this represents a fundamental problem for handwriting recognition. Nev-

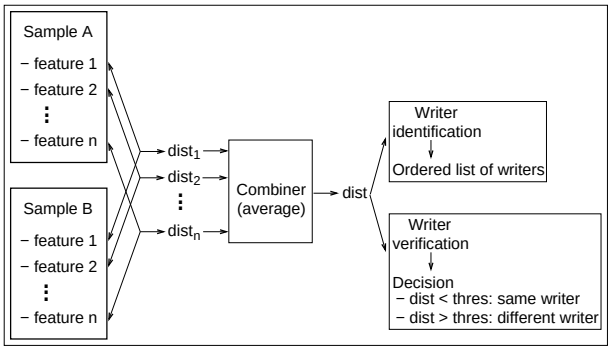

Figure 6. Feature fusion method: the distances generated by the individual features are averaged (using simple or weighted average) and the result is then used in writer identification and verification.

ertheless, the ink fraglets generated by our imperfect segmentation procedure can still be effectively used for writer identification. The essential idea is that the ensemble of these simple graphemes still manages to capture the shape details of the allographs emitted by the writer.

4. Feature matching and fusion for writer identification and verification

A large number of distance measures between the feature vectors were tested, we will report only on the best performing ones. The χ^2 distance is used for matching a query sample q and any other sample i from the database:

$$\chi^2_{qi} = \sum_{n=1}^{Ndims} \frac{(p_{qn} - p_{in})^2}{p_{qn} + p_{in}} \qquad (2)$$

where p are entries in the PDF, n is the bin index and $Ndims$ is the number of bins in the PDF.

Writer identification is performed using nearest-neighbor classification in a "leave-one-out" strategy. For a query sample q, all the other samples $i \neq q$ are ordered in a sorted hit list with increasing distance to the query q, using a selected feature. Ideally the first ranked sample should be the pair sample produced by the same writer. If one considers, not only the nearest neighbor (Top 1), but rather a longer list of neighbors starting with the first and up to a chosen rank (e.g. Top 10), the chance of finding the correct hit (the recall) increases with the list size.

Writer verification is performed in the classical Neyman-Pearson framework of statistical decision theory. By varying the decision threshold, Receiver Operating Characteristic (ROC) curves are computed for all features. The Equal Error Rate (EER) is used to quantify in a single number the writer verification performance.

The considered features are not totally orthogonal, but nevertheless they do offer different points of view on a handwritten sample. It is therefore natural to try to combine them for improving performance, this being the main focus of the present paper. In our feature combination scheme, the final unique distance between any two handwritten samples is computed as the average (simple or weighted) of the distances due to the individual features participating in the combination (see Fig. 6). In feature combinations, Hamming distance performed best:

Table 2. Writer identification and verification performance of individual features on the Large dataset (900 writers, 2 samples per writer, lowercase handwriting). The features are explained in Table 1.

	Feature	Identification		Verification
		Top 1	Top 10	EER
f1	$p(\phi)$	43	72	7.1
f2	$p(\phi_1, \phi_2)$	80	91	4.8
f3h	$p(\phi_1, \phi_3)$ h.	65	84	5.9
f3v	$p(\phi_1, \phi_3)$ v.	59	82	9.1
f4	$p(g)$	76	92	5.8
f5h	$p(rl)$ h.	8	29	16.6
f5v	$p(rl)$ v.	10	34	12.1

$$H_{qi} = \sum_{n=1}^{Ndims} |p_{qn} - p_{in}| \qquad (3)$$

The χ^2 distance, due to the denominator, gives more weight to the low probability regions in the PDFs and maximizes performance for each individual feature. Hamming distance generates comparable distance values for the different features and offers a common ground with slight advantages in feature combinations.

5. Results

Table 2 gives the writer identification and verification performance of the individual features considered here. There are important differences in performance among the different features: the best performer is the contour-hinge PDF (feature *f2*: Top-1 80%, Top-10 91%, EER 4.8%), followed by the grapheme PDF (feature *f4*: Top-1 76%, Top-10 92%, EER 5.8%). The contour-angle combination features *f2*, *f3h* and *f3v* deliver significant performance improvements over the basic directional PDF *f1*. This confirms the general principle that joint probability distributions do capture more information from the input signal. And, despite their higher dimensionalities, reliable probability estimates can be obtained when a few handwritten text lines are available (more than three in our dataset).

The run length PDFs, despite having the worst performance among the echelon of features selected in this paper, in fact do perform better than a number of other known writer identification features, e.g. entropy, wavelets, autocorrelation (see [11] for a wide analysis).

In brief, the results show that the contour-based angle-combination PDFs (*f2*, *f3h*, *f3v*) and the grapheme-emission PDF (*f4*) outperform the other features. They constitute the gist our text-independent approach to writer identification and verification.

The features studied in the paper can be grouped into 3 broad categories (see table 1): contour-based directional PDFs (*f1*, *f2*, *f3h*, *f3v*), grapheme emission PDF (*f4*) and run-length PDFs (*f5h*, *f5v*). We will analyze combinations of features within and between these broad feature groups. As stated earlier, feature fusion is performed by distance averaging. Assigning distinct weights to the different fea-

Table 3. Writer identification and verification performance of feature combinations on the Large dataset.

Feature combination	Identification		Verification
	Top 1	Top 10	EER
f3: *f3h* & *f3v*	73	89	5.0
f5: *f5h* & *f5v*	33	63	7.5
f1 & *f4*	81	94	3.3
f1 & *f5*	67	90	3.6
f2 & *f4*	86	95	2.9
f3 & *f4*	84	95	3.9
f3 & *f5*	80	94	3.7
f4 & *f5*	83	95	3.2
f1 & *f4* & *f5*	85	96	2.8
f2 & *f4* & *f5*	87	96	2.6
f3 & *f4* & *f5*	87	96	3.3

tures participating in the combination yields only very small performance improvements as will be shown further. This lead us to prefer simplicity and robustness here and report feature combination results obtained by plain distance averaging. This is equivalent to using the cumulative distance obtained over all the combined features.

First, we consider the natural combinations *f3h* with *f3v* and *f5h* with *f5v* (first two rows of table 3). Features *f3* and *f5* are therefore obtained by combining the two orthogonal directions of scanning the input image. Compared to their single horizontal or vertical counterparts, the fused features perform markedly better and they will be used, as such, in future combinations.

It is important to note that further combining directional features (*f1* & *f2*, *f1* & *f3*, *f2* & *f3* or *f1* & *f2* & *f3*) did not produce extra improvements over the performance of the best feature involved in the combination. Rather, the experimental results show that improvements are obtained by combining features from different feature groups. In the results given in table 3, the combined performance exceeds the performances of all individual features involved in the combination. The best performing feature combinations fuse directional, grapheme and run-length information yielding writer identification rates of Top-1 85-87% and Top-10 96% with an EER around 3% in verification.

Fig. 7 shows the results obtained by taking a weighted combination of features *f3* and *f4*: $d = \lambda d_3 + (1 - \lambda) d_4$, where λ is the mixing coefficient. Only marginal improvements are attainable over the performance corresponding to simple distance averaging at $\lambda = 0.5$. These results are, in fact, representative for extensive weight optimization tests carried on different combinations and generating, in the end, very small additional improvements. When combining 3 features, the performance landscape as a function of the two mixing coefficients has a broad peak in the neighborhood of $\lambda_1 \approx \lambda_2 \approx 0.3$.

Such a direct feature combination by simple distance averaging is possible in our case because the fused features are PDFs (that sum up to 1) and, for a chosen pair of samples, the Hamming distances produced by the different features lie roughly within the same range. We also tried Borda rank combination schemes and an SVM distance combiner and with rather dismal results.

After feature extraction, feature matching / fusion and performance calculation, our programs generate HTML files containing numerical results (distances, ranks, thresholds) and hyperlinks to the written samples. Fig. 8 shows a hit list generated by our system, dubbed GRAWIS for Groningen Automatic Writer Identification System. A web browser can then be used to visualize these HTML files. For a chosen query sample, writer identification searches can be run using a battery of different features or feature combinations. Fig. 9 shows a *false reject* error and Fig. 10 shows a *false accept* error. The examples were selected to illustrate problematic cases where the *within-writer* variability arguably exceeds the *between-writer* variability, at the fringes of the Bayes decision boundary in the writer verification task.

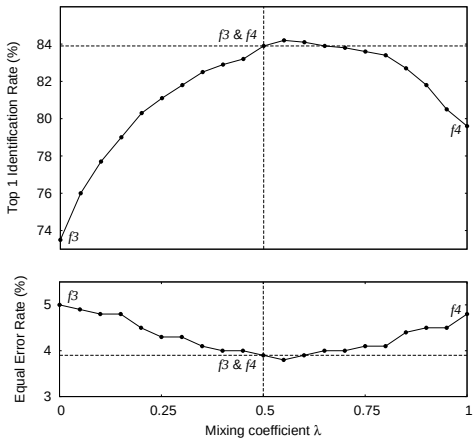

Figure 7. Writer identification and verification performance for a weighted combination of *f3* and *f4*. Only marginal improvements are obtainable over the performance levels of the simple average (the horizontal lines) corresponding to $\lambda = 0.5$. These results are representative for extensive weight optimization tests carried out for different feature combinations.

6. Conclusions

In the writer identification and verification methods presented here, the computer is completely unaware of what has been written in the samples. Our features are independent of the textual content of the handwritten images: the writing is merely seen as a texture characterized by joint directional PDFs that operate at the scale of the ink-trace width or as a simple stochastic shape-emission process characterized by a grapheme occurrence PDF that operates at the scale of characters.

Our features capture different aspects of handwriting individuality and operate at different levels of analysis and different scales. Combining textural and allographic features yields very high writer identification and verification performance. The presented fusion method based on simple distance averaging diminishes the risk of a biased solution, while capturing most of the achievable increases in writer identification and verification performance.

References

[1] B. Arazi. Handwriting identification by means of run-length measurements. *IEEE Trans. Syst., Man and Cybernetics*, SMC-7(12):878–881, 1977.

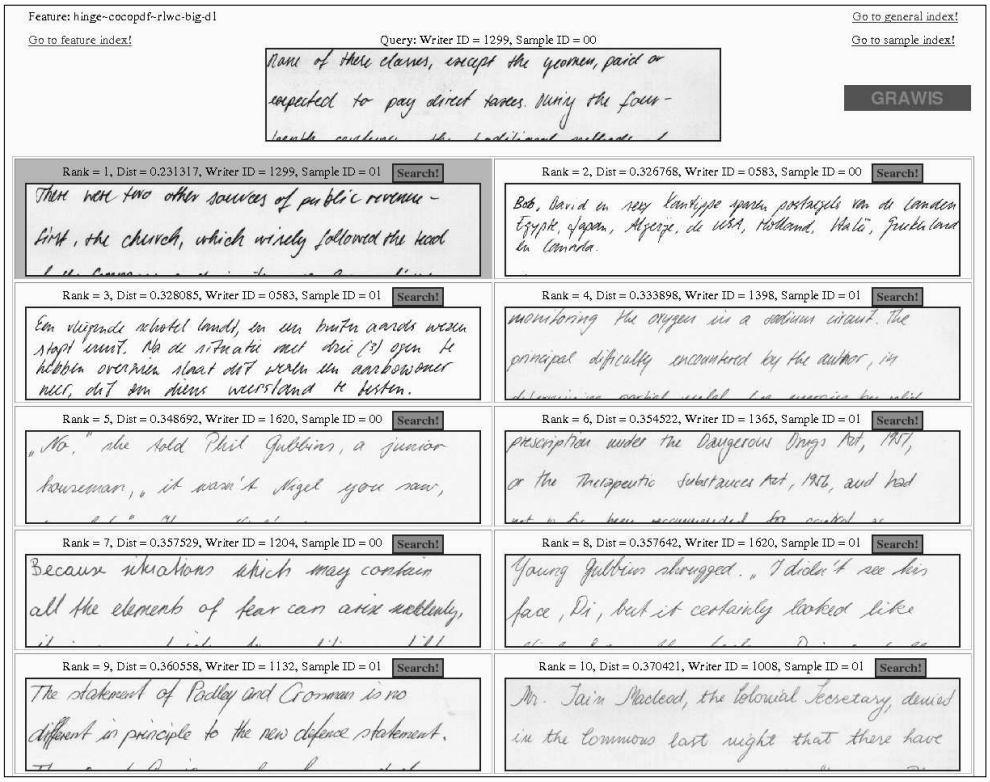

Figure 8. A successful writer identification hit list generated by GRAWIS using the best performing feature combination *f2* & *f4* & *f5*. The query sample is in the top-center position and the best-matching sample (rank 1) was written by the same writer. A uniform handwriting style can be observed across the query sample and at the top of the hit list.

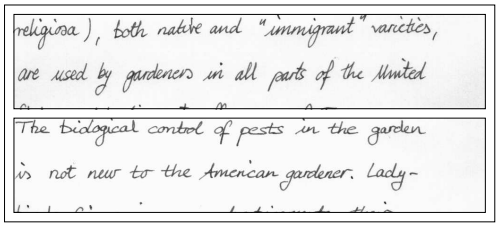

Figure 9. Example of a *false reject* error in writer verification: the two samples were written by the same person, but the system wrongly decided the opposite.

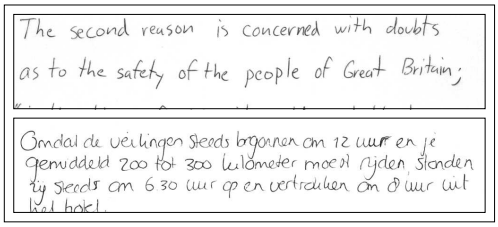

Figure 10. Example of a *false accept* error in writer verification: the two samples were written by different persons and the system took the wrong decision.

[2] A. Bensefia, T. Paquet, and L. Heutte. A writer identification and verification system. *Pattern Recognition Letters*, 26(10):2080–2092, 2005.

[3] M. Bulacu and L. Schomaker. A comparison of clustering methods for writer identification and verification. In *Proc. of 8th ICDAR*, pages 1275–1279, 2005.

[4] M. Bulacu, L. Schomaker, and L. Vuurpijl. Writer identification using edge-based directional features. In *Proc. of 7th ICDAR*, pages 937–941, 2003.

[5] I. Dinstein and Y. Shapira. Ancient hebraic handwriting identification with run-length histograms. *IEEE Trans. Syst., Man and Cybernetics*, SMC-12(3):405–409, 1982.

[6] F. Maarse, L. Schomaker, and H.-L. Teulings. Automatic identification of writers. In G. van der Veer and G. Mulder, editors, *Human-Computer Interaction: Psychonomic Aspects*, pages 353–360. Springer, New York, 1988.

[7] U.-V. Marti and H. Bunke. The IAM-database: an english sentence database for off-line handwriting recognition. *Int. J. on Doc. Analysis and Recognition*, 5(1):39–46, 2002.

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

[9] H. Said, T. Tan, and K. Baker. Personal identification based on handwriting. *Pattern Recognition*, 33:149–160, 2000.

[10] A. Schlapbach and H. Bunke. Using HMM-based recognizers for writer identification and verification. In *Proc. of 9th IWFHR*, pages 167–172, 2004.

[11] L. Schomaker and M. Bulacu. Automatic writer identification using connected-component contours and edge-based features of uppercase western script. *IEEE Trans. on PAMI*, 26(6):787–798, 2004.

[12] L. Schomaker, M. Bulacu, and K. Franke. Automatic writer identification using fragmented connected-component contours. In *Proc. of 9th IWFHR*, pages 185–190, 2004.

[13] L. Schomaker and L. Vuurpijl. Forensic writer identification: A benchmark data set and a comparison of two systems. Technical report, Nijmegen: NICI, 2000.

[14] S. Srihari, S. Cha, H. Arora, and S. Lee. Individuality of handwriting. *J. of Forensic Sciences*, 47(4):1–17, 2002.

[15] B. Zhang, S. Srihari, and S. Lee. Individuality of handwritten characters. In *Proc. of 7th ICDAR*, pages 1086–1090, 2003.

[16] E. Zois and V. Anastassopoulos. Morphological waveform coding for writer identification. *Pattern Recognition*, 33(3):385–398, 2000.