IJDAR
DOI 10.1007/s10032-007-0041-5

ORIGINAL PAPER

# Binarization, character extraction, and writer identification of historical Hebrew calligraphy documents

**Itay Bar-Yosef · Isaac Beckman · Klara Kedem · Itshak Dinstein**

Received: 14 February 2005 / Revised: 2 January 2007 / Accepted: 30 January 2007


**Abstract** We present our work on the paleographic analysis and recognition system intended for processing of historical Hebrew calligraphy documents. The main goal is to analyze documents of different writing styles in order to identify the locations, dates, and writers of test documents. Using interactive software tools, a data base of extracted characters has been established. It now contains about 20,000 characters of 34 different writers, and will be distinctly expanded in the near future. Preliminary results of automatic extraction of pre-specified letters using the erosion operator are presented. We further propose and test topological features for handwriting style classification based on a selected subset of the Hebrew alphabet. A writer identification experiment using 34 writers yielded 100% correct classification.

**Keywords** Binarization · Character extraction · Writer identification · Document analysis · Historical documents

I. Bar-Yosef (✉) · I. Beckman · K. Kedem
Computer Science Department,
Ben Gurion University, Beer-Sheva 84105, Israel
e-mail: itaybar@cs.bgu.ac.il

I. Beckman
e-mail: beckman@cs.bgu.ac.il

K. Kedem
e-mail: klara@cs.bgu.ac.il

I. Dinstein
Electrical and Computer Engineering Department,
Ben Gurion University, Beer-Sheva 84105, Israel
e-mail: dinstein@ee.bgu.ac.il

## 1 Introduction

Paleography is the study of ancient handwritten manuscripts. Among other things, it deals with dating and localizing ancient and medieval scripts, and with studying the development of the letter shapes. The work reported in this paper is part of a project to develop algorithms and tools for computerizing paleographic analysis of old Hebrew calligraphy scripts. (see Fig. 1 for example image.)

The Hebrew ancient handwriting used in our research, is influenced both by time and place—different regions over different periods of time use different styles of the same alphabet. In this sense, paleographic research is related to script and writer identification (see Sect. 5). One of the important aspects of paleographic research is visual analysis of character shapes. The main problem in quantitative shape analysis of handwritten characters is the development of features that can deal with the variations of the character's forms. The variations between styles, and the changing morphology of the letters are sometimes represented by tiny detail differences.

The first published work regarding the use of image processing for paleographic research (as far as we know) was published in 1971 [1]. Colette Sirat [2] is the author of another early publication reporting the use of computer image processing methods for paleographic research. Features based on run-length histograms were used in [3] for style identification of ancient Hebrew handwriting. An expert system using document analysis strategies for authentication of Hebrew manuscripts is reported in [4].

We present an overall system for paleographic analysis and recognition of old Hebrew calligraphic

**Fig. 1** A fourteenth century example image

documents. First, we present our novel approach for document binarization. Since our writer identification approach is based on style analysis of selected letters, we propose a segmentation-free approach for extracting the letters automatically. Finally, we describe the writer identification approach based on style analysis of the selected letters (note that the origin and date of the documents can be derived from the writer's identity). Our paper is organized as follows: Sect. 2 describes our binarization method, Sect. 3 presents the database and ground truth generation. In Sect. 4 we lay out our novel character extraction method based on the erosion operator. Section 5 describes our method for writer identification, and Sect. 7 summarizes our work presented here.

## 2 Binarization method

In general, historical document images are of poor quality because the documents have degraded over time due to storage conditions, and to the quality of the written parchment. As a result, the foreground and background are difficult to separate. The problem is particularly difficult because many documents have varying contrast, smudges, variable background intensity and presence of seeping ink from the other side of the document. We use the multi-stage algorithm presented in [5]. In the first stage, an initial binary image $B$ is obtained by applying a global threshold. This suffices for noise free characters. Then, an evaluation procedure determines which of the connected components in $B$ are well separated from the background, and which components need to be refined as we describe below.

*Note 1* Throughout this paper the term *character* refers to the graphic representation of a letter. The term *letter* refers to the alphabet, i.e., the letter Aleph, the letter Bet, etc.

### 2.1 Quality evaluation

For each connected component in $B$, we create a *seed image*, which contains the low intensity pixels of the character. For each $CC_i$, i = {1,...,N}, where $N$ is the number of connected components, let $m_i$ be the mean gray scale value (in the original image) of pixels belonging to $CC_i$. Let $\mathrm{SD}_i$ be the *seed image* of $CC_i$ using $m_i$ as a local threshold:

$$\mathrm{SD}_i = \begin{cases} 1, & \text{if } CC_i <= m_i \\ 0, & \text{otherwise} \end{cases}$$

In good characters the transition between the seed and the background forms a narrow band of pixels. In noisy characters this transitions are wider and irregular (see Fig. 2 for example). Denote by $CC = \{CC_1, CC_2, \ldots, CC_N\}$ the set of connected components in $B$. We compute $\mathrm{SD}_i$, the seed image of $CC_i$, i = $\{1, \ldots, N\}$. Denote by $T_i$ the pixels belonging to $CC_i$ but not to $\mathrm{SD}_i$, $T_i = CC_i - \mathrm{SD}_i$. The pixels of $T_i$ belong to the transition between the seed and the background. For each pixel in $T_i$, we compute its distance to the closest pixel in $\mathrm{SD}_i$ as

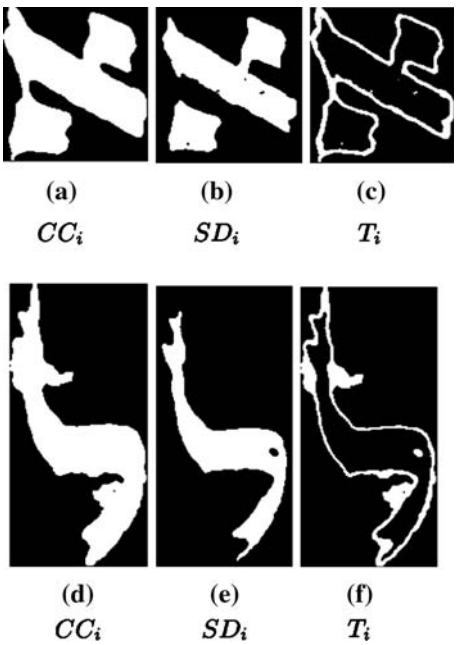


**Fig. 2** **a–c** A well segmented character. The mean and variance of the corresponding set $DT_i$ are $\mu = 4.01$ and $\sigma = 2.121$. **d–f** A character with some patches of noise. The mean and variance of the corresponding set $DT_i$ are $\mu = 6.02$ and $\sigma = 30.31$

follows. Within the bounding box of $CC_i$, we treat $\mathrm{SD}_i$ as the foreground and the rest as background ($CC_i - \mathrm{SD}_i$). Then we apply a distance transform algorithm [6], which calculates for each background pixel (pixels in the set $CC_i - \mathrm{SD}_i$) the distance to the closest foreground pixel (pixels in the set $\mathrm{SD}_i$). Denote by $DT_i$ the set of distances calculated for $T_i$. We use the *variance* of $DT_i$ as a measure to discriminate between good characters and noisy characters. Figure 2 shows two characters ($CC_i$), their corresponding seed image ($\mathrm{SD}_i$) and the difference between the character and its seed image ($T_i$).

Notice Fig. 2a–c, in which a well segmented character is presented. As can be seen, the set $T_i$ is composed of a narrow band of pixels, uniformly scattered around $\mathrm{SD}_i$. The mean value of the set $DT_i$ is $\mu = 4.01$ and the variance is $\sigma = 2.121$. Fig. 2d–f depict a noisy character. The set $T_i$ for this character is in part narrow as that of a well segmented character, and in part wider and irregular where patches of noise are present. The mean value of the set $DT_i$ for this character is $\mu = 6.02$ and the corresponding variance is $\sigma = 30.31$. As illustrated in these figures, the variance $\sigma_i$ has a much higher value for noisy characters than for well segmented ones. Let $\sigma_i$ be the variance of the set $DT_i$, and let $\sigma_{\mathrm{mean}}$ be the average of $\sigma_i$, i={1,…,N}.

If $\sigma_{\mathrm{mean}} \geq \sigma_{\mathrm{ths}}$, where $\sigma_{\mathrm{ths}}$ is an empirically based threshold, we assume that the document is entirely degraded, and all of its components, $CC_i$, are classified as noisy. Otherwise, the document is composed of both good and noisy characters. In this case, every component $CC_i$, with $\sigma_i \leq \sigma_{\mathrm{mean}}$ is classified as a well segmented character. The rest of the components are processed using the local method described in the next section. Notice that the penalty on classifying a good character as a noisy one, is the extra computation time involving the calculation of the local method.

### 2.2 The local method

Even when a character is noisy or faded, the seed body of the character is easy to detect. This fact led us to adopt a region growing scheme in which we first detect the seed image of the characters, and then apply a growing process that expands the character to its final form (see Fig. 3).

For each $CC_i$ classified as a noisy character, we define the seed image as in Sect. 2.1. The Growing process is an iterative process in which during each iteration a set of candidate pixels is observed. Each pixel from this set is tested whether it can join the foreground or not. The process is terminated when no pixel is added to the foreground. The Algorithm goes as follows. Starting from the seed images $\mathrm{SD}_i$, repeat until the foreground set does not change:


**Fig. 3** Different stages of the binarization algorithm: **a** Grayscale image. **b** After global thresholding. Pixels above the threshold are shown in *white*, the rest are shown in their original *gray value*. **c** Corresponding seed image. **d** Final binary image

1. *Find all candidate pixels* The candidate pixels are background pixels which are 8-connected to the growing foreground.
2. *Assign candidate pixels* For each candidate pixel $p$, consider its $7 \times 7$ neighborhood: let $M_{\mathrm{f}}$ be the average gray scale value of the foreground pixels in this window, $M_{\mathrm{b}}$ be the average gray scale value of the background pixels in this window. Assign $p$ to the class whose average is closest to the gray level of $p$ (according to $M_{\mathrm{f}}$ and $M_{\mathrm{b}}$).

Figure 4a shows an original manuscript. Its binary image produced by this algorithm is shown in Fig. 4b.

The results of the binarization algorithm were evaluated subjectively, where in each document the percentage of correctly segmented characters was counted. Our approach was evaluated on the 34 historical documents described in the next Section. These documents contained approximatley 20,000 characters, were the percentage of correctly segmented characters was 94%, where in a substantial set of documents the average percentage reached up to 98%. Most of the problems occured in documents where the character strokes had almost disapeared, and the accuracy of the adaptive local method was too sensitive to handle the rapid intensity changes. In few cases, the presence of bleedthrough caused small fragments due to the seed creation procedure, as can be seen in the middle of line 3 in Fig. 4b. For more details on the evaluation of the binarization method see [5].

## 3 Calligraphic letter database

Our data consist of 34 calligraphic manuscripts from the archive of *the School of Library, Archive and*

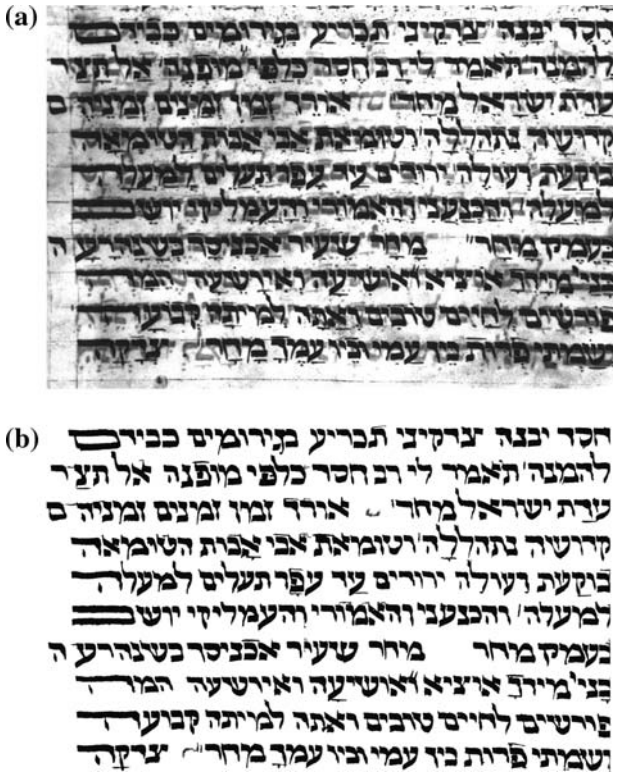


**Fig. 4** **a** Grayscale image. **b** After binarization

*Information Studies, the Hebrew university, Jerusalem, Israel*. The manuscripts are from the fourteenth to sixteenth century, written in different parts of the world. The Hebrew alphabet, consists of 22 letters, five of them have special forms when appearing at the end of a word.

### 3.1 Ground truth generation

In the document image analysis (DIA) research area, the term 'ground truth' refers to various attributes associated with the document. Parameters like the respective letter *ASCII* code, the bounding box coordinates of characters, the size of each character, etc., are associated with characters. Information like the document's writer identity, or any other global attributes, characterize the whole document. We have developed a visualization tool that enables interactive generation of the ground truth.

Consider the ground truth generation for a given gray scale manuscript. The user first enters the global information regarding the document. This includes the writer's identity, the date and place of the document writing, and any other global relevant information. Then we apply the binarization algorithm presented previously, and a standard connected component labelling operation. The labelled components are displayed, and

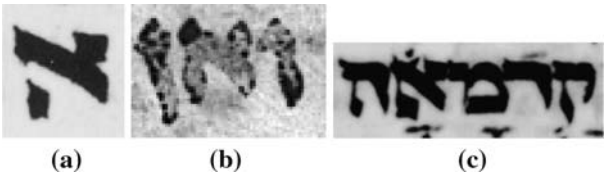


**Fig. 5** Typical problems with ancient documents: **a** broken, **b** defected and **c** merged characters

the user can tag each connected component with the respective letter *ASCII* code. In cases where the user identifies a set of connected components that belong to the same character, the user can link the components and assign the tag to set of the connected components. The program computes the bounding box for the connected components of each character, their center of gravity, and their sizes. The sub-image within the bounding box is extracted and entered with the respective information into the database. The visualization tool was developed using *MATLAB*, and we use *Access* as the data base.

Since historical documents often suffer from ink loss and smear, this causes a large number of broken and merged characters. A good *OCR* system for such documents must handle well these phenomena. Following this observation, for each character, along with its *ASCII* code, bounding box coordinates and size, we specify whether it is broken, merged or degraded (see Fig. 5 for example).

## 4 Extraction of selected letters

In Sect. 5 we describe our approach for writer identification of ancient manuscripts. Our method is based on style analysis of several selected letters. Since there is no transcription of the manuscripts, in order to automate the identification process we have developed a segmentation-free approach for extracting these letters. In this section, we describe our approach.

There are several papers dealing with retrieval of complicated characters or extraction of pre-defined symbols. A system for retrieval of Chinese calligraphic characters is reported in [7], where characters are represented by an approximated point context. In Saykol et al. [8], features based on angular and distance span of shapes are used for symbol extraction. The symbols are maintained in a codebook for the purpose of content-based image retrieval of Ottoman documents. A segmentation-free approach for recognition of arabic text is presented in [10]. Text primitives are extracted using mathematical morphology in order to recognize words.

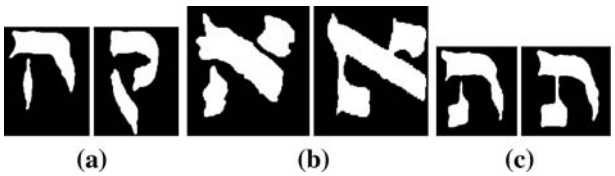


**Fig. 6** **a** Some of the Hebrew letters are composed of more than one connected component (CC), for example the letters Kuf and Hei. **b**, **c** Although most of the Hebrew letters are composed by one CC, their strokes are often disjoint and appear as several CC's

They report promising results for symbol extraction and word recognition.

We propose a segmentation-free approach for extraction of pre-specified letters based on the well-known erosion operator (for another use of the erosion operator, see Haralick et al. [11]). The extraction process is composed of several stages: structuring element generation, character extraction, character validation and structuring element adaptation.

### 4.1 Object-based erosion features

Let $I$ be a set of the foreground pixels in a binary image. Each object in $I$ is represented by one or more connected components. See, for example Fig. 6, in which sets of connected components represent some Hebrew Calligraphic characters. Denote by $\Omega = \{\omega_1, \ldots, \omega_N\}$ the set of object classes, each class representing a letter, where $N$ is the number of letters. For each object class $\omega_i, i = \{1, \ldots, N\}$, we generate a structuring element $S_i$, such that the number of translations in which $S_i$ is contained in an object class $\omega_i$ is maximal, and the number of translations in which $S_j$ is contained in an object class $\omega_i$ when $j \neq i$ is minimal.

The erosion operation $\ominus$, causes objects to shrink. The amount and the way that they shrink depends upon the choice of the structuring element. We show that when a suitable structuring element is used, the connected components of substantial area in the binary image $D_n = I \ominus S_n$, are most likely associated with objects belonging to class $\omega_n$. Consider the following definition of the erosion operation:

$$D_n = I \ominus S_n = \{(r,c) | (S_n)_{(r,c)} \subseteq I\}$$

For each foreground pixel $(r,c)$ in $D_n$, the structuring element $S_n$ translated by $(r,c)$ is contained in $I$. Denote the set of all foreground connected components of $I$ by $C = \{C_1, C_2, \ldots, C_M\}$, and the set of all foreground connected components in image $D_n$ by $CD^n = \{CD_1^n, CD_2^n, \ldots, CD_L^n\}$ (see Fig. 10b).

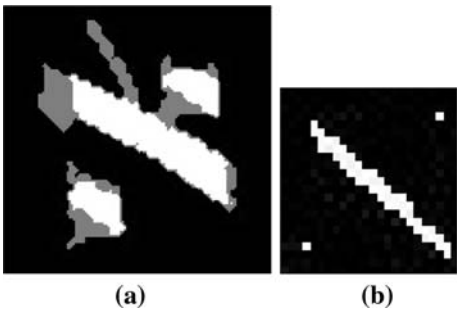


**Fig. 7** **a** The letter Aleph (*gray*) with the superimposed dilated structuring element (*white*). **b** The structuring element $S_n$, consisting of three connected components

The following two claims shed light on relevant properties of the relations between elements of $CD^n$ and $C$, based on the number of connected components in $S_n$.

*Claim 1* If the structuring element $S_n$ is connected, then for each connected component $CD_i^n$ in the eroded image $D_n$, there exists a connected component $C_k$ such that $CD_i^n = C_k \ominus S_n$.

*Proof* For each pair of connected (neighboring) pixels $(p,q) \in S_n$ and $(r,s) \in S_n$, the set $(S_n)_{(p,q)} \cup (S_n)_{(r,s)}$ is also connected.

*Claim 2* If $S_n$ is a union of $K$ connected components, $S_n = \bigcup_{k=1}^{K} S_{n,k}$, then for each connected component $CD_i^n$, there are at most $K$ connected components in $C$, such that

$$CD_i^n = \left\{ \bigcup_{j=1}^{K_i} C_{i,j}^n \right\} \ominus S_n \equiv C_i^n \ominus S_n, 1 \leq K_i \leq K,$$

where $C_{i,j}^n$ is the $j$th component among the set of connected components representing the $i$th object of class $\omega_n$.

*Proof* $S_n = \cup_{k=1}^{K} S_{n,k}$, where $S_{n,k}$ is a connected component. According to *Claim 1*, there is one connected component, say $C_{i,k}^n$, such that $CD_{i,k}^n = C_{i,k}^n \ominus S_{n,k}$. This is true for $k = \{1, 2, \ldots, K\}$. $CD_i^n = \bigcap_{k=1}^{n} C_{i,k}^n$ is the connected component containing all translations for which $(S_n)_{(r,c)} \subseteq C_i^n$. Since some of the $K$ components $C_{i,k}^n$ can be inter-connected, $C_i^n$ contains at most $K$ connected components.

Figure 7 illustrates a letter "Aleph" represented by two connected components (in gray color). Superimposed on the letter (in white) is the structuring element dilated by the respective $CD_i^n$ component. Notice that erosion followed by dilation is the well known open operator (for more details see Haralick et al. [9]).

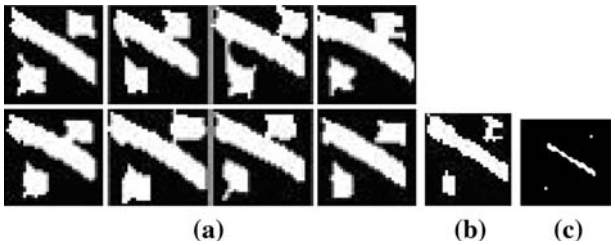


**Fig. 8** **a** A set of eight Alephs (training set). **b** Their intersection $CS_n$. **c** The structuring element $S_n$

The object based erosion features are defined as $E_i^n = \#\{CD_i^n\}$. $E_i^n$ is the number of pixels belonging to the $i^{th}$ connected component in the binary image $D_n$. It is the number of possible translations of $S_n$ such that $S_n$ is included in $C_i^n$. A high value of $E_i^n$ indicated that the component $C_i^n$ may represent an element belonging to $w_n$.

## 4.2 Generating the structuring element

The structuring element $S_n$ for class $\omega_n$ is generated in the following manner. Let $C_i^n, i = \{1, \ldots, T_n\}$ be $T_n$ sets of connected components, representing a training set of $T_n$ elements of class $\omega_n$. The selected sets are normalized to a standard height, and their widths are stretched by the same factors as their heights. Then, we calculate the maximum intersection (under translation) of these sets, and denote it by $CS_n$. $CS_n = \max(\bigcap_{i=1}^{T_n} C_i^n)$. The set of pixels belonging to $CS_n$ is contained in each one of the training set elements. The structuring element $S_n$ is a pseudo medial axis of $CS_n$ (i.e., strokes are replaced by thin lines, and small blobs are replaced by few pixels at and around the center of mass). Figure 8 illustrates the process of generating a structuring element for the letter Aleph.

**Fig. 9** An example of false detection

## 4.3 The letter extraction process

The extraction process is composed of several stages. In the first stage, we use the erosion operator to extract the candidate letters. As can be seen in Fig. 9, there are some cases where the structuring element $S_n$ is contained in a combination of several characters. Therefore, in the second stage a validation procedure is invoked in order to decide for each extracted character, whether it belongs to class $\omega_n$ or not. In order to make the extraction process robust and insensitive to different writing styles, an adaptation process of the structuring element $S_n$ is applied in the last stage.

*Letter extraction* The extraction process of objects belonging to class $\omega_n$, is as follows. Given a gray scale image, we first binarize it using the algorithm presented in Sect. 2. This results in a binary image $I$. Following the binarization, we normalize $I$'s height such that lines height in I will be equal on all documents. This is done inorder to make the structuring elements $S_n$, $n = \{1, \ldots, N\}$ insensitive to different object sizes. Then we apply the erosion operator on $I$, using the structuring element $S_n$ of the training set. The eroded image $D_n$ contains a set of connected components, $CD_i^n$, each representing a match between $S_n$ and the corresponding component in $I$, $C_i^n$ (see Fig. 10b).

*The validation process* For each $C_i^n, i = \{1, \ldots, N_x\}$, where $N_x$ is the number of letters extracted with $S_n$, we compute a measure $V_i^n$ for validation as follows.

$$V_i^n = \frac{\#\max(C_i^n \cap CS_n\}}{\#CS_n}$$

where $CS_n$ is the maximum intersection (under translation) between the training objects of class $\omega_n$ ($0 \leq V_i^n \leq 1$). $C_i^n \notin \omega_n$ if $V_i^n \geq THS$. We used in our experiments $THS = 0.9$.

*Adaptation of the structuring element* When dealing with writing styles different from the style of the training set, the extraction process often yields poor results. In order to adapt to the new style, we generate

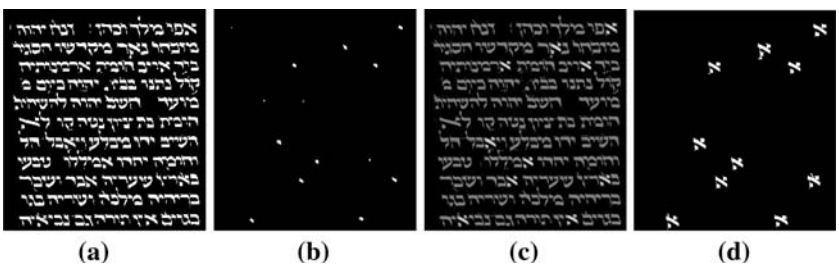


**Fig. 10** **a** The input binary image containing the set $I$. **b** $I$ eroded by the structuring element. **c** The set $I$ after opening by the structuring element $S_n$ (*in white*) superimposed on the input image (*in gray*). **d** The extracted Alephs

a new structuring element based on the extracted letters. After extraction and validation of $C_i^n$, we use the extracted characters as a training set for generating a new structuring element $S_n$ as described in Sect. 4.2.

We summarize the overall letter extraction algorithm as follows:

1. *Structuring element generation*
   - Let $C_i^n$, $i = \{1, 2, \ldots, T_n\}$ be the characters of the training set of a certain letter $\omega_n$.
   - Normalize $C_i^n$ to a standard height.
   - Compute the maximum intersection (under translation). $CS_n = \max(\bigcap_{i=1}^{T_n} C_i^n)$
   - Compute the pseudo medial axis of $CS_n$. $S_n = \text{Medial}(CS_n)$
   - $S_n$ is the structuring element for the letter $\omega_n$.
2. *Character extraction*
   - Compute $D_n = \{I \ominus S_n\}$
   - Compute $E_i^n = \#\{CD_i^n\}$, for i=1 to the number of connected components in $D_n$.
   - $C_i^n$ are the corresponding extracted characters from $I$.
3. *Validation measure*
   - Compute $V_i^n = \frac{\#\max(C_i^n \bigcap CS_n)}{\#CS_n}$.
4. *Structuring element adaptation*
   - Define the new training set $T$. $C_i^n \in T$, if $E_i^n \geq$ THS.
   - Repeat steps 1–3 to extract the new $C_i^n$.
   - $C_i^n \in \omega_n$ if $V_i^n \geq$ THS and $E_i^n \geq E_i^m$ for all $m \neq n$.

Section 6.1 summarizes the experimental results. 96% correct extraction rate of the letter *Aleph* is achieved with twenty six manuscripts written by different writers.

## 5 Writer identification

The writer identification problem is treated by comparing questioned handwriting with samples of handwriting obtained from known sources for the purposes of determining the identity of the writer. In other words, it is the examination of the design, shape and structure of handwriting to determine authorship of given handwriting samples. There are two main approaches to the off-line method, namely, *text-dependent* and *text-independent*. The *text-independent* approach uses feature sets whose components describe global statistical features extracted from the entire image of a text, or extracted from a region of interest, therefore may be called texture analysis approaches. The *text-dependent* approach uses features extracted from one or a limited group of characters. In Said [13], a text-dependent approach is presented, based on texture analysis of randomly extracted text blocks with Gabor filters. Schomaker et al. [14], evaluate the performance of edge-based directional features in comparison to several non-angular methods. Srihari et al. [15,16] performed a study on the individuality and discrimination power of characters and words by combining micro and macro features. They showed that different handwritten characters, have different discrimination power.

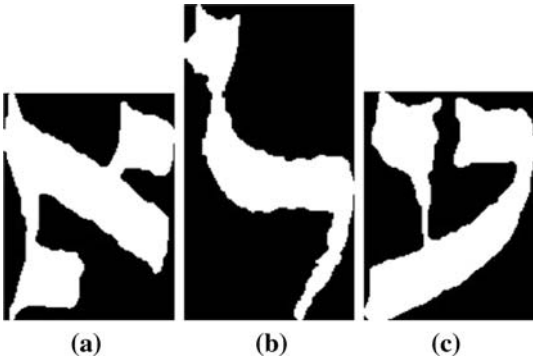


**Fig. 11** The letters used for writer identification: **a** Aleph, **b** Lamed **c** Ain

We propose a text-dependent approach based on a selected set of letters—the letters Aleph, Ain and Lamed (see Fig. 11). For each letter, a feature vector is extracted, and using dimension reduction techniques, the most discriminative features are selected for writer identification.

### 5.1 Feature extraction

Given a binary image representing a character, we decompose its shape into regions, using the convex hull of the character. Decomposing the character's shape, reduces it's complexity, and simplifies the description process. The convex hull $H_S$ of an arbitrary set $S$ is the smallest convex polygon containing $S$. The set difference $H_S$-$S$ is called the *convex deficiency* $D$ of the set $S$. We use the set $D$ to extract shape information from the character (Fig. 12).

Let $B$ be the binary image of a letter in our document. Denote by $S$ the set of all pixels where $B(i, j) = 1$ and denote by $D$ the convex deficiency of $S$. The set $D$ consists of a number of disjoint connected components included in $H_S$. These connected components represent the character concavities. In each letter, some of the components, depending on the character shape, are of substantial size. The rest of the components are due to noise and are insignificant. We will refer to the large

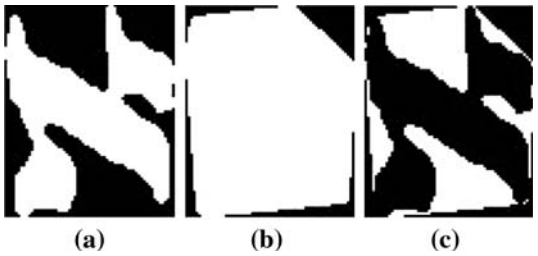


**Fig. 12** The white pixels are **a** the set $S$, **b** the set $H_S$, **c** the set $D$

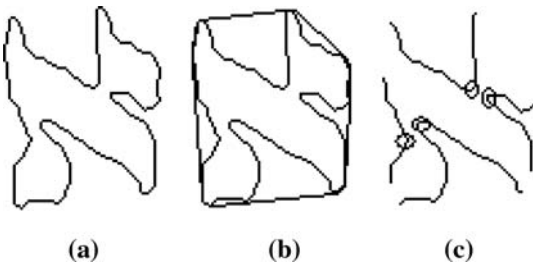


**Fig. 13** **a** An Aleph letter. **b** Superimposed on its convex hull. **c** The *dominant background sets* internal boundary segments. As can be seen, the character deficiency partitions its internal boundary into concavity segments. In each segment, the point with highest curvature is marked with a *circle*

connected components as *dominant background sets*. In the letter Aleph, there are four dominant background sets, while in each of the letters Lamed and Ain, there are two substantial components. The number of dominant background sets is independent of character size and orientation.

Denote the sets of pixels belonging to the dominant background sets by $D_i$, i = { 1, . . . , n}, where n is the number of dominant sets, according to the following:

Let $D_1$ be the component for which the $y$ coordinate of its center of mass is maximal, and number $D_i$, i={2, . . . ,n} in a clockwise order. We use geometrical and topological parameters of the character $B$ and of the sets $\{D_1, \ldots, D_n\}$ as features for classifying the handwriting style. We use the following notations:
Let $S$ be a set of pixels. Denote by $|S|$—the number of pixels in the set $S$ and denote by Major($S$), Minor($S$) the diameters of the major/minor axis of the ellipse having the same second moment as $S$ respectively.

For each set $D_i$, $i = \{1, \ldots, n\}$, the following features are extracted:

$\mathbf{F_{1i}} = \frac{|D_i|}{|H_S|}$—the ratio between the area of each *dominant background set* and the convex hull.

$\mathbf{F_{2i}} = \frac{\text{Minor}(D_i)}{\text{Major}(D_i)}$—the aspect ratio of the enclosing ellipse.

$\mathbf{F_{3i}}$—concavity features. The character's internal boundary, within $D_i$, is divided into two segments by the highest curvature point in that segment (see Fig. 13c). The length ratio of these two segments is used as a rough estimation of $D_i$'s curvature.

$\mathbf{F_{4i}}$—Compactness = $4\pi \times \text{area}(D_i)/\text{perimeter}(D_i)$ The compactness is defined as the ratio of the area of an object to the area of a circle with the same perimeter.

$\mathbf{F_{5i}}$—Moment features. The 2D moment of order $(p + q)$ of an image $B(x, y)$ is defined as $m_{pq} = \sum_x \sum_y x^p y^q B(x, y)$, for $p, q = 1, 2, \ldots$.

Central moments are defined as $\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y - \bar{y})^q B(x, y)$, where $\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_0 0}$. The central moments are translation invariant. In order to obtain scale invariance, we use the normalized central moment, $\eta_{pq} = \frac{\mu_{01}}{\mu_{00}^{\gamma}}$, where $\gamma = \frac{p+q}{2} + 1$, for ($p, q$=0, 1, 2). The last set of features are extracted from the character image $B$(as they were defined for $D_i$ in $\mathbf{F_{1i}}$, $\mathbf{F_{2i}}$ above):

$$\mathbf{F_6} = \frac{|B|}{|H_S|} \text{—area ratio}$$

$$\mathbf{F_7} = \frac{\text{Minor}(B)}{\text{Major}(B)} \text{—Ellipse aspect ratio}$$

$$\mathbf{F_8} = \frac{\text{height}(B)}{\text{width}(B)} \text{—Image's aspect ratio}$$

In total, the number of extracted features, depending on the chosen letter is 13*$n$+3, where n is the number of dominant background set.

### 5.2 Dimensionality reduction

Several papers use feature selection and extraction techniques to find the most discriminative features. Wang et al. [17], uses principal component analysis (PCA) followed by linear discriminant analysis (LDA) to lower the feature dimension and to find the most discriminative features for writer identification. In [18], a script identification approach is reported based on feature subset selection from a large set of features.

Each of the letters used in our experiment has its own set of features that maximizes the discrimination between writers. In order to find the best set of features, we implement two well established techniques for dimensionality reduction. The first is feature selection algorithm which searches for the most discriminative features. The second is the *Fisher linear discriminant analysis*, which finds a linear transformation that maximizes the classes separability.

#### *5.2.1 Feature selection*

The problem of feature selection is defined as follows: given a set $Y$ of $d$ features, select a subset $X$ of size $m$ that leads to the smallest classification error according to

a criterion function $J$. A natural choice for the criterion function is $J = 1 - P_e$, where $P_e$ denotes the classification error. A survey on feature selection algorithms can be found at [19–21]. Feature selection methods fall into categories of *filter* methods, which use feature selection as a preprocessing step to classification, and *wrapper* methods, which use classification internally as a means of selecting features. The simplest wrapper method is forward selection (FS). It starts with the empty set and greedily adds attributes one at a time. At each step FS adds the attribute that, when added to the current set, yields the best result. Once an attribute is added, FS cannot remove it later. Backward selection (BS) starts with all attributes and greedily removes them one at a time in a same manner. A powerful and widely used selection algorithm is the Sequential forward floating selection (SFFS) [22]. It is characterized by the changing number of features included or eliminated at different stages of the algorithm. We use the SFFS algorithm in the feature selection process.

#### *5.2.2 Linear discriminant analysis*

LDA is a popular method for dimensionality reduction that searches for a linear transformation which maps a $q$-dimension vector $x$ onto an $r$-dimensional ($r \leq q-1$) vector $Y = W^{\mathrm{T}}X$, while retaining a maximum amount of discrimination information. For the $c$-class problem, let $m_i i = 1, \ldots, c$ be the $i$th class mean vector, with $n_i$ samples. The within-class scatter matrix is defined as $S_{\mathrm{w}} = \sum_{i=1}^{c} p_i \sum_{j=1}^{n_i} (x_i - m_i)(x_i - m_i)^{\mathrm{T}}$, and the between-class scatter matrix are defined as $S_{\mathrm{b}} = \sum_{i=1}^{c} p_i \sum_{j=1}^{n_i} (m - m_i)(m - m_i)^{\mathrm{T}}$, where $m$ is the total mean vector, $P_i$ is the a priori class probabilities of the $i$th class.

According to Fisher's criterion, the transformation matrix $W$ can be obtained by maximizing the ratio $\frac{\det(W^{\mathrm{T}} S_{\mathrm{b}} W)}{\det(W^{\mathrm{T}} S_{\mathrm{w}} W)}$. It has been shown that the optimal transformation matrix can be found by solving the generalized eigenvalue problem $(S_{\mathrm{b}} - \lambda_i S_{\mathrm{w}} \mathrm{W}) = 0$. The eigenvectors corresponding to the $r$ largest eigenvalues then make up the rows of $W$.

### 5.3 Classification

The classification we have employed is based on one letter at a time. We have used the letters Aleph, Ain and Lamed. Given an unknown document (writer), we extract the characters Aleph, Ain and Lamed. The selected features are computed in a manner discussed in Sect. 5.2, and a classifier is applied. We used in our experiments both the K-nearest neighbors (KNN) classifier with $K = \{1, 5\}$, and the Linear Bayes classifier, assuming normal distributions [23].

Denote by $N_i$ the number of characters identified as writer $i$ by the classification procedure, then the writer of the manuscript is identified as writer $k$, if $k = \arg\max(N_k)$.

Thirty four documents were used in our experiments, each written by different writer. A correct classification rate of 100% was achieved. The experimental results are reported in Sect. 6.2.

## 6 Experimental results

### 6.1 Letter extraction experiments

The letter extraction process was evaluated on a set of *twenty two* documents. *Eight* documents were chosen randomly, from which the training set $T$ was generated. After the generation of $S_n$ from $T$, we applied the letter extraction process on each of the twenty two documents. The results of the experiment are summarized in Table 1.

The 5.35% classification error is illustrated in the following figures. Part of the errors occur due to degradation of characters, as can be seen in Fig. 14a. The structuring element does not fit inside the character due to missing parts or holes. A reconstruction procedure which identifies and reconstructs the degraded characters can solve this problem. Another kind of error is shown in Fig. 14b, depicting a stretched character at the end of a line. This is a common left justification in Hebrew calligraphy. These Alephs should be detected with a special structuring element.

The false detection errors mostly occur when the structuring element $S_n$ is contained in more then one

**Table 1** Letter extraction experimental results

| | |
|---|---|
| Number of documents | 22 |
| Number of characters | 19125 |
| Total number of alephs | 1477 |
| correct classification as alephs | 1398 |
| Percentage of correct classification as alephs | 94.65% |
| No. of false classification as alephs | 10 |
| Percentage of false classification as alephs | 0.052% |

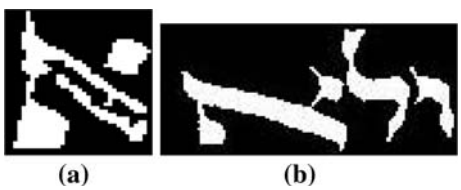


**Fig. 14** Examples of typical misclassification errors: **a** A degraded Aleph. **b** A stretched Aleph

**Fig. 15** An example of false detection. The structuring element $S_n$ is contained in more then one character

character, as depicted in Fig. 15. Although the validation procedure should handle these situations, there are some cases where more information is needed in order to correctly classify the character (i.e., word level information).

### 6.2 Writer identification experiments

Thirty four documents were used in our experiments. Twenty Aleph, Ain and Lamed characters were extracted from each document. Two sets of experiments were conducted: The first experiment conducted for each letter (Aleph, Ain and Lamed) in a "leave one out" manner. For example, the $34 \times 20 = 680$ Aleph characters were divided into a training set of 646 characters and a test set of 34 characters, one from each document. This classification was repeated 20 times, thus each character was classified once.

In the second experiment, each class was divided into two sets: a training set of fifteen characters, and a test set containing the remaining five characters (for each letter).

In both experiments, we compare the effectiveness of the dimension reduction techniques—LDA verses SFFS.

Experiments show that the Linear Bayes classifier outperformed the KNN classifier in all categories. Table 2 shows the results of the first experiment. Using the SFFS selection algorithm to select the best set of features, followed by the LDA transform to further lower the feature dimensionality, performed slightly better then direct use of LDA.

Table 3 shows the results of the second experiment according to the decision in Sect. 5.3. At the first experiment, combining the SFFS selection algorithm and LDA performed the best. The best results were obtained by using the letter *Aleph*, which has a more intricate shape.

**Table 2** Results of writer identification, experiment 1. This experiment was conducted in a "leave one out" manner

| Letter | Feature dimension | LDA (%) | SFFS (%) | SFFS + LDA (%) | Correct writer (%) |
|---|---|---|---|---|---|
| Aleph | 15 | 86 | 85 | 88 | 100 |
| Aleph | 10 | 82 | 79 | 83 | 100 |
| Lamed | 15 | 79 | 79 | 82 | 100 |
| Lamed | 10 | 77 | 75 | 79 | 100 |
| Ain | 15 | 76.4 | 76.7 | 76.4 | 100 |
| Ain | 10 | 76.9 | 69 | 76.4 | 100 |

The characters were divided into a training set of 646 characters and a test set of 34 character, one from each writer. The classification was repeated 20 times, thus each character was classified once. Although not all characters were classified correctly, we have reached 100% correct identification (the classification was based on majority rule)

**Table 3** Results of writer identification, experiment 2

| Letter | Feature dimension | LDA (%) | SFFS (%) | SFFS + LDA (%) | Correct writer (%) |
|---|---|---|---|---|---|
| Aleph | 15 | 84 | 84.4 | 85 | 100 |
| Aleph | 10 | 84 | 77 | 84.4 | 100 |
| Lamed | 15 | 77.5 | 76 | 80 | 100 |
| Lamed | 10 | 78.1 | 74 | 78.6 | 100 |
| Ain | 15 | 76.6 | 75 | 76.6 | 100 |
| Ain | 10 | 75.1 | 68 | 75 | 100 |

In this experiment, each class of characters (according to the writer) was divided into two sets: a training set of 15 characters, and a test set containing the remaining 5 characters (for each letter). Based on the majority rule, we have reached 100% correct identification

## 7 Summary

In this paper we have presented our work on Paleographic analysis of old Hebrew calligraphic scripts. We have developed an adaptive binarization method, which shows very good results. In Sect. 4 we presented a character extraction algorithm based on the erosion operator. We intend in the near future to extend the structuring element generation into a learning process in which for each writing style, the optimal structuring element is generated. In addition, we intend to automate the training set generation based on maximizing the intersection of the training elements. In Sect. 5 we presented a writer identification method based on extraction of a geometric parameters from several letters, followed by selection of the most discriminative features. Experimental results on 34 writers, yielded 100% correct identification. However, the result of our method should be examined on a much larger database. We plan to expand the database by at least an order of magnitude in the near future.

**Acknowledgment** We thank Professor Malachi Beit-Arie and Dr. Edna Engel of the Hebrew University for their collaboration in this project. We thank Professor R. M. Haralick of the Graduate Center of CUNY for the help and discussions while Professor I. Dinstein visited his laboratory in New York. We also want to thank the anonymous reviewers for their helpful and educative comments. Our work was supported in part by the Paul Ivanier

Center for Robotics and Production Management, Ben-Gurion University, Israel.

## References


1. Fournier, J.M., Vienot, J.C.: Fourier transform holograms used as matched filters in hebraic paleography. Isr. J. Technol. 281–287 (1971)
2. Sirat, C.: L'examen des 'critures: L'oeil et la machine, Paris, Editions du Centre National de la Recherche Scientifique (1981)
3. Dinstein, I., Shapira, Y.: Ancient hebraic handwriting identification with run-length histograms. IEEE Trans. Syst. Man Cybern. **12**, 405–409 (1982)
4. Likforman-Sulem, L., Maitre, H., Sirat, C.: An expert vision system for analysis of Hebrew characters and authentication of Manuscripts. Pattern Recognit. **24**(2), 121–137 (1991)
5. Bar-Yosef, I.: Input sensitive thresholding for ancient Hebrew manuscript. Pattern Recognit. Lett. **26**, 1168–1173 (2005)
6. Breu, H., Gil, J., Kirkpatrick, D., Werman, M.: Linear time Euclidean distance transform algorithms. IEEE Trans. Pattern Anal. Machine Intell. **17**(5), 529–533 (1995)
7. Zhuang, Y., Zhang, X., Wu, J., Lu, X.: Retrieval of Chinese calligraphic character image. In: 5th Pacific Rim Conference on Multimedia, Tokyo, Japan. pp. 17–24. Part I, (2004)
8. Saykol, E., Sinop, A.K., Gudukbay, U., Ulusoy, O., Cetin, A.E.: Content-based retrieval of historical Ottoman documents stored as textual images. IEEE Trans. Image Process. **13**(3), 314–325 (2004)
9. Haralick, R.M., Sternberg, S.R., Zhuang, X.: Image analysis using mathematical morphology. IEEE Trans. PAMI **9**(4), 532–550 (1987)
10. Al-Badr, B., Haralick, R.M.: A segmentation-free approach to text recognition with application to Arabic text. IJDAR **1**(3), 147–166 (1998)
11. Schauf, M., Akoy, S., Haralick, R.M.: Model-based shape recognition using recursive mathematical morphology. 14th International Conference on Pattern Recognition, pp. 202–204 (1998)
12. Beit-Arie, M.: Paleographical Identification of Hebrew Manuscripts: Methodology and Practice, in idem, The Making of the Medieval Hebrew Book, pp. 15–44. The Magnes Press, The Hebrew University, Jerusalem (1991)
13. Said, H.E.S., Tan, T.N., Baker, K.D.: Personal identification based on handwriting. Pattern Recognit. **33**(1), 149–160 (2000)
14. Bulacu, M., Schomaker, L.R.B., Vuurpijl, L.G.: Writer identification using edge-based directional features. International Conference on Document Analysis and Recognition, pp. 937–941 (2003)
15. Zhang, B., Srihari, S.N., Lee, S.: Individuality of handwritten characters. ICDAR **2003**, pp. 1086–1090
16. Zhang, B., Srihari, S.N.: Analysis of Handwriting Individuality Using Word Features. ICDAR '01, p. 1142
17. Wang, X., Ding, X., Liu, H.: Writer identification using directional element features and linear transform. In: International Conference on Document Analysis and Recognition, pp. 942–945 (2003)
18. Ablavsky, V., Stevens, M.R.: Automatic feature selection with applications to script identification of degraded Documents. In: International Conference on Document Analysis and Recognition, pp. 750–754 (2003)
19. Molina, L.C., Belanche, L., Nebot, A.: Feature selection algorithms: a survey and experimental evaluation. In: Proceedings of the International Conference on Data Mining, pp. 306–313 (2002)
20. Kittler, J.: Feature set search algorithms. Pattern Recognit. Signal Process. pp. 41–60 (1978)
21. Jain, A.K., Zongker, D.: Feature selection: evaluation, application and small sample performance. IEEE Trans. Pattern Anal. Machine Intell. **19**, 153–158 (1997)
22. Pudil, P., Novovicova, J., Kittler, J.: Floating search methods in feature selection. Pattern Recognit. Lett. **15**, 1119–1125 (1994)
23. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley, New York (2000)