IMPROVING AUTOMATIC WRITER IDENTIFICATION

Laurens van der Maaten Eric Postma

IKAT, Universiteit Maastricht, P.O. Box 616, 6200 MD Maastricht

Abstract

State-of-the-art systems for automatic writer identification from handwritten text are based on two approaches: a statistical approach or a model-based approach. Both approaches have limitations. The main limitation of the statistical approach is that it relies on single-scale statistical features. The main limitation of the model-based approach is that the codebook generation is time-consuming. We attempt to improve automatic writer identification by overcoming the limitations of both approaches. For the statistical approach we evaluate multi-scale statistical features and find one of them to improve the identification performance of single-scale features. For the model-based approach we show that the usage of Kohonen maps for codebook generation is unnecessary, and that a randomly generated codebook is more efficient. We conclude that multi-scale features may enhance identification performances and that random codebooks are to be preferred over Kohonen-based codebooks.

1 Introduction

The main task of forensic handwriting analysis is the identification of the writer of handwritten text. An example of handwritten text by two different writers is shown in Figure 1. Usually, the text has to be assigned to one of a list of writers, e.g. suspects in a criminal case. Currently, writer identification is performed by forensic handwriting experts. A recent study revealed that the judgements of these experts lack reliability [2]. The important, sometimes even decisive, role that these judgements play in criminal courts, prompts for a more objective way of handwriting analysis. Artificial intelligence offers approaches for realising the automatic assignment of handwritten text to a writer.

We distinghuish two main approaches to automatic handwriting analysis: (1) the statistical approach and (2) the model-based approach. The statistical approach entails a statistical analysis of features extracted from the handwritten text [1]. The model-based approach involves the use of pre-defined models of small strokes of handwriting called graphemes [6].

The outline of the remainder of this paper is as follows. Section 2 describes both approaches in more detail. Section 3 presents an overview of our extensions and adjustments to both approaches. Then, in section 4, our experiments are described. Section 5 discusses our results. Finally, section 6 presents our conclusions.

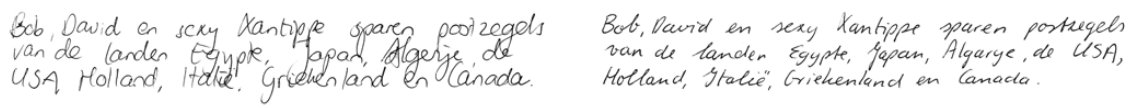

Figure 1: Examples of handwritten text created by two different writers.

2 Approaches to writer identification

In this section we discuss the statistical approach and the model-based approach to writer identification. Both the statistical and the model-based approaches consist of two stages: a feature-extraction

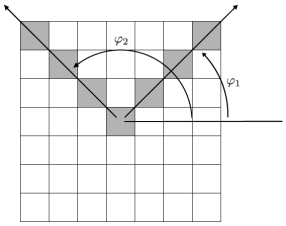

Figure 2: Angle pair $p(\varphi_1, \varphi_2)$.

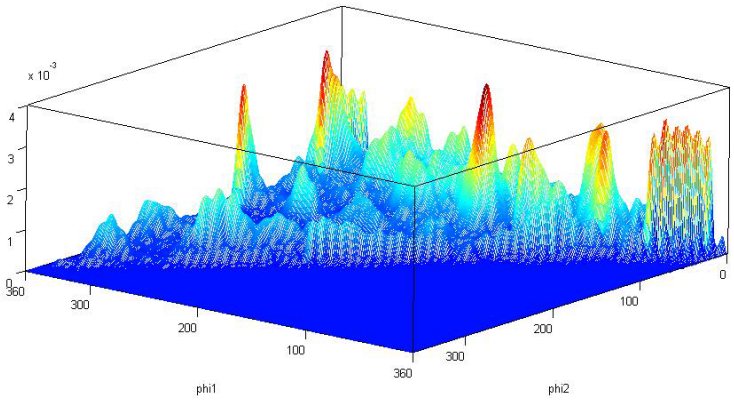

Figure 3: Three-dimensional histogram plot of the edge-hinge distribution showing the frequency of occurrence of angle pairs.

stage and a classification stage. In the feature-extraction stage, features are extracted from handwriting and are stored in feature vectors. In the classification stage, the feature vectors are mapped onto classes representing the writers.

2.1 Statistical approach

Research in automatic writer identification has mainly focused on the statistical approach. This has led to the specification and extraction of statistical features such as run-length distributions, slant distribution, entropy, and edge-hinge distribution. An overview of statistical features is given by Bulacu *et al.* [1]. They found that the edge-hinge distribution feature outperforms all other statistical features. Therefore, we focus on this feature.

Edge-hinge distribution is a feature that characterizes the changes in direction of a writing stroke in handwritten text. The edge-hinge distribution is extracted by means of a window that is slid over an edge-detected binary handwriting image. Whenever the central pixel of the window is *on*, the two edge fragments (i.e. connected sequences of pixels) emerging from this central pixel are considered. Their directions are measured and stored as pairs. A joint probability distribution $p(\varphi_1, \varphi_2)$ is obtained from a large sample of such pairs. An example of an angle pair is shown in Figure 2.1. Figure 2.1 shows an example of an edge-hinge distribution.

For the edge-hinge feature, Bulacu *et al.* [1] found an identification performance of 63% on the *Firemaker* dataset using 250 distracting writers. The main limitation of the edge-hinge feature is that it only evaluates changes in direction on a single scale, rather than on multiple scales.

2.2 Model-based approach

The model-based approach relies on a codebook of models of graphemes. Graphemes are small strokes of handwriting, which are extracted by applying a robust segmentation algorithm on a handwriting image. Graphemes differ from the edge fragments used for the construction of edge-hinge distributions because of the used segmentation algorithm.

In Schomaker *et al.* [6], a codebook of graphemes is generated by training a Kohonen SOFM [3]

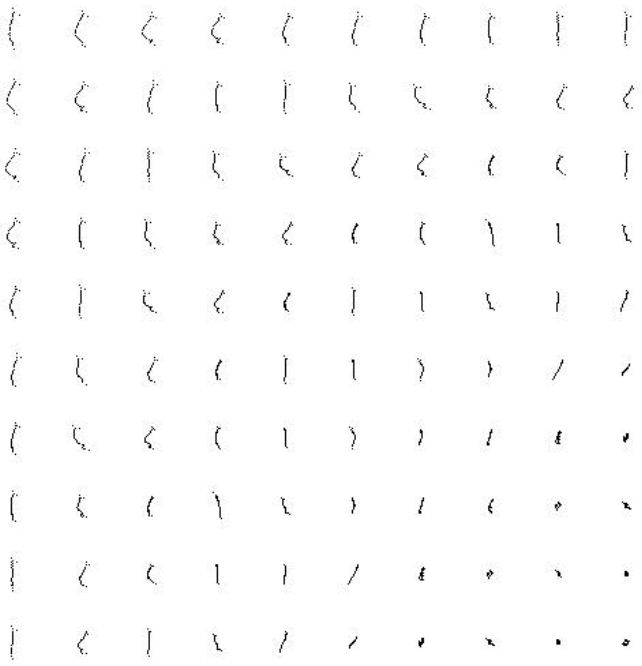

Figure 4: Grapheme codebook created by Kohonen SOFM.

on a large number of grapheme contours. The resulting codebook, of which an example is shown in Figure 4, is used to construct feature vectors in the following way. Given a fragment of handwriting, all graphemes are extracted and matched to the grapheme models in the codebook. The matching is based on the Euclidean distance between the grapheme contours. For each grapheme model, the number of successful matches is counted, yielding an approximation of a probability density function. In this approach, the writer is viewed as a grapheme generator that yields a characteristic probability density function. This function is taken as a feature for writer identification.

When combining this feature with the edge-hinge feature, Schomaker *et al.* [6] achieved a writer identification performance of 97% on the *Firemaker* dataset using 150 distracting writers. A limitation of this approach is the long training time of the Kohonen SOFM. Schomaker *et al.* [6] report a training time of up to 122 hours on a Beowulf high-performance cluster with 128 nodes. In addition, a Kohonen SOFM may get stuck in local minima.

3 Improving approaches to writer identification

In the previous section we have described two approaches to writer identification and identified their limitations. In this section, we propose extensions to both approaches in an attempt to overcome these limitations.

In the statistical approach, we specify and test two new statistical features. In the model-based approach, we compare Kohonen-trained grapheme codebooks, as presented by Schomaker *et al.* [6], with grapheme codebooks constructed by random selection.

3.1 Improving the statistical approach

For the statistical approach we identified one limitation: statistical features are obtained using a single scale. We try to overcome this limitation by defining two new multi-scale features.

The first feature is a variation on the edge-hinge distribution feature. The edge-hinge distribution, as described in section 2.1, is a feature measured on a single scale. However, characterizations on multiple scales often outperform single-scale characterizations. This idea of multiresolution provides its strength to i.e. wavelet analysis. Therefore, we experimented with combinations of

edge-hinge distributions created using different fragment lengths (i.e. window sizes). Henceforth, we will refer to the resulting features as edge-hinge combinations.

Second, we performed experiments using wavelet features. Wavelet features have shown to produce promising results in various digital imaging applications [7]. Additionally, wavelet analysis is a multi-scale analysis technique. However, wavelet analysis has not yet been applied to automatic writer identification. In order to extract wavelet features, first line segmentation is performed using horizontal run-lengths. A 50×50 pixel window is slid over the middle of these lines, and wavelet transforms are applied on the contents of this window. A number of wavelet features (i.e. coefficients obtained by performing wavelet transforms of different level, type or direction) is selected. On the resulting feature vectors, PCA is performed, conserving approximately 90% of the variance in the data. This results in a 50-dimensional feature vector.

3.2 Improving the model-based approach

In subsection 2.2, we observed that grapheme features are effective writer-specific features. However, their main limitation is that the construction of grapheme codebooks is a time-consuming process. We try to overcome this limitation with grapheme codebooks constructed by random selection. Graphemes are extracted in the same way as described by Schomaker *et al.* [6]. However, no time-consuming Kohonen SOFM training is performed. Instead of training we randomly draw a number of graphemes from the large set of graphemes. The selected graphemes form the codebook that is used for the construction of grapheme features.

4 Experiments

We performed experiments to evaluate the performance of both improved approaches on the *Firemaker* dataset. This set contains handwritings of 250 Dutch writers. In our experiments, we use a part of the Firemaker dataset consisting of two pages of standard written text per writer.

4.1 Methodology

Our experiments were performed using the following validation method. We trained our classifiers on the first pages of each writer. The identification performances are estimated using a test set consisting of all second pages. The performance of an approach is estimated by its identification performance, which is the percentage of writers that an approach correctly identifies. A higher identification performance means that an approach performs better. Identification performances are measured for various list sizes, where the list size is the number of writers the classifier is allowed to select.

In all experiments, classification is performed using a 1-nearest neighbour classifier (using chi-square distance). For experiments in the statistical approach we used 250 distracting writers. In the model-based approach, we used 150 distracting writers, since the handwritings of 100 writers are necessary to create a grapheme codebook.

4.2 Experiments in the statistical approach

The identification performances of individual features (both existing and new) in the statistical approach for various list sizes are shown in Figure 5.

Identification performances obtained using wavelet features are not shown in the Figure 5. The best performance using wavelet features was achieved by a combination of 4-, 5-, and 6-level horizontal and vertical Daubechies D4 coefficients. This feature achieves an identification performance of 13% (for list size 1).

Identification performances of edge-hinge combinations are not shown in Figure 5 either. Edge-hinge combinations outperform the original edge-hinge distribution with a maximum of 12%. The identification performances for edge-hinge combinations (for list size 1) are shown in Table 1.
Table 1 reveals that edge-hinge combinations outperform edge-hinge distributions by approximately 11% .

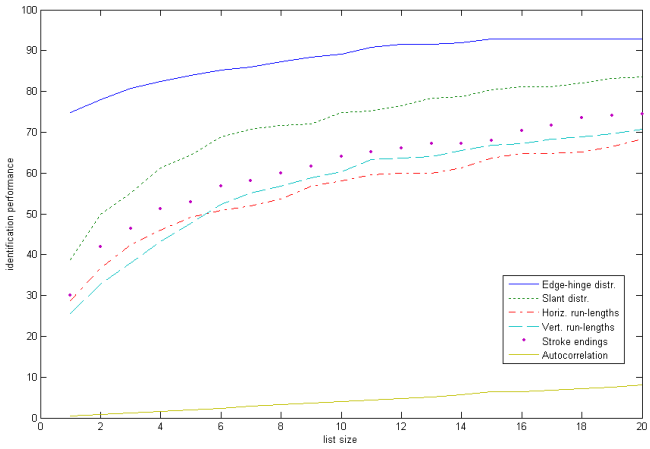

Figure 5: IP's of various features.

Table 1: IP's of edge-hinge combinations.

Fragment lengths	*Identification perf.*	*Fragment lengths*	*Identification perf.*
$\{3\}$	68%	$\{5,7\}$	74%
$\{5\}$	70%	$\{5,9\}$	77%
$\{7\}$	70%	$\{7,9\}$	72%
$\{9\}$	69%	$\{3,5,7\}$	80%
$\{3,5\}$	77%	$\{3,7,9\}$	78%
$\{3,7\}$	77%	$\{5,7,9\}$	76%
$\{3,9\}$	79%	$\{3,5,7,9\}$	81%

4.3 Experiments in the model-based approach

We compared the identification performances of codebooks constructed by Kohonen SOFM training with codebooks constructed by random selection. The results achieved by both construction methods are shown in Table 2(a) and 2(b). The performances reveal that the use of random codebooks produces roughly the same identification performances as the use of Kohonen-trained codebooks. However, the use of random grapheme codebooks yields a large computational gain. The construction of a random grapheme codebook takes approximately 2 seconds, whereas we measured a computation time of 146 hours for the construction of a codebook of size 500 by SOFM training (on a dual Xeon 3.2GHz).

Table 2: IP's using Kohonen SOFM and random codebooks.

Map size	*Identification perf.*
100	78%
200	79%
300	79%
400	79%

(a) IP's using Kohonen codebooks.

Map size	*Identification perf.*
100	72%
200	79%
300	82%
400	77%

(b) IP's using random codebooks.

4.4 Combining both approaches

In our last experiments, we combined edge-hinge combinations with features from the model-based approach. Combining features, which are produced using a codebook of size 400, with the edge-hinge combination $\{3, 5, 7, 9\}$ increases the identification performance to 97% (using 150 distracting writers).

This result is the same as the result presented in Schomaker *et al.* [6], despite the fact that edge-hinge combinations outperform the edge-hinge distribution. This might be caused by the ceiling effect in the dataset, since an identification performance of 97% corresponds to 6 incorrect classifications only. Therefore, a larger dataset is necessary to establish a significant improvement of our combined approach over that of Schomaker *et al.* [6].

5 Discussion

Four phenomena can be observed from the results presented above.

First, we observe that introducing multi-scale analysis to the edge-hinge distribution, yielding edge-hinge combinations, improves results with a maximum of 11%. This improvement is achieved despite the high dimensionality of the resulting feature vectors. This improvement is caused by the multi-scale property of edge-hinge combinations.

Second, we observe that the codebook shown in Figure 4 looks different than the codebook presented by Schomaker *et al.* [6]. This may be caused by a difference in the used representation of the graphemes.

Third, we observe that random codebooks achieve the same results as Kohonen-trained codebooks. The time-consuming, self-organizing map training is therefore unnecessary. As long as certain elements are included in a codebook, it performs equally good as a codebook created by a self-organizing map. Since codebook sizes are reasonably large, these elements can be included in the codebook by random selection. Our results suggest that the two-dimensional topological ordering of codebook vectors, as performed by Kohonen's SOFM, does not aid writer classification.

Last, the wavelet features we examined showed disappointing results. We can probably improve identification performances by using different transforms, such as the Gabor wavelet transform or the banana wavelet transform [4]. However, wavelet features will not be able to equal identification performances as achieved by edge-hinge features or grapheme features.

6 Conclusions

In our research, we achieved the same identification performance as presented in Schomaker *et al.* [6]. As a result, we were unable to show that our improved approaches lead to a better performance. Nevertheless, our research leads to three important new observations.

First, in the construction of a grapheme codebook, the (time-consuming) training of a self-organizing map as reported elsewhere is not necessary to obtain high identification performances. Second, identification performances of the edge-hinge distribution can be improved by combining edge-hinge features from different window sizes. Third, wavelet features perform worse in the handwriting domain than in other digital imaging domains.

Especially the first observation is important, since it provides a handle for further research. We surmise that in the model-based approach results can be improved by changing the way the codebook is constructed, since it is not likely that constructing a codebook by random selection creates an optimal model. Therefore, experiments using other clustering methods (e.g., k-mediods clustering) should be performed. Furthermore, combining edge-hinge combinations with the approach by Schlapbach *et al.* [5] might produce interesting results.

References

[1] M. Bulacu, L. Schomaker, and L. Vuurpijl. Writer identification using edge-based directional features. In *Proceedings of ICDAR 2003*, pages 937–941, Edinburgh, UK, 2003.

[2] M. Kam, G. Fielding, and R. Conn. Writer identification by professional document examiners. *Journal of Forensic Sciences*, 42:778–785, 1997.

[3] T. Kohonen. *Self-organization and associative memory:* 3^{rd} *edition*. Springer-Verlag New York, Inc., New York, NY, USA, 1989.

[4] G. Peters, N. Krüger, and C. von der Malsburg. Learning object representations by clustering banana wavelet responses. In *Proceedings of the* 1^{st} *STIPR*, pages 113–118, Prague, 1997.

[5] A. Schlapbach and H. Bunke. Off-line handwriting identification using hmm based recognizers. In *Proceedings of the* 17^{th} *ICPR*, pages 654–658, Cambridge, UK, 2004.

[6] L. Schomaker, M. Bulacu, and K. Franke. Automatic writer identification using fragmented connected-component contours. In *Proceedings of the* 9^{th} *IWFHR*, pages 185–190, Tokyo, Japan, 2004.

[7] J.S. Walker. *A Primer on Wavelets for Scientists and Engineers*. CRC Press, Boca Raton, FL, USA, 1999.