IJDAR (2010) 13:243–259
DOI 10.1007/s10032-010-0123-7

ORIGINAL PAPER

A combination of features for symbol-independent writer identification in old music scores

Alicia Fornés · Josep Lladós · Gemma Sánchez · Xavier Otazu · Horst Bunke

Received: 28 October 2009 / Revised: 24 May 2010 / Accepted: 8 June 2010 / Published online: 27 June 2010

Abstract The aim of writer identification is determining the writer of a piece of handwriting from a set of writers. In this paper, we present an architecture for writer identification in old handwritten music scores. Even though an important amount of music compositions contain handwritten text, the aim of our work is to use only music notation to determine the author. The main contribution is therefore the use of features extracted from graphical alphabets. Our proposal consists in combining the identification results of two different approaches, based on line and textural features. The steps of the ensemble architecture are the following. First of all, the music sheet is preprocessed for removing the staff lines. Then, music lines and texture images are generated for computing line features and textural features. Finally, the classification results are combined for identifying the writer. The proposed method has been tested on a database of old music scores from the seventeenth to nineteenth centuries, achieving a recognition rate of about 92% with 20 writers.

A. Fornés (✉) · J. Lladós · G. Sánchez · X. Otazu
Computer Vision Center, Department of Computer Science,
Universitat Autònoma de Barcelona, Edifici O,
08193 Bellaterra, Spain
e-mail: afornes@cvc.uab.es

J. Lladós
e-mail: josep@cvc.uab.es

G. Sánchez
e-mail: gemma@cvc.uab.es

X. Otazu
e-mail: xotazu@cvc.uab.es

H. Bunke
Institute of Computer Science and Applied Mathematics,
University of Bern, Neubrückstrasse 10,
3012 Bern, Switzerland
e-mail: bunke@iam.unibe.ch

Keywords Writer identification · Graphics recognition · Optical music recognition

1 Introduction

Analysis of historical documents has attracted growing interest in the last years. The aim is the conversion of these documents into digital libraries, helping in the diffusion and preservation of artistic, technical and cultural heritage. An interesting application in this field is writer identification, i.e. the classification of the document in terms of the writer.

Writer identification (w.i.) is focused on the identification of the author of a piece of handwriting among a set of writers. Traditionally, the off-line approaches for writer identification in text documents can be divided in text-dependent and text-independent. In the first group of methods [1] the individual characters or words are compared with a known transcription. These approaches require either a prior knowledge of the textual content, or the recognition of the handwritten text. Contrary, in the second group, the writer identification can be performed with the meaning of the text being unknown [27,30].

There is an active research community concerning handwritten text documents, specially in forensic analysis, including not only writer identification [6,28] but also writer verification [2] and dating manuscripts [12]. Nevertheless, the identification of the author of a handwritten document of graphical contents is still a challenge. Music scores are an example of hybrid documents (because they contain both graphics and text) with an important research community working in this topic. Concerning historical music documents, an important application is the retrieval of anonymous documents for their analysis, and the validation of their authorship. In fact, musicologists identify the writer (or composer) through a deep analysis of the music score, analyzing not only the handwriting style, but also the rhythm, melody, and harmony of the composition. Similarly, an automatic writer identification approach should recognize and analyze the whole composition. Unfortunately, this task becomes extremely complex, because an Optical Music Recognition system should recognize the music notation of very complex documents, coping not only with the variability of the handwriting style, but also with the degradation of historical documents. For that reason, an automatic writer identification approach for old music scores is still required for helping musicologists in their human task, which is time consuming and prone to errors. It must be said that although an automatic writer identification approach will never substitute the musicologists in this task, it will help them to discard an important amount of writers of the database. In this context, the handwriting style of the hand-drawn music symbols can be used for determining the authorship of a music score.

To the best of the authors' knowledge, only one approach to writer identification in music scores has been proposed [5,14,22]. The authors introduce a method that analyzes the music score and then extracts some features that represent structural information of the music symbols and notes. The process of automated identification requires a definite level of handwriting content understanding, in other words, the recognition of music symbols. Only once the object features are given in the form of structural descriptions, the matching algorithm can determine the identity of the structure for the following writer identification step. However, due to the difficulties in the automatic recognition of hand-drawn music symbols, only the staff removal and a graphical primitive analysis has been performed. The next steps, namely object recognition and writer identification, are still not implemented. For this reason, no results are shown in the papers, and as far as we know, this project has not been continued.

Obviously, the recognition of old handwritten music scores is extremely difficult, not only because of the recognition of hand-drawn symbols, but also because of paper aging and degradation. In this work we propose to avoid the recognition step, obtaining a *symbol-independent* writer identification method. Consequently the system is faster and more robust, avoiding the dependence on a good symbol recognizer. In fact, we have borrowed some ideas from the text-independent writer identification approaches for text and applied them to old musical scores, where instead of letters of the alphabet, music symbols are analysed.

Most compositions in the last centuries were sacred music, and consequently, contain lyrics (text) for the chorus and the solists. In these scores, the writer identification methods for handwritten text documents could be applied to lyrics. However, there are two main reasons for avoiding the use of lyrics. Firstly, it has been noticed that in some cases, the writer of the lyrics and the writer of the music notation is not the same; and secondly, many music compositions are for instruments only, and consequently, they contain no text nor lyrics. For this reason, a method which performs the writer identification based on the extraction of features only from music symbols is required.

In [10] and [11] we presented two different approaches for writer identification in old music scores. The first one, inspired by writer identification approaches applied to text lines, extracts features for every music line. Once the musical score is transformed into individual normalized handwritten music lines, 100 features are computed for every line, including basic measurements (such as slant and width of the writing), connected components, lower and upper contour of the line and fractal features. The second method was inspired by holistic writer identification approaches that analyze the whole music score image as a texture. Textural features are then extracted rather than focusing on a set of predefined local features. After the preprocessing of the music sheet, music texture images are generated from music symbols, and then Gabor features and Gray-Scale co-ocurrence matrices (GSCM) are extracted and used as features. In both approaches, the classification was performed using the k-NN classifier. The experimental results showed that every individual approach achieved quite good identification rates (73 and 76%, respectively), but in some cases, the information extracted was not enough for a reliable writer identification.

In the current paper we propose the combination of features for performing writer identification in graphical documents, such as old music scores. The proposed architecture (see Fig. 1) combines two writer identification approaches, improving the global identification rate. Firstly, the input image is preprocessed in order to binarize, deskew and remove the staff lines and lyrics. The resulting image is then the input for the two writer identification approaches. The first method performs specific preprocessing and normalization operations in order to obtain music lines; then it generates long lines and extracts 98 line features. The second method generates texture images and extracts textural features, using Gabor filters and GSCM. Once the features have been extracted, both approaches individually apply a k-NN classifier, and then we combine the results for the final classification. For this purpose, the combination of results is performed using the Majority Voting or Borda Count method, so that each element (line or texture) gives votes to the nearest neighbor classes [21]. Finally, the votes of the two approaches are taken into account for the final identification of the writer.

The remainder of the paper is structured as follows. The preprocessing step is presented in Sect. 2, in which the music score is binarized, and staffs and lyrics are removed. In Sect. 3 the first feature extraction method is fully described, consisting in normalizing the music line and extracting line features. Section 4 presents the second feature extraction method,

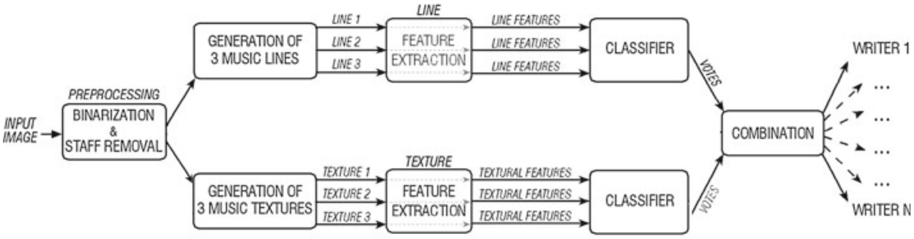

Fig. 1 Stages of the ensemble architecture for combining the two writer identification approaches

which generates texture images and extracts textural features. Section 5 presents the combination of the two writer identification approaches. Experimental results are presented and discussed in Sect. 6. Finally, Sect. 7 concludes the paper and proposes future work.

2 Preprocessing

The preprocessing phase consists in binarizing the image, removing the staff lines and lyrics. First, the image is binarized using Niblack, and deskewed using the Hough Transform. Then, a coarse staff approximation is obtained by using median filters (with a horizontal mask), and joining the resulting horizontal segments. Afterwards, a contour tracking process is used for an accurate detection of the staff lines. Finally, staff lines and lyrics are removed. The process is outlined in this section. For further details, see [9].

The input gray-level scanned image (at a resolution of 300 dpi) is first binarized with the adaptive binarization technique proposed by Niblack [24]. Then, filtering and morphological operations are applied to reduce noise. Afterwards, the image is deskewed in order to make the recognition of staff lines easier. For this purpose, the Hough Transform is used to detect lines and obtain the orientation of the each staff, which is independently rotated if necessary.

Once the image has been binarized and deskewed, the next step consists in removing the staff lines. For the task of writer identification, the staff lines are useful only if they are drawn by hand. In most of the music sheets of our database, however, they are printed. For that reason, staff lines are removed from the score. The extraction of staff lines is difficult (even if they are printed) because of paper degradation and the warping effect. For that reason, a robust system for detecting staffs is required, coping with distortions and gaps in staff lines.

The steps for staff removal are the following. Firstly, a coarse staff approximation is obtained using horizontal runs as seeds to detect a segment of every staff line. This approximation is computed by applying median filters (with a horizontal mask) to the skeleton of the image. Remaining objects are only segments of the staff lines and horizontally-shaped symbols. Afterwards, staff lines are reconstructed, and each segment is discarded or joined with others according to its orientation, distance and area. Secondly, a contour tracking process is performed from left to right and right to left, following the best fitting path according to a given direction. In order to cope with gaps in staff lines and to avoid deviations (wrong paths) in the contour tracking process, the coarse staff approximation mentioned above is consulted. Then, those segments that belong to the staff lines (their width is similar to the average of the width of staff lines, which has been previously computed) are removed. Finally, those elements that are not connected to any staff line are considered as lyrics and removed from the image. For further details, see [9].

3 Writer identification approach based on music lines

The first writer identification (w.i.) approach is inspired by the text-independent writer identification methods applied to text lines, such as the approach proposed by Bouletreau et al. [3], which uses fractal features; the method proposed by Schlapbach and Bunke [29], which uses Hidden Markov Models; and Schomaker and Bulacu [31], which propose the use of connected component contours and edge-based features.

Concretely, our method is based on the one proposed by Hertel and Bunke's [17], which computes 100 features from text lines, using connected components, enclosed regions, and fractals, among others. It is an extension of the approach proposed by Marti et al. [23], in which 12 features were used. In our first approach (see [10]), we generated as many music lines as staffs were found in the music score, and 100 line features were computed. Contrary, in the current approach we generate three long music lines (see Fig. 2), and then compute 98 features from these lines. The individual steps of the method are described next.

3.1 Normalization

The information about the location of staff lines previously obtained is used for segmenting the music sheet into lines. Afterwards, the lines must be aligned with respect to a horizontal reference line. This step will be called normalization.

The normalization typically performed in handwritten text can not be applied here, because in musical scores, the height of every music line will vary depending on the melody of the composition. In music notation, notes are located further up

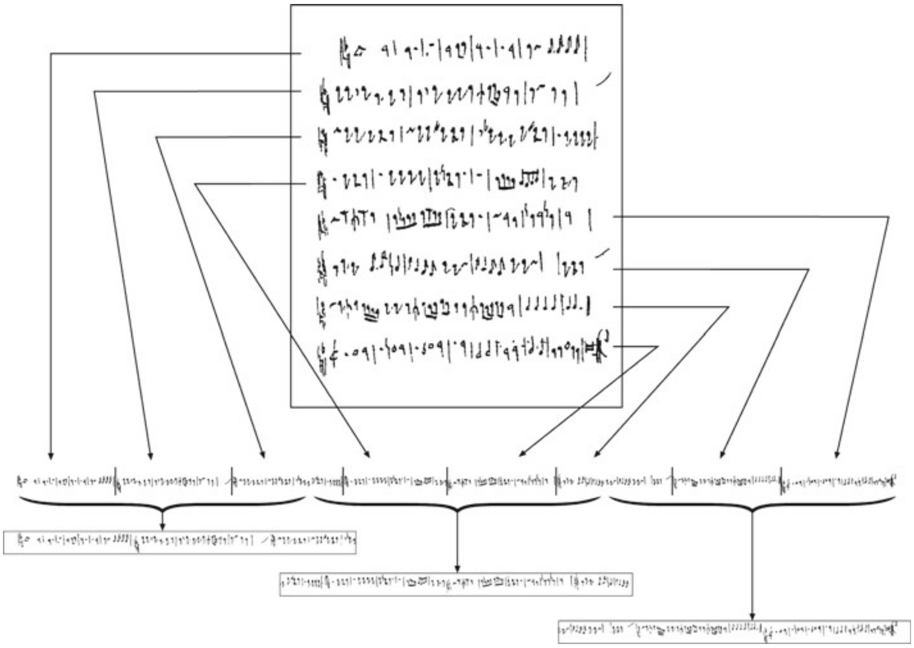

Fig. 2 Obtaining *three music lines* for each page. Once the *staff lines* are removed, each *music line* is normalized and joined in a single music line. Then, this line is split in three lines of approximately equal length, which will be the input music lines

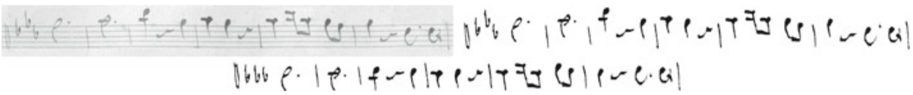

Fig. 3 Preprocessing step: original music line in *gray scale*, binarized music line (without *staff lines*), and *normalized line*, in which all the *music symbols* are aligned with respect to a *horizontal reference line*

and down in the staff for reaching higher or lower frequency. Therefore, melodies with both treble and bass notes would result in a line with a larger height. This fact can be confusing for the writer identification system, which could wrongly identify heights of large extend in lines (melodies with bass and treble notes) as a typical feature of a specific writer. Since our goal is to obtain a melody independent writer identification method, it must not rely on melody information. For this purpose, the music notes are rearranged with respect to a horizontal reference line. Thus, the normalization step computes the centroid of every connected component of the line, and uses this centroid for aligning the component with a horizontal reference line (see Fig. 3).

To obtain the music line that will be used for the computation of features, the first option consists in generating as many music lines as staffs are present in the music sheet. Thus, each music staff line was preprocessed, normalized and used as an input music line. Using this option the number of staffs in each music sheet will indicate the number of music lines that will be generated. But, with this option, one can easily introduce noise to the writer identification process, because some music sheets contain short staffs. In this case, music lines do not contain enough music symbols to compute reliable features.

For avoiding this problem, each music page will generate exactly three long music lines, independently of the number of music staff lines that it contains. With this option, after the preprocessing and normalization steps, all the music lines will be joined in one single music line. Afterwards, this long music line will be split in three equal parts, which will be the three input music lines for the feature extraction stage (see Fig. 2). It must be said that we have decided to generate exactly three lines after analyzing the amount of music symbols that usually appear in the music sheets. Nevertheless, a long music line could also be split in a different number of lines, depending on the amount of music symbols of music sheet.

3.2 Feature extraction from music lines

Once the musical score is transformed into normalized handwritten individual music lines, 98 features are computed for every line. Previous work by Hertel and Bunke [17] was performed for writer identification in handwritten text documents, in which 100 features where extracted. These features include basic measures (such as slant and width of the writing), connected components, enclosed regions, lower and upper contour of the line and fractal features.

The basic idea is to use 98 of the 100 Hertel's features, adapting them to music lines, within the specific normalization described in the previous section. The two features that have been omitted in our approach are the enclosed

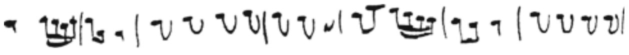

Fig. 4 Example of a music line without any *closed loops*

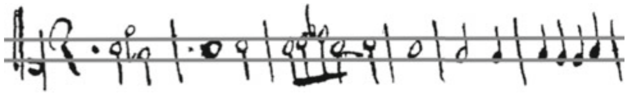

Fig. 5 The three main zones detected in a *music line*: Upper, Middle and LowerZone

regions measures, which measure the roundness of the loops. These measures are very useful in handwritten text, because closed loops can be of circular, elliptical or rectangular shape, depending on the writing style. Contrary, the probability of finding closed loops in music notation is low (see Fig. 4). In fact, just a few music symbols contain loops (e.g. whole and half note, and accidentals), and in addition, these symbols are not frequent, specially in fast rhythms and tempos. Consequently, they appear only in a small subset of music lines, and for this reason, they can not be used for writer identification.

A brief description of the features used in the work described in this paper is given below. For a full description we refer to [17] and [23].

3.2.1 Basic measures

The basic features taken into account are the following: the writing slant, the height of the main three zones and the width of the writing.

For obtaining the slant angle, the contour of the writing is computed and an angle projection is created by accumulating the different angles along the contour. All angles are weighted by the length of the corresponding line. From the histogram, the mean and standard deviation are computed.

The three writing zones are called the UpperZone, the MiddleZone and the LowerZone. They are determined by the top line, the upper baseline, the lower baseline and the bottom line. To determine these lines, a horizontal projection histogram of the music line is computed, and an ideal histogram with variable position of the upper baseline and the lower baseline is matched against this histogram. Then, the following ratios (for avoiding absolute values) are used as features: U/M, U/L and M/L, where U is the height of the UpperZone, M is the height of the MiddleZone and L is the height of the LowerZone (see Fig. 5).

The width of the writing is obtained by selecting the row with most black-white and white-black transitions. In this row, the median m_l of the lengths of the foreground runs is computed. Finally, this value is used for obtaining the ratio, M/m_l (where M is the height of the Middlezone), which will be used as a feature.

3.2.2 Connected components

Some authors write musical notes in a continuous stroke while others break it up into a number of components. Thus, from every binary image of a line of music, connected components are extracted. Then, the average distance between two successive bounding boxes is computed. The system computes the average distance of two consecutive connected components and also the average distance between the graphical primitives belonging to the same connected component. Moreover, the average, median, standard deviation of the length of the connected components are used as features.

3.2.3 Lower and upper contour

A visual analysis of the upper and lower contours of the music lines reveals that they differ from one writer to another. Some writings show a rather smooth contour whereas others are pointed with more peaks. This kind of information is utilized for writer identification.

For selecting the lower and the upper contour of a line, gaps must be removed, and discontinuities in the y-axis are eliminated by shifting these elements along the y-axis (see Fig. 6). Once the continuous lower and upper contour (called characteristic contours) are obtained, the following features are extracted: slope of the characteristic contour (obtained through linear regression analysis), the mean squared error between the regression line and the original curve, the frequency of the local maxima and minima on the characteristic contour (if m is the number of local maxima and l is the number of local minima, then the frequency of local maxima is m/l and the frequency of local minima is l/m), the local slope of the characteristic contour to the left of a local maximum within a given distance, and the average value taken over the whole characteristic contour. The same features are computed for the local slope to the right of a local maximum, and for local minima.

3.2.4 Fractal features

The idea proposed in [3,4] is to measure how the area A of a handwritten line grows when a morphological dilation operation is applied on the binary image. The line is first thinned, and the dilation is performed using different kernels (disks of radius η for obtaining information invariant to rotation).

For each of these kernels, the area $A(X_\eta)$ of the dilated writing X_η is measured. The fractal dimension $D(X)$ is defined by

$$D(X) = \lim_{\eta \to 0} \left(2 - \frac{\ln A(X_\eta)}{\ln \eta} \right) \tag{1}$$

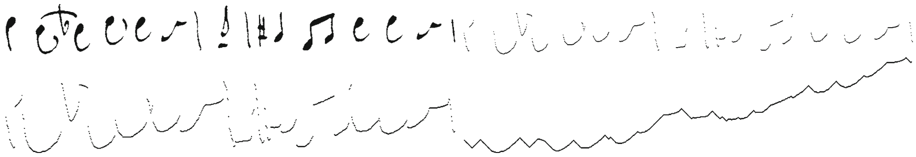

Fig. 6 Discontinuities are removed before computing the contour features. The *first row* shows the music image, the *second one* shows the lower contour extracted, then the lower contour after gap elimination, and finally, the lower contour after the elimination of discontinuities in the y-direction

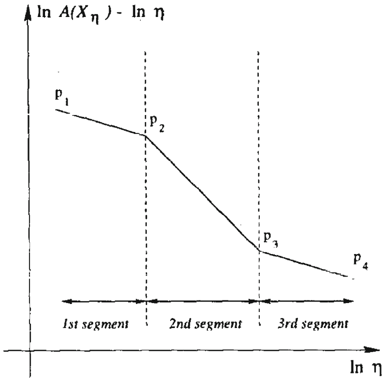

Fig. 7 Fractals: Approximation of the evolution *graph* by *three straight lines* (extracted from [23])

Then, we obtain the evolution graph plotting y as a function of x

$$x = \ln \eta; \quad y = \ln A(X_\eta) - \ln \eta \tag{2}$$

This function is approximated by three straight lines (see Fig. 7). The points $p_1, \ldots, p_4$ are found by minimizing the square error between the three line segments and the points of the evolution graph. Finally, the slopes of these three characteristic straight line segments are computed and used as features.

In addition to three disks kernels, 18 ellipsoidal kernels are used for getting more features. These ellipses are defined with increasing length of the ellipse's two main axes and the rotation angle. Thus, a total of 63 (= 21 × 3) fractal features are extracted.

We can conclude that with this first approach, we have adapted the text-independent writer identification approach proposed by Hertel and Bunke in [17] to music scores. In the preprocessing step, the image is binarized, de-skewed, staffs are removed and the lines of music symbols are normalized. Afterwards, 98 features (slant, connected components, upper and lower contours, and fractals) are computed. We have removed the closed-loop features from the set of 100 line features proposed for text [17] because there are not enough closed loops in a big amount of music symbols.

4 Writer identification approach based on music textures

The second approach is based on the computation of textural features. Textures provide important characteristics for object identification, playing an important role in image analysis and pattern classification [15,34]. Texture classification has been used in applications such as biomedical image processing, content based image retrieval, the analysis of satellite images, etc. Some authors [13,18,27] considers writer identification as a texture identification problem. They generate a uniform texture from text lines and compute their textural features. Some works [18,35] demonstrate that these features can also been used for script and language identification. Concretely, Peake and Tan [25] propose the generation of texture images from printed text and the use of Gabor filters and grey level co-ocurrence matrices as textural features for script and language identification. Similarly, Said et al. demonstrate in [27] that textural features can also be successfully used for writer identification.

In our approach, we generate texture images from music symbols before the computation of textural features.

4.1 Generation of texture images

Once the input image is preprocessed, the music symbols are used for generating texture images. It must be noted that textural features directly applied to the music score without any staff removal are not effective, because the frequency of the staff lines affects the values of the textural features.

In [11] we presented four different methods for obtaining the texture images. Every method used a different spatial modification of music symbols to generate the textures. Results showed that *Resize Textures* were the best texture images for this problem. Resize texture consists in randomly choosing music symbols from the music score, and putting them in a reference line, with the same inter-symbol distance, and resizing them to the same size, but without preserving the aspect ratio in the resizing process. In this way, the appearance of the symbols is distorted (symbols are taller compared to their original shape), but one obtains compact texture images.

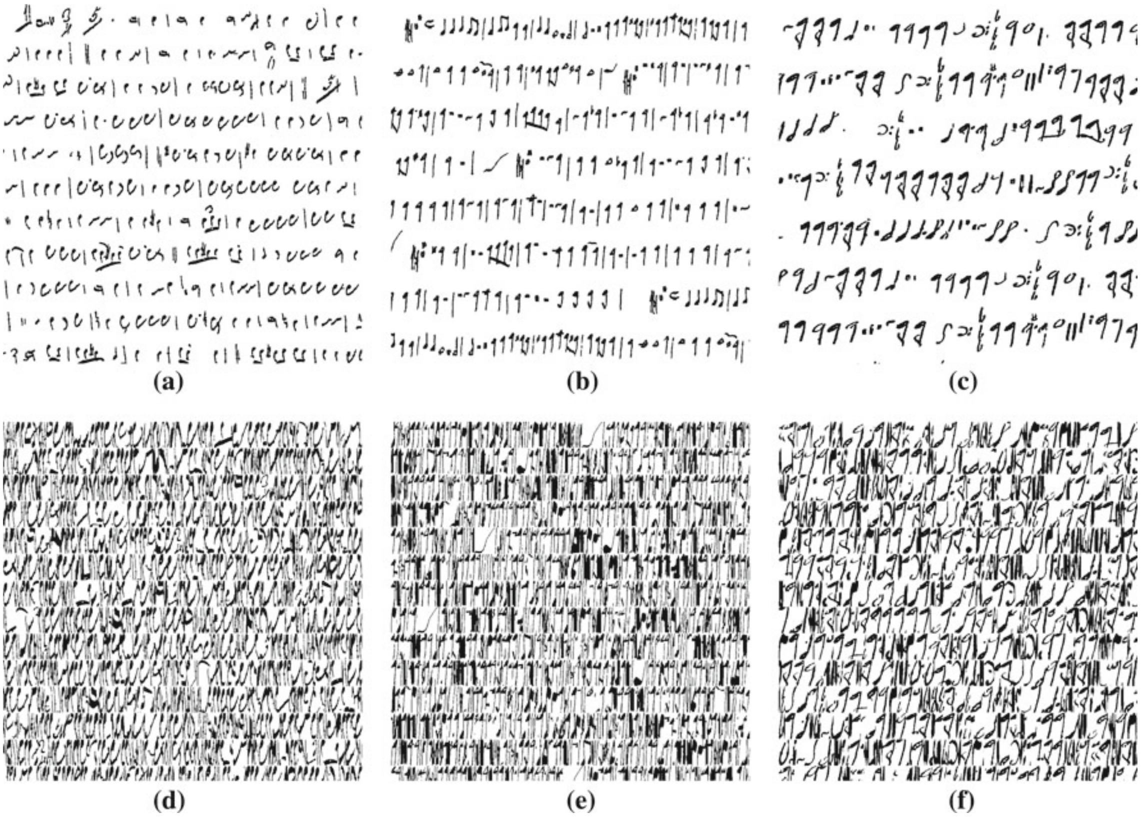

Fig. 8 Generation of texture images. Pieces of preprocessed music pages from three different writers. **a** Writer 1, **b** Writer 2, **c** Writer 3; and the corresponding Resize texture images generated. **d** Writer 1, **e** Writer 2, **f** Writer 3

The steps for the generation of the Resize textures are the following:

- Take all the music symbols obtained after the staff removal step.
- Resize all the music symbols to the same size, without preserving their aspect ratio.
- Take randomly the resized music symbols and put them in a reference line, with the same inter-symbol distance.
- Repeat the previous steps until all the music lines in the texture image are completed.

An example is shown in Fig. 8. It is important to remark that the three writers in this figure can be easily distinguished one from each other. Having a look at the resulting texture images, one can see that writer 1 tends to use more curves than straight lines, writer 2 tends to write in a rectilinear way (a lot of straight lines), and writer 3 tends to write with clearly observable slant.

4.2 Feature extraction from texture images

Once images of music textures have been generated, textural features can be computed. In [25] and [27], texture images are generated from text, and from these texture images one can obtain textural features. We have been inspired by this idea, generating music texture images to extract textural features.

The textural features computed in our approach are Gabor features and Gray-Scale co-ocurrence matrices. They are described next.

4.3 Gabor features

Prior to the definition of the Gabor features, we will briefly present the fundamentals of Gabor filters.

4.3.1 Gabor filters

The multi-channel Gabor filtering technique [32] is typically used for analyzing the frequential information of an input image. It can be seen as a windowed Fourier Transform where the window is a Gaussian function. This technique is based on psychophysical and physiological studies [7] which state that the processing of information in the human visual cortex involves a set of parallel and quasi-independent cortical channels. Every cortical channel can be modeled by a pair of Gabor filters $h_e(x, y; f, \theta)$ and $h_o(x, y; f, \theta)$. These filters show opposite symmetry and are computed as

$$\begin{cases} h_e(x, y; f, \theta) = g(x, y)\cos(2\pi f(x\cos\theta + y\sin\theta)) \\ h_o(x, y; f, \theta) = g(x, y)\sin(2\pi f(x\cos\theta + y\sin\theta)) \end{cases} \quad (3)$$

where $g(x, y)$ is a 2D Gaussian function, f is the frequency and θ is the spatial orientation.

The Gabor transform of image $P(x, y)$ can be computed as

$$\begin{cases} q_e(x, y) = FFT^{-1}\left[\hat{P}(u, v)H_e(u, v)\right] \\ q_o(x, y) = FFT^{-1}\left[\hat{P}(u, v)H_o(u, v)\right] \end{cases} \quad (4)$$

where $\hat{P}(u, v)$ is the Fourier Transform of the input image $P(x, y)$, and $H_e(u, v)$ and $H_o(u, v)$ are the Fourier Transform of the filters $h_e(x, y; f, \theta)$ and $h_o(x, y; f, \theta)$, respectively. In our case, we perform a combination of the two filters by

$$q(x, y) = \sqrt{q_e^2(x, y) + q_o^2(x, y)}. \quad (5)$$

For the sake of clarity, Fig. 9 shows an example of the application of different Gabor filters to an input image with music symbols. One can see that the important frequency components of this image are found in the horizontal orientation (Fig. 9b, d) instead of the vertical and diagonal ones (Fig. 9c, e, respectively).

4.3.2 Computation of gabor features

For the computation of the Gabor features, we have to specify the angle θ and the central frequency f, which define the location of the Gabor filter on the frequency plane. In [33], it has been shown that for an image of size $N \times N$, the important frequency components are found within $f \leq N/4$ cycles/degree. For this reason, the two parameters used are the radial frequency with values $f \in \{4, 8, 16, 32\}$ and the orientation with values $\theta \in \{0°, 45°, 90°, 135°\}$. The output corresponds to $4 \times 4 = 16$ images. Extracting the mean and the standard deviation we obtain a total of $16 \times 2 = 32$ features. In other words, textural features can be described as a two-dimensional vector $(\mu, \sigma) = \mathbf{v}(f, \theta)$.

4.4 GSCM features

Some authors [16,34] claim that neighborhood properties of a pixel can represent a texture. In this sense, the grey level co-occurrence and their distribution in the pixel neighborhood reflect the local activities of a texture, being one of the useful properties for texture description. They estimate image properties related to second-order statistics, allowing the discrimination of one texture from another. Haralick [16] proposes the use of Grey-Scale Co-ocurrence Matrices (GSCM), which describe pairs of grey levels with given distance and orientation. Although this method considers only the spatial distribution of each pair of grey level pixels, it has become a popular technique for characterizing grey scale textures [25].

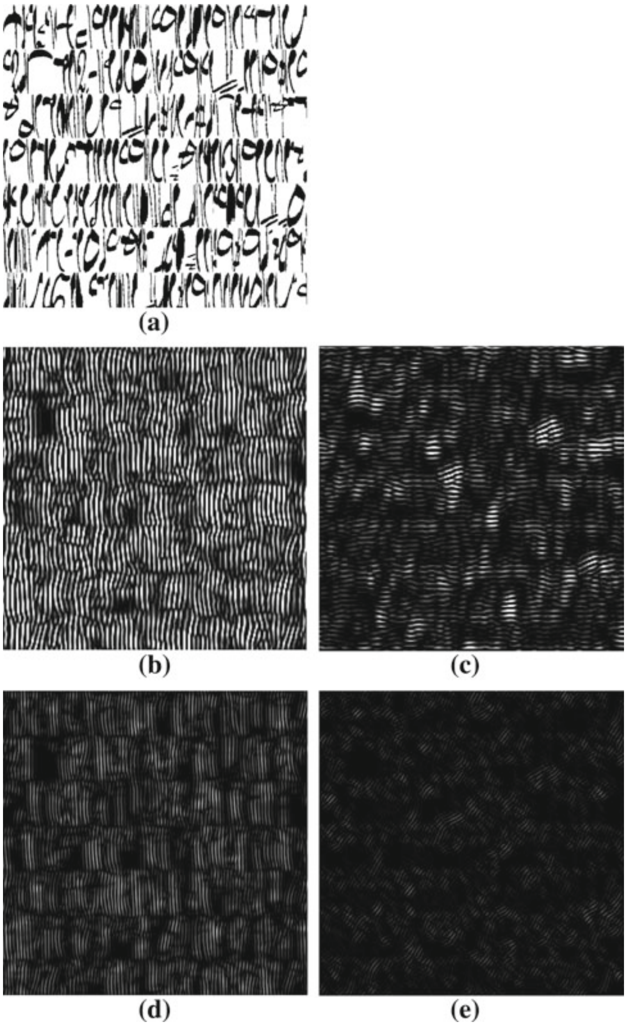

Fig. 9 Example of the application of different Gabor filters. **a** Original image, **b** and **d** results of applying a *horizontal Gabor filter* with low and high frequency, respectively, **c** Results of applying a *vertical Gabor filter* with low frequency and **e** diagonal high frequency

If an image contains G grey levels, for every distance d and angle θ we obtain a matrix of dimension $G \times G$, defined as $GSCM_{d,\theta}$, where $GSCM_{d,\theta}(a, b)$ corresponds to the number of pairs $(P1, P2)$ where $P1$ is a pixel of grey value a, $P2$ is a pixel of grey value b, and $P1$ and $P2$ are separated by distance d and angle θ. Whereas GSCM are of a high computational cost for grey level images, they are fast to compute for binary images, because there are only two grey values.

The parameters used in our method are the distance d with values $d \in \{1, 2, 3, 4, 5\}$; and the orientation $\theta \in \{0°, 45°, 90°, 135°\}$. The output corresponds to 20 matrices of dimension 2×2, and due to the diagonal symmetry, there are only 3 independent values in each matrix. In total we obtain $20 \times 3 = 60$ features.

We can summarize that this second symbol- independent writer identification approach generates texture images from the music scores in order to extract textural features (Gabor and GSCM). It is an adaptation of the text-idenpendent writer identification approach proposed in [27] to music scores. Consequently, the system is more robust, without the use of any symbol recognition method.

5 Combination of the two approaches

Once the line and textural features have been computed as described in Sects. 3 and 4, the individual classification methods according to Sects. 3 and 4 are applied, and then the results are combined for obtaining the final writer identification (w.i.). These steps are described next.

5.1 Classification of each of the two individual approaches

Once the first w.i. approach has completed its specific preprocessing (normalization and generation of s music lines) and has computed the 98 line features described in Sect. 3, the classification of each music line is performed with the k-NN classifier using the Euclidean Distance [8]. For a fair comparison between all the features, they are all normalized (values between 0 and 1) before applying the Euclidean distance. Since we obtain s music line images from every music page, and consequently they belong to the same music sheet, they should be assigned to the same class. For this reason, we combine the classification results of the s music lines. Since we are mainly focused on the feature extraction step, we have chosen two classic well-known methods for combining the results from both approaches: the Majority Voting and the Borda Count method.

Majority Voting and Borda Count are classical combination methods [21] which have been often used in the context of multiple classifier systems. In both methods, for each element to be classified we first determine the k nearest candidate classes. The list with the k candidates is sorted so that the first candidate has obtained the highest confidence rate. Majority Voting consists in assigning one vote to each of the k nearest candidate classes. Contrary, in the Borda Count method, the first ranked candidate obtains more votes for the class than the last ranked candidate.

Similarly, the second w.i. approach generates s texture images and computes the textural features described in Sect. 4. Then, the s texture images are classified with the k-NN classifier using the Euclidean Distance. Due to the fact that the s textural images are extracted from the same music sheet, they should be assigned to the same class. For this reason, the classification results are combined using the Majority Voting or the Borda Count method described above.

5.2 Combination of the results obtained from the two approaches

Once each writer identification approach has performed the voting step described above, we combine all the obtained votes. The combination is performed as follows. Firstly, for the line-features approach, each of the three input lines are classified and the corresponding k nearest neighbor classes receive votes. Secondly, each of the three input textures are also classified and the k-NN classes receive the corresponding votes. Finally, all the votes are combined, and the input music sheet will be classified as the class which has received the largest number of votes. Notice that both w.i. approaches have the same weight, because they give the same number of votes.

6 Experimental results

We have tested the methods on a data set consisting of 200 music pages (see an example in Fig. 10). These pages have been obtained from a collection of music scores of the seventeenth to nineteenth centuries, from the archive of Seminar of Barcelona and the archive of Canet de Mar (Spain). The data set contains 10 pages for each one of 20 different writers. Concerning the first approach, we have divided the long music line in one, two, three, four or five samples. Similarly, we have generated from one to five texture images. The size of each texture image is of 2,048 × 2,048 pixels (because of the high resolution of the scanned input images). Thus, from the 200 music pages, if we generate s music lines and s texture images per music sheet, we obtain a data set of $200 \cdot s$ lines and $200 \cdot s$ texture images (10 pages × 20 writers $\times s$).

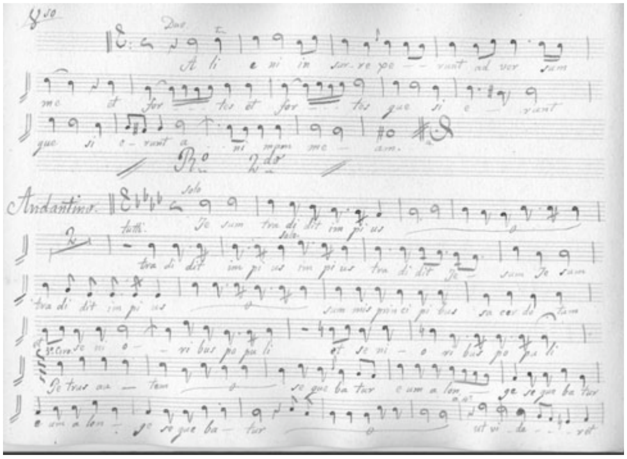

Fig. 10 Example of an old score of the composer Casanoves

Table 1 Writer identification rates and confidence values using line features with different number of samples (s = number of samples generated for each music page)

It shows the results for different values of the k-NN, and with the combination of features using the Majority Voting (MV), the Borda Count (BC) method, or without any combination. Best values for each case are indicated in bold

s	Combination	W.I.Rates (%) and confidence values				
		1-NN	3-NN	5-NN	7-NN	9-NN
1	None	61 (11)	**65 (10)**	60 (6)	57 (8)	56 (6)
2	None	65 (6)	67 (7)	64 (8)	66 (8)	64 (8)
2	MV	62 (9)	69 (9)	65 (10)	64 (10)	66 (12)
2	BC	62 (9)	**70 (7)**	68 (9)	67 (8)	67 (8)
3	None	67 (7)	70 (6)	68 (4)	68 (6)	69 (7)
3	MV	72 (7)	**76 (8)**	74 (8)	71 (8)	72 (12)
3	BC	72 (7)	75 (5)	75 (10)	73 (10)	73 (9)
4	None	70 (8)	69 (7)	69 (8)	68 (10)	68 (11)
4	MV	75 (7)	73 (9)	73 (11)	72 (13)	72 (12)
4	BC	**75 (7)**	74 (7)	74 (7)	72 (9)	71 (12)
5	None	69 (6)	68 (6)	70 (8)	72 (9)	68 (8)
5	MV	74 (6)	74 (9)	**78 (10)**	77 (12)	74 (11)
5	BC	74 (6)	75 (10)	75 (11)	77 (10)	76 (11)

For the experiments, we have used 5 test subsets, randomly chosen, containing one page per writer. This means that all the s music lines obtained from every page are used in the same test set. Similarly, all the s music texture images generated from every page are in the test set. In order to obtain independent test subsets, all the s lines or textures generated from one music test page belong to the same subset. For example, if we generate $s = 3$ samples for each image, there will be 60 images (3 elements × 20 pages) in each test subset, and the remaining 540 images in the training set, i.e. as prototypes for the k-NN classifier.

As previously explained, we obtain s music images (lines or textures) from every music page and, consequently, all these s elements belong to the same music sheet and should be assigned to only one class. For the purpose of comparison, our experiments also show the case where no combination of results is attempted (the s images extracted from a same page can be assigned to different classes).

Next, the results of each individual writer identification approach are shown. In addition, some Feature Selection Methods have been applied for discussing the suitability of the feature sets. Finally, the individual approaches are compared with the combination of the two approaches. The score is computed by means of stratified 5-fold cross validation, testing for the 95% confidence interval CI with a two-tailed t-test [19,21], computed as:

$$CI = \frac{1.96\,\sigma_{X_j}}{\sqrt{NT}} \tag{6}$$

where σ_{X_j} is the standard deviation of the performance of the tests X_j, and NT is the number of tests.

6.1 Results using music line features

Table 1 shows the writer identification results using line features with different number of samples ($s = \{1, 2, 3, 4, 5\}$) generated for each music page. Thus, from the 200 music pages, if we generate one music line per music sheet, we have a data set of 200 lines. In case we generate five music lines for each music page, we generate a total of 1,000 music lines (5 × 10 pages × 20 writers) for computing the feature vectors. From the experimental results, we can see that using one line per page we obtain a writer identification (W.I.) rate of 65%, using two lines we obtain a W.I. rate of 70%, which increases to 76% using three lines per page. Contrary, with the generation of four and five samples there is no significant improvement (75 and 78% respectively), whereas the number of samples generated per page increases considerably (1,000 with 5 samples instead of 600 with 3 samples), and consequently, the computational time. For this reason, we have decided to generate exactly three lines for each music page, obtaining a good trade-off between performance and computational cost.

Table 2 shows the writer identification results using line features for an increasing number of writers. From the database, the first 5 writers have been selected. The classification rates have been obtained for different values of k in the k-NN classifier and also for different combinations of the results obtained from the classification of the three music lines per page (None, Majority Voting and Borda Count). Iteratively, 5 writers have been added to the database, and the experiments have been repeated. It can be observed that 3-NN and 5-NN obtain in most cases better recognition rates than 7-NN. Results using Majority Voting or Borda Count are

Table 2 Classification results: writer identification rates (W.I.Rates) and confidence values (given in brackets), using 98 line features for different database sizes (w = number of writers) and different combination of results

w	Combination	W.I.Rates (%) and confidence values		
		3-NN	5-NN	7-NN
5	None	77.3 (12.1)	77.3 (6.7)	73.3 (4.1)
5	MV	84 (14.7)	76 (7.8)	72 (9.6)
5	BC	**84 (14.6)**	**84 (14.6)**	72 (9.6)
10	None	75.3 (7.9)	72 (8.9)	68 (8.4)
10	MV	80 (10.7)	74 (4.8)	68 (7.3)
10	BC	**80 (12.4)**	76 (10)	72 (11.4)
15	None	69.3 (7.7)	69.7 (4.7)	69.3 (5.4)
15	MV	**78.6 (8.6)**	74.6 (9.6)	70.6 (8.9)
15	BC	77.3 (8.9)	73.3 (10.9)	73.3 (10.9)
20	None	70.6 (5.9)	68.3 (4.3)	68.6 (6.1)
20	MV	**76 (8.4)**	74 (8.4)	71 (7.8)
20	BC	75 (5.4)	75 (10.3)	73 (9.6)

Best values for each case are indicated in bold

better than results using no combination at all. Concerning the scalability of the method, in the best case, the recognition rate of 84% for 5 writers decreases to 76% for 20 writers, showing that the method scales well.

6.2 Results using music textural features

Similar to the line features experiments, we show the writer identification results using textural features with different number of samples ($s = \{1, 2, 3, 4, 5\}$) generated for each music page. Thus, if we generate one music texture image per music sheet, we have a data set of 200 texture images, and if we generate five music texture images for each music page, we generate a total of 1,000 music texture images (5×10 pages $\times$ 20 writers). The writer identification rates obtained using Gabor, GSCM and both (Gabor and GSCM) set of features are shown in Tables 3, 4, and 5 respectively. We can see the results for different values of $k = \{1, 3, 5, 7, 9\}$ for the Nearest Neighbor, and also with the Majority Voting and Borda Count method for the combination of classification results. From the experimental results, we can see that the generation of three texture images for each page is a good trade-off between performance and computational cost (a W.I. rate of 73% with 600 texture images). Notice that the combination of the Gabor and GSCM features in one single vector of 92 features increases the final recognition rates (73% with 3 texture images). One can see that the Majority Voting and Borda Count method obtain the best classification results, and in most of the cases, the $k = 5$ value slightly increases the final classification rates.

As a summary, it can be said that the Resize Textures with the combination of both Gabor ad GSCM features are the best choice, obtaining a writer identification rate of 73% using Borda Count and the 5-NN classifier, or using Majority Voting and 3-NN. A Resize texture has the highest number of symbols for each texture image, and visually, the texture images belonging to the same writer are very similar. As a consequence, the intra-class distance is reduced, helping in the classification.

Concerning the scalability of the method, Table 6 shows the writer identification rates of music textures with the 92 textural features for different database sizes. It is important to notice that the writer identification rate decreases significantly when adding more writers to the database (from 96% with 5 writers to 73% with 20 writers) because the different writer styles become very close. In fact, the analyzed confusion matrices show that the disciples of the same musician, or writers that belong to the same place and time period, tend to have a very similar writing style (see Fig. 11).

6.3 Results using feature selection methods

The suitability of the line and textural features has been analyzed, because some of them could be unnecessary or even redundant. The goal of feature selection is to find the best subset of features that perform better than the whole set of features. We have performed the Sequential Forward Search (SFS), Sequential Backward Search (SBS), Sequential Floating Forward Search (SFFS), Sequential Floating Backward Search (SFBS) (see [20,26]). For the experiments, wrappers are used as objective function, where one of the five subsets is used as the test set and the others as the prototypes in the 5-NN classifier. To evaluate the quality of a selected feature subset, iteratively three subsets are used in the classifier and the remaining set is used to measure the quality of the considered feature subset. Once the algorithm finds the

Table 3 Writer identification rates and confidence values using the Gabor features (s = number of samples used for the classification)

It shows the results for different values of the k-NN, and with the combination of features using the Majority Voting (MV), the Borda Count (BC) method, or without any combination. Best values for each case are indicated in bold

s	Combination	W.I.Rates (%) and confidence values (with 32 Gabor features)				
		1-NN	3-NN	5-NN	7-NN	9-NN
1	None	49 (6)	53 (7)	53 (5)	**56 (9)**	56 (6)
2	None	58 (6)	59 (7)	60 (6)	62 (5)	62 (3)
2	MV	59 (7)	66 (6)	64 (8)	68 (7)	65 (3)
2	BC	59 (7)	65 (4)	65 (4)	68 (5)	**69 (5)**
3	None	57 (6)	59 (7)	59 (5)	62 (6)	61 (5)
3	MV	59 (7)	67 (11)	67 (7)	65 (7)	67 (8)
3	BC	59 (7)	67 (10)	64 (10)	66 (8)	**68 (7)**
4	None	56 (6)	58 (6)	58 (5)	58 (5)	60 (5)
4	MV	56 (8)	68 (7)	66 (7)	66 (7)	**69 (6)**
4	BC	56 (8)	65 (5)	69 (7)	68 (7)	68 (9)
5	None	55 (5)	59 (5)	59 (4)	59 (5)	58 (6)
5	MV	66 (9)	70 (7)	71 (7)	69 (4)	71 (6)
5	BC	66 (9)	69 (8)	71 (8)	**71 (7)**	71 (7)

Table 4 Writer identification rates and confidence values using the GSCM features (s = number of samples used for the classification), with different values of the k-NN, and with the combination of features using the Majority Voting, the Borda Count method, or without any combination

Best values for each case are indicated in bold

s	Combination	W.I.Rates (%) and confidence values (with 60 GSCM features)				
		1-NN	3-NN	5-NN	7-NN	9-NN
1	None	64 (6)	61 (8)	**65 (11)**	60 (12)	58 (7)
2	None	64 (7)	64 (9)	63 (7)	62 (9)	64 (10)
2	MV	65 (5)	65 (11)	61 (9)	62 (7)	62 (8)
2	BC	65 (5)	64 (10)	**65 (11)**	65 (11)	64 (12)
3	None	64 (8)	64 (8)	64 (8)	61 (7)	60 (9)
3	MV	64 (7)	64 (7)	65 (9)	61 (11)	62 (10)
3	BC	64 (7)	64 (7)	**66 (9)**	66 (9)	64 (12)
4	None	64 (10)	64 (9)	64 (9)	64 (8)	62 (8)
4	MV	65 (9)	65 (8)	63 (8)	65 (10)	61 (9)
4	BC	65 (9)	65 (10)	64 (8)	64 (8)	**66 (10)**
5	None	65 (11)	63 (10)	64 (9)	64 (9)	64 (9)
5	MV	63 (11)	64 (10)	66 (9)	66 (9)	65 (10)
5	BC	63 (11)	64 (10)	65 (10)	66 (9)	**67 (10)**

best feature subset, the fifth subset is used for the final writer identification rate.

Table 7 shows the line features results of feature selection algorithms for the dataset of 20 writers. The first row shows the baseline rate (with a 76% or identification rate), where all the features are used for the classification. The next ones show the writer identification rates using SFS, SBS, SFFS and SFBS feature set search methods. It is important to remark that results show that they do not improve the baseline. In fact, the SFS and SBS obtain about 65% of identification rate, which is remarkably lower that 76%, probably because the methods reach some local minima or maxima and cannot improve the final identification rate. In fact, in the SFS method, when a feature Y is selected, it will be for sure in the final solution set. In a similar way, if a feature Z is removed from the set in the SBS, it will never be considered again. For this reason, SFFS and SFBS reach higher identification rates (70 and 75%), because a feature W can be added and removed several times from the set of features during the training step.

In Table 8 results of Resize textures of feature selection algorithms are shown. The first row again shows the baseline rate, and the next ones show the results using SFS, SBS, SFFS and SFBS feature set search methods. Similarly to the

Table 5 Writer identification rates and confidence values using the Gabor and GSCM features (s = number of samples used for the classification), with different values of the k-NN, and with the combination of features using the Majority Voting, the Borda Count method, or without any combination

s	Combination	W.I.Rates (%) and confidence values (with 92 features)				
		1-NN	3-NN	5-NN	7-NN	9-NN
1	None	**65 (3)**	61 (6)	62 (7)	62 (7)	64 (6)
2	None	66 (6)	67 (7)	69 (9)	66 (8)	65 (7)
2	MV	65 (4)	71 (11)	69 (9)	67 (6)	61 (8)
2	BC	65 (4)	**71 (9)**	70 (10)	70 (12)	70 (12)
3	None	65 (7)	67 (7)	68 (6)	68 (8)	69 (6)
3	MV	68 (5)	**73 (9)**	72 (8)	71 (10)	71 (8)
3	BC	68 (5)	70 (8)	73 (11)	71 (11)	71 (11)
4	None	64 (6)	65 (6)	70 (5)	67 (7)	69 (7)
4	MV	69 (6)	71 (9)	**75 (13)**	72 (11)	71 (8)
4	BC	69 (6)	69 (8)	71 (9)	74 (10)	73 (9)
5	None	66 (5)	67 (6)	66 (6)	67 (6)	68 (7)
5	MV	70 (4)	70 (4)	**75 (6)**	74 (6)	74 (8)
5	BC	70 (4)	70 (4)	73 (5)	74 (6)	74 (6)

Best values for each case are indicated in bold

Table 6 Classification results of the resize textures: writer identification rates and confidence values using the 92 textural features (Gabor and GSCM) for different k-NN values and database sizes (w = number of writers), and different combinations of results

w	Combination	W.I.Rates (%) and confidence values		
		3-NN	5-NN	7-NN
5	None	93 (5.8)	93 (5.8)	89 (12.1)
5	MV	**96 (7.9)**	92 (9.6)	88 (15.7)
5	BC	92 (9.6)	92 (9.6)	92 (9.6)
10	None	81 (9.6)	81 (10.2)	82 (12.1)
10	MV	**86 (11.8)**	82 (15.7)	80 (13.9)
10	BC	84 (7.8)	84 (13.3)	82 (15.7)
15	None	67 (6.8)	68 (6.2)	69 (7)
15	MV	71 (9.8)	**75 (10.5)**	71 (9.8)
15	BC	69 (6.7)	73 (10.9)	72 (12.7)
20	None	67 (6.6)	68 (6.2)	68 (8.1)
20	MV	73 (9)	72 (8)	71 (8.4)
20	BC	70 (8.2)	**73 (11)**	71 (11.3)

Best values for each case are indicated in bold

feature selection methods applied to line features, SFFS and SFBS reach better results than SFS and SBS. In this case, the identification results of SFFS and SFBS do not improve the baseline (SFBS reaches 71.6% which is a little bit lower than 73% of the baseline rate).

It must be said that, although feature selection methods do not reach any improvement over the baseline, the dimensionality reduction is significant (from the 98 line features and 92 textural features to the 20 and 11 features selected by SFBS). This fact shows that there are many dependent or irrelevant features in the original feature set, giving us the possibility to select a subset for obtaining similar results in this database.

6.4 Comparison of the two approaches

We have also compared the two *symbol-independent* writer identification approaches. Figure 12 shows the writer identification results of both methods. We have decided to compare the best result for each size of the database and method, independently of the value of k-NN and the combination method. Thus, for example, in some cases the best value is obtained using 3-NN with Majority Voting, and in others, the best rate is obtained using 5-NN with Borda Count. Having a look at the results, one can see that textural features reach higher performance for a database of few writers (86% of w.i.r. with 10 writers with textural features versus the 82% with line

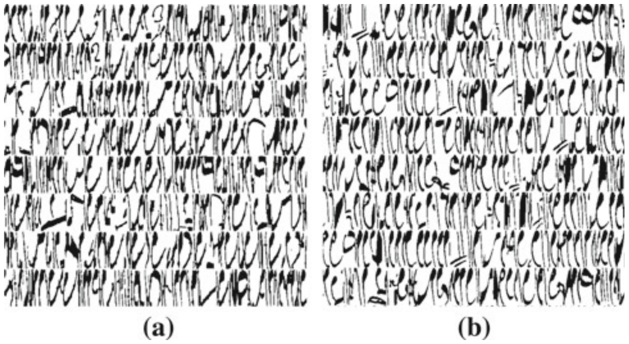

Fig. 11 Resize texture images from two writers: both texture images are very similar although they belong to different classes

Table 7 Classification results: writer identification rates for 20 writers using feature set search methods for line features. It shows the results for different values of the k-NN and different number of features (f = number of features), and also with the combination of features using the Majority Voting (MV), the Borda Count (BC) method, or without any combination

Method	Combination	f	W.I.Rates (%)	
			3-NN	5-NN
All features	None	98	70.6	68.3
All features	MV	98	**76**	74
All features	BC	98	75	75
SFS	None	43	60	58.3
SFS	MV	43	65	60
SFS	BC	43	65	60
SBS	None	54	65	58.3
SBS	MV	54	60	65
SBS	BC	54	65	60
SFFS	None	35	65	66.6
SFFS	MV	35	70	70
SFFS	BC	35	70	70
SFBS	None	20	66.6	68.3
SFBS	MV	20	70	**75**
SFBS	BC	20	**75**	**75**

Best values for each case are indicated in bold

features), whereas the line features lead to higher results for a database with 15 or 20 writers (76 vs. 73% of w.i.r. with 20 writers for line features and textural features respectively). Since the writer identification rate of 76% is not a significant improvement of the 73% rate, we can conclude that line and textural features reach similar identification rates for this specific database.

Table 8 Classification results: writer identification rates for 20 writers using feature set search methods for textural features

Method	Combination	f	W.I.Rates (%)	
			3-NN	5-NN
All Features	None	92	67	68
All Features	MV	92	73	72
All Features	BC	92	70	**73**
SFS	None	32	61.6	60
SFS	MV	32	70	60
SFS	BC	32	65	70
SBS	None	18	66.6	63.3
SBS	MV	18	65	65
SBS	BC	18	65	65
SFFS	None	28	65	66.6
SFFS	MV	28	70	65
SFFS	BC	28	70	70
SFBS	None	11	68.3	**71.6**
SFBS	MV	11	70	70
SFBS	BC	11	70	70

Best values for each case are indicated in bold

6.5 Combination of the two approaches

The ensemble architecture proposed in this paper has been evaluated on the same set of 200 music sheets (10 pages for each one of the 20 writers) that has been used for testing the two individual w.i. methods. For a fair comparison, these sets are identical to those that have been used for testing the writer identification methods based on music lines and textures. For the combination, we have used the Euclidean distance, the 5-NN classifier and the Borda Count combination method.

Table 9 shows the writer identification rate (w.i.r.) for each one of the 5 tests, and the final identification rate. The first two rows show the identification rates when using line features (76%) and textural features (73%). The third row corresponds to the combination of line and textural features, reaching a writer identification rate of 92%, which is a significant improvement in comparison with the two individual approaches.

It must be noticed that the proposed architecture extracts information about the music symbols, avoiding any dependence on the symbols' density. The number and the kind of symbols appearing in the music score is closely related to the rhythm and the melody of the composition. In our approach, we try to avoid this dependence. In the first w.i. approach, the value of features extracted from music lines is averaged (e.g. we compute the mean of the distance between connected components, or the mean of the slant of the music symbols). In the second w.i. approach, texture images are generated by randomly selecting symbols and resizing them. Thus, we minimize the effect of the different number and kind of symbols in the music sheets. As an example, Fig. 13 shows two music scores written by the same writer, which have been correctly classified. One can see that although the density

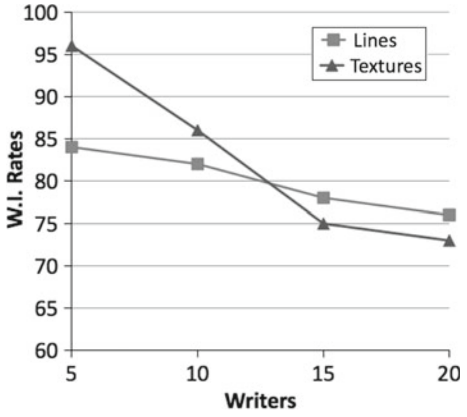

Fig. 12 Classification results: writer identification rates using 98 line features and 92 textural features for different database sizes

and the kind of symbols is different, the writing style is very similar (compare the shape of notes, clefs, ending signature).

The high performance of the combination of the two approaches is remarkable. It is much higher than the rates obtained from the two individual approaches. Thanks to the combination of results, when one approach has misclassified a test image, the other approach is sometimes able to correct this missclassification. In fact, it might occur that even though both classifiers made a wrong decision, an input test image is correctly classified due to the combination of the previous classifications.

7 Conclusions

This work has addressed the task of writer identification in old music scores, as an example of graphic documents. In this paper we have proposed the combination of two *symbol-independent* writer identification approaches, avoiding the dependence on good symbol recognition methods. It must be noticed that the writer identification methods for text documents can not be directly applied to graphical documents, such as music scores. We need a specific a preprocessing stage in order to remove the staff lines. In addition, the normalization of music lines is not similar to the text lines, because contrary to handwritten text, music symbols are distributed over the staff in different positions. Concerning the generation of texture images, we have investigated several approaches, concluding that one particular distribution of symbols in a texture is the best option.

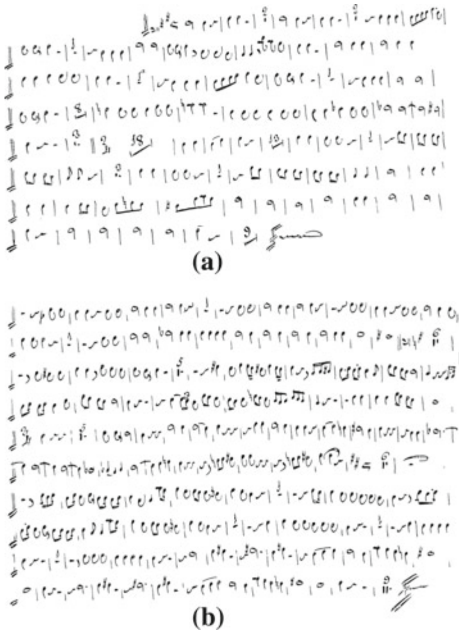

Fig. 13 Two music scores of the same writer: although the density of symbols is different, both music sheets are correctly classified as belonging to the same class

The steps of the system are the following. In the preprocessing step, the image is binarized, de-skewed, and staffs and lyrics are removed. Then, the first approach generates three normalized music lines and extracts 98 line features, while the second approach generates three texture images and computes 92 textural features (GSCM and Gabor features). The classification is performed using the Euclidean distance, the k-Nearest Neighbour classifier, and the Majority Voting or Borda Count combination method. For combining the results, each classifier gives votes to the nearest neighbor classes according to the confidence rate. Finally, the input music sheet is classified as the writer that has received the highest number of votes.

As in other application domains, the experimental results show that the combination of two different approaches, obtaining an identification rate of 92%, increases the writer identification rates obtained individually from the two

Table 9 Combination of the results of the two writer identification approaches

Experiment	W.I.Rates (%) and confidence values					
	Test					Average
	1	2	3	4	5	
Lines	65	75	95	70	70	**75 (10.3)**
Textures	65	55	80	85	80	**73 (11)**
Both	80	95	100	95	90	**92 (6.6)**

We show the identification rates of each test and the average with the corresponding confidence value. Best values for each case are indicated in bold approaches (76 and 73% respectively). The significant improvement of combination results demonstrates the suitability of the classifiers proposed architecture.

As a summary, we can conclude that for this specific database, the combination of the two approaches has demonstrated to be the best choice. As far as we know, this is one of the very few works addressing the problem of writer identification in handwritten documents of graphical contents. We believe that the work described in this paper constitutes a step forward in the field of graphics recognition. Further work will focus on increasing the size of the database, and adding specific features for musical notation to the current set of features. In fact, we are planning to extract information about the shape of music symbols, thus obtaining a *symbol-dependent* method. In this context, a semi-automatic approach could be the best option, because the user could supervise the symbol recognition step. Although the addition of the user in the loop would probably decrease the speed of the system (specially when working with large data sets), the system could reach higher identification rates.

Acknowledgments We would like to thank Prof. J.M.Gregori for his help in accessing the historical archives, and V.Kilchherr and Dr. A.Schlapbach for providing support in the experiments. This work has been partially supported by the Spanish projects CONSOLIDER-INGENIO 2010 (CSD2007-00018), TIN2008-04998, TIN2009-14633-C03-03 and 2007 BE-1 00086.

References

1. Bar-Yosef, I., Beckman, I., Kedem, K., Dinstein, I.: Binarization, character extraction, and writer identification of historical Hebrew calligraphy documents. Int. J. Doc. Anal. Recognit. **9**(2), 89–99 (2007)
2. Bensefia, A., Paquet, T., Heutte, L.: A writer identification and verification system. Pattern Recognit. Lett. **26**(13), 2080–2092 (2005)
3. Bouletreau, V., Vincent, N., Sabourin, R., Emptoz, H.: Synthetic parameters for handwriting classification. In: Proceedings of the 4th International Conference on Document Analysis and Recognition (ICDAR), pp. 102–106. IEEE Computer Society, Washington, DC, USA (1997)
4. Bouletreau, V., Vincent, N., Sabourin, R., Emptoz, H.: Handwriting and signature: One or two personality identifiers? In: Proceedings of 14th International Conference on Pattern Recognition (ICPR), vol. 2, pp. 1758–1760 (1998)
5. Bruder, I., Ignatova, T., Milewski, L.: Integrating knowledge components for writer identification in a digital archive of historical music scores. In: Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital libraries (JCDL), p. 397. ACM, New York, NY, USA (2004). doi:10.1145/996350.996463
6. Bulacu, M., Schomaker, L.: Combining multiple features for text-independent writer identification and verification. In: Proceedings of 10th International Workshop on Frontiers in Handwriting Recognition (IWFHR 2006), pp. 281–286 (2006)
7. De Valois, R.L., Albrecht, D.G., Thorell, L.G.: Spatial frequency selectivity of cells in macaque visual cortex. Vision Res. **22**(5), 545–559 (1982)
8. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley, London (2001)
9. Fornés, A., Lladós, J., Sánchez, G.: Primitive segmentation in old handwritten music scores. In: Liu, W., Lladós, J. (eds.) Graphics Recognition: Ten Years Review and Future Perspectives, vol. 3926 of Lecture Notes in Computer Science, pp. 279–290. Springer (2006)
10. Fornés, A., Lladós, J., Sánchez, G., Bunke, H.: Writer identification in old handwritten music scores. In: 8th International Workshop on Document Analysis Systems, pp. 347–353. Nara, Japan (2008)
11. Fornés, A., Lladós, J., Sánchez, G., Bunke, H.: On the use of textural features for writer identification in old handwritten music scores. Document analysis and recognition (ICDAR). In: Tenth International Conference on, vol. 2, pp. 996–1000 (2009)
12. Garain, U., Parui, S.K., Paquet, T., Heutte, L.: Machine dating of manuscripts written by an individual. J. Electron. Imaging SPIE. **17**(1), 011012 (2008)
13. Gazzah, S., Ben Amara, N.: Arabic handwriting texture analysis for writer identification using the dwt-lifting scheme. Document analysis and recognition, 2007. ICDAR 2007. In: Ninth International Conference on, vol. 2, pp. 1133–1137 (2007). doi:10.1109/ICDAR.2007.4377092
14. Göcke, R.: Building a system for writer identification on handwritten music scores. In: Proceedings of the IASTED International Conference on Signal Processing, Pattern Recognition, and Applications (SPPRA), pp. 250–255. Rhodes, Greece (2003)
15. Haralick, R.: Statistical and structural approaches to texture. In: Proceedings of the IEEE, vol. 67, pp. 786–804 (1979)
16. Haralick, R., Shnmugam, K., Dinstein, I.: Textural features for image classification. IEEE Trans. Syst. Man Cybern. Part B Cybern. **3**(6), 610–621 (1973)
17. Hertel, C., Bunke, H.: A set of novel features for writer identification. In: Audio- and Video-Based Biometric Person Authentication (AVBPA), pp. 679–687 (2003)
18. Hiremath, P., Shivashankar, S.: Wavelet based co-occurrence histogram features for texture classification with an application to script identification in a document image. Pattern Recognit. Lett. **29**(9), 1182–1189 (2008)
19. Johnson, R., Kuby, P.: Elementary Statistics. Duxbury Press, North Scituate (2006)
20. Kittler, J.: Pattern recognition and signal processing. In: Chen, C.H. (ed.) Feature Set Search Algorithms, pp. 41–60. Sijthoff & Noordhoff Intl Pub, Alphen aan den Rijn, Netherlands (1978)
21. Kuncheva, L.: Combining Pattern Classifiers: Methods and Algorithms. Wiley, London (2004)
22. Luth, N.: Automatic identification of music notations. In: Proceedings of the Second International Conference on WEB Delivering of Music (WEDELMUSIC) (2002)
23. Marti, U., Messerli, R., Bunke, H.: Writer identification using text line based features. In: Proceedings of the 6th International Conference on Document Analysis and Recognition (ICDAR), pp. 101–105 (2001)
24. Niblack, W.: An Introduction to Digital Image Processing. Prentice Hall, Englewood Cliffs (1986)
25. Peake, G., Tan, T.: Script and language identification from document images. Document Image Analysis, 1997. (DIA '97) Proceedings, Workshop on pp. 10–17 (1997). doi:10.1109/DIA.1997.627086
26. Pudil, P., Novovicová, J., Kittler, J.: Floating search methods in feature selection. Pattern Recognit. Lett. **15**, 1119–1125 (1994)
27. Said, H., Tan, T., Baker, K.: Personal identification based on handwriting. Pattern Recognit. **33**(1), 149–160 (2000)
28. Siddiqi, I., Vincent, N.: A set of chain code based features for writer recognition. Document analysis and recognition (ICDAR). In: Tenth International Conference on, vol. 2, pp. 981–985 (2009)
29. Schlapbach, A., Bunke, H.: A writer identification and verification system using HMM based recognizers. Pattern Anal. Appl. **10**(1), 33–43 (2007)

30. Schlapbach, A., Bunke, H.: Off-line writer identification and verification using Gaussian mixture models. In: Marinai, S., Fujisawa, H. (eds.) Machine Learning in Document Analysis and Recognition, Studies in Computational Intelligence, vol. 90, pp. 409–428. Springer, Berlin (2008)
31. Schomaker, L., Bulacu, M.: Automatic writer identification using connected-component contours and edge-based features of uppercase western script. IEEE Trans. Pattern Anal. Mach. Intell. **26**(6), 787–798 (2004)
32. Tan, T.: Texture feature extraction via visual cortical channel modelling. In: Pattern Recognition, 1992. Vol. III. Conference C: Image, Speech and Signal Analysis, Proceedings, 11th IAPR International Conference on, pp. 607–610 (1992). doi:10.1109/ICPR.1992.202060
33. Tan, T.: Written language recognition based on texture analysis. Image Processing, 1996. In: Proceedings, International Conference on, **1**, pp. 185–188, vol. 2 (1996) doi:10.1109/ICIP.1996.560633
34. Tuceryan, M., Jain, A.K.: The Handbook of Pattern Recognition and Computer Vision, 2nd edn. chap. Texture Analysis. pp. 207–248. World Scientific Publishing, Singapore (1998)
35. Zhu, Y., Tan, T., Wang, Y.: Font recognition based on global texture analysis. Pattern Anal. Mach. Intell. IEEE Trans. **23**(10), 1192–1200 (2001). doi:10.1109/34.954608