Journal of Imaging Science and Technology® 61(1): 010404-1–010404-9, 2017.

Writer Identification in Modern and Historical Documents via Binary Pixel Patterns, Kolmogorov–Smirnov Test and Fisher's Method

Arie Shaus and Eli Turkel
Department of Applied Mathematics, Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv 69978, Israel
E-mail: ashaus@post.tau.ac.il

Abstract. *The authors present a new method of writer identification, employing the full power of multiple experiments, which yields a statistically significant result. Each individual binarized and segmented character is represented as a histogram of 512 binary pixel patterns—3 × 3 black and white patches. In the process of comparing two given inscriptions under a "single author" assumption, the algorithm performs a Kolmogorov–Smirnov test for each letter and each patch. The resulting p-values are combined using Fisher's method, producing a single p-value. Experiments on both Modern and Ancient Hebrew data sets demonstrate the excellent performance and robustness of this approach.* © *2017 Society for Imaging Science and Technology.*
[DOI: 10.2352/J.ImagingSci.Technol.2017.61.1.010404]

INTRODUCTION

The current article deals with the challenging task of writer identification in historical documents. In what follows, we provide a short overview of the existing approaches to this task, and present the main contribution of this article.

Prior Work

The problem of computerized writer identification within historical documents exists in the literature for several decades.[1] Several features and their combination methods have been proposed for that purpose. The article Ref. 1 uses run-length histograms, combined via PCA (first two components). Article Ref. 2 continues the use of run-length distributions, supplementing them with allographic features (grapheme codebook generated using self-organizing map); the feature fusion is performed via simple or weighted averaging distances due to the individual features. Similar allographic features ("fraglets"), optionally supplemented with edge-directional feature ("hinge") are present in Ref. 3, with Hamming distance measures between the normalized features.

The article Ref. 4 presents another feature combination technique; extracting 8 types of features pertaining to various relations between foreground and background pixels of segmented characters, as well as their central moments. The features are selected via dimension reduction techniques such as sequential forward floating selection and linear discriminant analysis, classifying the reduced feature vectors via a linear Bayes classifier or K-nearest neighbors. Yet another set of classifiers based on grid microstructure, allograph level and topological features, combined via weighting procedure, is presented in Ref. 5. Article Ref. 6 provides a wealth of contour-based, oriented basic image, as well as SIFT, features classified by a voting procedure and SVM; Ref. 7 uses a similar setup, adding HOG features. An adaptation of SIFT features is also used in Ref. 8, with dimensionality reduced via PCA, resulting in a visual vocabulary. The features are clustered using a Gaussian Mixture Model and employing the Fisher kernel. A recent use of KDA in a setting involving both chain-code and edge-based directional features can be found in Ref. 9.

A thoroughly different approach is demonstrated in Ref. 10, operating on a segmented character level, and treating them as realizations of estimated "Ideal Prototypes." The identity or distinction among writers is made via several techniques, employing comparisons of the contours of the realizations to various ideals, and using heuristics and maximum likelihood estimations procedure combining information from different letters in order to find similar writers. A similar method is described in Ref. 11, with comparisons between character or ideal contours solved analytically.

A review of these articles, as well as surveys of the broader field of writer identification,[12,13] demonstrate the common denominators of most of these algorithms: a series of features (e.g., based on edge, allograph or topological information, or using "classical" computer vision features such as SIFT, HOG and Gabor filters) is extracted. Optionally, the dimensionality is reduced (e.g., via weighting, LDA, PCA or KDA methods), followed by writer classification of the resulting feature vectors (e.g., by employing KNN, SVM, MLE or the Fisher Kernel). Usually, the question is whether a given document, according to some metric, is written by the same author as the most closely matching document. Alternatively, several (e.g., 5 or 10) "closest" documents are fetched for the purpose of identifying at least one identical writer. The algorithm's performance is checked based on an existing ground truth.

Although some of these methods perform reasonably for their tasks and data sets, their typical output is an

Received July 18, 2016; accepted for publication Nov. 12, 2016; published online Dec. 8, 2016. Associate Editor: Zeev Zalevsky.
1062-3701/2017/61(1)/010404/9/$25.00

abstract distance between two given inscriptions, or else a table indicating the distances between several inscriptions. However, these distances do not yield any probabilistic information. Thus, it is difficult to interpret such an output outside a well ground-truthed framework. In particular the distances, by themselves, are insufficient for the different task of analyzing a corpus of many inscriptions, with an unknown number of authors.

The existing approaches can be contrasted with the direct predecessor of this article,[14] which proposes a statistical approach. The article used a sophisticated concatenation and subsequent combination of SIFT, Zernike, DCT, and K_d-tree, image projections, CMI[15–17] and L_1 features and distances. Subsequently, on an inscriptions' pair-wise basis, the writer identification analysis was performed *independently for each letter*. This resulted in different statistical *p-values*, estimating the probability of a single author producing the different letter instances of the two inscriptions. These independent p-values were later combined into a "meta" p-value via Fisher's Method (a brief explanation is provided below), typically resulting in more significant results. Contrary to the existing approaches, such an approach can be easily utilized in order to detect different authors within any given corpus, by detecting "meta" p-values below certain threshold.

Finally, we rely on previously developed document pre-processing techniques. In particular, we assume the existence of a suitable binarization, segmentation and (if needed) restoration of characters, whose quality is suitable for our needs. The inputs for the described method are individual black and white images of single characters, reflecting the original writing as reliably as possible (e.g., not thinned, no slant correction, etc.). Such automatic or semimanual techniques are described, for our data sets, in Refs. 17–19 and especially in Refs. 14, 20; for other approaches, consult the references mentioned in Ref. 4, 10, 11. For methods assessing the adherence of character's reconstruction to its image, as well as the general quality of the resulting binarization, see Refs. 15, 16, 21, 22. Additional details regarding the preparation of the different data sets are provided below.

The Main Contribution of this Article

In this research, we advance the ideas of Ref. 14 to the next level. The analysis is performed independently, not only on a level of a single letter, but also on the level of a single feature, unleashing the full statistical power of multiple experiments. The main changes from Ref. 14 are: an entirely different, and much larger set of features (using 512 different binary pixel patterns instead of a combination of 7 features); a two-step experimental process, working on both individual feature (by comparing the feature distributions via Kolmogorov–Smirnov test), as well as individual letter level in order to deduce the p-values, later to be combined via Fisher's method (potentially, thousands of experiments, equaling the number of letters multiplied by the number of features, are conducted!); and an improvement in the significance level of the results by lowering the p-value threshold. All these allow us to establish a robust platform for analyzing corpora of many inscriptions, with an unknown number of authors, while arriving at meaningful and statistically highly significant outcomes. A schematic comparison of the various handwriting analysis schemes is presented in Figure 1.

ALGORITHM'S DESCRIPTION

Preliminary Remarks

We use the common statistical convention of defining a "null hypothesis" H_0 and trying to *disprove* it. In our case, H_0 is "two given inscriptions were written by the same author." The probability for this event is the p-value, which will be estimated via the algorithm. If the p-value is *lower* than a pre-defined threshold, H_0 is rejected, and the competing hypothesis of "two different authors" is declared valid. On the other hand, an inability to reject the null hypothesis does not indicate its validity. In such a case we remain agnostic, not being able to say anything regarding the documents' authors.

The estimation of the p-value involves an activation of the Kolmogorov–Smirnov (KS) test, a classical nonparametric test, allowing for a comparison of two samples, not necessarily of the same size.[23] The main idea of KS is a comparison of *the empirical* distribution functions F_1 and F_2 (produced from the two samples), in order to calculate the observed statistic $D = \sup_x |F_1(x) - F_2(x)|$. The p-value of this statistic, under the hypothesis that the two samples stem from the same distribution, can be either calculated directly (via permutations) or approximated (our research utilizes the implementation[24]). For example, if the samples' sizes are large enough, and all the values within the first sample are smaller than the values of the second sample, the p-value should be low. A previous usage of Kolmogorov–Smirnov test in a signature verification setting can be seen in Ref. 25.

Another well-established technique used by the algorithm is Fisher's method for p-value combinations.[26] Given p-values p_i $(i = 1, \ldots, k)$ stemming from k independent experiments, the method allows one to estimate a combined p-value, reflecting the entire wealth of evidence at our disposal. The method utilizes the fact that $X^2_{2k} \sim -2\sum_{i=1}^{k} \ln(p_i)$, i.e., the sum produces a chi-squared distribution with $2k$ degrees of freedom. This allows for a calculation of a single combined ("meta") p-value. Intuitively, if several experiments produce low p-values (e.g., 0.1, 0.15 and 0.2), the probability for such an occurrence, by chance, is very small, and the combined p-value will also be low (possibly even lower than the original p-values; 0.071 for the last example). However, in the current article, the p-values of multiple experiments (stemming from different characters and features) are not necessarily independent, but are expected to be positively correlated. Thus, we are "overconfident" in the combined evidence against H_0. A common remedy to this problem is to demand more significant results, by substituting T with $T \cdot (k+1)/2k$ (k is the number of experiments), a common modification representing a mean of false discovery rates.[27] In our case, this demand can be satisfied simply by lowering

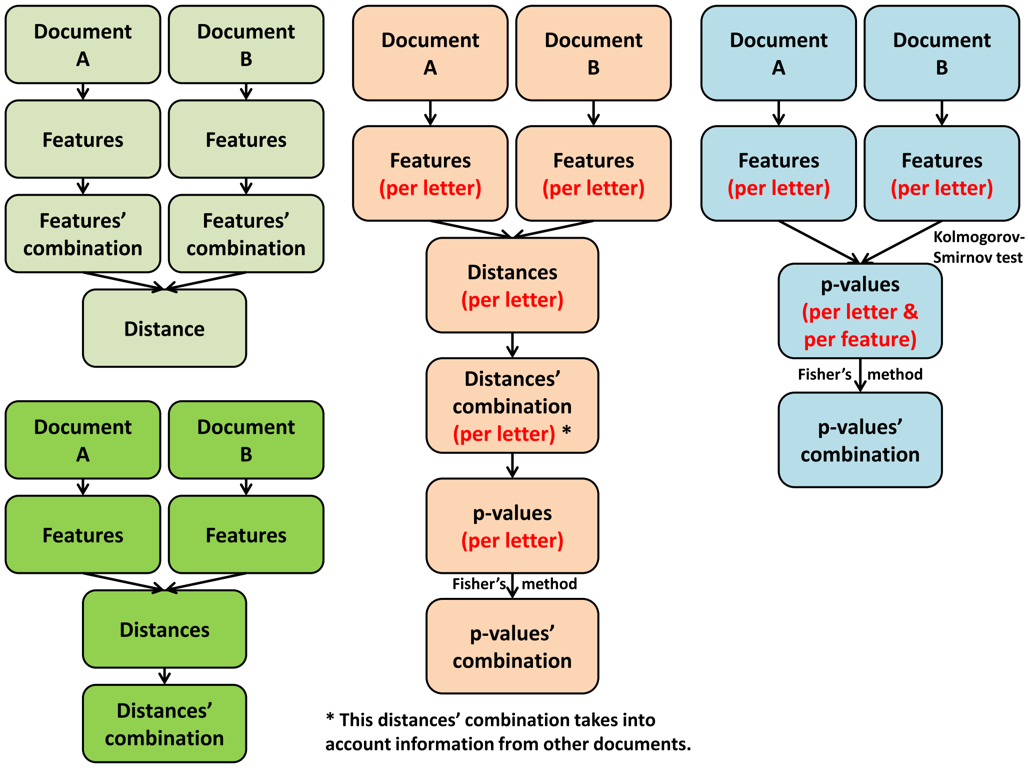


Figure 1. A comparison of handwriting analysis schemes. Left: common frameworks, producing an *abstract distance* between the documents as a final output. Center: the method of Ref. 14, performing the analysis on per-letter basis, yielding (number of letters) experimental *p-values* to be combined via Fisher's method. Right: the current technique, performing Kolmogorov–Smirnov tests for each feature and each letter, yielding (number of features) × (number of letters) experimental *p-values* to be combined via Fisher's method.

the threshold p-value T from 0.2 (as in Ref. 14) to 0.1 or even 0.05.

Prior Assumptions

We begin with two images of different inscriptions, denoted as I and J. The algorithm operates based on information derived at a character level. Herein, by a character we denote a particular instance of a given letter (e.g., there may be many characters, which are all instances of a letter *alep*). As remarked above, we assume that the inscriptions' characters are binarized and segmented into images $I^l_{i_l}$ ($i_l = 1, \ldots, M_l$, representing the instances of the letter l within I); and $J^l_{j_l}$ ($j_l = 1, \ldots, N_l$, representing the instances of the same letter l within J), belonging to appropriate letters ($l = 1, \ldots, L$). In the current research, the binarization and segmentation was performed automatically for Modern Hebrew, and in semimanual fashion for Ancient Hebrew documents.[14,20] The resulting characters' images were padded with a 1-pixel white border on each side.

Histogram Creation for each Character

Our features are the 3×3 binary pixel patterns, i.e., image patches of the individual characters (for additional information on pixel patterns, see the examples in Refs. 28, 29). There are $2^9 = 512$ optional patches of that size. All such possible patches are extracted from the images $I^l_{i_l}$ and $J^l_{j_l}$, in order to create normalized patches' histograms (counting frequencies of patches' occurrences), $H^l_{i_l}(p)$ and $G^l_{j_l}(p)$, respectively ($p = 1, \ldots, 512$).

A simple, yet illustrative, example of two such images and their respective histograms is seen in Table I. Remarkably, despite a similar overall shape of the character and only two pixels difference in the character images, 16 out of 19 meaningful histogram entities are different.

We note that the histograms only serve normalization purposes. In the following, the histograms themselves will **not** be compared. Instead, the comparison will take place on an individual feature (patch) level, across different characters.

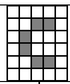
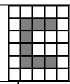
Same-Writer Statistics Derivation

The experiments are performed in the following fashion: for given inscriptions' images I and J with $I \neq J$:

1. An empty *PVALS* array is initialized.
2. For each letter $l = 1, \ldots, L$, with sufficient character instances present ($M_l > 0$, $N_l > 0$, $M_l + N_l \geq 4$; we verify there is enough statistics for a meaningful comparison, slightly lowering the requirements in Ref. 14):

   2.1 For each patch $p = 1, \ldots, 512$, with at least one nonzero term present in the histogram (i.e., $\exists i_l \cdot H^l_{i_l}(p) > 0$ OR $\exists j_l \cdot G^l_{j_l}(p) > 0$), perform a Kolmogorov–Smirnov (KS) nonparametric

Table I. Example of character histograms.

Patches	Characters' images, patches' counts and frequencies [a]			
				
	1	0.083	1	0.083
	1	0.083	0	0
	0	0	1	0.083
	1	0.083	0	0
	2	0.167	2	0.167
	0	0	1	0.083
	1	0.083	0	0
	1	0.083	0	0
	1	0.083	0	0
	0	0	1	0.083
	0	0	1	0.083
	0	0	1	0.083
	1	0.083	0	0
	1	0.083	0	0
	1	0.083	0	0
	0	0	1	0.083
	0	0	1	0.083
	1	0.083	1	0.083
	0	0	1	0.083

[a] Only the meaninful histogrm entries are provided. In both cases, the remaining entries contain zeros. In red – discrepancies between the two histograms.

test between the two samples $\{H^l_{il}(p)\}_{il=1}^{M_l}$ and $\{G^l_{jl}(p)\}_{jl=1}^{N_l}$: $pval^l_p = KS(\{H^l_{il}(p)\}_{il=1}^{M_l}, \{G^l_{jl}(p)\}_{jl=1}^{N_l})$.

2.2 Append the resulting $pval^l_p$ to the $PVALS$ array.

3. If the $PVALS$ array is empty (i.e., no experiments were performed due to the scarcity of data), OR if $I = J$, set:

$$SameWriterP(I, J) = SameWriterP(J, I) = 1.$$

4. Otherwise utilize the Fisher combination of all the $PVALS$ instances, and set:

$$SameWriterP(I, J) = SameWriterP(J, I) = FisherMethod(PVALS)$$

$SameWriterP(I, J)$ represents the deduced probability of having the same writer in both I and J (the H_0 hypothesis).

A toy-problem illustration of the whole scheme is shown in Figure 2. In this demonstration, two alep letters and four bet letters are segmented from the first document, while three alep and two bet letters are segmented from the second document. As a first step, patches histograms are extracted from the two documents. For illustration purposes, it is assumed that in both cases, only the first two patches yield a nonzero count. Since two types of relevant features and two different letters are involved, $2 \times 2 = 4$ Kolmogorov–Smirnov tests are performed, yielding four p-values. These are combined into a single p-value via Fisher's method.

Inscription I: patches' histograms

p	alep1	alep2	bet1	bet2	bet3	bet4
1	0.02	0.01	0.04	0.03	0.04	0.05
2	0.18	0.21	0.10	0.09	0.11	0.15
...						
512	0	0	0	0	0	0

Inscription II: patches' histograms

p	alep1	alep2	alep3	bet1	bet4
1	0.03	0.05	0.06	0.10	0.09
2	0.07	0.05	0.19	0.14	0.20
...					
512	0	0	0	0	0

Valid comparisons	Inscription I sample	Inscription II sample	KS test p-value
Letter=alep, p=1	0.02, 0.01	0.03, 0.05, 0.06	0.063
Letter=alep, p=2	0.18, 0.21	0.07, 0.05, 0.19	0.425
Letter=bet, p=1	0.04, 0.03, 0.04, 0.05	0.10, 0.09	0.047
Letter=bet, p=2	0.10, 0.09, 0.11, 0.15	0.14, 0.20	0.242

SameWriter(Inscription1, Inscription2) = FisherMethod(0.063, 0.425, 0.047, 0.242) = 0.0397

Figure 2. An example of the same-writer statistics derivation for two hypothetical inscriptions. Inscription I consists of two instances of the letter *alep*, and four instances of the letter *bet*, while Inscription II consists of three instances of the letter *alep*, and two instances of the letter *bet*. The only patches with enough statistics are patches numbers 1 and 2. Four comparisons of appropriate samples (for each letter and each patch) are performed via Kolmogorov–Smirnov test, resulting in four different *p*-values. These *p*-values are then combined via Fisher's method.

MODERN HEBREW EXPERIMENT

The Basic Settings

This experiment closely follows the setting described in Ref. 14. The data set (available at Ref. 30) contains a sampling of 18 individuals, $k = 1, \ldots, 18$. Each individual person filled in a Modern Hebrew alphabet table consisting of ten occurrences of each letter, out of the 22 letters in the alphabet (the number of letters and their names are the same as in the Ancient Hebrew in the next experiment; see Figure 3 for a table example). These tables were scanned and thresholded in order to create black and white images. Then their characters were segmented utilizing their known bounding box location (Fig. 3).

From this raw data, a series of "simulated" inscriptions were created. Due to the need to test both same-writer and different-writer scenarios, the data for each writer was split. Furthermore, in order to imitate a common situation in the Ancient Hebrew experiment, where the scarcity of data is prevalent (see below), each simulated inscription used only three letters (i.e., 15 characters; 5 characters for each letter), presenting a welcomed challenge for the new algorithm.

In total, 252 inscriptions were "simulated" in the following manner:

- All the letters of the alphabet except for *yod* (due to its small size), were split randomly into seven groups (three letters in each group), $g = 1, \ldots, 7$: *gimel*, *het*,

Figure 3. An example of a Modern Hebrew alphabet table, produced by a single writer; taken from Ref. 14.

resh; *bet*, *samek*, *shin*; *dalet*, *zayin*, *ayin*; *tet*, *lamed*, *mem*; *nun*, *sade*, *taw*; *he*, *pe*, *qop*; *alep*, *waw*, *kap*.

- For each writer k, and each letter belonging to group g, five characters were assigned into simulated inscription $S_{i,g,1}$, with the rest assigned to $S_{i,g,2}$.

In this fashion, for constant k and g, we can test if our algorithm arrives at wrong rejection for $S_{i,g,1}$ and $S_{i,g,2}$ (FP = "False Positive" error; 18 writers and seven groups producing 126 tests in total). In addition, for constant g, writer q s.t. $q \neq k$, and $b, c \in \{1, 2\}$, we can test if our algorithm fails to correctly reject the "same-writer" hypothesis for $S_{k,g,b}$ and $S_{q,g,c}$ (FN = "False Negative" error; 4284 tests in total).

Parameter Tuning and Robustness Verification

The algorithm described in the Algorithm's Description section provides an estimated probability for the H_0 hypothesis ("the two given inscriptions were written by the same writer"). However, two important parameters remain undecided. The first important parameter is the typical area of each character in pixels, leading to the optimal (or at least acceptable) performance. The second crucial parameter is the p-value threshold T, set for the purpose of rejecting the H_0.

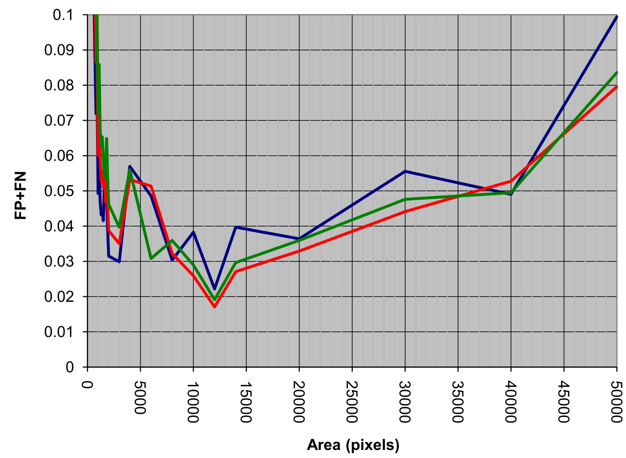


Figure 4. Testing the combined probability of FP + FN errors as a function of character area (in pixels) as well as different p value thresholds: 0.2 in blue, 0.1 in red and 0.05 in green. Taking into account the performance in Ref. 14 (FP + FN ≈ 0.043), all the tested thresholds and all the areas between 1000 and 40,000 pixels would yield reasonable and comparable performance. Slightly better results are achieved in the range of 8000–20,000 pixels, with 0.1 threshold.

As is common in statistics, lowering T can result in fewer FP errors, unfortunately increasing the likelihood for FN errors. Conversely, raising T might result in the opposite outcome. In order to minimize the FP and FN errors, a set of simulations was performed. The simulations measured the behavior of the sum FP + FN, with respect to the area of the character's image (ranging from 200 to 50,000 pixels), and to the chosen value of T (attempting the value 0.2 chosen by Ref. 14, as well as the values 0.1 and 0.05, as explained above).

The results of these simulations is shown in Figure 4. Taking into account the performance in Ref. 14 ($FP + FN \approx 0.043$), all the tested thresholds and all the areas between 1000 and 40,000 pixels yield a reasonable and comparable performance ($FP + FN < 0.05$), with slightly better results in the range of 8000–20,000 pixels, with $T = 0.1$. This wide range for acceptable areas indicates an excellent robustness of the algorithm (though the algorithm would probably result in better outcomes if the character images were of similar resolution). Since the mean area of the original character images was 17,367 pixels, well within the reasonable limits of our analysis, we have chosen the typical area of each character to be 17,000 pixels.

Experimental Results

The results of our configuration (for different values of T) are provided in Table II. The results are certainly better than the results of Ref. 14 on the same data set, with a much simpler configuration. As expected, FP error rate tends to zero as the threshold is lowered, while the FN increases slightly. The threshold value of $T = 0.1$ produced better results, with a combined FP + FN error of less than 2%.

Figure 5. Examples of ostraca (ink inscriptions on clay) from the Iron Age fortress of Arad, located in arid southern Judah. These documents are dated to the latest phase of the First Temple Period in Judah, ca. 600 BCE. The texts represent correspondence of local military personnel.

Table III. Letter statistics for Arad texts (adapted from Ref. 14).

	Letters						
Text	Alep	He	Waw	Yod	Lamed	Shin	Taw
1	4	5	3	7	3	3	8
2	6	3	3	5	3	1	7
3	2	4	5	4	4	3	3
5	5	3	1	3	4	2	4
7	1	2	1	4	6	8	5
8	2	1	2	1	4	4	2
16	6	3	9	5	10	3	2
17a	2	4	2	2	2	1	2
17b		1		2	1	1	2
18	2	4	4	5	6	6	3
21	5	4	6	6	12	5	2
24	9	10	5	8	4	4	7
31	3	7	6	4	1	1	
38	1	1	2	2	2	1	
39a	3	3	3	5	2	1	1
39b	3	1	1	4	1		
40	4	5	3	4		3	2
111	4	3	3	3	1	3	2

Table II. Results of modern Hebrew experiment.

Configuration	Results		
	FP	*FN*	*FP* + *FN*
Results of [14], $T = 0.2$	2.38%	1.94%	**4.32%**
Our setting, $T = 0.2$	0.79%	1.70%	**2.50%**
Our setting, $T = 0.1$	0.00	1.96%	**1.96%**
Our setting, $T = 0.05$	0.00	2.12%	**2.12%**

Confident in our newly obtained configuration (target area of ~17,000 pixels and $T = 0.1$), we proceed to the Ancient Hebrew experiment.

ANCIENT HEBREW EXPERIMENT

The Basic Settings

The Ancient Hebrew data set[31] stems from the Judahite desert fortress of Arad, dated to the end of the First Temple period (Iron Age), ca. 600 BCE—the eve of Nebuchadnezzar's destruction of Jerusalem. The fortress was unearthed half a century ago, with 100 ostraca (ink on clay) inscriptions found during the excavations.[32] The inscriptions represent the correspondence of the local military personnel. See Figure 5 for examples of Arad ostraca.

We concentrate on 16 (relatively lengthy) Arad ostraca, two of them two-sided, which brings the total number of texts for analysis to 18. The scarcity of data in this situation is common for these ancient texts. Ostraca images were utilized in order to segment and restore the characters stroke by stroke via a variational procedure (detailed in Ref. 20). Some examples in Ref. 14 required minimal human involvement. No further manipulation of the resulting characters' images (e.g., skeletonization, slant correction, etc.) was performed. Table III provides statistics of the most prominent letters, after reconstructing the most legible characters. It can be seen, that even by the modest quantitative standards set in the Algorithm's Description section, for some of the texts the comparisons are barely feasible.

Contrary to the situation in the modern context, now we do not possess any ground truth, indicating the identity of the writers across different inscriptions. Moreover, the experiment's requirements do not impose a strict partition of the texts by their authors. The task is limited to detecting the minimal number of hands within this group of texts. A previous study[14] demonstrated at least four different (pair-wise distinct) writers within the corpus (in fact six different "quadruplets" of texts), while bringing this number to six via textual considerations (not considered in the current article).

As already mentioned, following plausible results of the Modern Hebrew experiment, the characters were resized to 17,000 pixels, and the threshold was set to $T = 0.1$.

Experimental Results

The results of the experiment are presented in Table IV. The results indicate that there are at least five different (pair-wise distinct) writers within this group of texts. In fact, a closer look at Table IV reveals that three such

Table IV. Results of ancient Hebrew experiment, indicating separation of writers between texts (**in darker background**).

Text [a]	1	2	3	5	7	8	16	17a	17b	18	21	24	31	38	39a	39b	40	111
1	1	0.000104451	0.794037056	0.997286098	0.835555244	0.999986905	0.145123589	0.999994862	0.345685406	0.012774181	0.012535286	3.65925x10^{-12}	0.016073174	0.078182536	0.038794364	0.308694925	2.97103x10^{-06}	0.000119896
2	0.000104451	1	0.999999997	0.230810507	0.23103925	0.999837107	0.782882802	0.999377805	0.121018637	1.96989x10^{-08}	0.677687539	0.907018003	0.460683411	1.45921x10^{-07}	0.999999999	0.004777329	0.011757086	0.918537018
3	0.794037056	0.999999997	1	0.99999999	1	0.999999995	1	1	0.999917956	4.13858x10^{-07}	1	1	0.51882444	0.960818768	1	0.999999219	0.983116915	1
5	0.997286098	0.230810507	0.99999999	1	0.999999996	0.999999979	0.829655242	0.955254622	0.093782225	2.78436x10^{-17}	0.996839946	4.38907x10^{-10}	0.026192004	0.026797057	0.942902756	0.867249786	0.040887838	2.1339x10^{-05}
7	0.835555244	0.23103925	1	0.999999996	1	0.999049432	0.974036343	0.938473936	0.543854905	6.1151x10^{-25}	0.953218488	0.01448862	0.15146191	0.651369422	0.973278889	0.381286518	0.999602961	0.072003619
8	0.999986905	0.999837107	0.999999995	0.999999979	0.999049432	1	0.996867196	0.999999995	0.26481398	0.000160748	0.99999487	0.995623889	0.490501387	0.028988689	0.999933169	0.610834495	0.999979451	0.818431322
16	0.145123589	0.782882802	1	0.829655242	0.974036343	0.996867196	1	0.914322562	0.989523149	6.53819x10^{-37}	0.99999911	0.050144018	1.82513x10^{-08}	0.017541713	0.999999999	0.752972486	0.986154478	0.999998263
17a	0.999994862	0.999377805	1	0.955254622	0.938473936	0.999999995	0.914322562	1	0.98838179	0.999824883	0.989761454	0.916602116	0.999934002	0.971750814	0.999957681	0.999885898	0.750470353	0.999999995
17b	0.345685406	0.121018637	0.999917956	0.093782225	0.543854905	0.26481398	0.989523149	0.98838179	1	0.989432496	0.987354188	0.397038359	0.997936364	0.994674672	0.690349079	0.989449919	0.469371025	0.976381275
18	0.012774181	1.96989x10^{-08}	4.13858x10^{-07}	2.78436x10^{-17}	6.1151x10^{-25}	0.000160748	6.53819x10^{-37}	0.999824883	0.989432496	1	1.74207x10^{-37}	2.85314x10^{-23}	0.872302919	6.42961x10^{-05}	0.000351619	3.9727x10^{-05}	4.06656x10^{-14}	0.016251091
21	0.012535286	0.677687539	1	0.996839946	0.953218488	0.99999487	0.99999911	0.989761454	0.987354188	1.74207x10^{-37}	1	0.004905952	0.002746089	0.122737346	0.999999846	0.982566541	0.957807456	0.999809072
24	3.65925x10^{-12}	0.907018003	1	4.38907x10^{-10}	0.01448862	0.995623889	0.050144018	0.916602116	0.397038359	2.85314x10^{-23}	0.004905952	1	2.93422x10^{-09}	0.009250254	0.999999991	0.405106241	6.4106x10^{-12}	0.684834031
31	0.016073174	0.460683411	0.51882444	0.026192004	0.15146191	0.490501387	1.82513x10^{-08}	0.999934002	0.997936364	0.872302919	0.002746089	2.93422x10^{-09}	1	0.856161514	0.99999999	0.999884872	1.70082x10^{-09}	0.772576882
38	0.078182536	1.45921x10^{-07}	0.960818768	0.026797057	0.651369422	0.028988689	0.017541713	0.971750814	0.994674672	6.42961x10^{-05}	0.122737346	0.009250254	0.856161514	1	0.600342228	0.112003203	0.067411667	0.09199445
39a	0.038794364	0.999999999	1	0.942902756	0.973278889	0.999933169	0.999999999	0.999957681	0.690349079	0.000351619	0.999999846	0.999999991	0.99999999	0.600342228	1	0.999985794	0.046450883	1
39b	0.308694925	0.004777329	0.999999219	0.867249786	0.381286518	0.610834495	0.752972486	0.999885898	0.989449919	3.9727x10^{-05}	0.982566541	0.405106241	0.999884872	0.112003203	0.999985794	1	0.839063568	0.90756289
40	2.97103x10^{-06}	0.011757086	0.983116915	0.040887838	0.999602961	0.999979451	0.986154478	0.750470353	0.469371025	4.06656x10^{-14}	0.957807456	6.4106x10^{-12}	1.70082x10^{-09}	0.067411667	0.046450883	0.839063568	1	0.113447052
111	0.000119896	0.918537018	1	2.1339x10^{-05}	0.072003619	0.818431322	0.999998263	0.999999995	0.976381275	0.016251091	0.999809072	0.684834031	0.772576882	0.09199445	1	0.90756289	0.113447052	1

groups of five pair-wise distinct inscriptions exist, including the following texts: {1, 2, 18, 38, 40}, {1, 18, 24, 38, 40} and {5, 18, 24, 38, 40}. This can be contrasted with Ref. 14, where no such large pair-wise distinct groups were found, despite a higher threshold ($T = 0.2$). A simple simulation shows that given a random undirected graph of size 16 with an edge probability of 0.1, the probability for having at least three different cliques with at least five members is about 8×10^{-7}; hence the high statistical significance of the results.

CONCLUSIONS AND FUTURE DIRECTIONS

The current research demonstrates a relatively simple and easily implementable algorithm for the purpose of writer identification. The algorithm demonstrates highly significant results in a setting including a minimal amount of letters. It is fast and robust with respect to both the typical area of the character images, and the evaluated p-value thresholds.

Our approach goes against the common wisdom of combining the different features or metrics before the documents' comparisons takes place. Instead, we propose to perform as many individual experiments as possible on both the letter and feature levels, combining their results only in the end. Although individual building blocks of our algorithm have occasionally been utilized in the literature, our specific amalgamation of binary pixel patterns, Kolmogorov–Smirnov test, and Fisher's method has not been described previously in the literature with regard to writer identification.

Several future development directions and possible enhancements deserve mentioning:

- The size of the patch (3×3) can be altered, although even a 4×4 patch results in a histogram of length $2^{16} = 65{,}536$, which may be too sparse.
- More research regarding target character areas is required. In particular, a more fine-grained approach can be suggested, e.g., areas optimized individually for each letter; areas dependent on characteristic stroke width, etc.
- The algorithm can probably benefit from more aggressive filtering of the incoming input at step 2 (demanding more than four characters in comparison; more than one nonzero histogram item).
- Naturally, our algorithm can benefit from additional tests on other real-world scenarios and data sets, including other languages, writing systems and time periods. This should also include more noisy environments.

ACKNOWLEDGMENTS

The research received initial funding from the Israel Science Foundation—F.I.R.S.T. (Bikura) Individual Grant no. 644/08, as well as the Israel Science Foundation Grant no. 1457/13. It was also funded by the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement no. 229418, and by an Early Israel grant (New Horizons project), Tel Aviv University. This study was also supported by a generous donation from Mr. Jacques Chahine, made through the French Friends of Tel Aviv University. Arie Shaus is grateful to the Azrieli Foundation for the award of an Azrieli Fellowship. The research was made possible by the dedicated work of Ms. Ma'ayan Mor. The kind assistance of Dr. Shirly Ben-Dor Evian, Ms. Sivan Einhorn, Ms. Noa Evron, Ms. Shira Faigenbaum-Golovin, Ms. Myrna Pollak and Mr. Barak Sober is greatly appreciated. The authors wish to thank the reviewers for their valuable suggestions. The ostracon images are courtesy of the Institute of Archaeology, Tel Aviv University and of the Israel Antiquities Authority.

REFERENCES

[1] I. Dinstein and Y. Shapira, "Ancient Hebraic handwriting identification with run-length histograms," IEEE Trans. Syst. Man Cybern. **12**, 405 (1982).

[2] M. Bulacu and L. Schomaker, "Automatic handwriting identification on medieval documents," *Proc. 14th Int'l. Conf. on Image Analysis and Processing—ICIAP 2007* (IEEE Computer Society, 2007), pp. 279–284.

[3] L. Schomaker, K. Franke, and M. Bulacu, "Using codebooks of fragmented connected-component contours in forensic and historic writer identification," Pattern Recognit. Lett. **28**, 719 (2007).

[4] I. Bar-Yosef, I. Beckman, K. Kedem, and I. Dinstein, "Binarization, character extraction, and writer identification of historical Hebrew calligraphy documents," Int. J. Document Analysis Recognition **9**, 89 (2007).

[5] F. Aiolli and M. Giollo, "A study on the writer identification task for paleographic document analysis," *Proc. 11th IASTED Int'l. Conf. on Artificial Intelligence and Applications—AIA 2011* (ACTA Press, 2011).

[6] D. Fecker, A. Asi, W. Pantke, V. Märgner, J. El-Sana, and T. Fingscheidt, "Document writer analysis with rejection for historic Arabic manuscripts," *Proc. 14th Int'l. Conf. on Frontiers in Handwriting Recognition—ICFHR 2014* (IEEE, Piscataway, NJ, 2014), pp. 743–748.

[7] D. Fecker, A. Asi, V. Märgner, J. El-Sana, and T. Fingscheidt, "Writer identification for historical Arabic documents," *Proc. of the 22nd Int'l. Conf. on Pattern Recognition—ICPR 2014* (IEEE, Piscataway, NJ, 2014), pp. 3050–3055.

[8] S. Fiel, F. Hollaus, M. Gau, and R. Sablatnig, "Writer identification on historical Glagolitic documents," Proc. SPIE **9021** (2014).

[9] S. Al-Maadeed, A. Hassaine, A. Bouridane, and M. A. Tahir, "Novel geometric features for off-line writer identification," Pattern Anal. Appl. **19**, 699 (2016).

[10] M. Panagopoulos, C. Papaodysseus, P. Rousopoulos, D. Dafi, and S. Tracy, "Automatic writer identification of ancient Greek inscriptions," IEEE Trans. Pattern Anal. Mach. Intell. **31**, 1404 (2009).

[11] C. Papaodysseus, P. Rousopoulos, F. Giannopoulos, S. Zannos, D. Arabadjis, M. Panagopoulos, E. Kalfa, C. Blackwell, and S. Tracy, "Identifying the writer of ancient inscriptions and Byzantine codices. A novel approach," Comput. Vis. Image Underst. **121**, 57 (2014).

[12] L. R. B. Schomaker, "Writer identification and verification," *Advances in Biometrics: Sensors, Systems and Algorithms*, edited by N. Ratha and V. Govindaraju (Springer-Verlag, 2007), pp. 247–264.

[13] M. Sreeraj and S. M. Idicula, "A survey on writer identification schemes," Int. J. Comput. Appl. **26**, 23 (2011).

[14] S. Faigenbaum-Golovin, A. Shaus, B. Sober, D. Levin, N. Naaman, B. Sass, E. Turkel, E. Piasetzky, and I. Finkelstein, "Algorithmic handwriting analysis of Judah's military correspondence sheds light on composition of biblical texts," Proc. Natl Acad. Sci. **113**, 4664 (2016).

[15] A. Shaus, E. Turkel, and E. Piasetzky, "Quality evaluation of facsimiles of Hebrew First Temple period inscriptions," *Proc. 10th IAPR Int'l. Workshop on Document Analysis Systems—DAS 2012* (IEEE, Piscataway, NJ, 2012), pp. 170–174.

[16] A. Shaus, I. Finkelstein, and E. Piasetzky, "Avoiding the eye of the beholder: Automated ostraca facsimile evaluation," Maarav **17**, 7 (2010).

[17] A. Shaus, E. Turkel, and E. Piasetzky, "Binarization of first temple period inscriptions—performance of existing algorithms and a new registration based scheme," *Proc. 13th Int'l. Conf. on Frontiers in Handwriting Recognition—ICFHR 2012* (IEEE, Piscataway, NJ, 2012), pp. 641–646.

[18] A. Shaus, B. Sober, E. Turkel, and E. Piasetzky, "Improving binarization via sparse methods," *Proc. 16th Int'l. Graphonomics Society Conf.—IGS 2013* (International Graphonomics Society, 2013), pp. 163–166.

[19] A. Shaus, B. Sober, S. Faigenbaum-Golovin, A. Mendel-Geberovich, E. Piasetzky, and E. Turkel, "Facsimile creation: Review of algorithmic approaches," *Alphabets, Texts and Artefacts in the Ancient Near East, Studies Presented to Benjamin Sass*, edited by I. Finkelstein, C. Robin, and T. Römer (Van Dieren É diteur, Paris) , forthcoming.

[20] B. Sober and D. Levin, "Computer aided restoration of handwritten character strokes," arXiv preprint arXiv:1602.07038 .

[21] S. Faigenbaum, A. Shaus, B. Sober, E. Turkel, and E. Piasetzky, "Evaluating glyph binarizations based on their properties," *Proc. 2013 ACM Symposium on Document Engineering—DocEng 2013* (ACM, New York, 2013), pp. 127–130.

22 A. Shaus, B. Sober, E. Turkel, and E. Piasetzky, "Beyond the ground truth: Alternative quality measures of document binarizations," *Proc. 15th Int'l. Conf. on Frontiers in Handwriting Recognition—ICFHR 2016* (IEEE, Piscataway, NJ, 2016), pp. 495–500.

23 G. W. Corder and D. I. Foreman, *Nonparametric Statistics: A Step-by-Step Approach* (Wiley, 2014).

24 E. Jones, E. Oliphant, and P. Peterson, *et al. SciPy: Open Source Scientific Tools for Python* (2001-), https://www.scipy.org/citing.html.

25 E. Griechisch, M. I. Malik, and M. Liwicki, "Online signature verification based on Kolmogorov–Smirnov distribution distance," *Proc. 14th Int'l. Conf. on Frontiers in Handwriting Recognition—ICFHR 2014* (IEEE, Piscataway, NJ, 2014), pp. 738–742.

26 R. A. Fisher, *Statistical Methods for Research Workers* (Oliver and Boyd, Edinburgh, 1925).

27 Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: a practical and powerful approach to multiple testing," J. R. Stat. Soc. Ser. B **57**, 289.

28 T. Akiyama, N. Miyamoto, M. Oguro, and K. Ogura, "Faxed document image restoration method based on local pixel patterns," Proc. SPIE **3305**, 253–262 (1998).

29 V. Ratnakar, "RAPP, lossless image compression with runs of adaptive pixel patterns," *Conf. Record of the Thirty-Second Asilomar Conf. on Signals, Systems & Computers* (IEEE, Piscataway, NJ, 1998), Vol. 2, pp. 1251–1255.

30 http://www-nuclear.tau.ac.il/∼eip/ostraca/DataSets/Modern_Hebrew.zip .

31 http://www-nuclear.tau.ac.il/∼eip/ostraca/DataSets/Arad_Ancient_Hebrew.zip.

32 Y. Aharoni, *Arad Inscriptions* (Israel Exploration Society, Jerusalem, 1981).