Pattern Anal Applic
DOI 10.1007/s10044-008-0127-9

THEORETICAL ADVANCES

# Overlapping and multi-touching text-line segmentation by Block Covering analysis

**Abderrazak Zahour · Brunco Taconet · Laurence Likforman-Sulem · Wafa Boussellaa**

Received: 25 October 2007 / Accepted: 12 April 2008

**Abstract** This paper presents a new approach for text-line segmentation based on *Block Covering* which solves the problem of overlapping and multi-touching components. Block Covering is the core of a system which processes a set of ancient Arabic documents from historical archives. The system is designed for separating text-lines even if they are overlapping and multi-touching. We exploit the Block Covering technique in three steps: a new fractal analysis (*Block Counting*) for document classification, a statistical analysis of block heights for block classification and a neighboring analysis for building text-lines. The Block Counting fractal analysis, associated with a fuzzy C-means scheme, is performed on document images in order to classify them according to their complexity: tightly (closely) spaced documents (TSD) or widely spaced documents (WSD). An optimal Block Covering is applied on TSD documents which include overlapping and multi-touching lines. The large blocks generated by the covering are then segmented by relying on the statistical analysis of block heights. The final labeling into text-lines is based on a block neighboring analysis. Experimental results provided on images of the Tunisian Historical Archives reveal the feasibility of the Block Covering technique for segmenting ancient Arabic documents.

**Keywords** Block covering · Text-line segmentation · Overlapping and multi-touching lines · Block Counting · Ancient Arabic documents

## 1 Introduction

There is a huge amount of historical documents in libraries and National Archives that are waiting to be exploited electronically. Among them, many documents are written in Arabic with historical, philosophical, and scientific interest. The Tunisian Historical Archives [1] for instance own more than 85,000 documents including manuscripts. Like many archive institutions in the world, they have recently launched a project for digitizing and indexing this corpus. The final objective is to extract both document content (the text) and document logical structure from document images. It is thus necessary to develop a complete document analysis system including structure extraction and recognition stages. Text-line segmentation is one major component of a document analysis system devoted to historical documents.

Documents written in Arabic include specific characteristics such as the presence of many diacritical points and additional marks. Dots are used to distinguish some Arabic characters having the same basic shape. Other Arabic characters include special marks to modify the character accent. When diacritical symbols (dots, specials marks) are used, they appear above or below characters and they are

A. Zahour · B. Taconet
IUT, Université du Havre/GED, Place Robert Schuman,
76610 Le Havre, France
e-mail: zahour@univ-lehavre.fr

B. Taconet
e-mail: taconet@univ-lehavre.fr

L. Likforman-Sulem (✉)
TELECOM ParisTech/TSI and CNRS-LTCI, 46 rue Barrault,
75013 Paris, France
e-mail: likforman@telecom-paristech.fr

W. Boussellaa
Université de Sfax, REGIM, ENIS Route Soukra,
3038 Sfax (BPW), Tunisia
e-mail: wafa.boussellaa@gmail.com

**Table 1** Text-line segmentation methods for Arabic documents

| Methods | Overlapping | Simple-touching | Multi-touching | Constraint |
|---|---|---|---|---|
| Spaning tree [5] | Yes | No | No | |
| Attractive-repulsive forces [6] | Yes | Yes | No | Complete lines |
| Level set [7] | Yes | No | No | |
| Projection profile [8, 9] (surveys) | No | No | No | |
| Vertical strips [10, 11] | Yes | Yes | No | Not adaptative cutting out |

drawn as isolated entities. Diacritical symbols are positioned at a certain distance from the character. This makes the separating border of a text-line intricate; indeed, diacritical symbols can generate extra lines.

The writing is also highly cursive and produces many ascenders and descenders that generally touch each other through different text-lines, unless lines are very far from each other. In most cases, the space between text-lines is filled with many diacritical points, additional marks and touching components, which makes the segmentation of Arabic documents difficult.

Although many methods on handwritten line segmentation have been published in the literature for Latin and non-Latin scripts [2–4], only few papers are available on text-line segmentation of handwritten Arabic documents.

Abuhaiba et al. [5] propose a method based on the shortest spanning tree of a graph formed from a set of main strokes. Main strokes of extracted lines are arranged in the same order as they were written by following the path in which they are contained. Then, every secondary stroke is assigned to the closest main stroke. The proposed method can handle various line positions but does not handle touching and overlapping text-lines.

The approach based on attractive–repulsive forces is presented in [6] for extracting the baselines. It consists in iteratively adapting the $y$-position of a predefined number of baselines units. Pixels of the image act as attracting forces for baselines and already extracted baselines are acting as repulsive forces. The lines must have similar lengths. The result is a set of pseudo-baselines, each one passing through the body of the words. The method is applied to ancient Ottoman document archives (written in Arabic) and Latin texts. Some failures may happen when two neighboring text-lines are touching significantly and when text-lines are of different lengths. This method does not separate overlapping or touching components because the baseline only is searched and components are not assigned explicitly to text-lines.

In [7], text-lines are extracted by evolving an initial estimate using the level set method. The resulting lines may be broken into several segments due to large horizontal gaps between neighboring words and must be grouped by a post-processing step. Some failures may also happen when two neighboring text-lines are touching significantly or are connected in a few areas. Post-processing may be exploited to segment them horizontally to improve performance.

Several approaches described in recent surveys [8, 9] use the classical projection profile method and look for the minima of the profile to segment the image into text-lines. It is not easy, however, to extend these methods to handwritten documents where text-lines are connected, close to each other or skewed because there are no clear minima. A first adaptation of the projection profile method consists in dividing the document image into vertical strips [10, 11]. Projection profiles are then performed on each strip which makes the method tolerant to skew and line fluctuations.

These methods [7, 10, 11] can handle overlapping and simple touching components, i.e., components which are touching through at most two text-lines. However, they do not deal with multi-touching components. Table 1 shows method abilities to deal with overlapping and touching components.

The methods presented above are based on connected component analysis or on contour following. Our approach is different and based on the following principles. Let's consider a document image including overlapping and touching components (through two or more text-lines). Even in this case, there are enough words not overlapping or touching from which it is possible to derive statistics. Connected component analysis cannot solve this problem, because the height and the width of connected component enclosing boxes are two parameters that are statistically dependent. Moreover, widths of connected components are not related to inter-line spacing. In contrast to approaches based on connected components, our approach considers components including writing fragments of equal width; this width should be small enough in order that a fragment height should not be correlated to its width. The width should also not be too large; it should be typically the width of several characters, in order that a fragment should not be reduced to a diacritical mark. It is to be noted that words may be fragmented, but this is not important for text-line segmentation since words do not have to be extracted.

The simplest way for creating these writing fragments is to divide a document image into strips. Writing blocks are defined as boxes enclosing writing fragments. Within each strip, writing blocks are followed by empty blocks. With such block covering, one can obtain a high percentage of non-overlapping, non-touching writing fragments and a one-dimensional statistical analysis of block heights can be performed. Moreover, empty blocks represent the local inter-line spacing. The document image is thus covered by a set of blocks of same width but of variable height, which adapt to both text and inter-line space. This method we call *Block Covering,* relies on the choice of the strip width, on the statistical analysis of block heights and on the neighboring analysis of each block. Large blocks including overlapping and touching components are thus segmented using the estimated heights of average blocks and inter-line spacing. A main feature of our method is that it does not use any text-line following from baselines or contours. The robust statistical analysis of block heights (average and empty blocks) can solve multi-touching cases without using other contextual information.

We have considered documents including overlapping and touching (simple or multi-touching) components. A system dealing with any type of document should also include a classification step for classifying documents according to their line spacing: documents with small inter-line spacing (TSD, tightly spaced documents) and documents with large inter-line spacing (WSD, widely spaced documents). A new method for document classification is presented and called *Block Counting*. The method is based on fractal analysis of the writing and also uses blocks and block covering.

The paper is organized as follows. Sect. 2 describes the overall system. In Sect. 3, the fractal analysis and the classification stage for detecting tightly spaced documents (TSD) and widely spaced documents (WSD) are presented. Sections 4 and 5 are dedicated to the separation of touching and overlapping components and the assignation of all extracted blocks to alignments. Experimental results are reported in Sect. 6. Some conclusions are drawn in Sect. 7.

## 2 Block covering technique

The above methods of text-line segmentation rely on connected component analysis, or line following. In our previous paper [11], the document is cut into strips of fixed width; the local minima of the profile of partial projections are detected. From these minima, text lines are extracted by contour following and the segmentation process takes into account the global skew angle, estimated prior to segmentation. This method does not require any skew correction but it fails when the document includes multi-touching lines.

The present approach is different. It relies on blocks which cover, without overlap, the whole document image. These blocks are of two types: the blocks covering the local inter-line space (empty blocks) and the blocks covering writing fragments (non-empty blocks). The width of all blocks is fixed but the height is locally variable.

The majority of writing blocks include non-overlapping and non-touching components. Stable statistics can thus be extracted from these blocks. Similarly, the height of the majority of empty blocks corresponds to the local inter-line spacing. Using such blocks is more efficient than using connected components because both height and width of connected components vary and are statistically dependent. Moreover, the local inter-line spacing is hard to estimate from the set of connected components. Our method does not rely on any contour or baseline following. The multi-touching case is solved from statistics on average writing-block and empty-block heights. Prior to segmentation, it is necessary to compute the global skew angle and to correct it [12], in order for the blocks to be only oriented vertically and horizontally.

The Block Covering technique performs as follows. First, the image is cut out regularly into vertical strips of width $r$ (Fig. 1). To find the height and the position of a covering block, the shape is projected on the vertical axis. Then, the histogram of projections is cut out into intervals made of empty lines and intervals made of nonempty lines.

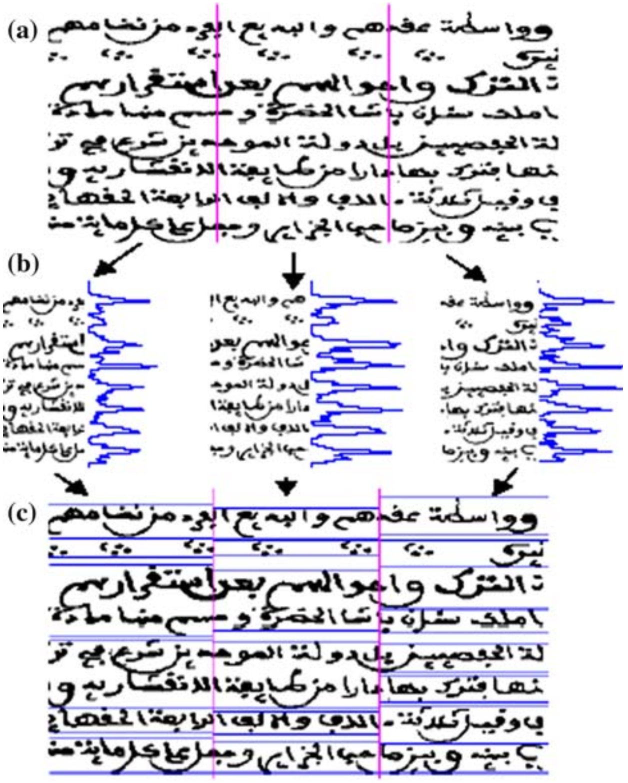


**Fig. 1** Block extraction on a sample document: **a** dividing the document into strips. **b** projection profiles within each strip **c** resulting text blocks (a text block is delimited by two *horizontal blue lines*)

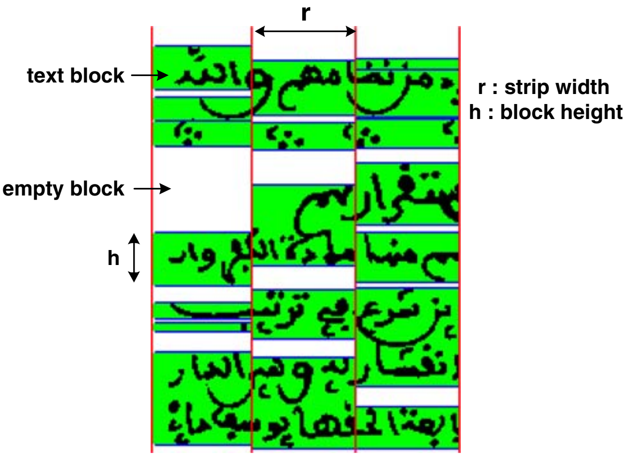


**Fig. 2** Block Covering with blocks of width $r$

An interval made up of nonempty lines defines the height and the position of a text covering block within the strip (Fig. 2). The width of all blocks is $r$, but the block height $h$ is variable and adapts to the fragments of the shape.

The unit of $r$ is set as the image width. So, $1/r$ represents the number of vertical strips (vsn), and $r$ is equal to 1/vsn. For example, let $L = 500$ pixels be the image width and let the number of vertical strips vsn be equal to 10. The absolute width of a strip is 50 pixels and $r$ is equal to 50/500, i.e., $r = 1/\text{vsn} = 1/10$. This way to set $r$ makes the strip width independent of image width.

A Block Covering only depends on the strip width $r$. We have used Block Covering in three steps:

- The fractal-based analysis yielding document classification ($r$ is varying) (see Sect. 3).
- The statistical analysis of block heights ($r$ is fixed optimally) (see Sect. 4).
- The neighboring block analysis ($r$ is fixed optimally) (see Sect. 5).

Figure 3 shows the block diagram of the text-line segmentation system proposed in this paper. We assume an input image already restored, deskewed and binarized including only text. These pre-processing steps are described in [13].

The document to be segmented is first analyzed and labeled according to the suitable class (TSD or WSD). For segmenting TSD documents, text blocks are classified into three categories: small text blocks correspond to diacritical marks, average blocks stand for the bodies of the words, overlapping or touching characters are gathered together into large blocks. Text-line extraction is carried out by segmenting large blocks and assigning each resulting text block to a single line. It can be noted that large blocks can be divided into more than two blocks for handling multi-touching lines. For WSD documents, text blocks are classified into two categories: small blocks and average blocks.

## 3 Document classification

This section describes the automatic classifier which sorts documents into two categories: WSD and TSD. From the fractal analysis performed on document images, two features are extracted. These features constitute the input vector of a fuzzy C-means classifier.

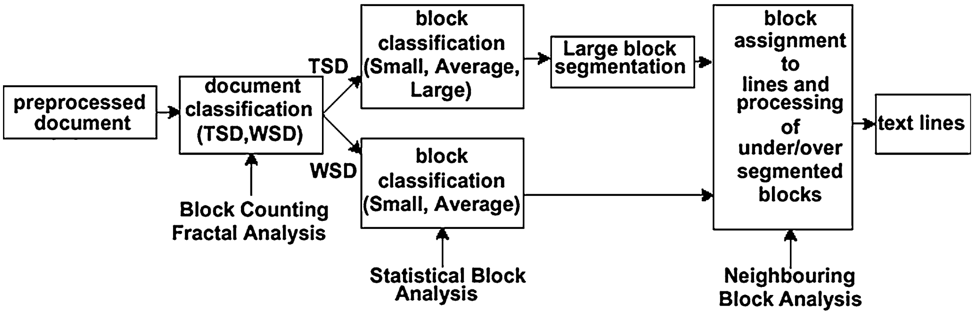
The fractal dimension resulting from fractal analysis is a measure of how writing occupies the space [20]. The fractal dimension resulting from the classical box-counting technique is a measure of how writing strokes occupy the geometrical space. The novel block-counting technique described below leads to a new fractal dimension which measures how the writing fragments are arranged into text lines. This measurement is used to classify documents as WSD or TSD.

### 3.1 Usual fractal methods

#### *3.1.1 Fractal dimension*

The dimension of an object describes the way in which it occupies the space and thus how its size is quantified. The size of an object can be measured by counting the number of rulers necessary to cover it. Let a ruler be of dimension $d$ and characterized by its size $r$. The size $h(r)$ of the object is:

**Fig. 3** System overview


preprocessed document
document classification (TSD,WSD)
TSD
WSD
Block Counting Fractal Analysis
block classification (Small, Average, Large)
block classification (Small, Average)
Statistical Block Analysis
Large block segmentation
block assignment to lines and processing of under/over segmented blocks
Neighbouring Block Analysis
text lines

$$h(r) = \gamma(d) r^d \tag{3.1}$$

where $\gamma(d)$ can be written with $\Gamma$ function:

$$\gamma(d) = \frac{\left(\Gamma(\frac{1}{2})\right)^d}{\Gamma(1 + \frac{d}{2})} \tag{3.2}$$

Hausdorff defines the $d$-dimensional measure $H_d(F)$ of a unit $F$, included in metric space, characterized by dimension $d$, as:

$$H_d(F) = \lim_{\varepsilon \to 0} \inf_{r < \varepsilon} \left\{ \sum h(r) \ : \ C(r) \right\} \tag{3.3}$$

where $C(r)$ represents any finished cover of $F$ by rulers of size $r < \varepsilon$. The term between braces indicates the whole of $F$ covers.

The accuracy of the measure increases as the size of the ruler decreases, this is why it is better to take the limit as epsilon approaches zero.

Hausdorff shows that for any unit $F$, there is a unique value of $D_H$ such that:

$$H_d(F) = \begin{cases} \infty & \text{if} \quad d < D_H \\ 0 & \text{if} \quad d > D_H \end{cases} \tag{3.4}$$

$D_H$ is known as the Hausdorff dimension; $H_D$ is the Hausdorff measure (when $d = D_H$).

Contrary to topological dimension, this dimension can be non integer.

The term fractal dimension was introduced by Mandelbrot in 1983. He stated that one criterion for a surface being fractal is its self-similarity. According to him, a set A in an Euclidean n-space, is said to be self-similar when A is the union of $N$ distinct (non-overlapping) copies of itself, each of which has been scaled down by a ratio $r$ in all coordinates. The fractal dimension $D$ of the set A is given by the relation:

$$D = \frac{\log N(r)}{\log(\frac{1}{r})} \tag{3.5}$$

As $r$ approaches 0, $D$ becomes the Hausdorff dimension.

#### 3.1.2 Fractal aspect of natural scenes

Natural scenes do not exhibit the deterministic self-similarity property. Instead, they exhibit some statistical self-similarity. If the set is scaled down by a ratio $r$ in all the dimensions, then it becomes statistically identical to the original one in a validity range. The fractal dimension is also given by Eq. 3.5.

Fractal geometry can yield features able to qualify the writing according to the complexity of graphics [15]. It was used successfully for the classification of writing families (Latin, Arabic, Chinese…) [16]; it allowed the identification of the Arabic fonts [17]. While the definition of fractal dimension by self-similarity is straightforward, it is difficult to estimate the fractal dimension directly from the image. In practice, two techniques are used: box counting and image dilation.

#### 3.1.3 Box counting dimension

The image is cut out into square boxes of side $r$. $N(r)$ represents the number of nonempty square boxes (containing at least one pixel of the shape). The Hausdorff measure is replaced by a simplified measure $H$:

$$H = N(r) r^D \tag{3.6}$$

where $N$ is the number of boxes of side $r$ which cover the shape. By taking the logarithm of the two members, the formula becomes:

$$\begin{aligned} &\log(H) = \log N(r) + D \log(r) \quad \textit{or} \\ &\quad \log N(r) = D \log\left(\frac{1}{r}\right) + \log(H) \end{aligned} \tag{3.7}$$

In practice, the fractal dimension $D$ is estimated by the slope of the least squares straight line, with $r$ taking values in the interval where the linearity is checked. The parameter $D$ is usually the only one retained as a fractal characteristic in the writing recognition domain.

#### 3.1.4 Image dilation dimension

At each point of the image, a dilation is carried out in a neighborhood where the topology is set preliminarily (generally in an isotropic way, the basic topologic cell is a disc; then $r$ represents the radius of the disc). $N(r)$ is the number of points of the dilated shape. With this method, one obtains Minkowski–Boulingand fractal dimension. In both cases, $r$ is the homothetic factor of the transformation of the basic topological cell. In practice the fractal dimension value differs according to the technique used for its estimation. In the literature, the box-dimension term is the fractal dimension related to the frequently used box-counting technique.

## 3.2 Block Counting method

#### 3.2.1 Presentation of the method

In the case of writing, fractal dimension is a measure of the way writing occupies the 2D geometrical space. Box counting and image dilation methods assume the isotropy of the scale variation (internal isotropic homothetic transformation), or more rarely assume the self-affine anisotropy. The proposed block-counting method is different. As a matter of fact, the ruler is a nonempty block as explained in Sect. 2: the width of the ruler is that of the

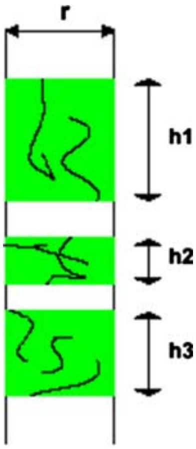


**Fig. 4** Ruler blocks. Width $r$ is fixed and heights $h1$, $h2$, $h3$ are adapted to the shape

strip, i.e., $r = 1/\mathrm{vsn}$, vsn being the number of vertical strips, but the ruler height is variable and adapts to the fragments of the shape.

The strip width $r$ of the Block Counting ruler can be very small, as it is for the box-counting ruler. Thus, the ruler has the aspect of a rectangular block, but only its width contributes to the quantification of the measure (Fig. 4).

The ruler is a block, not a box, so we call this technique "block-counting". According to the Hausdorff theory [14, 18], since $r$ is the only significant ruler parameter, the simplified Hausdorff measure for the block-counting analysis is:

$$H_{\mathrm{B}} = N_{\mathrm{B}}(r) * (r)^{D} \tag{3.8}$$

where $N_{\mathrm{B}}$ is the number of ruler blocks of width $r$, covering the shape.

By taking the logarithm of the two members:

$$\log N_{\mathrm{B}} = \log H_{\mathrm{B}} + D * \log(1/r) \tag{3.9}$$

The linearity of the graph is related to the self-similarity behavior: $D$ is interpreted as the slope of the straight line and log $H_{\mathrm{B}}$ as the origin ordinate.

### *3.2.2 Analysis of synthetic images*

The first image considered is shown in Fig. 5. It contains ten horizontal stripes which schematize text-lines. Any vertical strip of the cut-out image is self similar with the whole image. The number of blocks is proportional to the number of strips cut out into the image. The application of the formula (3.8) gives $D = 1$ and $H = 10$. $H$ represents the cumulated length of horizontal stripes. Figure 5 shows the graph of log(Nb) versus log(1/$r$).

One can notice that if these stripes undulate without overlapping, even if their thickness varies, $D$ locally remains equal to 1 and $H$ remains equal to 10.

The image in Fig. 6 consists of ten horizontal stripes with additional barbs of variable height. These barbs produce the overlapping and the joining of the adjacent stripes. This synthetic image schematizes the intricacy of the lines of text in a real document. Self-similarity is not verified for the greatest values of $r$ (Fig. 6). The points corresponding to these values accentuate the slope of the approximate straight line and thus make it possible to discriminate this case from the regular case. $H$ has a lower value and no longer represents the number of stripes. The linearity of the graph appears when $r$ is lower than $r0$ (here, $r0$ is equal to 1/20). The self-similarity is thus valid in this area. $H$ is equal to 10, which corresponds to the number of horizontal stripes in the image.

The analysis of the preceding cases shows that for the irregular case (overlapping, touching lines), the fractal dimension $D$ is higher than 1 and $H$ is different from the number of text-lines (equals to ten in previous examples). In the regular case, the fractal dimension $D$ is equal to 1 and $H$ is equal to the cumulated length of text-lines.

In both regular and irregular cases, one observes an alignment of points which defines the auto similarity straight line fractal dimension (the slope value) noted $D0$ and equal to 1. The origin ordinate is noted log$H0$. $H0$ can

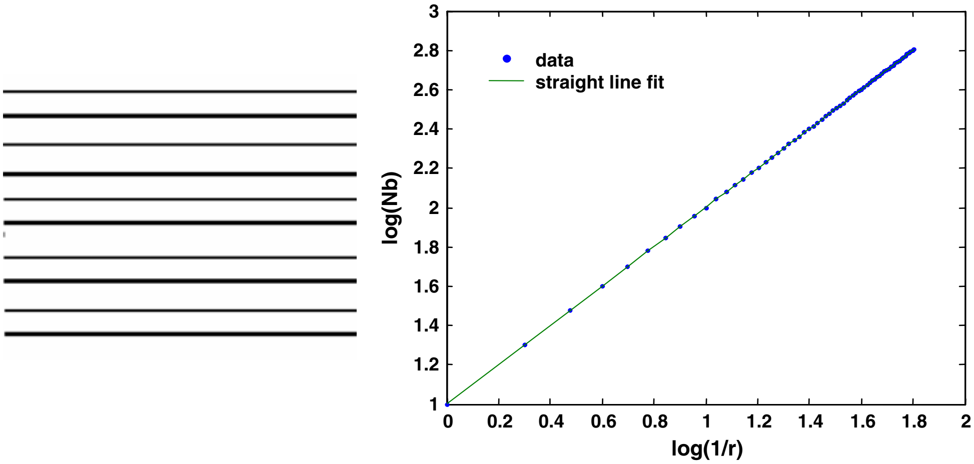


**Fig. 5** Image made up of ten horizontal stripes and graph of log(Nb) versus log (1/$r$)

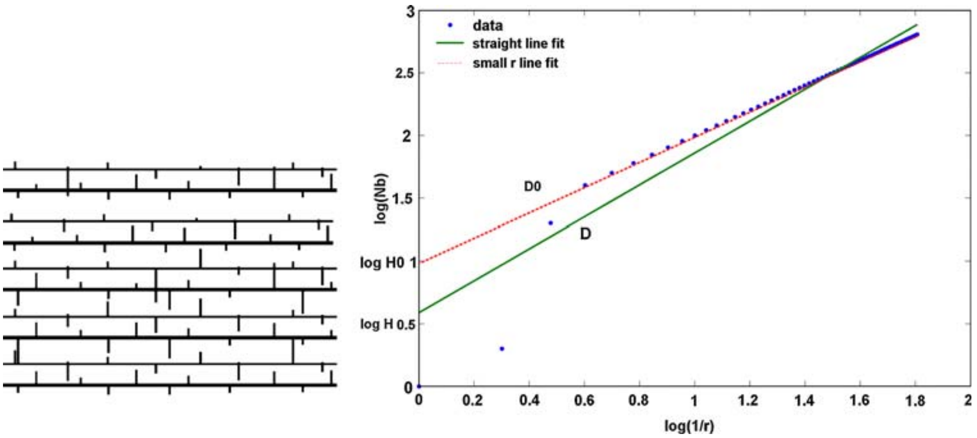


**Fig. 6** Image made up of ten horizontal stripes with additional barbs and the graph of log (Nb) versus log (1/*r*). The straight line fit estimated from all points (*in green*) yields *D* = 1.3, *H* = 3.9. The small *r* line fit (*in red, dotted*) yields *D*0 = 1 and *H*0 = 10. Δlog*H* = 0.41

be seen as the cumulated length of all (even overlapping and touching) horizontal lines, because $D0 = 1$ is the topological dimension for lines. In the irregular case, the straight line fit estimated from all points has a slope *D* different from *D*0 and an origin ordinate log*H* different from log*H*0. *D* is the mean fractal dimension and the measurement *H* can be seen as a mean cumulated length of separable lines, because *H* is theoretically equal to the number of blocks when there is no cutting out ($r = 1$, Eq. 3.9). In the regular case, the straight line fit estimated from all points merges with the auto-similarity straight line ($D = D0$, $H = H0$) (see Fig. 6). Thus, it is possible to discriminate the two classes (TSD, tightly spaced documents, WSD, widely spaced documents) with features *D*/*D*0 and log(*H*0/*H*) (denoted Δlog*H* in the following). These two features are dimension free; they do not depend on the size of the image. This technique used for synthesized images can be applied to real images if the graph of log(Nb) versus log(1/*r*) shows the same properties.

### 3.2.3 Analysis of real document images

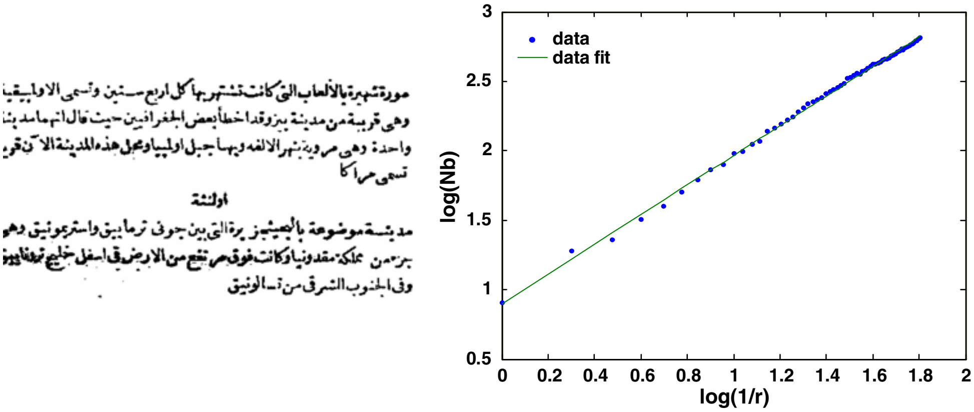
When all text lines are separable by projection, one can notice that the line of approximation has a slope value very close to 1, and that all feature points are almost aligned on this line (see Fig. 7). When the lines are overlapping or touching (Fig. 8), *D* is greater than 1 (here *D* = 1.76), a zone of linearity is observed for low values of *r* and Δlog*H* is significantly non null (here $\Delta\log H = 0.76$).

In real cases, *D* measures the degree of interpenetration of the text lines: the less separable the lines by projection are, (in other words the more they are tangled up), the higher the fractal dimension. Indeed, cutting out into thin strips reveals new blocks masked by projection in broader strips.

When the strips are sufficiently thin, the text-lines become separable within strips and the self-similarity appears. As a result, the linearity is seen in this zone, with a slope value very close to 1.

The graph for real documents has the same aspect as the graph for synthesized images. So, the two previous features are also suitable for real document images.

Moreover, for the set of documents we consider, a common linear zone is observed for all documents. The interval for this linear zone extends from $(1/r)_{\min}$ to $(1/r)_{\max}$. Above $(1/r)_{\max}$, the linearity is not guaranteed any more. This range is considered to be sufficient to plot with precision the straight line corresponding to the self-

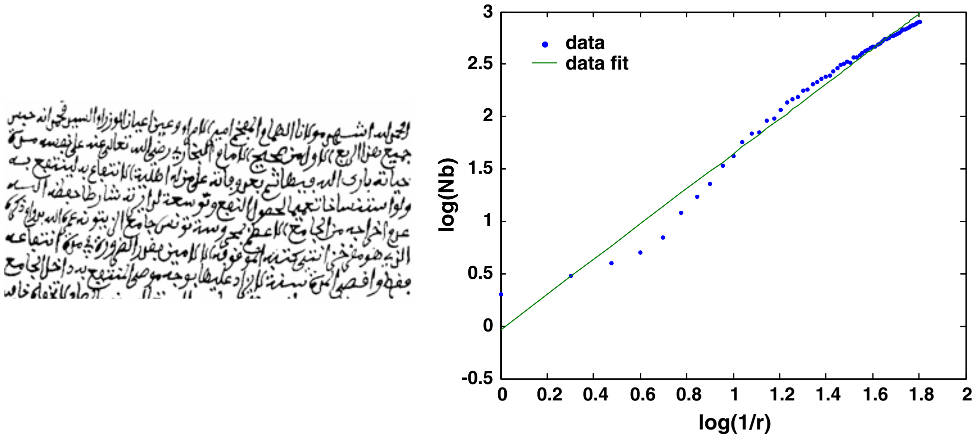

موورة شهيرة بالألعاب التى كانت تشتهر بها كل اربع سنين وتسمى الاولمبيقيا
وهى قريبة من مدينة بيزو وقد اخطأ بعض الجغرافيين حيث قال انهما مدينة
واحدة وهى مروية بنهر الالفه وبها جبل اوليمبيا ومحل هذه المدينة الآن قرية
تسمى مراكا

اولنثة

مدينة موضوعة بالجزيرة التى بين جونى ترمايق واسترمونيق وهو
جزء من مملكة مقدونيا وكانت فوق مرتفع من الارض فى اسفل خليج تورونايوس
وفى الجنوب الشرقى من تسالونيق

**Fig. 7** Sample document and graph of log(Nb) versus log(1/*r*). All lines are separable by projection. *D* = 1.05 and Δlog*H* = 0.06

**Fig. 8** Sample document and graph of log(Nb) versus log(1/$r$). Text lines are significantly touching and overlapping. $D$ = 1.76 and ΔlogH = 0.76



similarity zone. As previously, the straight line estimated from all points extends from 1 to $(1/r)_{\max}$.

### 3.3 Training and classification

The feature vector (Δlog$H$, $D$) is the input of an unsupervised fuzzy C-means classifier which classifies training samples into two categories (WSD or TSD). Figure 9 shows the resulting classification of the training documents.

The points in Fig. 9 are located in a diagonal area: indeed, when the fractal block dimension $D$ is close to 1, Δlog$H$ is very close to zero; when $D$ is large, Δlog$H$ is greater than zero. The WSD documents are characterized by a value of $D$ close to 1 and a value of Δlog$H$ close to 0. In contrast, TSD documents are characterized by a value of $D$ greater than 1 and a value of Δlog$H$ significantly non null. When training points are classified, only two points are almost equidistant from the centers of the two classes. For these two points, examining the matrix of the degrees of fuzzy membership gives values about 0.48 for class TSD and 0.52 for class WSD. For all the other points, the degree of principal membership is greater than 0.95. These two ambiguous points were thus withdrawn from the training set.

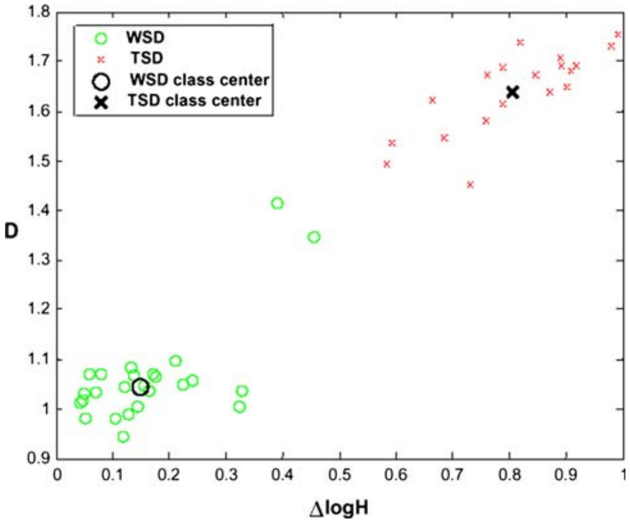


**Fig. 9** Classification of training documents into two classes (*WSD* or *TSD*) by fuzzy C-means. Two TSD documents are misclassified as WSD

The two classes, WSD and TSD, are determined from the training set and they are represented by the statistical model of Mahalanobis. Indeed, the distribution of points is much more compact for class WSD than for class TSD. The Mahalanobis distance takes into account this dispersion through the covariance matrix. A class $C_i$ is represented by mean vector $X_i$ covariance matrix $\Sigma_i$. The distance of a document $X$ to class $C_i$ is given by the formula:

$$d(X, C_i) = \sqrt{(X - X_i)^T \Sigma_i^{-1} (X - X_i)}$$

The two previous misclassified points are correctly classified, i.e., as TSD documents, when using this distance. The resulting WSD/TSD classification is similar to human classification.

## 4 Text-line segmentation

During the previous stage, a document has been labelled as WSD or TSD. In the current stage, the Block Covering technique is still used (see Fig. 1). However, the covering blocks are processed differently; previously they were covering blocks which had to be enumerated. They are now blocks that are statistically classified according to their height. The number of classes for blocks differs according to the label of the document: three classes for a TSD document and two classes for a WSD document. In this stage, the optimal Block Covering is automatically found. The optimal Block Covering yields the best histogram of block heights; small blocks correspond to diacritics. The height of average blocks corresponds to the height of text-lines and word bodies. Large blocks result from the fusion of several text blocks which overlap or touch each other. Such interpretation on the origin of block heights is valid when the writing size is rather homogenous. Blocks from the large block class are only included in TSD documents. Large blocks are then segmented into average blocks; then, the document including only small and average blocks is segmented into text-lines by a neighboring block grouping process.

### 4.1 Block classification of TSD documents

A TSD document includes three types of blocks. The optimal Block Covering has to be found for each document image. The best Block Covering yields the best classification; the classification method is the unsupervised K-means method in the 1D case. Too many strips produce too many average blocks while too few produce too many large blocks. It is thus important for the number of strips to be determined accurately. The optimization criterion must favor high intra-cluster density and low between-cluster density. The proposed criterion is inspired from [19]. This criterion is originally used for determining the optimal number of classes from data in a Kohonen-based self organizing map. In Wu's study, data are fixed and the optimal number of classes is to be determined. In our case, the number of classes is fixed (the three classes considered are: small, average and large) but data vary (block height and the number of blocks) according to the number of vertical strips. The number of vertical strips ($\mathrm{vsn} = 1/r$) is the parameter to be adjusted. The criterion must give both a high separation measure and a low interclass density. So it is proposed to maximize their product.

The overall clustering validity index, which is called in [19] *Composing Density between and with clusters* (CDbw), is defined here by:

$$\mathrm{CDbw}(\mathrm{vsn}) = \mathrm{intra\_den}(\mathrm{vsn}) * \mathrm{sep}(\mathrm{vsn})$$

The optimal number of vertical strip, denoted ovsn, is the one which maximizes the compromise criterion CDbw (vsn). The quality of intra-class clustering is denoted by *intra_den*. The cluster separation measure is denoted by sep(vsn). We defer the mathematical definitions and development of the CDbw criterion in Appendix A.

Figure 11 shows the variation of the quality measure CDbw for different values of vsn when dividing the TSD document in Fig. 10 into strips. The optimal value ovsn is equal to ten strips. Figure 12b shows the resulting classification of the text blocks of Fig. 12a. Small, average and large blocks are in purple, blue and red, respectively.

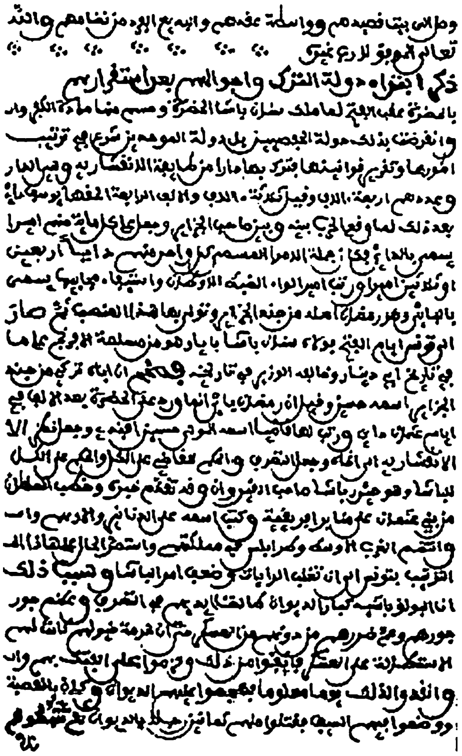


**Fig. 10** An ancient Arabic TSD document

### 4.2 Segmentation of large blocks

Large blocks result from multi-overlapping and multi-touching text-lines in the $y$ direction. Segmentation consists in dividing a large block into $n$ average blocks of same height $h$ and spaced by $n - 1$ empty inter-line blocks of height $e$, with $n$, $h$ and $e$ adapted to each large block. Each triplet $(n, h, e)$ can synthesize a large block of height $Hs$. $Hs$ thus equals $n * h + (n - 1) * e$ (see Fig. 13).

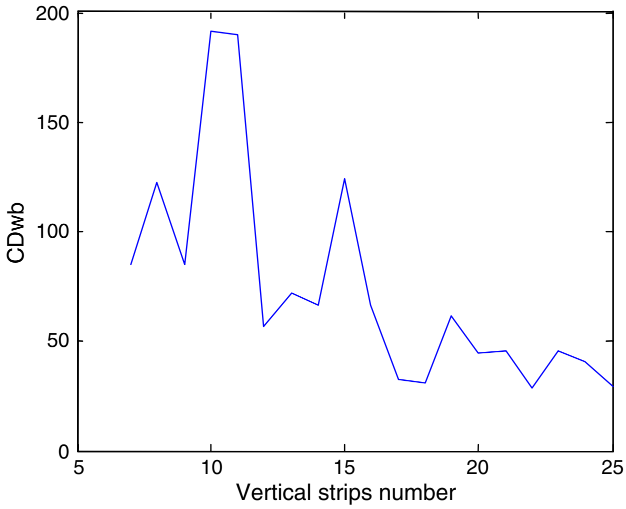


**Fig. 11** Clustering quality measure CDbw versus number of vertical strips vsn

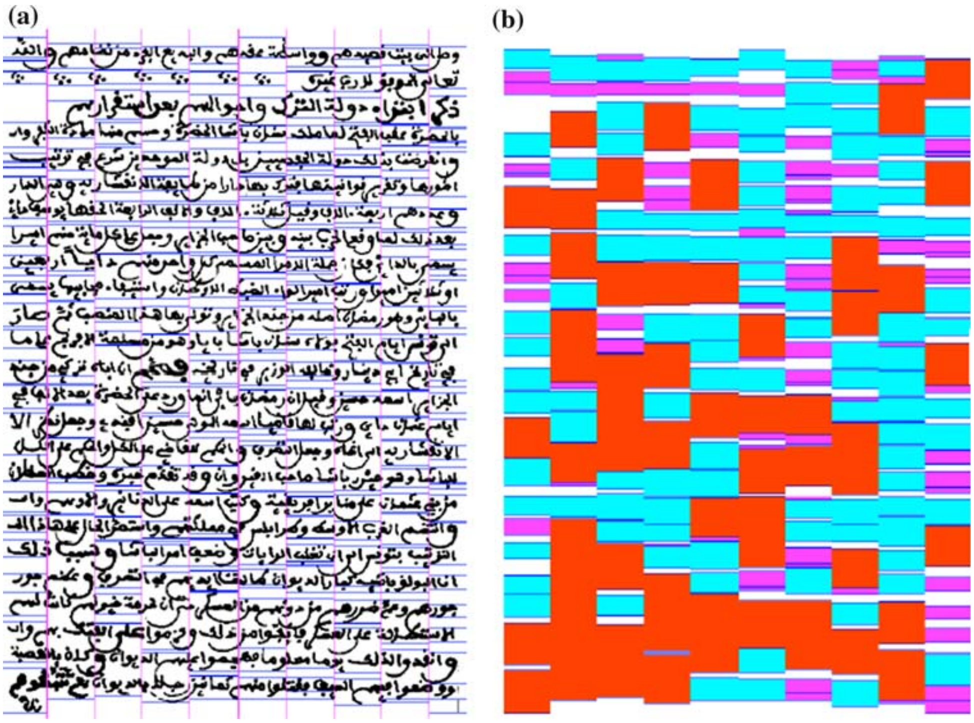


**Fig. 12** **a** Text blocks resulting from dividing a document image into ten vertical strips. **b** Classification of text blocks into small (*purple*), average (*blue*) and large (*red*) blocks

For a large block of height *Hs*, the (*n*, *h*, *e*) triplet is chosen in order to best fit the given block:

$$(h, n, e) = \underset{(h,n,e)}{\arg} \left(\min \| Hs - (n * h + (n-1) * e) \| \right)$$

The estimated line number *n* must satisfy $n \geq 2$.

The height *h* is that of a rather large average block: the range of *h* (in pixels) is [mean(*h*), mean(*h*+*stddev*(*h*))], considered to be statistically the most probable. mean*(h)* and *stddev(h)* are the first two statistical moments of the height of the average block class.

The inter-line block set is a subset of the empty blocks set. All empty blocks do not belong necessarily to the class of inter-line blocks. Indeed, an empty block may result from: an incomplete text-line, an indented paragraph, a margin at the beginning or at the end of the text. The inter-line blocks correspond to the smallest empty blocks. To separate the set of empty blocks into two classes (inter-line block, not inter-line block), a K-means classification is performed. The class with smallest height corresponds to the inter-line blocks. The first two statistical moments mean*(e)* and *stddev(e)* are then calculated on the set of inter-line blocks to determine the admissible interval for *e*: [sup(1, mean(*e*)−*stddev*(*e*)), *mean*(*e*)].

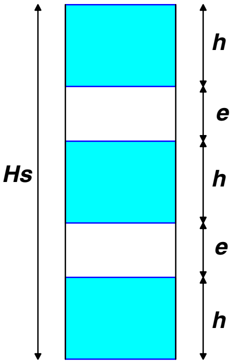


**Fig. 13** Synthesized large block of height *Hs* by three average blocks of height *h* separated by two inter-line blocks of height *e*. The corresponding triplet is *(h, e, n* = 3*)*

If several triplets satisfy the above condition, the triplet with maximum *h* value and minimum *e* value is selected.

Figure 14 shows an enlarged zone of a sample document and the resulting segmentation of two large blocks (in red). The very small distances between pairs of separating lines (in green) correspond to the height of the estimated inter-line spaces. In the following, pairs of separating lines will be replaced by a unique separating line at median position.

In Figure 15 a and b, we show the resulting segmentation of large blocks extracted from the ancient document in Fig. 10. There are many multi-overlapping and multi-touching cases represented by red zones. All blocks have been successfully segmented, even the largest one (bottom left) which has been segmented into seven average blocks. Zones including large blocks, lying in neighboring strips, are also segmented correctly.

### 4.3 Block segmentation of WSD documents

Documents classified as WSD should include no large blocks. Consequently, we could divide them into only one strip. However, a classical projection is not efficient enough for segmenting the document into text lines since diacritical points would produce wrong lines. Possible document

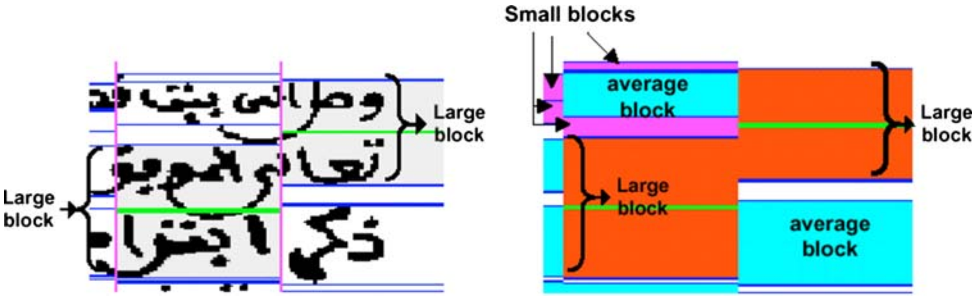


**Fig. 14** Segmentation of two large blocks

misclassification could also occur if documents include few overlapping or touching components. Consequently, we prefer to divide documents classified as WSD into four strips. This enhances the classical projection-based method.

Our method proceeds as follows: we search for the blocks within each of the four strips. The K-means algorithm classifies them into two classes; the resulting small blocks include diacritical marks, the average blocks include the main writing stream. Figure 16 shows an example of block classification of a WSD document.

## 5 Neighboring analysis of Block Covering for building text-lines

### 5.1 Block labeling

Neighboring analysis of Block Covering aims at labeling each block by assigning it to a single text-line. This process is iterative and adjacent strip pairs ($c_j$ and $c_{j+1}$) are first processed at each iteration from left to right. All blocks in strip *cj* which belong to a confident neighborhood configuration are labeled at the end of each iteration. The process iterates on strip pair ($c_{j+1}$, $c_{j+2}$) until reaching pair ($c_{N-1}$, $c_N$). Then a second process occurs from right to left from pair ($c_N$, $c_{N-1}$) to pair ($c_2$, $c_1$) in order to label remaining unlabeled blocks.

There are three configurations A to C for grouping two blocks through adjacent strips into the same text-line. These configurations are based on the vertical position relative to the two blocks (Fig. 17). Let $B_{j,k}$ be the *k*th block in strip *j* to be analyzed. When blocks are in configuration A, they are never grouped together. Grouping occurs for configurations B and C. But configuration B subdivides into configurations B1 and B2 when a block in one strip has more than one neighbor in the adjacent strip (Fig. 18). The possible assignment of these other neighbors has to be taken into consideration. A confident assignment

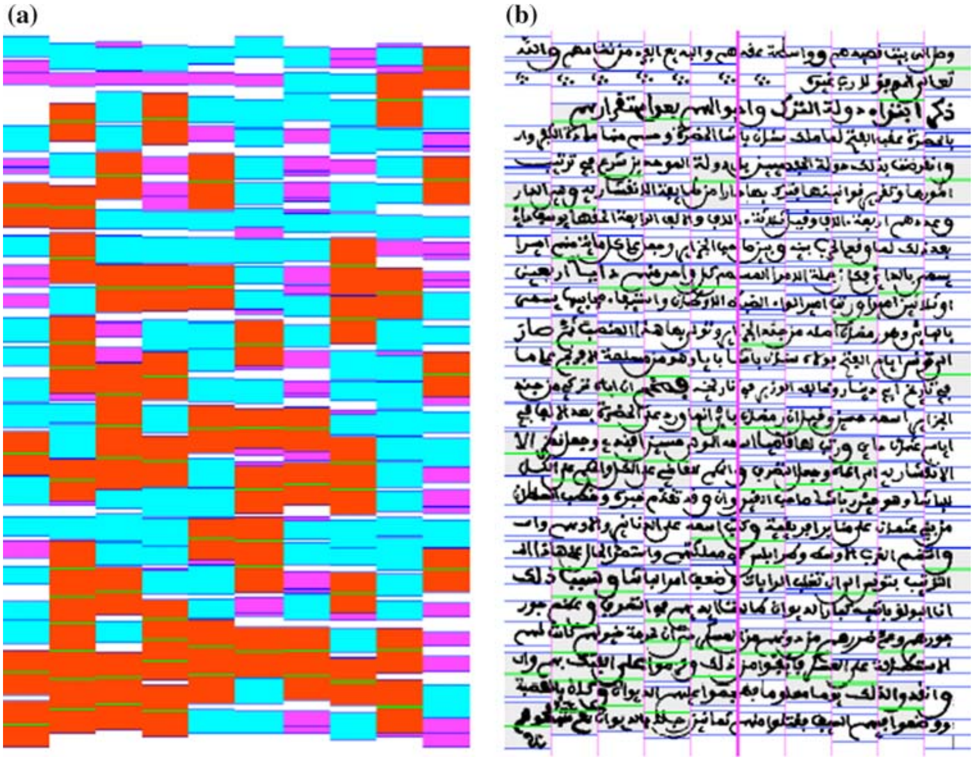


**Fig. 15** Large block segmentation of a sample document. **a** Large blocks (*in red*) are divided into several average blocks as necessary. **b** Sample document, large blocks (*in grey*) and separating lines

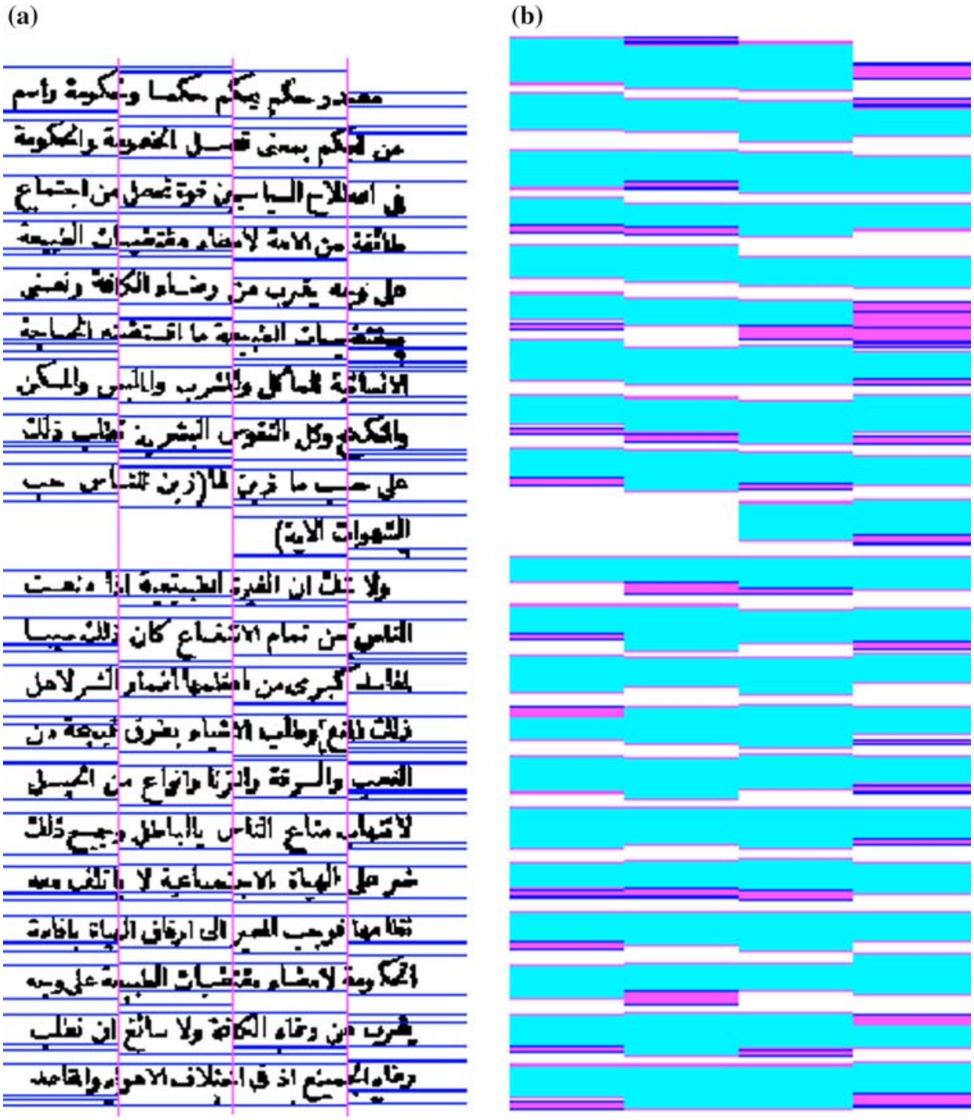


**Fig. 16** Block extraction and classification of WSD documents: small blocks (*purple*) and average blocks (*blue*)

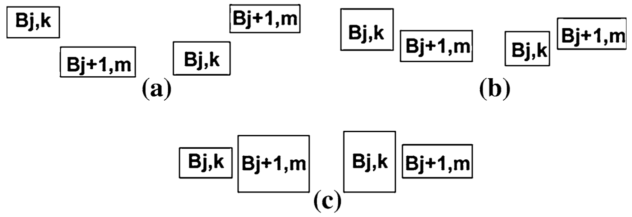


**Fig. 17** Grouping configurations for two blocks in adjacent strips. *Case A* no grouping. *Case B* $B_{j,k}$ is grouped with $B_{j+1,m}$ ($B_{j,k}$ and $B_{j+1,m}$ have only one neighbor on the right and on the left, respectively). *Case C* $B_{j,k}$ is grouped with $B_{j+1,m}$

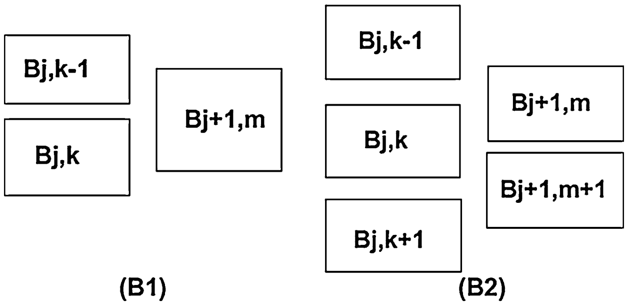


**Fig. 18** Case B subdivides into B1 and B2 in case of multiple neighbors. $B_{j,k}$ is grouped with $B_{j+1,m}$

depends on whether the other blocks are already labeled or not and decision depends on the overlapping degree of the blocks to label.

The text-line segmentation of the TSD document in Fig. 10 is shown in Fig. 19a. Two adjacent lines are shown in different colours. The only segmentation defects are due to large block segmentation: the block separating lines are passing through characters and small character parts are assigned to the adjacent line. But for WSD documents (Fig. 19b), there are no such defects as there are no large blocks and thus no separating lines.

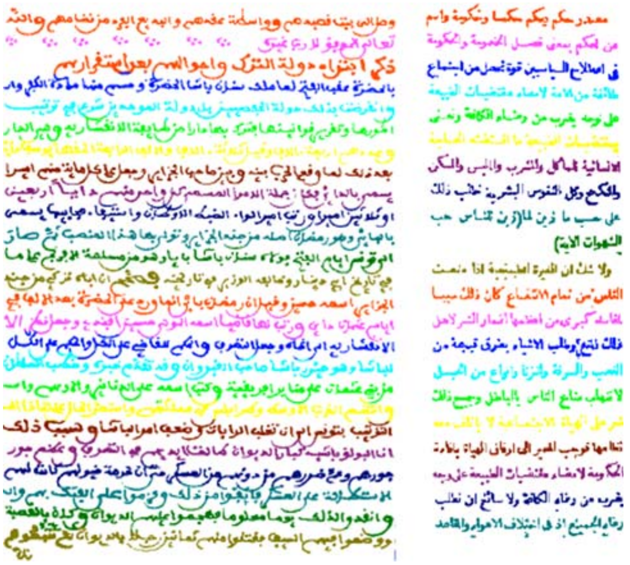


**Fig. 19** Text-line segmentation of the TSD document in Fig. 12a and of the WSD document in Fig. 16b

### 5.2 Post-processing of under and over-segmented blocks

When an average block is misclassified into a large block, it is generally over-segmented into two blocks. This is the most common case for over-segmentation. Other cases of segmentation errors are due to correctly classified large blocks segmented either into too few blocks (under-segmentation) or into too many blocks (over-segmentation) as shown in Fig. 20a and b.

That is why we enhance the previous grouping algorithm with the ability to detect incorrect segmentations and to correct them in most cases. For instance, in Fig. 20a a large block has been subdivided into two blocks $B_{j+1,m}$ and $B_{j+1,m+1}$. Let's assume that the labeling of blocks $B_{j,k}$, $B_{j,k+1}$ and $B_{j,k+2}$ has already occurred when processing previous strip pair $(j-1, j)$ and that labels are different for each block. The under-segmentation case is detected and block $B_{j,k+1}$ is not grouped to any block in strip $j+1$ even if it could be grouped to one of its two neighboring blocks in strip $j$. When such a case is detected, the large block is re-segmented with an additional constraint: the number of its compound average blocks is incremented by one. Grouping is then performed again to verify the accuracy of the new segmentation.

Figure 20b shows an over-segmentation case: a large block is segmented into three average blocks. When processing strip $j+1$, $B_{j+1,m+1}$ is an isolated block which is not grouped to any block in strip $j$ and cannot be grouped to any block in strip $j+2$. This case is detected and corrected in a similar way as previously.

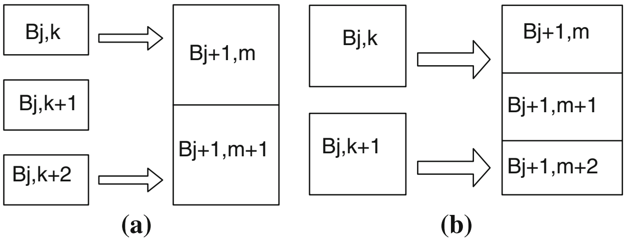


**Fig. 20** Cases of under segmentation (**a**) and over segmentation (**b**) of large blocks

**Table 2** General statistics

| | Documents | Training data set | Test data set | Lines | Text blocks |
|---|---|---|---|---|---|
| **TSD** | 120 | 21 | 99 | 2,842 | 28,083 |
| **WSD** | 40 | 23 | 17 | 805 | 7,239 |

## 6 Experiments

The text-line segmentation method presented above is implemented in C++ language. Our Arabic documents are extracted from the Tunisian Historical Archives [1] collection of digitized documents. This collection includes both handwritten and printed documents. We selected 160 pages from various periods and subjects (see Table 2). Spaces between lines vary from one document to another, but in each document there is only one main text-line direction. Selected documents are in color and their resolution is 300 dpi. They are binarized, restored and saved as TIFF compressed files. The main text-line direction is detected using the Hough transform and the document is rotated according to this direction.

The document type has to be estimated: a subset of 44 representative documents is used for training. They are described through the fractal analysis by the two features of the Block Counting method: the block dimension $D$ and $\Delta \log H$, the logarithm of the ratio between the two Hausdorff measurements. The fuzzy C-means unsupervised method yields the following classification results (see Table 3): 19 documents are classified as TSD, 23 as WSD and two rejected as ambiguous and removed from the training set. Pairs ($D$, $\Delta \log H$) from all samples of one class

**Table 3** TSD/WSD classification

| | Learning set accuracy (Fuzzy C-means)+(Mahalanobis) | Test set (Mahalanobis) |
|---|---|---|
| **TSD** | 21/21 | 99/99 |
| **WSD** | 23/23 | 17/17 |

**Table 4** Block classification

| | Text blocks | Optimal cutting out | Small and mean blocks | Large blocks | Rate (%) |
|---|---|---|---|---|---|
| **TSD** | 28,083 | 8–18 | 22,062/22,243 | 6021/5840 | 97.0 |
| **WSD** | 7,239 | 4 | 7,239/7,239 | 0/0 | 100 |

**Table 5** Large block segmentation of TSD documents

| Large blocks | Misclassified thus wrongly segmented | Well classified but wrongly segmented | Total of wrongly segmented | Rate (%) |
|---|---|---|---|---|
| 6,021 | 181 | 120 | 301 | 5.0 |

**Table 6** Block assignment to text-lines correct assignment rate

| | Before post-processing | After post-processing |
|---|---|---|
| TSD | 94.2% | 96.5% |

are used to build the statistical parametric model of this class (mean vector and covariance matrix).

For classifying a test document, the two Mahalanobis distances are computed and the smaller distance yields the estimated document type. The two ambiguous documents removed from the training set are now classified as TSD documents. This result confirms the relevance of the counting block method. The test set includes 116 documents subdivided into 17 documents of type WSD and 99 documents of type TSD. The document type is assessed through visual inspection. All WSD and TSD documents are classified into the correct class, which assesses the efficiency of the counting block method and of the two features extracted.

We now evaluate the text-line segmentation process on the whole set (120 samples) of TSD documents (Tables 4, 5, 6). These documents include a total amount of 2,842 text-lines. Block classification is achieved by dividing documents into a number of strips determined automatically, which varies from 8 to 18 according to the document processed. A volume of 28,083 blocks is thus generated. Some line fragments are misclassified into large blocks because their height is much larger than most average blocks. However, the correct classification rate of large blocks is rather high: 97% accuracy is obtained and assessed through visual inspection. Visual inspection is performed by a native Arabic writer who verifies block assignments from their colours: for instance in Fig. 15a, large blocks are in red and correspond to the grey blocks superimposed to the writing in Fig. 15b. More precisely 6,021 large blocks are detected: from them 181 (3%) are misclassified average blocks.

The correct segmentation rate for the 6,021 large blocks is 95%. The 5% error rate is due to misclassified average blocks (3%) and to incorrectly segmented real large blocks (2%). The rate of correct block assignment to text-lines is 94.2% before post-processing. After post-processing (correcting under and over segmentation cases), the rate increases to 96.5% and assessed through visual inspection; to each line corresponds one color and the colors of two neighboring lines are different (Figs. 19, 21). This last rate expresses the global correct text-line segmentation rate. Results on sample documents are shown in Fig. 21.

The analysis of failures shows two main sources of errors. First, the unsupervised block classification can misclassify blocks close to class frontiers. Second, block-assignment rules are based on block-to-block distances: errors can occur when these distances are very close. More generally, errors are due to document irregularities: writing height and inter-line spacing highly variable, as well as defects remaining after pre-processing (stains, marks, underlining,…).

## 7 Conclusions

We have presented a novel approach for text-line segmentation applied to ancient Arabic documents. A document, printed or handwritten, is first restored and binarized. Documents are assumed to include only one main text-line direction. But text-lines may be tightly spaced, undulate, overlap or touch each other through several adjacent lines. The processing is based on a Block Covering technique and we exploit this technique in three steps: Block Counting analysis for document type classification, statistical block height analysis for block classification and neighboring analysis for building text-lines.

Document type, which depends on line spacing, is identified with high accuracy through fractal analysis and a fuzzy C-means classifier. For tightly spaced documents (TSD) documents, the optimal Block Covering is determined by a statistical index based on intra and inter-class density. Large block segmentation uses the set of average blocks and inter-line spacing specific to the document under study. Neighboring analysis uses logical rules according to the list of block configurations. When the height of average blocks has large variability, large blocks may be under or over segmented. These cases are detected and corrected.

Results obtained on a set of 120 TSD documents show a 96.6% correct segmentation rate into text-lines. The 40 widely spaced documents (WSD) documents are correctly segmented with a 100% rate. It can be noted that the

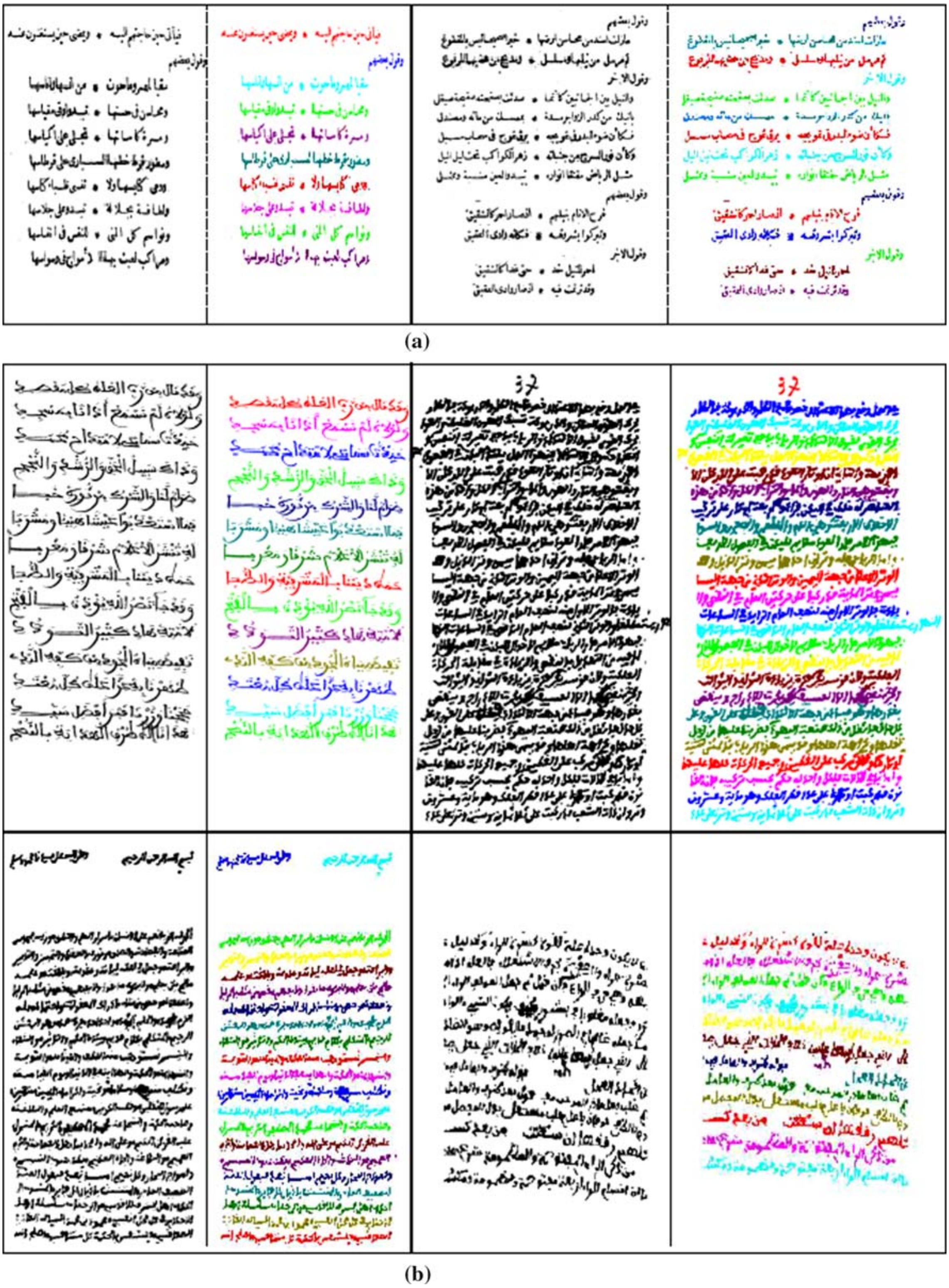


**Fig. 21** Text-line segmentation results. Original documents (black/white images), segmentation into text-lines (colored image). **a** WSD documents. **b** TSD documents

method does not use any implicit parameter or manual adjustments, but the height of writing is constrained to be rather homogenous.

The block-covering method may fail when blocks are misclassified. For instance a block classified as large must result from overlapping or multi-touching components and not from a writing component of whose height is much larger than the average writing height.

The method proposed here is included in a system which performs pre-processing on document images. The quality of the pre-processing directly influences global performance. All document irregularities (skew, stains, underlines, circle marks around words, inadvertent markings across text-lines, stamp marks, etc…) must be corrected.

## Appendix 1. Composing density between and with clusters (CDbw) criterion

CDbw is defined as the product:

$$CDbw(vsn) = intra\_den(vsn) * sep(mvsn)$$

*intra_den* expresses the quality of intra-class clustering. *sep* is the cluster separation measure.

We calculate now the terms of the product:

For a given number of vertical strips *vsn* = *1/r*, let $V_i = \{v_{i1}, v_{i2}, \cdots, v_{in_i}\}$ be the set of blocks of the *i*th class, and $n_i$ be the number of blocks in this class. The standard deviation *stddev* (*i*) of the *i*th class is defined as:

$$stddev(i) = \sqrt{\sum_{k=1}^{n_i} \frac{(h_{ik} - m_i)^2}{(n_i - 1)}}$$

with $h_{ik}$ being the height of the *k*th block of the *i*th class and $m_i$ being the average height of the blocks of the *i*th class.

The average *stddev* is:

$$stddev = \sqrt{\sum_{i=1}^{3} \frac{\|stddev(i)\|^2}{3}};$$

The quality of intra-class clustering, denoted by *intra_den* is defined as:

$$intra\_den(vsn) = \frac{1}{3}\sum_{i=1}^{3}\sum_{j=1}^{n_i} density(v_{ij});$$

with $density(v_{ij})$ defined as:

$$density(v_{ij}) = \sum_{l=1}^{n_i} f(v_{il}, v_{ij});$$

and $f\,(v_{il},v_{ij})$ defined as:

$$f(v_{il}, v_{ij}) = \begin{cases} 1 & \text{if } \|h_{il} - h_{ij}\| \leq stddev \\ 0 & \text{otherwise} \end{cases}$$

The interclass density *Inter_den* is defined as the number of blocks being in the close neighborhood of several classes. This density should be very low. It is defined as:

$$Inter\_den(vsn) = \sum_{i=1}^{3}\sum_{\substack{j=1\\ j\neq i}}^{3} \frac{\|m_i - m_j\|}{\|stddev(i) + stddev(j)\|} \times density(u_{ij});$$

$u_{ij}$ is a virtual block of height $h_{ij} = (m_i+m_j)/2$

*density* ($u_{ij}$) is defined as:

$$density(u_{ij}) = \sum_{k=1}^{n_i+n_j} f(v_k, u_{ij})$$

with $v_k$ belonging to the union set of blocks of classes *i* and *j*.

$f\,(v_k,u_{ij})$ is defined as:

$$f(v_k,u_{ij}) = \begin{cases} 1 \text{ if } \|h_k - h_{ij}\| \leq (\|stddev(i)\| + \|stddev(j)\|)/2, \\ 0 \text{ otherwise} \end{cases}$$

The cluster separation measure is defined as:

$$sep(vsn) = \sum_{i=1}^{3} \sum_{\substack{j=1\\ j\neq i}}^{3} \frac{\|m_i - m_j\|}{1 + Inter\_den}$$

## References

1. http://www.bibliotheque.nat.tn; http://www.archives.nat.tn
2. Kolcz A, Alspector J, Augusteyn M, Carlson R, Viorel Popescu G (2000) A line-oriented approach to word spotting in handwritten documents. Pattern Anal Appl 3:155–168
3. Lakshmi CV, Patvardhan C (2004) An optical character recognition system for printed Telugu text. Pattern Anal Appl 7:190–204
4. Likforman-Sulem L, Zahour A, Taconet B (2007) Text line segmentation of historical documents: a survey. IJDAR 9(2–4): 123–138
5. Abuhaiba ISI, Datta S, Holt MJJ (2005) Line extraction and stroke ordering of text pages. In: Proceedings of ICDAR'05, Seoul (South Korea), pp 390–393
6. Oztop E, Mulayim AY, Atalay V, Yarman-Vural F (1999) Repulsive attractive network for baseline extraction on document images. Signal Process 75:1–10
7. Li Y, Zheng Y, Doermann D (2006) Detecting text lines in handwritten documents. In: Proceedings of ICPR'06, Hong Kong, pp 1030–1033
8. Khorsheed MS (2002) Off-Line Arabic character recognition—a review. Pattern Anal Appl 5:31–45
9. Lorigo LM, Govindaraju V (2006) Off-line Arabic handwriting recognition—a survey. IEEE PAMI 28(5):712–724
10. Arivazhagan M, Srinivasan H, Srihari S (2007) A statistical approach to line segmentation in handwritten documents. In: Proceedings of Document Recognition and Retrieval XIV, IST&SPIE, San Jose
11. Zahour A, Taconet B, Mercy P, Ramdane S (2001) Arabic handwritten text-line extraction. In: Proceedings of ICDAR'01, 10–13 Sept., Seattle, USA, pp 281–285
12. Amin A, Fischer S (2000) A document skew detection method using the Hough transform. Pattern Anal Appl 3:243–253
13. Boussellaa W, Zahour A, El Abed H (2006) A concept for the separation of foreground/background in Arabic historical manuscripts using hybrid methods. In: Ioannides M, Arnold D, Niccolucci F, Mania K (eds) Proceedings of the 7th internat.

symp. on virtual reality, archaeology and cultural heritage VAST, pp 1–5

14. Dodson M, Kristensen S (2004) Hausdorff dimension and diophantine approximation. Fractal geometry and applications: a jubilee of Benoit Mandelbrot. Part 1. Proceedings of Sympos. Pure Math., vol 72, Part 1, Amer. Math. Soc., Providence, pp 305–347
15. Boulétreau V, Vincent N, Emptoz H, Sabourin R (2000) How to use fractal dimension to qualify writings and writers. Fractals Complex Geometry Patterns Scaling Nat Soc 8(1):85–98
16. Vincent N, Emptoz H (1995) A classification of writing based on fractals. In: Novak MM (ed) Fractal reviews in the natural and applied sciences. Chapman & Hall, London, pp 320–331
17. Ben Moussa S, Zahour A, Alimi MA, Benabdelhafid A (2005) Can fractal dimension be used in font classification. In: Proceedings of ICDAR 2005, Seoul (South Korea)
18. Hausdorff F (1919) Dimension und äußeres Maß. Math Ann 79:157
19. Wu S, Chow TWS (2005) Clustering of the self-organizing map using a clustering validity index based on inter and intra-cluster density. Pattern Recognit 37(2):175–188
20. Falconer K (1997) Techniques in fractal geometry. Willey, New York, ISBN 0–471-92287-0