Writer identification by means of loop and lead-in features

Vivian Blankers and Ralph Niels

Radboud University Nijmegen
Nijmegen, The Netherlands
vivianblankers@gmail.com, r.niels@nici.ru.nl

The research presented in this paper was conducted as a Bachelor project [1].

Abstract

Writer identification is an important issue in forensic investigations. In this paper, we propose a novel method for identifying a writer by means of features of loops and lead-in strokes of produced letters. Using a k-nearest-neighbor classifier, we were able to yield a correct identification performance of 98% on a database of 41 writers. These results are promising and have great potential for use in the forensic practice.

1 Introduction

Handwriting is one of the traces by which an individual can be identified. This is important in forensic investigations whenever handwriting is available, because identifying a human could help solving a crime [1].

Human forensic handwriting experts identify writers by comparing questioned handwriting to reference writings and then trying to find out who produced the questioned document. The experts compare these texts one by one. In cases where large databases need to be examined, however, human comparison becomes practically impossible. Also, human comparison is prone to subjective decisions [2]. Therefore, computer systems can be useful to help the experts identifying a writer.

To identify a writer, it is essential to assume that every writer has a unique handwriting [3]. Assuming this, it may be possible to find style specific writings which differentiate a writer from others. To find these styles, it is useful to zoom in on certain parts of a written letter, or allograph, to compare it with other versions of that same letter, and to search for a unique style [4]. Until now, this is done by human experts who judge whether parts of the letter, the suballographs, match. To do this, they compare the global shapes, the trajectories, and zoom in on specific parts like loops, crossings, lead-in and lead-out strokes [4]. We believe that computer systems are able to compare these features more precisely and objectively.

This study concentrates on the features of loops and lead-ins. Loop features are found in the loops of ascenders, as they appear in the letters 'l', 'k' and 'b' and descenders, as they appear in the letters 'g' and 'j' (see Figure 1). Lead-in features are found in almost all letters
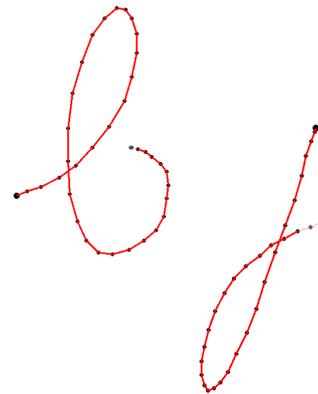
[1] http://www.forensischinstituut.nl

Figure 1. The loops of ascender of a 'b' and the descender of a 'j'.

of the alphabet, especially in cursive handwriting. There has been some research on loops before, but that primarily concentrated on the different sizes of a loop for one writer [5], or was eventually classified as irrelevant [6]. However, Pervouchine & Leedham [6] use another definition of a loop than is used in this study; hidden and lost loops (loops without a gap between the strokes) were not included in their study, whereas online data (which will be explained in Section 2) is used in this study, which enables us to locate and use hidden and lost loops.

As far as we know, there has not been any similar kind of research on lead-in strokes before.

It appears that there is a similarity in the different loops and lead-in strokes produced by a single writer. This means that the loops occurring in the 'l's they write, will be quite the same every time they write this letter 'l', and may even bear resemblance to the loops they produce in other letters containing ascenders, like the 'k', 'b' or even letters containing descenders, e.g. 'j' or 'g'. Equally, the lead-in stroke of an 'a' could correspond with the lead-in strokes of a 'c' or a 'd', for example.

Two main questions which will be answered in this paper:

1. Can a writer be identified by comparing one of the letters he or she wrote to a database of *equal* letters of which the identity of the writer is known, using

only features of the loops and lead-in strokes? For example: If a 'b' of an unknown writer is available, can the writer be found by comparing that 'b' to the 'b's in a labeled database?

2. Can a writer be identified by comparing one of the letters he or she wrote to a database of *similar* letters of which the identity of the writer is known, using only features of the loops and lead-in strokes? For example: If a 'b' of an unknown writer is available, can the writer be found by comparing that 'b' to the 'h's, or the 'k's in a labeled database?

Subquestions concentrate on either loop, lead-in features or both, by taking into account only subsets of the features.

2 Method

2.1 Data collection

A selection of 41 writers (volunteers and paid volunteers) from the *Plucoll* database [7] of handwritten letters was used to test our technique. For this study the separate characters were used. The data in the *Plucoll* set was recorded using a WACOM PL100-V tablet, with an electromagnetic wireless pen, a sample rate of 100 Hz, a resolution of 0.02 mm/unit, and an accuracy of 0.1 mm. This data is online data, which means that time and pressure data are available. Data is recorded not only when the pen is on, but also when it hovers above the tablet. If the pen is on the tablet, the produced trajectory is called *pen-down*, while data recorded with the pen above the tablet is called *penup*. Note that in scanned images of ink, also called *offline* data, *penup* data cannot be distinguished.

2.2 Loop

A loop is defined as a closed curve, where the ends cross each other at some intersection point (see Figure 2). This study concentrates only on the loops of ascenders and descenders. We took into account loop-features of the letters 'b', 'd', 'f', 'g', 'h', 'j', 'k', 'l', 'p', 'q' and 'y'.
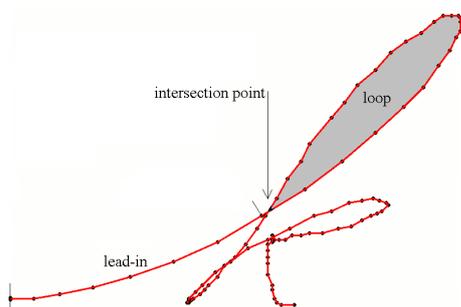

Figure 2. The ascending loop and the lead-in stroke of a 'k'.

2.2.1 Loop features

To compare writers based on the loops and lead-in strokes they produce, a program which calculates vari-

ous features was developed. The most important features are described below. Only the letters that contain lead-in strokes and/or loops were considered.

Length

The length of the total trajectory of the letter is calculated by adding up the Euclidian distances between each couple of succeeding coordinates. The relative length of the loop, with respect to the length of the total letter, is also calculated.

Average speed

Using the length of the letter, the average speed of writing can be calculated. As the frequency of the tablet was 100 Hz, which means there is a coordinate registered every 0.01 seconds, the average speed can be calculated using Eq. 1.

$$average\ speed = \frac{length\ in\ mm}{0.01 * number\ of\ coordinates} \quad (1)$$

The relative writing speed in the loop, with respect to the speed in the total letter, is also calculated.

Area

The area of the loop is calculated by using the fact that the loop has become a polygon of coordinates after it was transformed from a continuous signal of the pen, to a discrete representation in the computer. The area of a polygon can be computed by listing the coordinates $(x1, y1), (x2, y2), ..., (xn, yn)$ counterclockwise and using Eq. 2.

$$Area = \frac{1}{2} * (x1y2 - x2y1 + ... + xny1 - x1yn) \quad (2)$$

Width/height-ratio

The width/height-ratio gives the ratio between the width and the height of a loop. The width of the loop is calculated by finding the difference between the minimum and the maximum x-value of a loop, and the height is calculated by finding the difference between the maximum and the minimum of the y-values of the loop (see Figure 3).
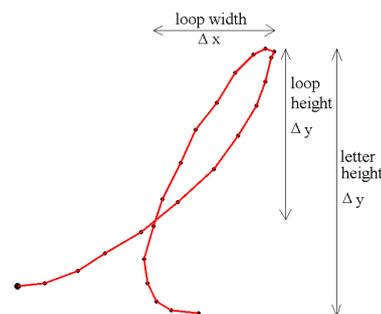

Figure 3. The width and height of a loop and letter.

Relative height

The relative height is the ratio between the height of the loop and the total height of the letter (see Figure 3).

Direction

The direction of a loop is the angle between the x-axis and the vector between the intersection point and the highest point of the loop (in case of an ascender) or the lowest point of the loop (in case of a descender). See Figure 4. The loop of an ascender has a direction between 0 and 180 degrees, while the loop of a descender usually has a direction between 180 and 360 degrees. The direction is calculated using Eq. 3.

$$Direction = \arctan \frac{\Delta \, y \, of \, the \, loop}{\Delta \, x \, of \, the \, loop} \qquad (3)$$
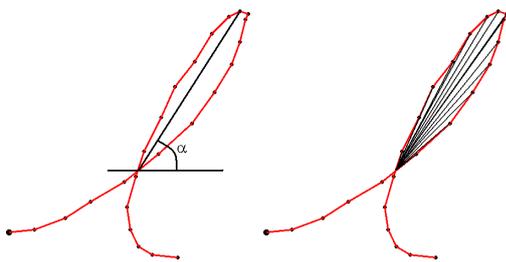

Figure 4. The direction and the average direction of a loop.

Average direction and standard deviation

The average direction of a loop is the average angle between the x-axis and each of the coordinates in the loop (see Figure 4) using Eq. 3. This is calculated by adding up those angles, and dividing the result by the number of angles. The standard deviation is calculated to quantify how much the loop directions differ from the mean. With this information the broad and narrow loops can be distinguished (see Figure 5).
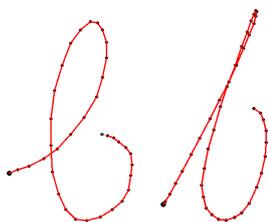

Figure 5. A broad and a narrow loop.

Curvature

The curvature of a trajectory is defined by the average angle between each couple of succeeding vectors (see Figure 6). Given two vectors $\vec{a}$ and $\vec{b}$, the angle between them is calculated using Eq. 4.

$$Angle = \arccos \left(\frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}||\mathbf{b}|} \right) \qquad (4)$$
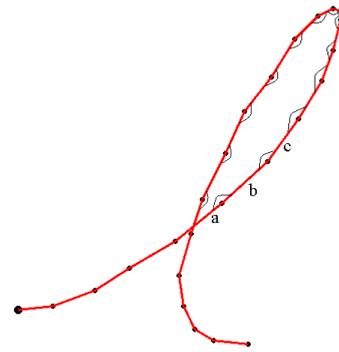

Figure 6. The curvature of a loop.

2.3 Lead-in

Multiple definitions can be used for lead-in strokes. There are different kinds of lead-ins, depending on the letter it belongs to. The overall similarity is that the lead-in stroke always starts at the first coordinate of the letter.

For most ascenders, we defined the lead-in stroke as the part of the letter before the loop (the end of the lead-in stroke is the coordinate where the loop begins). This group contains the letters 'b', 'h', 'k', 'l', 't' (if they have a loop). If these letters do not have a loop, the lead-in stroke ends at the top of the letter (the maximum y-coordinate) (see the 'b's in Figure 7).

The next group of lead-in strokes contains the 'e' and the 'f'. In this group, the lead-in ends at the beginning of the loop if there is one. If the letters lack a loop, no lead-in is defined.

The biggest group of letters, the 'i', 'j', 'm', 'n', 'p', 's', 'u', 'v', 'w', 'y' and 'z', have a lead-in stroke going up, so the lead-in ends where the trajectory starts going down. The 'a', 'c' 'd' and 'q' all have a lead-in which goes to the right, and ends when the trajectory starts going left. The lead-in of an 'a' only has one other constraint, which is that it has to go up, as some people write an 'a' in typewriter fashion ('a' instead of 'a'), where the first part is not a lead-in. The lead-in of a 'c' has another constraint: when people make the letter 'c' look like an 'e', by adding a lead-in, the lead-in ends at the top (maximum y-coordinate) of the letter. This is no problem for finding the lead-in, as our technique already knows which letter it is analyzing.
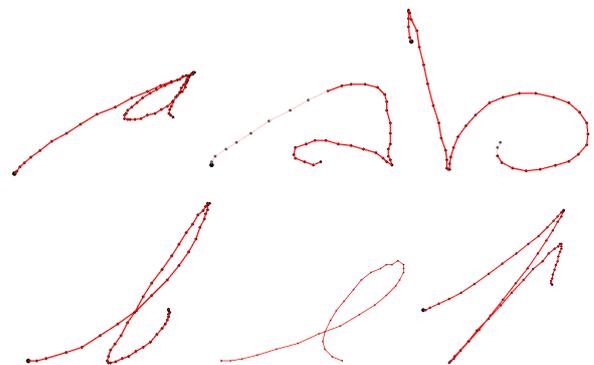

Figure 7. Different types of lead-ins.

2.3.1 Lead-in features

Length

After calculating the absolute length of the lead-in stroke, we calculate the relative length for the lead-in stroke with respect to the total letter.

Average speed

We calculate the writing speed of the lead-in stroke as described in Section 2.2.1. After calculating the absolute speed, the relative writing speed for the lead-in stroke is calculated.

Direction

The direction of a lead-in stroke is the angle between the x-axis and the vector between the first and last co-ordinate of the stroke. Because most lead-in strokes are directed towards the upper right, the angle is usually between 0 and 90 degrees. The direction is calculated as described in Section 2.2.1.

Average direction and standard deviation

The average direction of a lead-in stroke is calculated by summing the angles between the succeeding vectors in the stroke, and dividing the result by the number of vectors (see Figure 8).
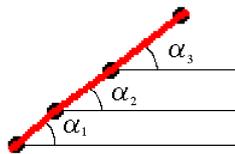

Figure 8. The average direction of a lead-in stroke.

The standard deviation indicates how much the directions differ from the mean.

Curvature

The curvature is the average angle between two vectors of a lead-in stroke, calculated as described in Section 2.2.1.

2.4 Analysis

The total dataset (see Section 2.1) was randomly divided over two subsets of equal size, a *trainset* and a *test-set*. The *trainset* was used to optimize the parameters of the k-nearest-neighbor classifier that was used to perform writer identification tests on the *testset*.

2.4.1 Training

The k-nearest neighbor algorithm is a method for classifying objects based on closest training examples in the feature space. The training examples are mapped into multidimensional feature space, by means of the Euclidean distances. The training phase of the algorithm consists of storing the feature vectors and writers of the training samples. In the actual classification phase, the same features as before are computed for the test letter

(whose writer is unknown). Distances from the test vector to all stored vectors are computed and k closest samples are selected. The new point is predicted to belong to the most numerous writer within the selection [2], (unweighted majority voting). We also implemented a weighted majority voting algorithm, where the nearest vector gets a higher score than the second nearest, et cetera. To optimize the quality of the to-be-obtained results, we used the *trainset* to decide the best value of k (we tried $k = 0, .., 20, 25, 30, 35, 40, 45, 50$), and the best decision algorithm. Therefore, we offered all letters from the *trainset* to the classifier in a leave-one-out manner, and counted how often the classifier returned the correct writer. Weighted majority voting turned out to generate the best results, and the optimal value of k was different for each letter.

2.4.2 Testing

For the tests, we created 9 different groups from of the *testset*, as can be seen in Table 1.

Table 1. The different test groups.

group	features	letters
combination	loop and lead-in features	b, d, f, h, j, k, l, p, q, y
ascenders	loop features	b, d, f, h, k, l
descenders	loop features	g, j, p, q, y
loops	loop features	ascenders and descenders
a-leadins	lead-in features	a, c, d, q
b-leadins	lead-in features	b, h, k, l, t
e-leadins	lead-in features	e, f
i-leadins	lead-in features	i, j, m, n, p, s, u, v, w, y, z
all leadins	lead-in features	all letters, except g, o, r, x

Two different tests were executed. One to find how well a writer could be identified when comparing a letter only to *equal* letters (e.g., comparing questioned 'b's only to 'b's in the database), and one to find out how well this could be done comparing the letters to *similar* letters, or letters belonging to the same group (see Table 1).

Comparing to equal letters

For every letter, the feature-vector is compared to all feature-vectors of that kind, e.g. a 'b'-loop is compared to all other 'b'-loops, using the k-nearest-neighbor classifier and the optimal k found in the previous step. We performed this test in a leave-one-out manner for all letters.

To test whether the identification performance would increase given more available data, we have performed tests on different amounts of available letters. Random selections of letters produced by the same writer were offered to knn, and for each letter, the k nearest samples were listed. The lists for each letter were combined, and the writer with the highest weight was returned by the system. Sets of $1, 2, .., 9, 10$ and $15, 20, 25, 30$ random letters were used.

Comparing to similar letters

To find out to which extend similar letters can be used as a query for writer identification, we performed another test. This test does not compare feature-vectors with feature-vectors of the same letter, but with feature-vectors of similar letters: letters from the same group (see

---

[2]http://en.wikipedia.org/wiki/KNN

Table 1). For example: the lead-in feature-vector of an 'a' is compared to the lead-in feature-vectors of all letters 'c' (both contained in the *a-leadins* group). Similarly, the loop feature-vector of a 'b' is compared to the loop feature-vectors of all 'h's (the *ascenders* group), and also to all loop feature-vectors of all the 'p's in that group (the *loops* group).

In this test, we used $k = 10$ and weighted majority voting for all combinations. This test was done for all groups but the *all-leadins* group, since the feature values differed so much that no useful results could be expected.

3 Results

To test how well our system is able to find the correct writer given an unknown handwritten letter, we counted how often the classifier was able to find the correct writer.

3.1 Equal letters

With 30 available letters per writer, our program correctly identified 85.85% of the lead-in letters. Loop letters yield a higher performance: with 30 available ascender-loop letters, the program correctly classified 96.05% of the writers, whereas descender-loop letters gives a little less performance: 94.76%. With 30 available letters, mixed ascender and descender loop letters, the performance is 98.05%. Given 30 letters that contain both loops and lead-in strokes, a score of 95.05% is obtained. Note that chance level for 41 writers is 2.44%.

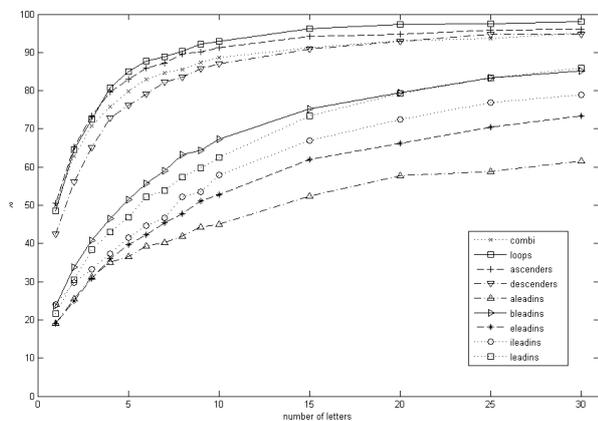The results for different amounts of available letters are summarized in Figure 9.

Figure 9. Results of the *equal letters* test.

3.2 Similar letters

In the second test, we evaluated how well our system is able to identify the writer of a letter using a database of only *similar* letters (i.e., letters that belong to the same group, as described in Table 1). We have performed evaluations for each combination of letters in all groups (i.e., we have used 'g's as query in a database of 'j's, a database of 'p's, a database of 'q's and a database of 'y's, et cetera). The average inner group performances can be found in Table 3.2.

Table 2. Average inner group performance of the *similar letters* test.

group	performance
combination	13.64
ascenders	31.82
descenders	18.04
loops	13.43
aleadins	10.60
bleadins	14.73
eleadins	6.50
ileadins	7.21

4 Discussion

In this paper, we have presented a novel method for writer identification based on suballographic properties of handwriting. We have tested our features in two different settings (*equal* letters and *similar* letters) using a knn classifier. The obtained results show that our system is very well able to identify a writer given a database of equal letters (e.g., a database of 'b's when the query is a 'b'). Correct identification percentages of over 95 percent are achieved if 30 letters of a person are available, but even with less data, very useful results are obtained (over 90% if less than 10 letters are available). Given a database of similar, but not equal, letters (e.g., a database of 'h's when the query is a 'b'), the obtained results are a bit disappointing. The test shows, however, that the loops and leadin-strokes produced by writers differ between letters, even if the letters belong to the same group. This is an important insight that can be used by human forensic experts.

It should be noted that the data we used for our tests is somewhat different from the data used in forensic practice. The most important difference is that the data we used was recorded on an electronic tablet, which gives us the possibility to analyze data produced when the pen was above the paper. Although preliminary tests have showed that the results of pure pendown data are very useful too [1], it should be mentioned that the amount of data that can be used by our system decreases, since a lot of the loops that we see in our data partly consists of penup coordinates. Furthermore, our database consists of only 41 writers, while the size of practical databases are much bigger. Further research has to show how the results will turn out with more writers. On the other hand, if a bigger and more homogeneous dataset would be available, it would be possible to calculate more features, and reach better results for the lead-in strokes.

The relatively weak results of the second test can be explained by the fact that, in contrast with the first test, we have only offered the system one letter at a time, while in the first test, we offered up to 30 samples. Furthermore, we did not optimize the value of k in the k-nearest-neighbor classifier. If we take this into account, the results sometimes come quite close to the other test. Nevertheless, further research could be done on comparing a letter to similar kinds of letters.

To conclude, these results are very promising, and even though it is only a limited study, these results will undoubtedly stimulate further research which could be used

in practice.

References

[1] V. Blankers, "Writer identification by means of loop and lead-in features," Bachelor's Thesis, Radboud University Nijmegen, 2007.

[2] A. Bensefia, T. Paquet, and L. Heutte, "Grapheme based writer verification," in *Proceedings of the 11th Conference of the International Graphonomics Society (IGS2003)*, Scottsdale, USA, 2003, pp. 274–277.

[3] S. N. Srihari, S.-H. Cha, H. Arora, and S. Lee, "Individuality of handwriting: a validation study," in *Proceedings of the Sixth International Conference on Document Analysis and Recognition(ICDAR2001)*, 2001, pp. 106–109.

[4] R. Niels, L. Vuurpijl, and L. Schomaker, "Introducing Trigraph - trimodal writer identification," in *Proc. European Network of Forensic Handwr. Experts*, Budapest, Hungary, 2005.

[5] R. Marquis, F. Taroni, S. Bozza, and M. Schmittbuhl, "Size influence on shape of handwritten characters loops," in *Forensic Science International*, 2007, in Press.

[6] V. Pervouchine and G. Leedham, "Extraction and analysis of forensic document examiner features used for writer identification," in *Pattern Recognition*, vol. 3, March 2007, pp. 1004–1013.

[7] L. Vuurpijl and L. Schomaker, "Finding structure in diversity: a hierarchical clustering method for the categorization of allographs in handwriting," in *ICDAR '97: Proceedings of the 4th International Conference on Document Analysis and Recognition*. Washington, DC, USA: IEEE Computer Society, 1997, p. 387.