Pattern Recognition 47 (2014) 2255–2265

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Writer identification using oriented Basic Image Features and the Delta encoding

Andrew J. Newell [a,b,*], Lewis D. Griffin [a,b]

[a] Department of Computer Science, University College London
[b] CoMPLEX, University College London

ARTICLE INFO

Article history:
Received 30 January 2013
Received in revised form
21 September 2013
Accepted 30 November 2013
Available online 12 December 2013

Keywords:
Writer identification
Handwriting
Texture
oBIF Columns
Delta encoding

ABSTRACT

We describe how oriented Basic Image Feature Columns (oBIF Columns) can be used for writer identification and how this texture-based scheme can be enhanced by encoding a writer's style as the deviation from the mean encoding for a population of writers. We hypothesise that this deviation, the Delta encoding, provides a more informative encoding than the texture-based encoding alone. The methods have been evaluated using the IAM dataset and by making entries to two top international competitions for assessing the state-of-the-art in writer identification. We demonstrate that the oBIF Column scheme on its own is sufficient to gain a performance level of 99% when tested using 300 writers from the IAM dataset. However, on the more challenging competition datasets, significantly improved performance was obtained using the Delta encoding scheme, which achieved first place in both competitions. In our characterisation of the Delta encoding, we demonstrate that the method is making use of information contained in the correlation between the written style of different textual elements, which may not be used by other methods.

1. Introduction

Writer identification is the problem of determining authorship of written script based upon its characteristic style rather than its lexical content. The most common example is handwritten text, where the writing technique and anatomy of an individual give rise to a particular style of writing, but the problem can also include other written forms such as music scores [1–3].

The problem is well established in forensic science, where human experts have made comparisons to establish the identity of the author of documents in legal cases (see [4] for a review.) The problem has been approached using computer vision methods since the 1970s [5] and interest remains strong, as demonstrated by multiple competitions associated with major conferences [6,7,3]. The primary application remains the automation of writer identification for forensic science, but the techniques can also be applied to historical document analysis [8–11] and profiling [12,13].

Analysis of handwriting by human experts has typically relied upon the identification of distinguishing elements of an individual's writing, which can then be used to compare a new document against examples of known authorship [4]. Several automated methods have attempted to copy this process, by extracting elements of the text and then comparing them with the same elements found in labelled examples of writing [14,15]. These methods have the advantage of being able to make direct comparisons between the most informative elements of the text across different passages. However, they ignore correlations, if they exist, between the style of different text elements.

Other methods, drawing on advances in texture recognition in computer vision, have treated images of handwriting as textures, without the need to recognise individual words or characters within the text. As these methods encode the whole text, any correlation between the style of different elements of text will be encoded. On the other hand, as such methods take no account of the words in the text, comparisons of passages of text may be impaired if the distribution of characters or words differs significantly between the two passages.

In this work we present a method for writer identification that makes use of a texture-based encoding and takes account of constituent words in the text. In doing this we attempt to create a style vector, referred to as the Δ encoding, for each writer that is independent of the passage of text. This is done by calculating an estimate of the mean encoding across a population of authors and subtracting this from the encoding for each author. The texture-based encoding we use is the oBIF Column histogram scheme [16] which has previously been used for character recognition.

The paper is structured as follows. In Section 2, we summarise previous approaches to automated writer identification. We describe the oBIF Column encoding scheme in Section 3

* Corresponding author.
E-mail address: a.newell@ucl.ac.uk (A.J. Newell).

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.11.029

and evaluate it using the IAM dataset. In Section 4 we introduce the Delta encoding scheme. To evaluate our method, we made entries to both the 2011 ICDAR [6] and 2012 ICFHR [17] Arabic writer identification competitions, both of which were designed to identify the state-of-the-art in writer identification techniques. This is followed by a full characterisation of the method using the datasets from the handwriting competitions. We then look at how the new method compares to a more standard texture-based approach, without the delta stage. In Section 5, we look at the differences between texture-based and traditional methods and discuss the implications for the problem of writer identification. Finally, we provide a discussion and summary of the results.

2. Related work

Automated writer identification has typically been divided into the online problem and the offline problem. Both versions use images of handwriting samples, but the online version also includes data on the path of the pen (e.g. [18,19]) and is therefore considered an easier task. In this work we are concerned solely with the offline problem.

Offline writer identification has been investigated for a range of languages and alphabets. Early studies, such as those by Srihari et al. using 1500 writers [14,15], established the feasibility of automated writer identification for English writers. This has since been demonstrated for a range of languages and alphabets, including Arabic [20–25], Farsi [26], Kannada [27], Bangla [28], Chinese characters [29–32], music scores [1–3,33] and multiscript applications [34].

Approaches to writer recognition can generally be divided into two groups, which we refer to as allographic and texture-based methods. Allographic methods extract elements of the text, such as characters or bigrams, and make comparisons in a like-for-like manner. For example, instances of the bigram *th* may be extracted from a passage of text and compared with other instances (or allographs) of *th* found in a training set of images. The first stage in these methods is the detection and segmentation of the allographic elements [35–41]. Once segmented these elements may be used as greyscale images [35,36] or encoded according to characteristics such as gradient histograms and concavity [42,43,14,15]. A classifier, such as Nearest Neighbour, is then used to assign each element from the test set to an author from the training set.

As with manual methods used in forensic science, allographic methods can be used to select certain specific features that are the most discriminative for writer identification. For example, Bhardwaj et al. have shown that certain bigrams, *th* and *he*, and characters, *d* and *f*, were the most discriminative in their study of English writer identification [44]. The ability to extract such highly informative elements of the text may be advantageous when large quantities of text are available from each author. However, allographic methods have two major disadvantages. First, they involve comparing particular elements of the text like-for-like, these elements have to be present in both pieces of text. It would be impossible, for example, to compare two words which had no common letters. Therefore, when dealing with small amounts of text, we would expect the performance of allographic methods to suffer. Second, such techniques are unable to make comparisons across different characters. If, for example, there was a correlation between how authors wrote the letter *a* and the letter *e* then this would not be picked up by these methods.

Texture-based methods may overcome some of these problems. With these methods, a piece of handwriting is encoded without attempting to identify the content of the text and thus comparisons are made across the whole piece of text. Many different texture encoding schemes have been used from simple analysis of the ink texture [45] to those that draw on modern texture recognition methods from computer vision. Some make use of the fact that handwriting is generally made up of line segments and use edge-based [46,47] or directional features [48,49]. Some methods use features specific to writing, such as chain codes [50], geometrical features [51] or run length measurements [5]. Other methods use a filtering approach, such as Gabor filtering [52], Hermite features [53] or wavelet-based approaches [54,20,55,56]. The current best-performing texture-based methods use grey level co-occurrence matrix (GLCM) features [57] and local phase quantisation (LPQ) features [58].

These methods allow for comparisons to be made between pieces of text regardless of their content. This has the advantage that it does not matter whether the pieces of text being compared contain the same allographic elements, potentially reducing the quantity of text required to perform writer identification. Brink et al. have investigated how much text is required for four sets of texture-based features, proposing a minimum of 100 characters [59].

However, unless such methods are absolutely invariant to the content of the text there will be always be a certain level of effective noise present. This may disappear as the quantity of text becomes large, and the content of the text converges on the mean for all text, but will place an upper bound on performance for small quantities of text.

In an attempt to overcome this shortcoming, several methods have made use of both allographic and textural features [42,60,21,61,10,62]. Such methods can typically use information from the document, bigram and character levels (e.g. [10]), combined with a suitable learning framework that enables the information from different sources to be combined for successful recognition. Other methods that do not fit easily into the above classification include those that use Hidden Markov Models [63] or fractals [64].

3. The oBIF Column scheme for writer identification

We begin by describing the oBIF Column encoding scheme which forms the starting point for our method of writer identification. The oBIF Column scheme has been used previously for character recognition [16] and texture recognition [65,66]. Here we describe the scheme in full and explain how it has been adapted for use in writer identification.

When the oBIF Column system is applied to the problem of character recognition, the first stage is to encode the image of the character to be recognised into oriented Basic Image Features (oBIFs) at two scales. In this process, each location is classified according to the local symmetry type and local orientation using a bank of six Derivative-of-Gaussian (DtG) filters of a size determined by the scale parameter, σ. There are seven possible symmetry types which are *slope*, *dark line*, *light line*, *dark rotational*, *light rotational*, *saddle-like* and *flat*.

The *slope* type is accompanied by a signed orientation and the *line* and *saddle* types are accompanied by an unsigned orientation. These orientations are quantised and the number of possible unsigned orientations is given by the parameter ϕ. This means that there are 2ϕ possible orientations for the *slope* type and ϕ possible orientations for each of the *dark line*, *light line* and *saddle* types. The *dark rotational*, *light rotational* and *flat* types have no orientation. This results in a total of $5\phi+3$ different possible oBIF types. The value of ϕ is tuned for each application, but previous work has indicated that a value of 4 is adequate for near-optimal performance [16]. Using this value we end up with 23 oBIF types.

The oBIF calculation takes an additional parameter, ε, which determines whether a location is classified as *flat*. The oBIF