JOURNAL OF FORENSIC SCIENCES

J Forensic Sci, May 2017, Vol. 62, No. 3
doi: 10.1111/1556-4029.13345
Available online at: onlinelibrary.wiley.com

PAPER

QUESTIONED DOCUMENTS

John J. Miller,[1] Ph.D.; Robert Bradley Patterson,[1] Ph.D.; Donald T. Gantz,[1] Ph.D.; Christopher P. Saunders,[2] Ph.D.; Mark A. Walch,[3] M.P.H., M.Arch.; and JoAnn Buscaglia,[4] Ph.D.

A Set of Handwriting Features for Use in Automated Writer Identification*,†

ABSTRACT: A writer's biometric identity can be characterized through the distribution of physical feature measurements ("writer's profile"); a graph-based system that facilitates the quantification of these features is described. To accomplish this quantification, handwriting is segmented into basic graphical forms ("graphemes"), which are "skeletonized" to yield the graphical topology of the handwritten segment. The graph-based matching algorithm compares the graphemes first by their graphical topology and then by their geometric features. Graphs derived from known writers can be compared against graphs extracted from unknown writings. The process is computationally intensive and relies heavily upon statistical pattern recognition algorithms. This article focuses on the quantification of these physical features and the construction of the associated pattern recognition methods for using the features to discriminate among writers. The graph-based system described in this article has been implemented in a highly accurate and approximately language-independent biometric recognition system of writers of cursive documents.

KEYWORDS: forensic science, document examination, handwriting, writer identification, writing biometric, automated biometric

The most common task of automated writer identification is a closed-set biometric identification, which assumes that the writer of a document of unknown writership is one of the writers contained within a reference database and modeled by the biometric system. Closed-set identification is distinct from a fundamental forensic writer identification problem, which is to verify that a document of questioned writership came from a specific person (in some instances, to the exclusion of all other possible writers). In this article, we will use the term "writer identification" to be synonymous with closed-set biometric identification.

A writer's biometric identity can be characterized through the distribution of measurements of physical features, with this distribution commonly referred to as a "writer's profile." This article describes features based on skeletonized handwritten text and a method, isomorphic graph classification (see Methods for details), to provide a unified context for effectively using these features in computer-based writer identification. The features and methods described in this article are substantially similar to those used in the proprietary product FLASH ID® (Sciometrics LLC, Chantilly, VA, USA). Because FLASH ID® is proprietary, we cannot provide specific details of the system's functions; however, the careful reader should be able to implement a similar system based on the information in this article.

This method does not use these features for character recognition purposes; that is, the developed method uses the measured features to indicate who wrote a document, rather than what is written on it. The goal of this article is to demonstrate the combination of topology and geometry as an effective method for organizing features based on skeletons of graphemes. Even simple features like distances and angles can be very effective in identifying the writer of a document if there are meaningful ways to organize and compare them.

The skeletons and exemplars of the handwriting in this article are taken from cursive writing samples in the "FBI 500" dataset; these handwriting samples, which were collected by the FBI Laboratory to facilitate studies such as this, contain cursive and handprinted exemplars of a modified London Letter from volunteer writers. The original London Letter was first published by A.S. Osborn in 1929 (1) and was designed to obtain a standard text that contains two or more examples of each letter of the Roman alphabet in upper and lower case, all of the numerals and some various punctuation marks. The FBI modification added two sentences at the end of the paragraph that included specific letter combinations of interest such as "qu," "th," "ll," and the special characters "&" and "$." The full text of the paragraph collected for the FBI 500 dataset and the complete handwritten paragraph from the writer from which the examples were

[1] George Mason University, Document Forensics Laboratory, Volgenau School of Engineering, Nguyen Engineering Building, 4400 University Drive, Fairfax, 22030 VA.

[2] Department of Mathematics and Statistics, South Dakota State University, AME Building, Box 2225, Brookings, SD 57006, USA.

[3] Sciometrics LLC, The Gannon Technologies Group, 14150 Parkeast Circle, Suite 140, Chantilly, VA 20151.

[4] Federal Bureau of Investigation Laboratory Division, Counterterrorism and Forensic Science Research Unit, Quantico, VA 22135.

*This is publication number 14–27 of the FBI Laboratory Division. Names of commercial manufacturers are provided for identification purposes only, and inclusion does not imply endorsement of the manufacturer or its products or services by the FBI. The views expressed are those of the authors and do not necessarily reflect the official policy or position of the FBI or the U.S. government.

†Funded under a contract award to The Gannon Technologies Group by the FBI Laboratory Division.

Received 11 Mar. 2015; and in revised form 19 July 2016; accepted 30 July 2016.

taken is given in Appendix A. Many of the exemplars in the figures of this article are taken from the writings of one writer from the FBI 500 dataset.

We will describe features of handwriting in terms of topology and geometry. Topology refers to the connectedness pattern of the edges of the skeleton of a grapheme and leads to the concept of isomorphic graph classification. Topological properties are not changed when a grapheme is stretched, moved, rotated, or otherwise distorted without changing its connectedness. Geometry refers to actual physical measurements derived from the locations of the pixels in the skeleton of the grapheme. Geometric features do change if the image is distorted. The purpose of the two types of information is that the topological information gives a meaningful frame on which to place the geometric information so that effective comparisons of graphemes can be made.

Handwriting can be segmented into graphemes, which can be done manually or using an automated process. Graphemes are simply collections of the pixels of an image of a handwritten document and may correspond to character types (letters or numerals), or may consist of parts of characters or parts of groups of characters. Graphemes can then be "skeletonized" or "thinned" by one of many available algorithms (e.g., see (2)). Skeletonization leads to a loss of information about the thickness of the writing, but that information can be recovered as a separate set of features at a later point if desired. A skeleton is a set of curves that are each one pixel wide. Fig. 1*a* illustrates skeletons for four graphemes corresponding to characters taken from the writings of a particular individual. These skeletons and all exemplars of the handwriting in Fig. 1 are taken from the writings of one anonymous writer from the "FBI 500" dataset.

A skeleton leads immediately to a configuration of edges and nodes. A node is a point at which lines or pathways intersect, branch, or terminate, and edges are the lines between two nodes. In Fig. 1*a*, the leftmost two skeletons each have four nodes and three edges with the same topology (isomorphic) to a "T" shape. The two skeletons on the right each have six nodes and five edges isomorphic to an "H" shape. Although each pair of skeletons is isomorphic, the shapes of the characters are quite different. Shape is described easily here, namely the character type: the two leftmost shapes are "u" and "n", and the two rightmost shapes are "m" and "u". However, note that the features based on the skeleton of the first (left) "u" do not correspond to features based on the skeleton of the second (right) "u" because the topological structures of the two characters are different.

In the more general situation where graphemes do not correspond to character types (Fig. 1*b*), a concept of "shape code" is required. In Fig. 1*b*, the first grapheme is composed of parts of the "sin" in "business," the second grapheme corresponds to parts of "he" in "there," and the third grapheme corresponds to parts of "re" in "there." Each skeleton in Fig. 1*b* is isomorphic to an "H" shape. However, although the features based on the second and third skeletons might be comparable (although certainly many would take very different numerical values), the first skeleton has a fundamentally different shape and its features would not be comparable to the features of the other two skeletons (see Methods for a way of describing shape for graphemes that do not correspond to characters).

This article demonstrates the isomorphic graph class as an effective method for organizing features based on skeletons of graphemes. Even simple features can be very effective in identifying the writer of a document; we present both simple and more complex features in Results and Discussion.

This article is organized as follows: Background contains a review of some features used by other researchers. Methods provides a thorough description of isomorphic graphs classes; illustrates graphemes, their associated isomorphic graphs, and shape codes; and describes how graphemes can be created corresponding to characters (which may require analyst input) or created from very small bits of handwriting called protographemes (which requires no analyst input). Results and Discussion contains descriptions of features based on skeletons for edges and for loops, illustrates features selected by a statistical procedure to compare one type of character for two different writers, and describes the performance of classifiers based on a subset of the features described in this article and compares that performance with the results of other researchers. Conclusions summarizes the results and suggests future research in this area.

Background

Review of Features Used by Other Researchers

Various types of features derived from image analysis, pattern recognition techniques, graphemes, and forensic document examination have been used for automated writer identification. For coverage of early approaches, please refer to (3). In our descriptions below, we have used the language and descriptions of the researchers we cite. Because different authors may have used the same term to refer to somewhat different concepts, we refer the reader to the original cited documents for clarification of details. In all cases below, "grapheme" refers to a collection of pixels. As shown in Fig. 1*a*,*b*, these pixels may or may not form a recognizable character.

Writer identification methods may use features measured on words, lines, and whole documents. One approach measures features of single words based on morphological waveform coding (4). Others focus on features of individual lines by studying their connected components, loops, and contours (5,6). Features of individual lines also serve as input to hidden Markov and Gaussian mixture models to identify writers (7,8). Partial and whole document images also offer features: several techniques use them with morphological waveforms (9), fractal analysis (10), Zipf's law (11), Gabor filters and correlation measurements (12), and chain codes (13) to identify writers. Additionally, distributions

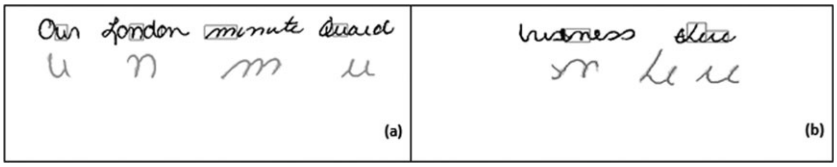


FIG. 1—*Sample skeletons for graphemes (a) based on characters and (b) not based on characters.*

of contour directions and run-lengths of black or white pixels from entire documents yield features (14–16).

At a finer level of detail, features may come from graphemes. The distribution of common graphemes defines features in several studies: Bensefia et al. (17–19) introduce the use of individual graphemes as features. Rather than measuring features summarized from an entire block of text, they find invariant classes of allographs by clustering graphemes according to correlation measurements of their pixels. These "writer invariants" from across a population of writers then define a feature space.

Bulacu (16), Bulacu and Schomaker (20–22), and Schomaker, Franke and Bulacu (23) also build on the use of graphemes introduced by Bensefia et al. (17–19) to form allographic features. Instead of writer invariants, they construct grapheme codebooks by clustering the graphemes from an entire population of writers. The graphemes in the codebook then define the bins of a grapheme emission probability density function.

Forensic document examiners have identified elements for discriminating handwriting (24,25). Srihari et al. (26) adapt similar features to arrive at computational measurements that they label as macro- and micro-features. Macro-features capture aspects such as darkness, slope, slant, and height at a global level. Micro-features pertain to images of allographs and include gradient, structural, and concavity measurements. Srihari et al. (27) develop style features similar to micro-features measured on whole words and pairs of letters.

Leedham and Chachra (28) present new computational features of handwritten digits for identifying writers. Leedham and Pervouchine (29) advance computational features of the letters "d," "y," and "f" and the grapheme "th" based on those used in forensic document examination, which relate to distances and angles of individual graphemes. Walch and Gantz (30,31) introduce features of graphemes described by isomorphic graph classes as illustrated in this article; the features in this article represent extensions of these early prototypes used by Walch and Gantz.

The choice of features and the choice of analysis method are closely tied. Selected features may be successful for one analysis method but unsuccessful for others. Some researchers summarize the grapheme level features at the document level using distributions (16–23). Others create a score for each grapheme and sum those scores to obtain an overall document score (26–36,38). The connection between choice of features and analysis method is demonstrated in more detail below.

Methods

The methods described below are used in systems by the authors of this article to organize and characterize graphemes. The concepts of isomorphic class and geometric class are the basis of these systems. The initial part of the current section is spent developing these two concepts. Once graphemes can be compared within a particular topological class and geometric class, then features can be defined and measured as shown in the rest of the section. The graphemes can then be compared using those features.

Isomorphic Graphs

Recall that skeletons are sets of lines one pixel in width. These lines can be connected in many different ways, but always form what is called a planar graph in mathematics. The lines in such graphs are called edges and places where lines terminate, or three or more lines intersect are called nodes. Sample skeletons from actual handwriting are given in Fig. 1*a*,*b*.

Graphs are called isomorphic if they have the same connectedness pattern among the nodes. This pattern is not changed if the graphs are stretched, moved, rotated, or otherwise distorted. We will describe an isomorphic graph class by giving an exemplar schematic to which all members of the class are isomorphic. Five potential schematic skeletons for a class with four nodes and three edges are given in Fig. 2*a*. The nodes and edges have been drawn in an exaggerated fashion to draw attention to the fact that bends in edges, even severe ones, do not create nodes. Each skeleton in Fig. 2*a* is isomorphic to the basic "T" shape of the first schematic.

Figure 2*b* contains four potential schematics for a class with six nodes and six edges. All schematics in Fig. 2*b* are isomorphic. This figure illustrates the fact that edges may move inside or outside a perceived loop and still maintain the connectedness pattern. It also illustrates that the connectedness pattern is not affected by whether lines are very straight or highly curved. Properties like curvedness are geometric and are captured as features (see Results and Discussion). Handwriting will be segmented into graphs, which are then assigned to isomorphic classes (Fig. 3).

It may be desirable to insert "pseudo-nodes" (analogous to knots in a spline model) into a grapheme to give better feature measurements. Such pseudo-nodes may be particularly useful for "S"-shaped graphemes (see Appendix B for details on pseudo-nodes).

Using isomorphic graph classes is a critical step in organizing graphemes so that feature measurements on graphemes may be compared in a meaningful manner. However, such comparisons require that the nodes of a graph be numbered in a consistent manner so that the geometric measurements have meaning. For instance, a feature like "the angle from node 1 to node 2" would be useless unless a consistent and organized set of rules for numbering nodes was used; a set of such rules is provided in Appendix C. The set of rules in Appendix C uses both topological aspects of connectedness and geometric aspects of location and angle to number the nodes so that comparisons can be made. Using the rules in Appendix C, Fig. 4*a*,*b* shows the node numbering for the skeletons in Fig. 1*a*,*b*.

Creation of Graphemes

Graphemes may be segmented from handwritten material corresponding to characters. This can be done manually by an

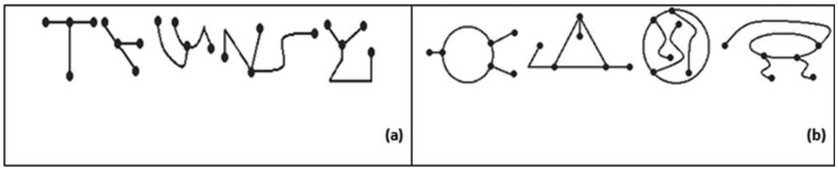


FIG. 2—*Examples of isomorphic schematic skeletons with (a) four nodes and three edges and (b) six nodes and six edges.*

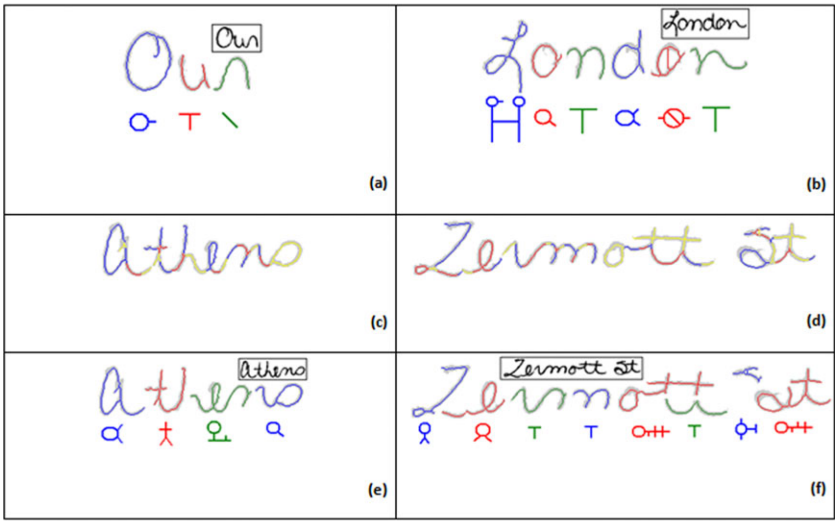


FIG. 3—*Graphemes and isomorphic schematics from the words (a) "Our" and (b) "London"; protographemes created from the words (c) "Athens" and (d) "Zermott St"; (e) graphemes created from the protographemes in (c); and (f) graphemes created from the protographemes in (d).*

analyst or automatically by computer algorithms. Because computer algorithms for character recognition are not completely accurate, the character segmentations described in this article were done manually. Figure 3*a,b* illustrates character segmentations from the first two words of the London paragraph shown in Appendix A. The isomorphic graph schematics are also given, because we must know that to make meaningful comparisons of features.

Manual segmentation is time-consuming and expensive; a fully automated method for creating graphemes is needed to implement an automated handwriting identification system. The outline of such a method is given below. The process begins by taking each line of the handwritten document and oversegmenting it into small contiguous "chunks" of pixels called protographemes. Any automated process or set of rules that creates these protographemes can be used. One set of rules is described in Bucalu (20); that set of graphemes could potentially be tuned to lead to more and smaller chunks of pixels and thus create protographemes. Figure 3*c* illustrates the output of one such automated process yielding 21 protographemes for the word "Athens" from the paragraph in Appendix A. Figure 3*d* illustrates 41 protographemes created from the words "Zermott St." Different protographemes would be created using different algorithms, so it is important to use the same algorithms throughout the process, both for training and for testing documents.

Once protographemes have been created, graphemes are made by combining contiguous protographemes. Decisions about how many consecutive protographemes to combine, whether to allow overlapping graphemes, whether to allow crossing white space, and other details can be made by the designers of any system. Different decisions in these matters will lead to different performance of the resulting system. Figure 3*e,f* illustrates one way to create graphemes from the protographemes in Fig. 3*c,d*, in which each grapheme is composed of between three and seven protographemes. Again, the isomorphic graph schematics are given for each resulting grapheme, because that information is crucial to continued analysis of the graphemes. These figures demonstrate that the resulting graphemes will seldom correspond to characters.

In addition to combining the protographemes to create graphemes, it is also possible to generate additional graphemes from embedded forms within graphemes. Small features such as "spurs" and "holes" within grapheme edges may or may not be significant, and it is often not possible to determine their importance at the grapheme level. By "toggling" certain small edges related to these artifacts, new graphemes can be created that have been simplified by removal of the artifacts. In this context, "toggling" means systematically removing certain small edges. The rules for toggling are to set a threshold size for the edges to be considered and to establish a maximum proportion of the overall grapheme that can be removed (or toggled "off"). For instance, all edges less than five pixels in length can be chosen as candidates for removal and the maximum amount of the overall graph to be eliminated by edge removal can be set to 20%. Given these parameters, various combinations of the selected pixels will be systematically toggled "off" (removed) yielding new simplified graphemes that are no more than 20% smaller than the original. Toggling is especially useful in instances where material is sparse such as postal addresses and short notes.

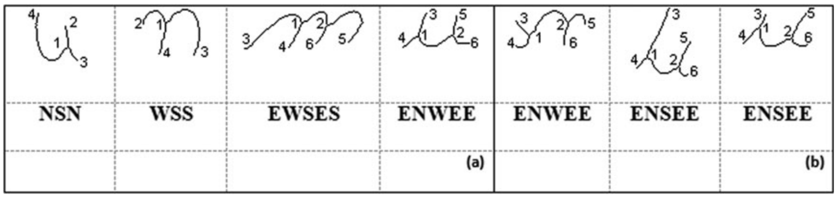


FIG. 4—*Graphemes (a) from Fig. 1*a *and (b) from Fig. 1*b*, with node numberings and shape codes.*

Once graphemes are created, the isomorphic graph class determined, and the nodes numbered, features can be computed. However, meaningful comparisons still cannot be made unless the shapes of the graphemes are taken into account. The assignment of shape is discussed in the next section.

Graphemes and Shape

Even when two skeletons belong to the same isomorphic graph class, their features may not lend themselves to meaningful comparisons. Consider the skeletons in Fig. 1*a*: the leftmost two belong to the same isomorphic graph class (schematic "T") and the rightmost two belong to the same isomorphic graph class (schematic "H"). Nevertheless, we would not want to compare their features. Similarly, all three skeletons in Fig. 1*b* belong to the same isomorphic graph class (schematic "H"), but the first has a fundamentally different shape than the other two. A method of assigning a shape class to each grapheme is needed.

For graphemes based on characters, the most obvious and fundamentally useful concept of shape is simply the character type (letter of the alphabet with case, numeral, or symbol). For example, in Fig. 3*a*,*b*, the graphemes correspond to the character types "O," "u," "r," "L," "o," "n," "d," "o," "n," respectively. Only the two "n"s would be in the same isomorphic shape class group (abbreviated isoshape group). All others would be in unique isoshape groups to be compared with other characters in the same groups.

For graphemes based on protographemes, shape must be determined in some other manner. One way is to use a "shape code," which can be created in many ways. A set of rules that will lead to a unique shape code using an alphabet of four letters corresponding to directions of the compass is given in Appendix D.

The shape codes (and numbered nodes) for the four graphemes from Fig. 1*a* are shown in Fig. 4*a*. These graphemes are based on characters, so there is no need to compute shape codes for them. However, this figure is included to aid the reader in understanding how shape codes are created. Figure 4*b* contains the resulting shape codes (and the numbered nodes) for the three graphemes from Fig. 1*b*. Because these graphemes were created automatically from protographemes, these shape codes are essential in allowing for meaningful comparison of features among writers. As noted previously, the shape codes for the second and third graphemes are the same, but different from the shape code for the first grapheme.

Given that features will now be compared only for graphemes with the same isomorphic graph class and the same shape class (either character type or shape code), we can now define the features that can be used for automated writer identification.

Features Based on Skeletons

There are many very simple features that may be computed once certain key points within the skeleton have been defined. Distances between points, angles between points, offsets (horizontal and vertical) between points, and sines and cosines (which are scaled offsets) between points may all be easily computed. All distances and offsets are measured in units of pixels; angles are measured from 0° to 360° with 0° being "east" and angles increasing counterclockwise.

There are two classes of key points we consider: quarter points and centroids. Quarter points are based on counting pixels in the skeleton from the lower numbered node to the higher numbered node and then using the rules in Appendix E to find the quarter points, which divide the pixels into four parts. The centroids are just the "centers of mass" of the pixels and are computed by simple averaging of the pixel locations in each direction (x or y) for each edge and for the entire grapheme.

Figure 5 contains the letter "n" from Fig. 1*a* with (*a*) the quarter points for each edge marked and with (*b*) the centroids for each edge and the entire grapheme indicated. There are 17 key points even for this simple form; this means there are 136 pairs of points, each of which could lead to one distance, one angle, two offsets, a sine, and a cosine, yielding 816 potential features of this type. For graphemes with more edges, there are correspondingly more such potential features. A user may opt not to use (or not even to compute) some of these features, but there is a rich list from which to choose.

Features Based on Bezier Fits

Bezier curves are used extensively in computer graphics and typography. A Bezier curve is a particular type of smooth curve that can be used to approximate the path of a skeleton. The advantage of a Bezier curve is that it has a precise mathematical form, which can be used to compute values, such as tangents and curvature measures, for the curve at important points (e.g., quarter points defined previously). These values, which are exact for the Bezier curve, then yield an approximate value for the pixels to which the curve was fitted. Appendix F gives some details of how the curves are fit and how tangents and curvature values are calculated from the fitted curve.

Figure 6*a* shows Bezier curves fit to the edges of a letter "a" taken from the paragraph in Appendix A. To show potential variations on the basic technique, the curves were fit to the entire edge in the left-hand diagram and separately to halves of the edge in the right-hand diagram. Note that the curves are smoother in the left-hand diagram but, in the right-hand diagram, the curves correspond more closely to the pixels. The user may opt to use either type of fit as appropriate.

Important information about the trajectory of an edge may be found by computing the tangents to the Bezier curve at the quarter points and recording the angles of these lines. Fig. 6*b* shows the tangent lines for the Bezier curves in the left-hand diagram of Fig. 6*a*. The values of the formula for mathematical curvature can be recorded for each quarter point as well as the average value over all pixels in the edge. These features give numerical

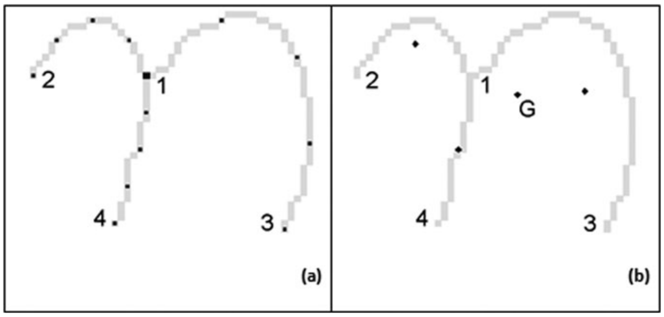


FIG. 5—*Key points within a skeleton illustrating (a) the quarter points for each edge marked and (b) the centroids for each edge and the entire grapheme indicated.*

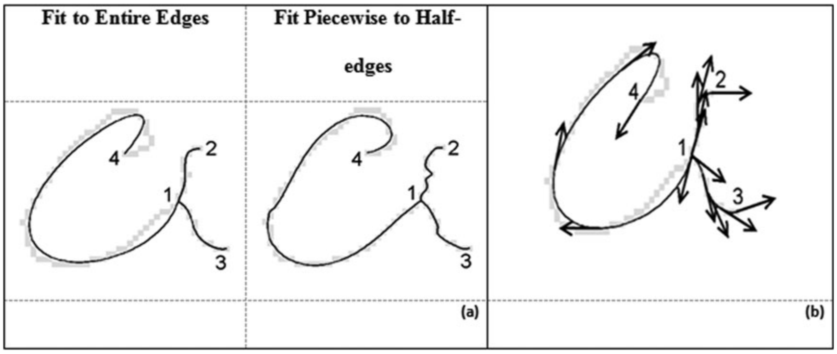


FIG. 6—*(a) Bezier curves fit to edges of a letter "a" and (b) tangents to Bezier fit at quarter points using entire edge fit.*

values to the concept of "curvedness," which can be input into statistical procedures.

Features for Loops

For loops, we consider several features that attempt to ascertain the orientation: the curvature, the roundness, and the size of the loop. Figure 7*a* shows the word "Lloyd" written by the writer of the paragraph in Appendix A and contains four loops and two pseudo-loops (loops 4 and 6 are composed of two and three edges, respectively and are, therefore, not true loops). However, we include the two pseudo-loops so that we may illustrate the different values that result for our features for different shaped loops.

Our first two features measure the orientation (slant) and eccentricity (lack of roundness) of the loop, both of which are computed by considering the coordinates of the pixels in the loop skeleton as bivariate statistical data and computing the principal components (see (40)) for that data. The angle of the vector of the first principal component gives the orientation of the loop. A measure of eccentricity is given by the following score: $e_1 = (1 - \lambda_{\min}/\lambda_{\max})^{\frac{1}{2}}$ (see (40)). This score takes on a value of zero when the figure is perfectly symmetric and a value approaching one as the loop degenerates into a straight line. As a perfect square shape is symmetric, we give a further measure of lack of roundness; this is computed by taking the centroid of the pixels in the loop and computing the distance from the centroid to each pixel.

An alternative measure of eccentricity, e_2, is computed by taking the ratio of the standard deviation of the distances divided by the mean of the distances all multiplied by the square root of three. This score again takes a value of zero for a perfect circle and approaches one as the loop degenerates to a straight line.

We can also fit a Bezier curve to the loop and compute the average value of the mathematical curvature for the pixels in the loop. We scale that value by dividing by the number of pixels in the loop to eliminate the scale dependence of the mathematical curvature described in Appendix F. Figure 7*b* gives enlargements of the curves in Fig. 7*a*, along with the fitted Bezier curves and the principal component axes. Table 1 contains the values of the scores for each feature for each loop.

The angle of the first principal component can illustrate the slant of the loops. The eccentricities show clearly that Loop 3 is least round and Loop 6 is most round. The mean curvatures are greatest for the smallest loops, whereas the scaled mean curvatures show that Loop 6 (which has the least proportion of pixels

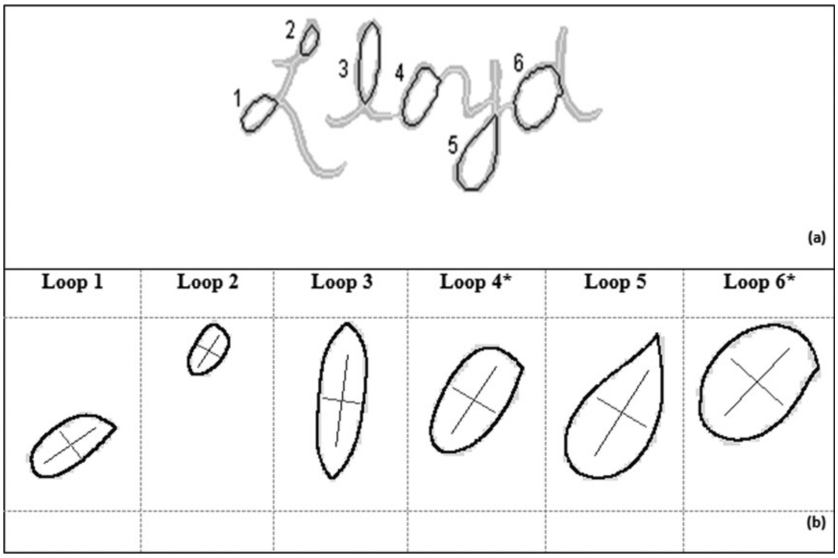


FIG. 7—*Illustration of (a) the six "loops" taken from the word "Lloyd" and (b) the loop feature details. Loops 4 and 6 are not true loops; they are combinations of two and three edges, respectively, and are included for illustrative purposes only.*

TABLE 1—*Loop feature values for the loops in Fig. 7*b.

Feature	Loop 1	Loop 2	Loop 3	Loop 4	Loop 5	Loop 6
Angle of first PC	37.26°	57.96°	82.41°	57.00°	58.30°	47.80°
Eccentricity 1(e_1)	0.84	0.77	0.91	0.81	0.84	0.67
Eccentricity 2 (e_2)	0.44	0.38	0.64	0.40	0.45	0.25
Mean curvature	0.0906	0.1565	0.0380	0.0687	0.0468	0.0718
Scaled mean curvature	0.0240	0.0282	0.0161	0.0247	0.0209	0.0323

where the Bezier curve is almost straight) has the greatest mean curvature, and Loop 3 (which has the greatest proportion of pixels where the Bezier curve is almost straight) has the least mean curvature.

Results and Discussion

The features defined above can be used as a foundation of a statistical procedure for writer identification. We illustrate such a statistical procedure in this section.

Feature Selection by Statistical Procedure

The large number of features presented in the previous section yield many potential features to be used in automated writer identification. Therefore, some sort of statistical procedure must then be used to select the most useful features. This section contains an example of the features selected by one particular algorithm for two writers, neither of which is the true writer of the paragraph in Appendix A (or of the examples presented from that writer's paragraph). We are restricting this example to just graphemes of the lower case letter "n" written using the "T" isomorphic schematic (see Fig. 3b for two examples of similar graphemes and Fig. 5a).

Figure 8 contains 12 examples taken from each of Writer 1 and Writer 2. Note that there is variability in the graphemes within the same writer, but there will also be some differences between the two writers in the measurements of some geometric features (e.g., distances and angles).

The statistical procedure that selected the features illustrated below is described in some detail in Appendix G. The selected features using the grapheme from Fig. 5a (not written by either Writer 1 or Writer 2 so as not to bias the interpretation) are illustrated in Fig. 9. Feature 1 refers to the angle from the central node to the "mid-pixel" of edge 1–2 and hence measures the "arch" of that edge. Larger angles (closer to 180°) describe an edge that is "flatter," and smaller angles (closer to 90°) describe an edge that is "more arched." Feature 2 refers to the angle from the central node to the centroid of edge 1–4. As this edge is quite straight, this feature measures the "slant" of this edge. Larger angles (closer to 270°) represent more "vertical" edges, and smaller angles (less than 270°) represent more "slanted" edges. Feature 3 refers to the distance from node 2 to the centroid of the entire grapheme. This feature is related mostly to the horizontal distance from node 2 to that centroid and hence to the length of edge 1–2; however, it can also be increased by edge 1–3 being longer. Feature 4 refers to the number of pixels in edge 1–3 and hence is a direct measure of its length.

Table 2 contains the mean values of the four selected features in the training data for each writer, and Table 3 gives the interpretation of those means. It must be noted that these means only indicate tendencies. There is much variability among the graphemes for each writer, so there is no absolute separation between writers for any of these features. The values for each feature in the training data are presented in Fig. A4*a* in Appendix G, which illustrates graphically the variability in these graphemes. The reader should note that the tendencies described in Table 3 can be seen in the graphemes in Fig. 8.

Using the statistical procedure described in Appendix G, 22 of the 29 test graphemes (75.9%) for Writer 1 are correctly assigned, and 55 of the 67 test graphemes (82.1%) for Writer 2 are correctly assigned. Five graphemes were assigned to neither writer. An example of a test grapheme written by Writer 1 that is incorrectly assigned to Writer 2 is shown in Fig. 10*a*. An example of a test grapheme written by Writer 2 that is incorrectly assigned to Writer 1 is shown in Fig. 10*b*. By comparing these graphemes to those in Fig. 8, we see that the statistical procedure yields results consistent with predictions using visual examination of these graphemes.

Figure 10*c* contains two graphemes that were not assigned to either writer by our method. The first is an example of an apparently anomalously written grapheme, whereas the second is caused by the "whisker" of the skeleton due to the small ink blob (which might be from "pen drag"). Given that no person writes exactly the same way twice (intrawriter variability) and a

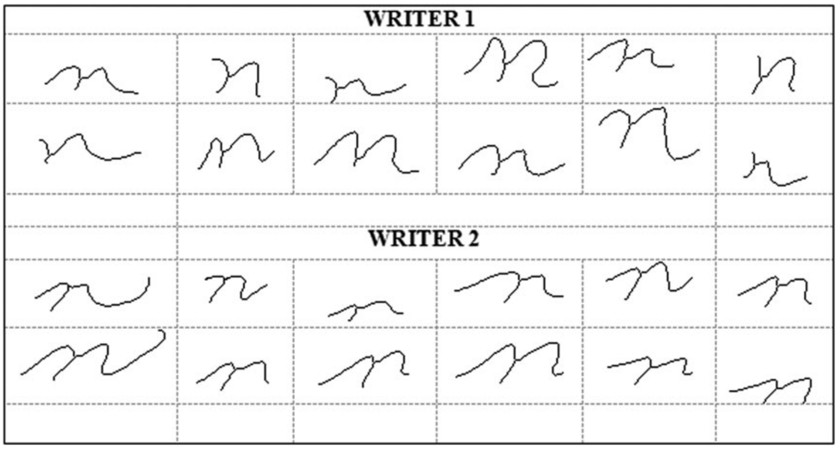


FIG. 8—*Twelve examples of the letter "n" using the "T" isomorphic schematic taken from writing samples from each of Writer 1 and Writer 2.*

relatively small sample of handwriting, it is not surprising to find a few instances of graphemes that may not be assigned to the true writer. However, writer identification with this method is not based on a single grapheme, but the totality of the geometric features measured in the handwriting sample.

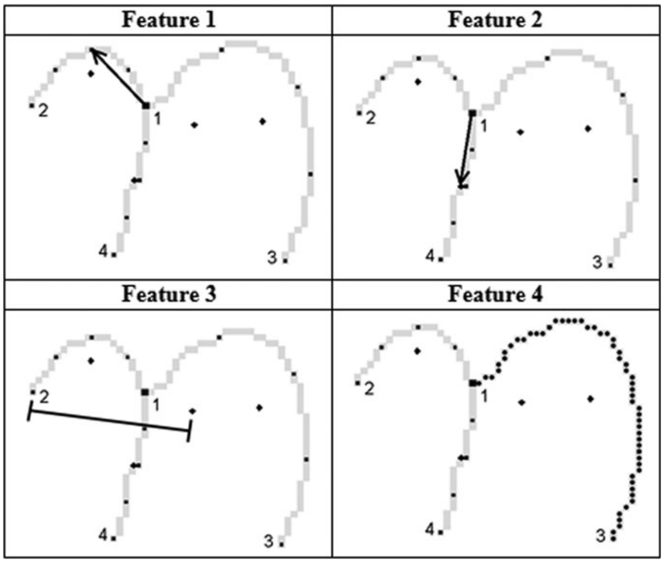


FIG. 9—*Illustration of four features selected by a statistical algorithm to compare Writer 1 and Writer 2. Feature 1 is the angle from node 1 to second quarter point of edge 1–2. Feature 2 is the angle from node 1 to centroid of edge 1–4. Feature 3 is the distance between node 2 and the centroid of entire grapheme. Feature 4 is the number of pixels in edge 1–3.*

TABLE 2—*The mean values of the selected features in the training data.*

Feature Number	Mean Value of Feature	
	Writer 1	Writer 2
1	125.63°	162.22°
2	243.44°	234.43°
3	17.83 pixels	20.42 pixels
4	50.84 pixels	48.28 pixels

TABLE 3—*Interpretation of the writer differences for the selected features in the training data.*

Feature Number	Feature Mainly Refers To	Referred Edge Tends to be	
		For Writer 1	For Writer 2
1	Edge 1–2	More “arched”	“flatter”
2	Edge 1–4	More “vertical”	More “slanted”
3	Edge 1–2	“shorter”	“longer”
4	Edge 1–3	“longer”	“shorter”

Writer Identification Performance

The previous part of this section worked with only one topological class and one geometric class with only two writers. We have also tested using many writers, many topological classes and many geometric classes. We have tested the features described above and the statistical procedures described in Appendix G (with modifications) and presented the results elsewhere (see 30–39). We summarize them here.

Walch et al. (34) found that, in testing documents using a subset of the features above, many statistical methods could give reasonable performance, but our method using pairwise comparisons (which we call the “Competitive Matrix” method), gave the best performance. In a test (34) using 100 possible writers, all 194 test documents are classified correctly. In another test using 300 possible writers, all 590 test documents are classified correctly. In both cases, a 100.0% correct assignment rate is obtained.

The data in Saunders et al. (39) are based on graphemes that were manually segmented to correspond to characters, and only using characters that were written using subset of basic isomorphic schematics were used. Walch et al. (36) used graphemes based on computer segmentation; shape codes were used in place of character type to divide graphemes into “like–with-like” comparisons. In that study, there were 100 possible writers with 200 test documents, and all but one document tested was classified correctly (a 99.5% correct assignment rate).

Conclusions

This article has introduced a system of features based on isomorphic graph class and shape that can be useful in automated handwriting identification. Many of the features described herein have been used in some part by other researchers, although not in the framework of topological and geometric class. Our work embedded these features using a framework based on topological classification and geometric classification of graphemes into a system; this system allows meaningful “like-with-like” comparisons of similar features among different writers.

The exploitation of features also is integrally related to a statistical procedure that can select the important features for comparing any two writers. We have given an example of features selected for two particular writers that shows that the features selected by a statistical procedure can be observed visually. We have described results that demonstrate that the integrated system of features and statistics can produce excellent identification results. The features presented in this article may used by other

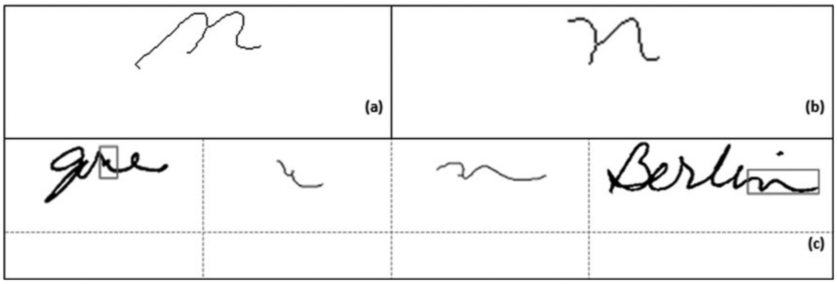


FIG. 10—*Illustration of examples of the letter “n” from (a) Writer 1 misclassified to Writer 2; (b) Writer 2 misclassified to Writer 1; and (c) Writer 1 classified to neither writer.*

researchers in conjunction with other types of statistical systems to achieve improved identification results in automatic handwriting identification.

Acknowledgments

The authors wish to thank Cliff Sutton and Peter Tytell for assistance with background literature and Gabriel Watts and forensic document examiners who provided background information on forensic handwriting comparisons that aided feature selection and system development.

References

1. Osborn AS. Questioned documents, 2nd edn. New York, NY: Boyd Printing Co., 1929.
2. Martin A, Tosunoglu S. Image processing techniques for machine vision. Proceedings of the 29th Florida Conference on Recent Advances in Robotics (FCRAR 2016); 2000 May 4–5; Boca Raton, FL. Boca Raton, FL: Florida Atlantic University, 2000.
3. Plamondon R, Lorette G. Automatic signature verification and writer identification – the state of the art. Pattern Recogn 1989;22(2):107–31.
4. Zois EN, Anastassopoulos V. Morphological waveform coding for writer identification. Pattern Recogn 2000;33(3):385–98.
5. Marti U, Messerli R, Bunke H. Writer identification using text line based features. In: Proceedings of the Sixth International Conference on Document Analysis and Recognition; 2001 Sept 10–13; Seattle, WA. Los Alamitos, CA: IEEE Computer Society, 2001.
6. Hertel C, Bunke H. A set of novel features for writer identification. In: Kittler J, Nixon MS, editors. AVBPA'03 Proceedings of the Fourth International Conference on Audio and Video-Based Biometric Person Authentication; 2003 June 9–11; Guildford, U.K. Berlin, Heidelberg, Germany: Springer-Verlag, 2003;679–87.
7. Schlapbach A, Bunke H. A writer identification and verification system using HMM based recognizers. Pattern Anal Appl 2007;10(1):33–43.
8. Schlapbach A, Bunke H. Off-line writer identification using Gaussian mixture models. In: ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition; 2006 Aug 20–24; Hong Kong. Hong Kong: IEEE Computer Society, 2006;992–5.
9. Said H, Tan T, Baker K. Personal identification based on handwriting. Pattern Recogn 2000;33(1):149–60.
10. Seropian A, Grimaldi M, Vincent N. Writer identification based on the fractal construction of a reference base. In: Proceedings of the Seventh International Conference on Document Analysis and Recognition – Volume 2; 2003 Aug 3–6; Edinburgh, U.K. Los Alamitos, CA: IEEE Computer Society, 2003;1163–7.
11. Pareti R, Vincent N. Global method based on pattern occurrences for writer identification. In: Lorette G, editor. Proceedings of the Tenth International Workshop on Frontiers in Handwriting Recognition; 2006 Oct 23–26; La Baule, France. Los Alamitos, CA: IEEE Computer Society, 2006.
12. Siddiqi I, Vincent N. Combining contour based orientation and curvature features for writer recognition. In: Jiang X, Petkov , N , editors: Proceedings of the 13th International Conference on Computer Analysis of Images and Patterns, CAIP 2009; 2009 Sept 2–4; Münster, Germany. Berlin, Germany: Springer-Verlag, 2009;245–52.
13. Siddiqi I, Vincent N. A set of chain code based features for writer recognition. In: Proceedings of the Tenth International Conference on Document Analysis and Recognition; 2009 July 26–29; Barcelona, Spain. Los Alamitos, CA: IEEE Computer Society, 2009;981–5.
14. Bulacu M, Schomaker L. Writer style from oriented edge fragments. In: Petkov N, Westenberg MA, editors. Proceedings of the 10th International Conference on Computer Analysis of Images and Patterns, CAIP 2003; 2003 Aug 25–27; Groningen, The Netherlands. Berlin, Germany: Springer-Verlag, 2003;679–87.
15. Bulacu M, Schomaker L, Vuurpijl L. Writer identification using edge-based directional features. In: Proceedings of the Seventh International Conference on Document Analysis and Recognition – Volume 2; 2003 Aug 3–6; Edinburgh, U.K. Los Alamitos, CA: IEEE Computer Society, 2003;937–41.
16. Bulacu M. Statistical pattern recognition for automatic writer identification and verification [dissertation]. Groningen, The Netherlands: University of Groningen, 2007.
17. Bensefia A, Nosary A, Paquet T, Heutte L. Writer identification by writer's invariants. In: Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition; 2002 Aug 6–8; Ontario, Canada. Los Alamitos, CA: IEEE Computer Society, 2002;274–9.
18. Bensefia A, Paquet T, Heutte L. Information retrieval based writer identification. In: Proceedings of the Seventh International Conference on Document Analysis and Recognition – Volume 2; 2003 Aug 3–6; Edinburgh, U.K. Los Alamitos, CA: IEEE Computer Society, 2003;946–50.
19. Bensefia A, Paquet T, Heutte L. A writer identification and verification system. Pattern Recogn Lett 2005;26(13):2080–92.
20. Bulacu M, Schomaker L. A comparison of clustering methods for writer identification and verification. In: Proceedings of the Eighth International Conference on Document Analysis and Recognition; 2005 Aug 29- Sept 1; Seoul, Korea. Washington, DC: IEEE Computer Society, 2005;1275–9.
21. Bulacu M, Schomaker L. Combining multiple features for text-independent writer identification and verification. In: Lorette G, editor. Proceedings of the Tenth International Workshop on Frontiers in Handwriting Recognition; 2006 Oct 23–26; La Baule, France. Los Alamitos, CA: IEEE Computer Society, 2006;281–6.
22. Bulacu M, Schomaker L. Text-independent writer identification and verification using textural and allographic features. IEEE Trans Pattern Anal Mach Intell 2007;29(4):701–17.
23. Schomaker L, Franke K, Bulacu M. Using codebooks of fragmented connected-component contours in forensic and historic writer identification. Pattern Recogn Lett 2007;28(6):719–27.
24. Huber RA, Headrick A. Handwriting identification: facts and fundamentals. Boca Raton, FL: CRC Press, 1999.
25. Morris R. Forensic handwriting identification: fundamental concepts and principles. London, U.K.: Academic Press, 2000.
26. Srihari S, Cha S, Arora H, Lee S. Individuality of handwriting. J Forensic Sci 2002;47(4):1–17.
27. Srihari S, Huang C, Srinivasan H. On the discriminability of the handwriting of twins. J Forensic Sci 2008;53(2):430–46.
28. Leedham G, Chachra S. Writer identification using innovative binarised features of handwritten numerals. In: Proceedings of the Seventh International Conference on Document Analysis and Recognition – Volume 1; 2003 Aug 3–6; Edinburgh, U.K. Los Alamitos, CA: IEEE Computer Society, 2003;413–6.
29. Pervouchine V, Leedham G. Extraction and analysis of forensic document examiner features used for writer identification. Pattern Recogn 2007;40(3):1004–13.
30. Walch M, Gantz D. Pictographic Matching: a graph-based approach towards a language independent document exploitation platform. In: Lubbes K, Ronthaler M, editors. Proceedings of the First ACM Hardcopy Document Processing Workshop; 2004 Nov 12; Washington, DC. New York, NY: The Association for Computing Machinery, 2004;53–62.
31. Walch M, Gantz D. Pictographic recognition technology applied to distinctive characteristics of handwritten Arabic text. In: SDIUT'05. Conference Proceedings of the 2005 Symposium on Document Image Understanding Technology; 2005 Nov 2–4; Adelphi, MD. College Park, MD: University of Maryland, 2005;173–84.
32. Gantz D, Miller J, Walch M. Multi-language handwriting derived biometric identification. In: SDIUT'05. Conference Proceedings of the 2005 Symposium on Document Image Understanding Technology; 2005 Nov 2–4; Adelphi, MD. College Park, MD: University of Maryland, 2005;197–209.
33. Gantz D, Miller J, Walch M. Application of pictographic recognition technology for spotting handwritten Chinese words. In: Proceedings of the Summit on Arabic and Chinese Handwriting (SACH06); 2006 Sept 27–28; College Park, MD. College Park, MD: University of Maryland, 2006;75–86.
34. Walch M, Gantz D, Miller J, Saunders C, Lancaster M, Buscaglia J. Evaluation of the individuality of handwriting using FLASH ID – a totally automated language-independent system for handwriting identification. In: Proceedings of the 60th Annual Scientific Meeting of the American Academy of Forensic Sciences; 2008 Feb 18–23; Washington, DC. Colorado Springs, CO: American Academy of Forensic Sciences, 2008;388.
35. Gantz D, Miller J, Saunders C, Lancaster M, Buscaglia J. Statistical characterization of writers for identification. In: Proceedings of the 60th Annual Meeting of the American Academy of Forensic Sciences; 2008 Feb 18–23; Washington, DC. Colorado Springs, CO: American Academy of Forensic Sciences, 2008;390–1.
36. Walch M, Gantz D, Miller J, Buscaglia J. Evaluation of the language-independent process in the FLASH ID system for handwriting identification.

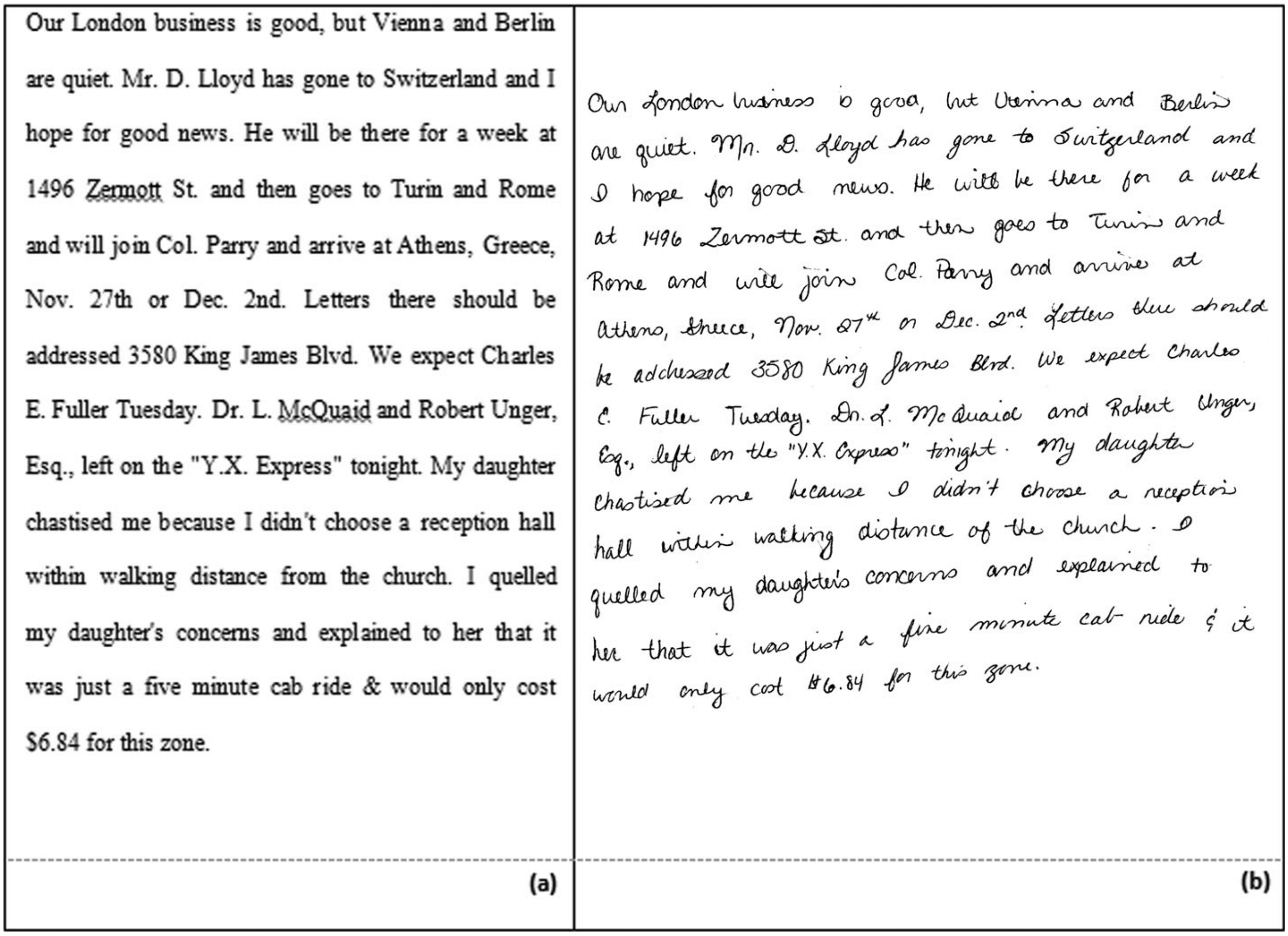
In: Proceedings of the 61st Annual Scientific Meeting of the American Academy of Forensic Sciences; 2009 Feb 16–21; Denver, CO. Colorado Springs, CO: American Academy of Forensic Sciences, 2009;381–2.

37. Saunders C, Hepler A, Davis L, Buscaglia J. Estimation of likelihood ratios for forensic handwriting analysis. Sci Justice 2010;50(1):32.
38. Gantz D, Miller J, Saunders C, Walch M, Buscaglia J. New results for addressing the open set problem in automated handwriting identification. In: Proceedings of the 62nd Annual Meeting of the American Academy of Forensic Sciences; 2010 Feb 22–27; Seattle, WA. Colorado Springs, CO: American Academy of Forensic Sciences, 2010;431–2.
39. Saunders C, Davis L, Lamas A, Miller J, Gantz D. Construction and evaluation of classifiers for forensic document analysis. Ann Appl Stat 2011;5(1):381–99.
40. Johnson R, Wichern D. Applied multivariate statistical analysis, 6th edn. New York, NY: Pearson, 2008;430–69.
41. Joy KI. Bernstein polynomials. Davis, CA: Department of Computer Science, University of Davis, CA, 1996; http://www.idav.ucdavis.edu/education/CAGDNotes/Bernstein-Polynomials.pdf
42. Brown K. Reflections on relativity. MathPages.com 2013;348–65.

Additional information and reprint requests:
JoAnn Buscaglia, Ph.D.
Research Chemist
FBI Laboratory
Counterterrorism and Forensic Science Research Unit
2501 Investigation Parkway
Quantico
VA 22135
E-mail: joann.buscaglia@ic.fbi.gov

Appendix A: Modified London Letter

Figure A1*a* contains the full text of the modified London Letter that was used in collection of the “FBI 500” dataset. Figure A1*b* contains a sample paragraph written by one writer from that dataset. Several examples are taken from this paragraph to illustrate various features and principles.



Our London business is good, but Vienna and Berlin
are quiet. Mr. D. Lloyd has gone to Switzerland and I
hope for good news. He will be there for a week at
1496 Zermott St. and then goes to Turin and Rome
and will join Col. Parry and arrive at Athens, Greece,
Nov. 27th or Dec. 2nd. Letters there should be
addressed 3580 King James Blvd. We expect Charles
E. Fuller Tuesday. Dr. L. McQuaid and Robert Unger,
Esq., left on the "Y.X. Express" tonight. My daughter
chastised me because I didn't choose a reception hall
within walking distance from the church. I quelled
my daughter's concerns and explained to her that it
was just a five minute cab ride & would only cost
$6.84 for this zone.

(a)

Our London business is good, but Vienna and Berlin
are quiet. Mr. D. Lloyd has gone to Switzerland and
I hope for good news. He will be there for a week
at 1496 Zermott St. and then goes to Turin and
Rome and will join Col. Parry and arrive at
Athens, Greece, Nov. 27th or Dec. 2nd. Letters there should
be addressed 3580 King James Blvd. We expect Charles
E. Fuller Tuesday. Dr. L. McQuaid and Robert Unger,
Esq., left on the "Y.X. Express" tonight. My daughter
chastised me because I didn't choose a reception
hall within walking distance of the church. I
quelled my daughter's concerns and explained to
her that it was just a five minute cab ride & it
would only cost $6.84 for this zone.

(b)

FIG. A1—*(a) The full text of the modified London Letter used in collection of the “FBI 500” dataset and (b) a sample paragraph written by one volunteer writer from that collection.*

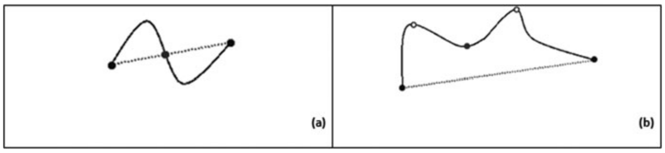


FIG. A2—*Examples of (a) a pseudo-node and (b) a more complicated pseudo-node situation.*

Appendix B: Insertion of "Pseudo-Nodes"

For some graphemes with a sigmoid shape, it may be useful to insert pseudo-nodes into the skeleton so as to make the features more useful. The letter "s" and the numerals "2," "3," and "5" may benefit from such insertions. Figure A2*a* shows an example of such a pseudo-node and illustrates a rule for adding such pseudo-nodes. This rule states that if the line connecting the end nodes crosses the skeleton, place a pseudo-node at the point or points of crossing.

Shapes with more complicated recurving may benefit by including a pseudo-node even if the line connecting the end points does not cross the skeleton. Figure A2*b* illustrates one such possible situation with the solid pseudo-node. Other complicated situations may benefit from adding more than one pseudo-node. Figure A2*b* also illustrates this situation with the hollow pseudo-nodes (or all three pseudo-nodes). Once pseudo-nodes have been added to a skeleton, they should be treated exactly like the original nodes for all purposes in the analysis including the assignment of the grapheme to an isomorphic class.

Appendix C: Description of Rules for Numbering Nodes

The following set of rules yields a consistent rubric for numbering nodes in any grapheme

1 Find the highest degree node or nodes (the degree of a node is the number of edges entering it).
  a If only one such node exists, number it 1. Go to step 2, else go to step 1b.
  b If more than one such node exists, find the leftmost one. If a tie still exists, take the bottommost one. Number that node 1. Go to step 2.

2 Find the highest degree node or nodes remaining.
  a If the degree of the node or nodes is greater than one, then number them by the angle they make with node 1. Begin by considering the direction west and proceed by numbering clockwise. In case of a tie, assign the lower number to the node closer to node 1. Iterate step 2.
  b If the degree of the node or nodes equals one, then divide them into groups based on the number of the node to which they connect. Following the order of the resulting groups, within each group, measure the angle from the central node of that group to the fifth pixel of the edge connecting the central node to the node of degree one. Begin by considering the direction "west" and proceed to number the nodes of degree one clockwise. If all vertices are numbered, stop.

Appendix D: Rules for Assigning Shape Codes to Graphemes

The shape codes we propose are based on the angles from node 1 to each numbered node in succession. Angles are measured counterclockwise starting with "due east" as 0°. The code for a given angle then corresponds to "compass direction." Hence, angles between −45° and 45° become "E," while those between 45° and 135° become "N," those between 135° and 225° become "W," and those between 225° and 315° become "S." Figure A3 illustrates this correspondence.

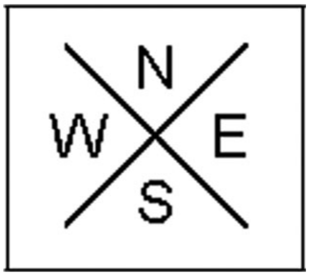


FIG. A3—*Correspondence between angles and shape codes.*

A given grapheme will then have a shape code consisting of the letters ENWS of length one less than the number of nodes in the grapheme.

Appendix E: Rules for Determining Quarter Points of an Edge

The following rule is used to determine the coordinates of the quarter points for an edge. Because an edge is by definition one pixel wide, the pixels can be consecutively numbered from 1 to *n* (where *n* is the number of pixels in the edge) beginning at the lower numbered node. The goal is to divide the interval 1 to *n* into four equal parts and then find the corresponding coordinates of pixels. The zeroth quarter point is defined to be the point numbered one. The fourth quarter point is the point numbered, *n*. It remains to assign the first, second, and third quarter points. Consider the ratios $(n + 3)/4$, $(2n + 2)/4$ and $(3n + 1)/4$, and divide the interval from 1 to *n* into four equal parts. We now need the coordinates of the points indexed by those fractions in the edgeline.

Suppose that one of the above ratios has a decimal value of *w. d*, where *w* is the whole number part and *d* is the decimal part. The desired *x* coordinate is just the following weighted average: $(1\text{-}d)\ x_w + dx_{w+1}$. The *y* coordinate is computed similarly. For example, suppose that $n = 22$, which leads to the ratios 6.25, 11.5, and 16.75, thereby dividing 1–22 into four equal parts. The *x* coordinates of the five quarter points are x_1, $0.75x_6 + 0.25x_7$, $0.5x_{11} + 0.5x_{12}$, $0.25x_{16} + 0.75x_{17}$, and x_{22} respectively.

Appendix F: Fitting a Bezier Curve to an Edge or Loop—Tangents and Curvature

A Bezier curve fit to a collection of pixels is obtained in the following way: number the pixels consecutively from 1 to *n* (where *n* is the number of pixels in the edge or loop). In the case of an arc, begin at the lower numbered node. In the case of a loop, begin where the loop intersects its edge. If the loop does not intersect an edge, begin at its lowest point. Now create a parameter *t* = $(i - 1)/(n - 1)$, where i is the pixel number. Now regress the *x* and *y* coordinates separately on the Bernstein polynomials (see (41)) of degree *d* (where *d* is usually chosen to be 3 or 4). One can choose whether or not to force the Bezier curves to go through the end points of the arc or beginning/ending point of the loop. In the work presented here, we used degree four curves, which were forced to go through the end points.

If we have a parametric curve of the form$\{x\ (t),\ y\ (t)\}$, we can find the slope of the tangents to that curve using the relationship $\mathrm{d}y/\mathrm{d}x = \dot{y}/\dot{x}$, where $\dot{y}$ and $\dot{x}$ represent differentiation by t (see (42)). To obtain the tangents at the quarter points, we only need to evaluate at dy/dx at t = 0, 1/4, 2/4, 3/4, and 1. A quantity called mathematical curvature is inversely proportional to the radius of a certain tangent circle and its formula is given by $C = |\dot{x}\ddot{y} - \dot{y}\ddot{x}|/(\dot{x}^2 + \dot{y}^2)^{\frac{3}{2}}$ (details are given in (42)). C can be evaluated at the quarter points or evaluated at the values of t corresponding to each pixel and then averaged.

Appendix G: Details of Statistical Use of Features

The statistical procedure that was used to select the four features described in "Feature Selection by Statistical Procedure" has been described in previous work by the authors (see (30–36,38)). We began by working with a set of 199 features (a subset of the many potential features described in Methods). We then used an iterative process to select ten features that best compared Writer 1 to the 99 other writers. (This is done by comparing Writer 1 pairwise via stepwise linear discriminant analysis to each other writer and finding the most commonly used features selected in these comparisons.) The resulting ten features are called Writer 1's "Biometric Kernel." We then used stepwise linear discriminant analysis to select the four features used to compare Writer 1 with Writer 2.

The dotplots in Fig. A4*a* show that the tendencies described in Tables 2 and 3 are subject to a great deal of variability. However, statistical procedures are constructed to deal with just such variability. The canonical variable used by linear discriminant analysis to assign test graphemes to writers is a scaled, weighted combination of the four feature measurements. A dotplot of the canonical variable scores for the 86 test graphemes is given in Fig. A4*b*. There remains a great deal of variability, but the separation of the scores for the two writers is much more apparent than in the raw feature data.

We used the following rule to assign writership to a test grapheme. If the canonical variable score was closer to Writer 1's canonical mean in the training data (−1.60) and within ±2 of that mean, then assign the grapheme to Writer 1. If the canonical variable score was closer to Writer 2's canonical mean in the training data (+0.80) and within ±2 of that mean, then assign the grapheme to Writer 2. Otherwise, assign the grapheme to neither writer. In terms of Fig. A4*b*, these rules translate to the following rules: if the canonical score is to the left of the double line and inside the dashed red lines, assign to Writer 1. If the canonical score is to the right of the double line and inside the dashed green lines, assign to Writer 2. Otherwise, assign to neither writer.

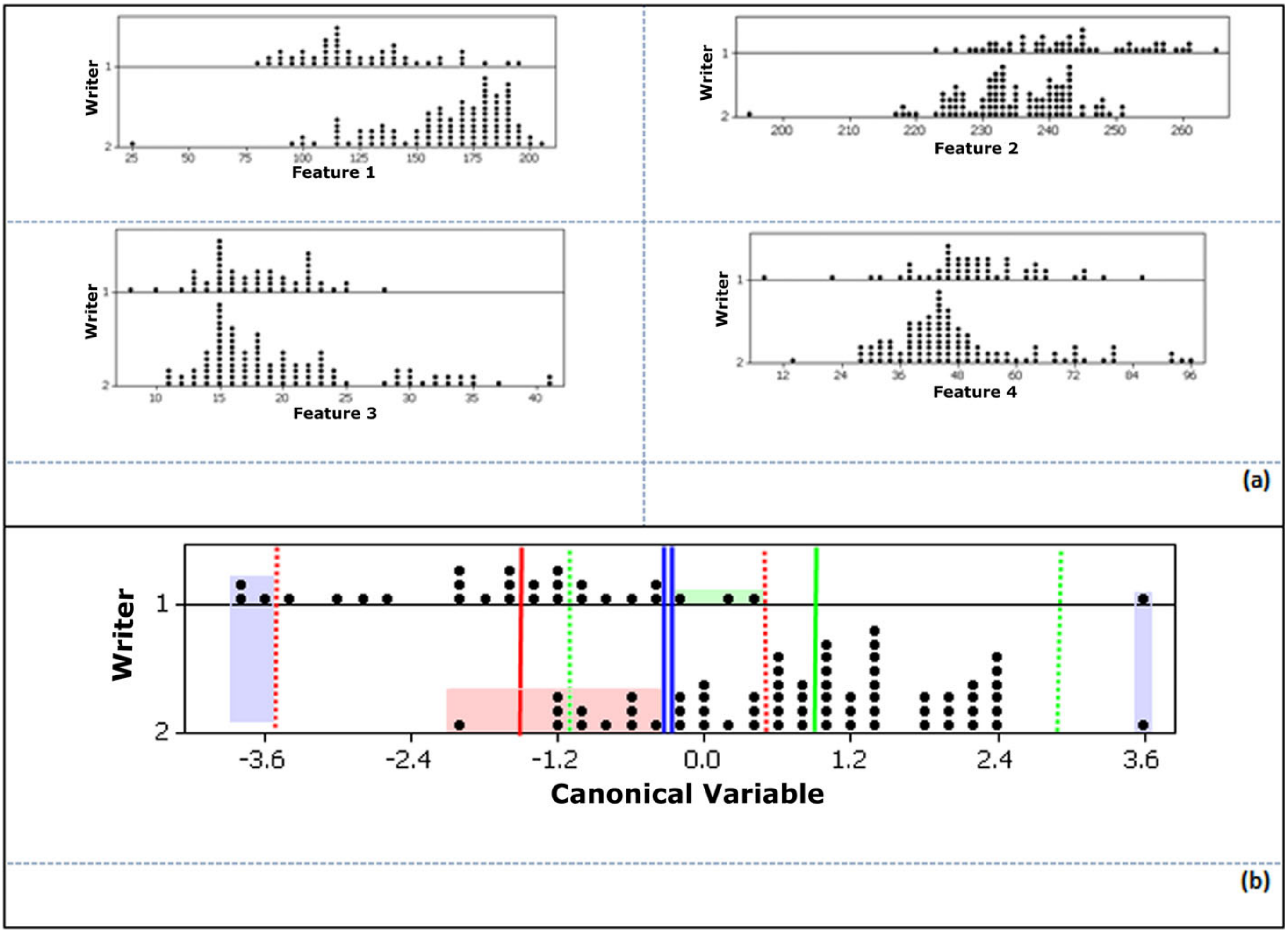


FIG. A4—*Dotplots of (a) the training data for each selected feature by writer and (b) the test data for the canonical variable by writer.*

These rules lead to erroneously assigning the three graphemes of Writer 1 whose dots are to the right of the double line (in the light green box) and the 11 graphemes of Writer 2 whose dots are to the left of the double line (in the light red box). The five graphemes in the light blue boxes on the extremes of the figure are assigned to neither writer.

We used a rule that allowed assignment to neither writer in the example even though we knew that all graphemes are written by one writer or the other. Removing the option would result in four of the five unassigned graphemes in this example being assigned to the correct writer. Hence, this option is not advantageous for the results in this example. However, in other applications of this basic methodology, such as comparing all pairs of writers from 100 or 300 writers, it is most likely that a test grapheme is written by neither of the pair of writers. Therefore, having the option of assigning to neither writer is very advantageous in those situations.